HITACHI
Inspire the Next

ノンストップデータベース

HiRDB Version 10 構造型データベース機能

解説・手引・文法・操作書

3020-6-578-50

前書き

■ 対象製品

●適用 OS：Red Hat(R) Enterprise Linux(R) Server 7（64-bit x86_64），Red Hat(R) Enterprise Linux(R) Server 8（64-bit x86_64），Red Hat(R) Enterprise Linux(R) Server 9（64-bit x86_64）

P-8462-C5A1 HiRDB Structured Data Access Facility Version 10 10-07

P-8462-AEA1 HiRDB Structured Data Access Facility/Developer's Kit Version 10(64) 10-07

P-F8462-C5A11 HiRDB Structured Data Access Facility Extension for XDM/SD type Version 10 10-04

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI，Cosminexus，DABroker，DBPARTNER，DocumentBroker，HA モニタ，HiRDB，Job Management Partner 1，JP1，OpenTP1，uCosminexus，XDM は，株式会社 日立製作所の商標または登録商標です。

ActiveX は，マイクロソフト 企業グループの商標です。

Amazon Web Services，AWS，Powered by AWS ロゴ，Amazon EC2 は，Amazon.com, Inc. またはその関連会社の商標です。

Linux は，Linus Torvalds 氏の米国およびその他の国における登録商標です。

Oracle(R)，Java 及び MySQL は，Oracle，その子会社及び関連会社の米国及びその他の国における登録商標です。

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat は，米国およびその他の国における Red Hat, Inc.の登録商標です。

Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat Enterprise Linux は，米国およびその他の国における Red Hat, Inc.の登録商標です。

UNIX は，The Open Group の登録商標です。

その他記載の会社名，製品名などは，それぞれの会社の商標もしくは登録商標です。

■ 発行

2023 年 10 月　3020-6-578-50

■ 著作権

All Rights Reserved. Copyright (C) 2019, 2023, Hitachi, Ltd.

変更内容

変更内容（3020-6-578-50） HiRDB Structured Data Access Facility Version 10　10-07

追加・変更内容	変更個所
次に示す製品の適用 OS に Red Hat Enterprise Linux Server 9 を新規にサポートし，Red Hat Enterprise Linux Server 6 のサポートを終了しました。 • HiRDB Structured Data Access Facility Version 10 • HiRDB Structured Data Access Facility/Developer's Kit Version 10(64) • HiRDB Structured Data Access Facility Extension for XDM/SD type Version 10	1.4.1，4.7.1，16.3
検索結果が NOT FOUND の場合のレコード位置指示子に関する説明を追加しました。	2.5.3(1)，2.5.3(4)(b)，2.5.4(1)(a)，2.5.5(2)，2.5.6(5)
SD FMB の SDB データベースでの，排他制御モードの組み合わせの例を変更しました。	2.9.2(2)
排他自動解除機能の適用範囲を拡大しました。	2.9.9(9)，9.3.1(2)(a)
SDB データベース情報長の見積もり式を変更しました。	3.4.2(2)(a)
各サーバプロセスが使用するプロセス固有領域のメモリ所要量の見積もり式を変更しました。	3.4.3，3.4.3(2)(b)
SDB データベースを操作する API の排他モード 2 が'D'の場合の排他資源数の見積もり式を変更しました。	3.8.2(1)(b)
ルートレコード格納ページの共用排他機能の適用範囲を拡大しました。	9.3.1(2)(b)
SDB 定義文の 1 行の長さの最大値を 600 バイトに変更しました。	11.4.1，11.10
二次インデクスの横分割のサポートに伴い，SD FMB の SDB データベースの SDB データベース格納定義で次の変更をしました。 • DEPENDING ON 句を追加しました。 • WITHIN 句のレコードを複数の RD エリアに格納する場合の規則を変更しました。 • SECONDARY INDEX 句の規則を変更しました。 • ORDER KEY 句の構成要素名の規則を追加・削除しました。 • ORDER KEY 句のデータ種別による構成要素の指定可否を変更しました。 • WITHIN 句に二次インデクスの RD エリア分割キー値を指定できるようにしました。	11.7.2(2)，11.7.2(3)(l)，11.7.2(3)(m)，11.7.2(3)(o)，11.7.2(3)(q)，11.7.2(3)(r)
二次インデクスの横分割のサポートに伴い，レコードを格納する RD エリアと二次インデクスを格納する RD エリアの対応をチェックするようにしました。また，論理エラー番号に 15，16，17 を追加しました。	12.6.2(1)，12.7，12.8.2，12.8.3，17.3.1(4)，17.4.5(4)，17.4.6(4)
UAP 統計レポートのアクセスパス情報の出力項目に Set Name を追加しました。	16.5.5(2)(a)，16.5.5(2)(b)

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは，HiRDB Structured Data Access Facility（以降，HiRDB/SD と略します）の機能と使い方について説明したものです。

■ 対象読者

HiRDB/SD を使って，構造型データベース（以降，構造型 DB と略します）を設計する方，構造型 DB システムを構築および運用する方を対象としています。

このマニュアルの記述は，次に示す知識があることを前提にしています。

- 構造型 DB の基礎的な知識
- Linux のシステム管理の基礎的な知識
- HiRDB の導入，設計，および運用に関する基礎的な知識

■ マニュアルの構成

このマニュアルは，次に示す章と付録から構成されています。

第 1 章　HiRDB/SD の概要

HiRDB/SD の特長や構成などについて説明しています。

第 2 章　HiRDB/SD のアーキテクチャ

HiRDB/SD のアーキテクチャについて説明しています。

第 3 章　HiRDB/SD のデータベース設計

HiRDB/SD のデータベース設計について説明しています。

第 4 章　HiRDB システムの構築

HiRDB/SD を使用する際の HiRDB システムの構築について説明しています。

第 5 章　運用

構造型 DB 機能を使用する HiRDB の運用方法について説明しています。

第 6 章　高速系切り替え機能の設定と運用

高速系切り替え機能の設定方法と運用方法について説明しています。

第 7 章　セキュリティ監査機能の設定と運用

セキュリティ監査機能の設定方法と運用方法について説明しています。

第 8 章　ディザスタリカバリシステムの構築と運用【4V FMB，4V AFM】
ディザスタリカバリシステムの構築方法と運用方法について説明しています。

第 9 章　システム定義
システム共通定義，ユニット制御情報定義などについて説明しています。

第 10 章　コマンド
運用コマンドおよびユティリティについて，マニュアル「HiRDB Version 10 コマンドリファレンス」との差異を中心に説明しています。

第 11 章　HiRDB/SD 定義ユティリティ（pdsdbdef）
HiRDB/SD 定義ユティリティの機能と使い方について説明しています。

第 12 章　HiRDB/SD データベース作成ユティリティ（pdsdblod）
HiRDB/SD データベース作成ユティリティの機能と使い方について説明しています。

第 13 章　HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
HiRDB/SD データベース再編成ユティリティの機能と使い方について説明しています。

第 14 章　HiRDB/SD データベースアクセスユティリティ（pdsdbexe）
HiRDB/SD データベースアクセスユティリティの機能と使い方について説明しています。

第 15 章　DML プリプロセサ（pdsdbcbl）【SD FMB】
DML プリプロセサの機能と使い方について説明しています。

第 16 章　HiRDB クライアントの環境設定
インストールや UAP の作成と実行に必要な環境定義の方法などについて説明しています。

第 17 章　DML リファレンス【SD FMB】
SD FMB の SDB データベースを操作する DML の機能と文法について説明しています。

第 18 章　メッセージ
HiRDB が出力するメッセージについて説明しています。

付録 A　未サポート機能の一覧
HiRDB のマニュアルに記載している機能のうち，未サポートの機能，および機能制限がある機能について説明しています。

付録 B　ディクショナリ表
ディクショナリ表について説明しています。

付録 C　ユティリティの最大同時実行数およびコマンドの同時接続数
ユティリティの最大同時実行数，およびコマンドの同時接続数について説明しています。

付録 D　HiRDB/SD の最大値・最小値
HiRDB/SD の最大値および最小値について説明しています。

付録 E　HiRDB/SD で起動するプロセス
HiRDB/SD で起動するプロセスについて説明しています。

付録 F　単調増加ファイル
HiRDB/SD の使用によって単調増加するファイルについて説明しています。

付録 G　ユティリティが出力するファイル
次のユティリティが出力するファイルについて説明しています。
・HiRDB/SD 定義ユティリティ（pdsdbdef）
・HiRDB/SD データベース作成ユティリティ（pdsdblod）
・HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

付録 H　ユティリティの排他制御モード
次のユティリティの排他制御モードについて説明しています。
・HiRDB/SD 定義ユティリティ（pdsdbdef）
・HiRDB/SD データベース作成ユティリティ（pdsdblod）
・HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

付録 I　ユティリティ実行時の留意事項
次のユティリティ実行時の留意事項について説明しています。
・HiRDB/SD 定義ユティリティ（pdsdbdef）
・HiRDB/SD データベース作成ユティリティ（pdsdblod）
・HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB, 4V AFM】
インナレプリカ機能を使用する場合の HiRDB/SD 定義ユティリティ（pdsdbdef）を実行するための条件，および注意事項について説明しています。

付録 K　検索範囲決定の契機，横分割時の検索範囲，および入力情報とキーの定義の関係
検索範囲決定の契機，横分割時の検索範囲，および入力情報とキーの定義の関係について説明しています。

付録 L　SDB データベースを操作する API または DML と，データページの排他解除の有無の関係

SDB データベースを操作する API または DML と，データページ（ページまたはサブページ）の排他解除の有無（排他自動解除機能の対象となるデータページ）の関係について説明しています。

付録 M　複数レコードの検索時の，データ格納エリアに対するレコードの格納形式【4V FMB】

複数レコードの検索時の，データ格納エリアに対するレコードの格納形式について説明しています。

付録 N　障害調査のために必要な情報

問題解決支援または Q&A のサポートサービスを御利用になる場合に必要となる情報について説明しています。

付録 O　予約語の一覧

システムに登録されている予約語について説明しています。

付録 P　4V FMB または 4V AFM の SDB データベースと SD FMB の SDB データベースの機能差一覧

4V FMB または 4V AFM の SDB データベースと，SD FMB の SDB データベースの機能差について説明しています。

付録 Q　用語解説

このマニュアルで使用している用語について説明しています。

■ 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

HiRDB マニュアル

- HiRDB Version 10 構造型データベース機能（UAP 開発編）（3020-6-579）
- HiRDB Version 10 解説（3020-6-551）
- HiRDB Version 10 システム導入・設計ガイド（UNIX(R)用）（3020-6-552）
- HiRDB Version 10 システム定義（UNIX(R)用）（3020-6-554）
- HiRDB Version 10 システム運用ガイド（UNIX(R)用）（3020-6-556）
- HiRDB Version 10 コマンドリファレンス（UNIX(R)用）（3020-6-558）
- インナレプリカ機能 HiRDB Staticizer Option Version 10（3020-6-563）
- HiRDB Version 10 ディザスタリカバリシステム　構築・運用ガイド（3020-6-564）
- HiRDB Version 10 UAP 開発ガイド（3020-6-560）
- HiRDB Version 10 SQL リファレンス（3020-6-561）

- HiRDB Version 10 メッセージ（3020-6-562）

以降，HiRDB Version 10 のマニュアル名は，Version 10,（UNIX(R)用）を省略して表記しています。

関連製品

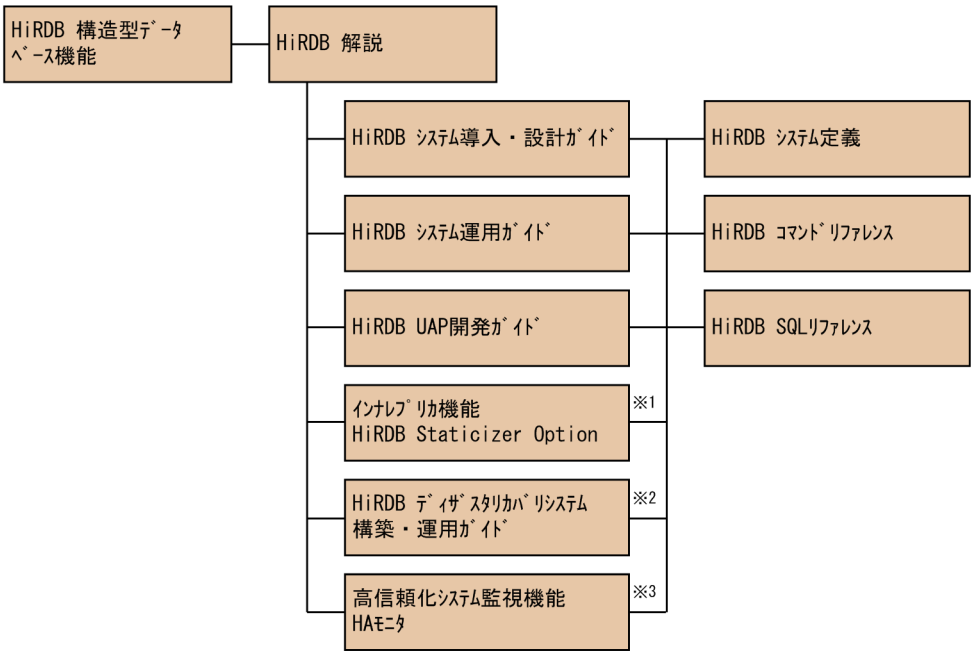
- TP1/Server Base Enterprise Option プログラム作成の手引（3000-3-983）
- TP1/Financial Service Platform 使用の手引（3000-3-K01）
- TP1/Financial Service Platform プログラム作成の手引（3000-3-K02）
- OpenTP1 Version 7 分散トランザクション処理機能　OpenTP1　運用と操作（3000-3-D53）
- 高信頼化システム監視機能 HA モニタ Linux(R)編（3000-9-132）

以降，HA モニタのマニュアル名は，Linux(R)編を省略して表記しています。

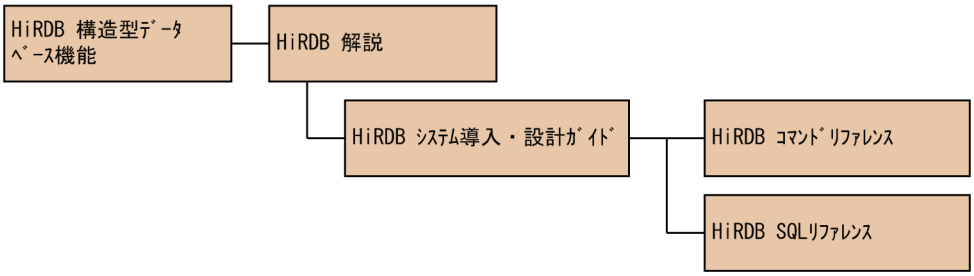
■ 利用者ごとの関連マニュアル

利用者ごとの関連マニュアルを次に示します。左側のマニュアルから順にお読みいただくことをお勧めします。

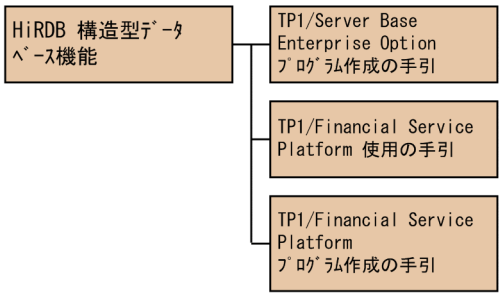
システム管理者が利用するマニュアル



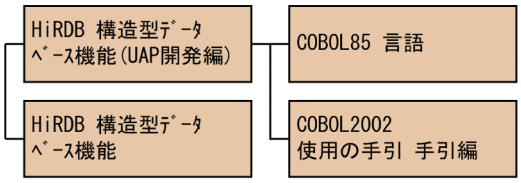
SDBデータベースの作成者が利用するマニュアル



UAP開発者が利用するﾏﾆｭｱﾙ ※4



UAP開発者が利用するﾏﾆｭｱﾙ ※5



注※1　ｲﾝﾅﾚﾌﾟﾘｶ機能を使用する場合にお読みください。
注※2　ﾃﾞｨｻﾞｽﾀﾘｶﾊﾞﾘｼｽﾃﾑを構築する場合にお読みください。
注※3　系切り替え機能を使用する場合にお読みください。
注※4　TP1/FSPが提供するﾗｲﾌﾞﾗﾘ関数を使用して構造型DBを操作するUAPを作成する場合に利用するﾏﾆｭｱﾙです。
注※5　DMLを使用して構造型DBを操作する，COBOL言語の埋込み型UAPを作成する場合に利用するﾏﾆｭｱﾙです。

■ このマニュアルでの表記

このマニュアルでは製品名称および名称について次のように表記しています。ただし，それぞれのプログラムについての表記が必要な場合はそのまま表記しています。

製品名称または名称	表記	
HiRDB Version 10	HiRDB または HiRDB サーバ	
HiRDB Structured Data Access Facility/Run Time Version 10(64)	HiRDB Structured Data Access Facility/Run Time	HiRDB クライアント
HiRDB Structured Data Access Facility/Developer's Kit Version 10(64)	HiRDB Structured Data Access Facility/Developer's Kit	
HiRDB/Run Time(64)	HiRDB/Run Time	
HiRDB/Developer's Kit(64)	HiRDB/Developer's Kit	
HiRDB Advanced High Availability Version 10	HiRDB Advanced High Availability	
HiRDB Staticizer Option Version 10	HiRDB Staticizer Option	
HiRDB Structured Data Access Facility Extension for XDM/SD type Version 10	HiRDB Structured Data Access Facility Extension for XDM/SD type	
JP1/NETM/Audit - Manager	JP1/NETM/Audit	

製品名称または名称	表記
JP1/Performance Management	JP1/PFM
Linux(R)	Linux
Red Hat(R) Enterprise Linux(R) Server 7（64-bit x86_64）	
Red Hat(R) Enterprise Linux(R) Server 8（64-bit x86_64）	
Red Hat(R) Enterprise Linux(R) Server 9（64-bit x86_64）	
OpenTP1/Server Base Enterprise Option	TP1/EE
uCosminexus TP1/Financial Service Platform	TP1/FSP
XDM/SD E2	XDM/SD
システムマネジャ	MGR
フロントエンドサーバ	FES
ディクショナリサーバ	DS
バックエンドサーバ	BES

- HiRDB 運用ディレクトリのパスを$PDDIR と表記します。
- TCP/IP が規定する hosts ファイル（/etc/hosts ファイルも含む）を hosts ファイルと表記します。
- TP1/EE および TP1/FSP をまとめて TP1/FSP と表記します。

■ このマニュアルで使用する略語

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字の表記
ADO	ActiveX(R) Data Objects
AFM	Attached File Management Program
Amazon EC2	Amazon Elastic Compute Cloud
API	Application Programming Interface
BES	Back End Server
BOM	Byte Order Mark
COBOL	Common Business Oriented Language
CSV	Character-Separated Values
DAM	indexed Direct Access Method
DAT	Digital Audio Tape

英略語	英字の表記
DB	Database
DIC	Dictionary Server
DML	Data Manipulate Language
DRBD	Distributed Replicated Block Device
DS	Dictionary Server
EX	Exclusive
FES	Front End Server
FMB	File Manager for Banks
FTP	File Transfer Protocol
ID	Identification number
IP	Internet Protocol
JAR	Java Archive File
JDBC	Java Database Connectivity
JIS	Japanese Industrial Standard code
JP1	Job Management Partner 1
LOB	Large Object
LRU	Least Recently Used
MAM	Main memory Access Method
MGR	System Manager
MIB	Management Information Base
ODBC	Open Database Connectivity
OLE	Object Linking and Embedding
OLTP	On-Line Transaction Processing
OOCOBOL	Object Oriented COBOL
OS	Operating System
PP	Program Product
PR	Protected Retrieve
PU	Protected Update
RAID	Redundant Arrays of Inexpensive Disk

英略語	英字の表記
RD	Relational Database
SAM	Sequential Access Method
SJIS	Shift JIS
SR	Shared Retrieve
SU	Shared Update
TAM	Table Access Method
UAP	User Application Program
XDM/RD E2	Extensible Data Manager / Relational Database Extended Version 2
XDS	Extended Data Server
XML	Extensible Markup Language

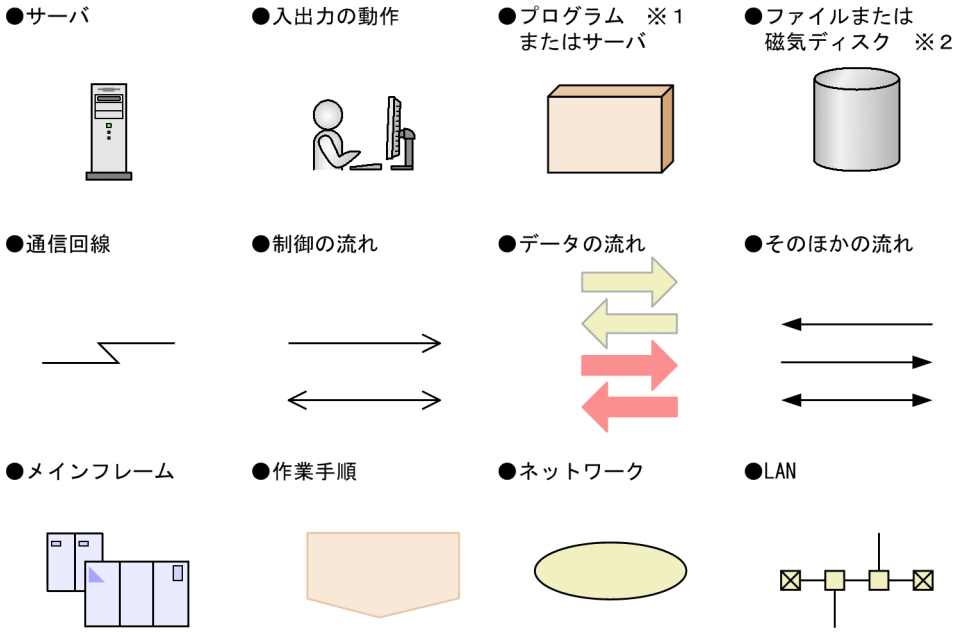
■ ログの表記

OS のログを syslogfile と表記します。syslogfile は，/etc/rsyslog.conf でログ出力先に指定しているファイルです。一般的には，次のファイルが syslogfile となります。

OS	ファイル
Linux	/var/log/messages

■ 図中で使用する記号

このマニュアルの図中で使用する記号を次のように定義します。



注※1

一部の図では，プログラムを単に四角で囲んで（影を付けないで）記載しています。

注※2

バックアップファイルには，磁気ディスク装置のほかに磁気テープ装置，カセット磁気テープ装置（CMT）およびデジタルオーディオテープ装置（DAT）が使用できますが，このマニュアルでは磁気ディスク装置だけを記載しています。

■ このマニュアルで使用する記号

形式および説明で使用する記号を次に示します。ここで説明する文法記述記号は，説明のための記号なので実際には記述しないでください。

文法記述記号	意味
〔　〕	この記号で囲まれている項目は省略できます。 （例）pdfbkup 〔-y〕 これは，pdfbkup と指定するか，または pdfbkup -y と指定できることを示します。
…	この記号直前の項目を繰り返して指定できます。 （例）pdclose -r RD エリア名 〔，RD エリア名〕… これは，-r オプションの RD エリア名を繰り返し指定できることを示します。
|	この記号で区切られた項目を選択できます。 （例）pdloginit -d sys | spd これは，-d オプションに sys か spd のどちらかを指定できることを示します。
{　}	この記号で囲まれている複数の項目のうちから，1 つを選択できます。

文法記述記号	意味
	（例）pdstart 〔{-x ホスト名｜-s サーバ名}〕 これは，-x ホスト名，-s サーバ名の 2 つのオプションのうち，どちらか 1 つを指定することを示します。
（下線）	この記号で示す項目は，省略時の仮定値です。 （例）pdcat 〔-｛a ｜ b ｜ ab｝〕 これは，オプションの指定を省略した場合，-ab オプションが仮定されることを示します。
{{　　}}	この記号で囲まれた複数の項目を 1 つの単位として，繰り返し指定できます。 （例）{{subschema -s SDB データベース名}} これは，「subschema -s SDB データベース名」を繰り返し指定できることを示します。
～	この記号のあとにユーザ指定値の属性を示します。
《　　》	ユーザが指定しなかった場合に仮定される値を示します。
＜　　＞	ユーザ指定値の構文要素記号を示します。
((　　))	ユーザ指定値の指定範囲を示します。

■ このマニュアルで使用する構文要素記号

このマニュアルで使用する構文要素記号を次に示します。

構文要素記号		意味
＜英字＞		アルファベット（A～Z，a～z）と下線（_）
＜英字記号＞		アルファベット（A～Z，a～z）と＃，@，￥
＜英数字＞		英字と数字（0～9）
＜英数字記号＞		英字記号と数字（0～9）
＜符号なし整数＞		数字（0～9）
＜ 16 進数字＞		数字（0～9）と（A～F，a～f）
＜識別子＞		先頭がアルファベットの英数字
＜識別子＞	RD エリア名を指定する場合	先頭が英字記号で始まる英数字記号，下線（_），ハイフン（-），および空白 ハイフンおよび空白を含む場合は，引用符（"）で囲んでください。
	ホスト名を指定する場合	アルファベット（A～Z，a～z），数字（0～9），ピリオド（.），ハイフン（-），下線（_），および@で構成される文字列
	グローバルバッファ名またはパスワードを指定する場合	先頭が英字記号で始まる英数字記号
	認可識別子を指定する場合	英数字記号

構文要素記号		意味
	SDB データベースで使用する名前（SDB データベース名，レコード型名など）を指定する場合	指定できる文字の規則については，「11.4.1(1)　名前の規則」を参照してください。
＜記号名称＞		先頭が英字記号で始まる英数字記号
＜文字列＞		任意の文字の配列
＜パス名＞※		/，英数字，ピリオド（.），#，@で構成される文字列
＜ HiRDB ファイル名＞		アルファベット（A～Z，a～z），数字（0～9），ピリオド（.），下線（_），および@で構成される文字列（最大 30 文字）

注

すべて半角文字を使用してください。

注※

パス名は使用している OS に依存します。

■ このマニュアルで使用する計算式の記号

このマニュアルで使用する計算式の記号の意味を次に示します。

記号	意味
↑↑	計算結果の値を小数点以下で切り上げることを示します。 （例）↑34÷3↑の計算結果は 12 となります。
↓↓	計算結果の値を小数点以下で切り捨てることを示します。 （例）↓34÷3↓の計算結果は 11 となります。
MAX	計算結果の最も大きい値を選ぶことを示します。 （例）MAX（3×6，4 ＋ 7）の計算結果は 18 となります。
MIN	計算結果の最も小さい値を選ぶことを示します。 （例）MIN（3×6，4 ＋ 7）の計算結果は 11 となります。

■ このマニュアルをお読みいただく際の留意事項

このマニュアルでは，構造型 DB（SDB データベース）の定義方法や運用方法について説明しています。このマニュアルで説明している SDB データベースには，次に示す 3 つの種類があります。

・4V FMB

メインフレームの TMS-4V/SP がサポートしている FMB のデータベースに相当します。

・4V AFM

メインフレームの TMS-4V/SP がサポートしている AFM（DAM，MAM，TAM，または SAM）のデータベースに相当します。4V AFM は，4V DAM，4V MAM，4V TAM，および 4V SAM の SDB データベースの総称です。

・SD FMB

メインフレームの XDM/SD E2 がサポートしている階層型モデルのデータベースに相当します。

章，節，項などのタイトルに次の表記がある場合，該当する SDB データベースだけに関する説明であることを意味しています。

【4V FMB】

この表記がある個所の説明は，4V FMB の SDB データベースを使用しているときに該当する記述であることを意味しています。

【4V AFM】

この表記がある個所の説明は，4V AFM の SDB データベースを使用しているときに該当する記述であることを意味しています。

なお，【4V DAM】，【4V MAM】，【4V TAM】，【4V SAM】と表記されている場合は，それぞれ 4V DAM，4V MAM，4V TAM，4V SAM の SDB データベースを使用しているときに該当する記述であることを意味しています。

【SD FMB】

この表記がある個所の説明は，SD FMB の SDB データベースを使用しているときに該当する記述であることを意味しています。

（例）

「2.3.10　二次インデクス【4V FMB，4V AFM】」

上記のように表記されている場合，4V FMB または 4V AFM の SDB データベースを使用しているときに，該当する記述であることを意味しています。そのため，SD FMB の SDB データベースを使用している場合は，「2.3.10　二次インデクス」の説明をお読みいただく必要はありません。

■ KB（キロバイト）などの単位表記について

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト）はそれぞれ 1,024 バイト，$1{,}024^2$ バイト，$1{,}024^3$ バイト，$1{,}024^4$ バイトです。

目次

前書き　2
変更内容　4
はじめに　6

1　HiRDB/SD の概要　38
1.1　メインフレームのデータベース資産をオープン化する HiRDB/SD　39
1.2　HiRDB/SD の特長　41
1.2.1　オープンシステムで稼働する構造型 DB　41
1.2.2　メインフレーム上の構造型 DB や UAP の移行負担を軽減できる　43
1.2.3　同一トランザクション内で構造型 DB とリレーショナル DB の両方にアクセスできる　43
1.3　HiRDB/SD を使用する際のシステム構成　44
1.4　動作条件　46
1.4.1　使用できるプラットフォーム　46
1.4.2　使用できる文字コード　46
1.5　HiRDB/SD の関連プログラムプロダクト　47
1.6　HiRDB/SD に関連する付加 PP　48
1.6.1　HiRDB Staticizer Option　48
1.6.2　HiRDB Advanced High Availability　48
1.6.3　HiRDB Structured Data Access Facility Extension for XDM/SD type　48
1.7　HiRDB Structured Data Access Facility（HiRDB/SD）の位置づけ　49
1.7.1　HiRDB/SD を使用することで追加される機能範囲　49
1.7.2　このマニュアルの記載範囲と HiRDB マニュアルとの関係　49

2　HiRDB/SD のアーキテクチャ　51
2.1　HiRDB/SD のアーキテクチャ　52
2.1.1　HiRDB/SD を使用する HiRDB のシステム構成　52
2.2　データベースの格納構造　53
2.2.1　HiRDB ファイルと HiRDB ファイルシステム領域　53
2.2.2　RD エリア　53
2.2.3　RD エリアの構造　55
2.3　構造型 DB の仕組み　60
2.3.1　HiRDB/SD がサポートしている構造型 DB の種類　60
2.3.2　スキーマ　60
2.3.3　SDB データベース　61
2.3.4　レコード　63

2.3.5 親子集合 66
2.3.6 ポインタによる親子集合の実現 68
2.3.7 SDB データベースの横分割 70
2.3.8 インデクスの概要 76
2.3.9 シーケンシャルインデクス 78
2.3.10 二次インデクス【4V FMB，4V AFM】 80
2.3.11 二次インデクス【SD FMB】 88
2.4 SDB データベースの定義 90
2.4.1 SDB データベースの定義の種類 90
2.4.2 SDB ディレクトリ情報の常駐化 90
2.5 SDB データベースの操作 93
2.5.1 SDB データベースの操作の概要（4V FMB または 4V AFM の場合） 93
2.5.2 SDB データベースの操作の概要（SD FMB の場合） 94
2.5.3 位置指示子 95
2.5.4 レコードの検索【4V FMB，4V AFM】 104
2.5.5 レコードの検索【SD FMB】 112
2.5.6 位置指示子の位置づけ【SD FMB】 114
2.5.7 レコードの格納 116
2.5.8 レコードの更新 119
2.5.9 レコードの削除 119
2.5.10 レコードの一括削除【4V DAM，4V SAM】 120
2.5.11 複数レコードの検索【4V FMB】 120
2.5.12 SDB データベースの操作と位置指示子の関係 127
2.5.13 個別開始実行要求オプションを指定した SDB データベースの操作【4V FMB，4V AFM】 136
2.6 レコードの配置制御（サブページ分割をしない場合） 139
2.6.1 親レコードと子レコードの近傍配置 140
2.6.2 レコードを格納するページの決定方法 143
2.7 レコードの配置制御（サブページ分割をする場合） 172
2.7.1 親レコードと子レコードの近傍配置 173
2.7.2 レコードを格納するサブページの決定方法 176
2.8 グローバルバッファ 206
2.9 排他制御 207
2.9.1 排他制御の単位 207
2.9.2 排他制御のモード 208
2.9.3 排他の期間 228
2.9.4 排他自動解除機能 228
2.9.5 排他自動解除機能を使用したレコードの検索例 230
2.9.6 TAM のデータベースの無排他検索機能【4V TAM】 240
2.9.7 排他モード指定による無排他検索機能【4V FMB，4V AFM】 241

2.9.8 SDB 用 UAP 環境定義の指定による無排他検索機能【SD FMB】 243
2.9.9 デッドロック 244
2.9.10 コミットしていない削除データの排他制御 258
2.9.11 重複するインデクスキー値を格納する際の実行中トランザクション終了待ち合わせ【4V FMB，SD FMB】 262
2.10 データベースの更新ログの取得方式 265
2.10.1 データベースの更新ログ取得方式の種類 265
2.10.2 データベースの更新ログ取得方式の指定方法 265
2.10.3 データベースの更新ログ取得方式による運用方法の違い 266
2.10.4 バックアップについての注意（重要） 267
2.11 システムファイル 268
2.11.1 作業表用ファイル 268
2.12 HiRDB システム定義 269
2.13 HiRDB の開始・終了 270
2.14 系切り替え機能 271
2.14.1 高速系切り替え機能とは 271
2.14.2 系切り替え機能の形態 271
2.14.3 モニタモードとサーバモード 271
2.15 機密保護機能 272
2.15.1 ユーザ権限の種類 272
2.15.2 機密保護機能の運用方法 272

3 HiRDB/SD のデータベース設計 273
3.1 HiRDB/SD の設計 274
3.1.1 システム設計 274
3.1.2 HiRDB ファイルシステム領域の設計 274
3.1.3 システムファイルの設計 275
3.1.4 RD エリアの配置 276
3.1.5 ユニット数またはサーバ数が多いシステムを構築する場合の考慮点 278
3.2 RD エリアの設計 279
3.2.1 ユーザ用 RD エリアを設計する際の考慮点 279
3.2.2 セグメントの設計 279
3.2.3 ページの設計 281
3.2.4 サブページの設計 284
3.2.5 レコード格納時の空き領域の作成（サブページ分割をしない場合） 289
3.2.6 レコード格納時の空き領域の作成（サブページ分割をする場合） 306
3.3 グローバルバッファの設計 328
3.3.1 グローバルバッファを割り当てる単位 328
3.3.2 グローバルバッファの割り当て方法 328
3.3.3 グローバルバッファを割り当てる際の考慮点 329

3.3.4 グローバルバッファの LRU 管理方式 330
3.3.5 グローバルバッファの先読み入力 331
3.4 メモリ所要量の見積もり 332
3.4.1 HiRDB の各ユニットのメモリ配置 332
3.4.2 共用メモリの見積もり 332
3.4.3 プロセス固有メモリの見積もり 338
3.5 ディスク容量の見積もり 346
3.5.1 システムログ量の見積もり 346
3.5.2 作業表用ファイルの容量の見積もり 355
3.5.3 監査証跡ファイルの容量の見積もり 356
3.5.4 システム用 RD エリアの容量の見積もり 358
3.5.5 ユーザ用 RD エリアの容量の見積もり 364
3.5.6 ユーザ用 RD エリア（追い付き反映キー対応表を格納する）の容量の見積もり【4V FMB】 373
3.6 ユティリティ実行時の容量の見積もり 375
3.6.1 ユティリティ実行時のファイル容量の見積もり 375
3.6.2 ユティリティ実行時のメモリ所要量の見積もり 380
3.7 OS のオペレーティングシステムパラメタの見積もり 386
3.7.1 カーネルパラメタの見積もり 386
3.7.2 メッセージキューおよびセマフォ所要量の見積もり 387
3.8 排他資源数の見積もり 388
3.8.1 定義系 SQL 実行時の排他資源数 388
3.8.2 SDB データベースを操作する API または DML の実行時の排他資源数 388
3.8.3 HiRDB/SD 定義ユティリティ（pdsdbdef）実行時の排他資源数 394
3.8.4 HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時の排他資源数 399
3.8.5 HiRDB/SD データベース再編成ユティリティ（pdsdbrog）実行時の排他資源数 400
3.8.6 HiRDB/SD データベースアクセスユティリティ（pdsdbexe）実行時の排他資源数 401

4 HiRDB システムの構築 402

4.1 システム構築手順 403
4.1.1 HiRDB を新規導入するときのシステム構築手順 403
4.1.2 ほかの製品と連携する場合の環境設定 403
4.2 HiRDB のディレクトリおよびファイル構成 404
4.2.1 最初に作成するファイル 404
4.2.2 単調増加ファイル 405
4.3 HiRDB のインストール 406
4.3.1 インストール前の準備 406
4.3.2 HiRDB のインストール手順 406
4.3.3 インストール後の作業 407
4.3.4 HiRDB のアンインストール 409

4.3.5 HiRDB のバージョンアップ 410
4.3.6 修正版 HiRDB Structured Data Access Facility への入れ替え 410
4.4 環境設定 411
4.4.1 HiRDB システム定義の作成 412
4.4.2 HiRDB ファイルシステム領域の作成 412
4.4.3 システムファイルの作成 412
4.4.4 システム用 RD エリアの作成 412
4.4.5 HiRDB の初期開始 412
4.4.6 ユーザ用 RD エリアの作成 412
4.5 SDB データベースの作成 414
4.5.1 SDB データベースを作成する前に必要な作業 414
4.5.2 SDB データベースの定義追加 416
4.5.3 SDB ディレクトリ情報ファイルの作成および配布 417
4.5.4 HiRDB の終了 418
4.5.5 SDB ディレクトリ情報の常駐領域サイズの見直し 418
4.5.6 HiRDB の開始 419
4.5.7 SDB データベースへのデータロード 419
4.6 インナレプリカ機能を使用する場合の環境設定【4V FMB，4V AFM】 420
4.6.1 HiRDB システム定義の指定 420
4.6.2 環境設定方法 420
4.6.3 レプリカ RD エリア作成時の留意事項 420
4.7 HiRDB Server と HiRDB Structured Data Access Facility の入れ替え手順 424
4.7.1 入れ替えに関する前提条件 424
4.7.2 単体版 HiRDB から統合版 HiRDB への入れ替え手順 424
4.7.3 統合版 HiRDB から単体版 HiRDB への入れ替え手順 430
4.8 マルチ HiRDB の設計 437
4.9 SDB 用 UAP 環境定義ファイルの準備【SD FMB】 438
4.9.1 SDB 用 UAP 環境定義ファイルの作成 438
4.9.2 SDB 用 UAP 環境定義ファイルを配置するサーバマシンの検討 439
4.9.3 SDB 用 UAP 環境定義ファイルを格納するディレクトリの作成 440
4.9.4 クライアント環境定義の作成 442

5 運用 444

5.1 運用項目の一覧 445
5.2 HiRDB の開始および終了 447
5.3 システムの状態監視 448
5.4 統計情報の取得 449
5.5 データベースのバックアップの取得 450
5.6 SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合） 452

5.6.1 SDB ディレクトリ情報の変更の流れ 452
5.6.2 SDB データベースの定義追加，定義変更，または定義削除の手順（概要） 453
5.6.3 SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細） 454
5.6.4 SDB データベースの定義追加 461
5.6.5 SDB データベースの定義変更 463
5.6.6 SDB データベースの定義削除 466
5.7 SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合） 467
5.7.1 SDB ディレクトリ情報の変更の流れ 467
5.7.2 SDB データベースの定義追加または定義変更の手順（概要） 468
5.7.3 SDB データベースの定義追加または定義変更の手順（各手順の詳細） 470
5.7.4 SDB データベースの定義追加 474
5.7.5 SDB データベースの定義変更 476
5.7.6 SDB ディレクトリ情報の状態遷移 478
5.7.7 障害発生時の対処（定義追加または定義変更前の状態に戻す場合） 480
5.8 SDB ディレクトリ情報ファイルの配布方法 486
5.8.1 SDB ディレクトリ情報ファイルを作成して配布する場合 486
5.8.2 SDB ディレクトリ情報ファイルの配布だけをする場合 487
5.8.3 特定のユニットに SDB ディレクトリ情報ファイルを配布する場合 488
5.9 全ユニットの SDB ディレクトリ情報が一致しているか確認する方法 490
5.9.1 常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合） 490
5.9.2 常用常駐領域の SDB ディレクトリ情報の確認方法（待機系ユニットの場合） 491
5.9.3 常用常駐領域の SDB ディレクトリ情報の不一致が発生したときの対処 492
5.10 HiRDB システム定義の変更 496
5.10.1 SDB 用 UAP 環境定義の変更方法【SD FMB】 496
5.11 SDB データベースの再編成（インナレプリカ機能を使用しない場合） 497
5.11.1 SDB データベースの再編成の概要 497
5.11.2 SDB データベースの再編成の手順 498
5.12 SDB データベースの再編成（インナレプリカ機能を使用したオンライン再編成）【4V FMB, 4V AFM】 502
5.12.1 オンライン再編成の流れ 502
5.12.2 オンライン再編成の手順 503
5.13 SDB データベースの再編成（インナレプリカ機能を使用した更新可能なオンライン再編成）【4V FMB】 507
5.13.1 更新可能なオンライン再編成の概要 507
5.13.2 運用前に考慮すること 512
5.13.3 更新可能なオンライン再編成の準備作業 518
5.13.4 更新可能なオンライン再編成の運用手順 528
5.13.5 更新可能なオンライン再編成を複数回実行する場合 536
5.13.6 追い付き反映キー対応表を格納している RD エリアを確認したい場合 538
5.13.7 更新可能なオンライン再編成の対象 RD エリアの状態を確認したい場合 538

5.13.8 追い付き反映処理の状態を確認したい場合 538
5.13.9 更新可能なオンライン再編成の操作を取り消したい場合 539
5.13.10 追い付き反映処理時に HiRDB/SD が内部的に発行する API のエラーをスキップしたい場合 540
5.14 RD エリアの運用 541
5.14.1 RD エリアの空き容量の確認 541
5.14.2 RD エリアの追加 542
5.14.3 RD エリアの容量拡張 542
5.14.4 RD エリアの削除 542
5.14.5 RD エリアの設定変更 542
5.15 SDB データベースの横分割数を増やす方法 544
5.15.1 横分割数を増やす手順（再定義をする場合） 544
5.15.2 横分割数を増やす手順（HiRDB の再起動を必要としない場合） 546
5.15.3 横分割数を増やす手順（HiRDB の再起動を必要とする場合） 547
5.16 セキュリティに関する運用 549
5.16.1 構造型 DB 機能に関するユーザ権限の設定 549
5.16.2 ディクショナリ表の参照権限の設定 551
5.17 スキーマの追加，削除 553
5.17.1 スキーマの追加 553
5.17.2 スキーマの削除 553
5.18 データベースの回復 555
5.18.1 データベース回復の流れ 555
5.18.2 例題 1（全 RD エリアをバックアップ取得時点に回復する場合） 557
5.18.3 例題 2（システム用 RD エリアとユーザ用 RD エリアをバックアップ取得時点に回復する場合） 560
5.18.4 例題 3（マスタディレクトリ用 RD エリアを除く RD エリアをバックアップ取得時点に回復する場合） 561
5.18.5 例題 4（全 RD エリアを最新の同期点に回復する場合） 563
5.18.6 例題 5（システム用 RD エリアとユーザ用 RD エリアを最新の同期点に回復する場合） 566
5.18.7 例題 6（マスタディレクトリ用 RD エリアを除く RD エリアを最新の同期点に回復する場合） 568
5.19 SDB データベースへのデータロード 571
5.19.1 SDB データベースへのデータロード手順 571
5.20 インデクスの再作成 574
5.20.1 インデクスの再作成手順 574
5.21 障害が発生したときの対処方法 577
5.22 障害が発生したときの対処方法（pdsdblod コマンドの異常終了時） 578
5.22.1 エラー発生位置と対処コード 578
5.22.2 エラーの対処方法 582
5.23 障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】 587
5.23.1 障害発生時の対処の流れ 587
5.23.2 システムログとオペランドの設定 587

5.23.3 システムログファイルの障害回復 587
5.23.4 更新可能なオンライン再編成の再編成状態の確認 587
5.23.5 障害発生時の対処方法 589
5.23.6 対処の詳細手順（その 1） 594
5.23.7 対処の詳細手順（その 2） 597
5.23.8 対処の詳細手順（その 3） 598
5.23.9 対処の詳細手順（その 4） 600
5.23.10 対処の詳細手順（その 5） 601
5.23.11 対処の詳細手順（その 6） 604
5.23.12 対処の詳細手順（その 7） 606
5.24 OS の時刻を変更する方法 609
5.25 一連番号の監視 610
5.25.1 注意事項 611
5.25.2 レプリカ RD エリアを使用する場合の運用 611
5.26 PRF トレース機能 614
5.26.1 PRF トレース情報の詳細と取得ポイント 614
5.27 システム構成を変更する方法 622

6 高速系切り替え機能の設定と運用 623
6.1 高速系切り替え機能の環境設定の流れ 624
6.2 システム構成例 625
6.3 IP アドレス（ホスト名）の構成例 626
6.4 共有ディスクの準備 627
6.5 HA モニタの環境設定 628
6.6 HiRDB の環境設定 629
6.6.1 前提条件および注意事項 629
6.6.2 HiRDB システム定義の作成 629
6.6.3 RD エリアの作成 630
6.6.4 グローバルバッファの定義 631
6.6.5 監査証跡ファイルの運用 631
6.7 系切り替え時のトランザクションエラーを少なくする設定（トランザクションキューイング機能） 633
6.8 サーバプロセスの異常終了が多発した場合に系を切り替える設定 634
6.9 RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える設定 635
6.10 停止中のユニットがある状態でもシステムマネジャユニットの系切り替えを実行する設定 636
6.11 高速系切り替え機能を使用している場合の運用 637
6.11.1 HiRDB の開始 637
6.11.2 HiRDB の終了 638
6.11.3 ユニットの状態を確認する場合 638
6.11.4 統計解析情報の取得 638

6.11.5 計画系切り替えの手順 638
6.11.6 運用上の注意事項 638
6.11.7 障害発生時のシステムの処理と HiRDB 管理者の処置 641

7 セキュリティ監査機能の設定と運用 642
7.1 セキュリティ監査機能の概要 643
7.1.1 セキュリティ監査機能とは 643
7.1.2 監査証跡の取得範囲 643
7.1.3 監査証跡の取得契機 644
7.1.4 監査証跡として取得する情報 649
7.1.5 監査証跡の参照 649
7.1.6 前提となるシステム構成 649
7.1.7 監査対象になるイベント 649
7.1.8 監査証跡表の自動データロード機能 650
7.2 監査証跡ファイルに出力される情報 651
7.3 監査証跡の出力パターン 652
7.3.1 レコードを検索した場合（FETCH） 652
7.3.2 レコードを格納した場合（STORE） 653
7.3.3 レコードを更新した場合（MODIFY） 654
7.3.4 レコードを削除した場合（ERASE） 654
7.3.5 一括削除をした場合【4V DAM，4V SAM】 656
7.3.6 複数レコードを検索した場合（FETCHDB ALL）【4V FMB】 657
7.3.7 レコードを取得した場合【SD FMB】 658
7.3.8 FIRST オプション機能適用時のデータベース操作と監査証跡レコードの関係 659
7.4 環境設定方法 661
7.4.1 監査証跡表を格納する RD エリアの作成 661
7.4.2 監査対象イベントの定義 661
7.4.3 監査対象イベントの定義の削除 668
7.5 運用方法 671
7.5.1 監査証跡の参照方法 671
7.6 監査証跡ファイルの運用 672
7.7 監査証跡表へのデータ登録 673
7.8 監査証跡表の列構成 674
7.8.1 監査証跡表の列に格納される情報についての留意事項（SDB ユティリティ操作イベントの場合） 674
7.8.2 監査証跡表の列に格納される情報についての留意事項（SDB データベース操作イベントの場合） 675
7.8.3 イベントタイプおよびイベントサブタイプの詳細 688
7.8.4 SQL コードまたは終了コードとイベント成否の詳細 689
7.9 監査証跡の絞り込み 691
7.10 障害が発生したときの対処方法 692

7.11 ほかの機能との関連　693
7.12 監査証跡のレコード項目　694
7.12.1 pdsdbdef コマンドの場合　694
7.12.2 pdsdblod コマンドの場合　696
7.12.3 pdsdbrog コマンドの場合　698
7.12.4 レコードの検索（FETCH）または位置指示子の位置づけ（FIND）の場合　700
7.12.5 レコードの格納（STORE）の場合　703
7.12.6 レコードの更新（MODIFY）の場合　704
7.12.7 レコードの削除（ERASE）の場合　706
7.12.8 レコードの一括削除の場合【4V DAM，4V SAM】　708
7.12.9 複数レコードの検索（FETCHDB ALL）の場合【4V FMB】　710
7.12.10 レコードの取得（GET）の場合【SD FMB】　712
7.13 ユティリティ実行時の監査証跡の出力先ユニット　715
7.14 バージョンアップ時の注意事項　716
7.15 注意事項　717

8 ディザスタリカバリシステムの構築と運用【4V FMB，4V AFM】　718
8.1 ディザスタリカバリシステムの概要　719
8.1.1 ディザスタリカバリシステムとは　719
8.1.2 サポートしているリモートサイトへのデータ反映方式（全非同期方式）　721
8.1.3 前提製品　723
8.2 ディザスタリカバリシステムの設計　724
8.3 ディザスタリカバリシステムの構築　725
8.3.1 ディザスタリカバリシステムの構築手順　725
8.4 ディザスタリカバリシステムの運用　727
8.5 SDB データベースの定義追加，定義変更，または定義削除するときの運用（HiRDB の再起動を必要とする場合）　728
8.5.1 SDB データベースの定義追加，定義変更，または定義削除の手順（概要）　728
8.5.2 SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）　729
8.5.3 障害が発生したときの対処方法　730
8.6 SDB データベースの定義追加または定義変更するときの運用（HiRDB の再起動を必要としない場合）　732
8.6.1 SDB データベースの定義追加または定義変更の手順（概要）　732
8.6.2 SDB データベースの定義追加または定義変更の手順（各手順の詳細）　733
8.6.3 障害が発生したときの対処方法　735
8.7 障害発生時の対処方法　737

9 システム定義　738
9.1 システム共通定義　739
9.1.1 オペランドの形式と説明　739

9.2 ユニット制御情報定義 741
9.2.1 オペランドの形式と説明 741
9.3 SDB 用 UAP 環境定義【SD FMB】 742
9.3.1 オペランドの形式と説明 742
9.3.2 SDB 用 UAP 環境定義の指定例 749
9.3.3 SDB 用 UAP 環境定義の文法規則 750
9.4 指定値の見直しが必要なオペランド 753
9.5 HiRDB の再開始時に指定値を変更できるオペランド 762
9.6 pdconfchk コマンドでチェックできるオペランド 763

10 コマンド 764
10.1 運用コマンドおよびユティリティの実行方法と実行可否 765
10.1.1 運用コマンドおよびユティリティの実行方法 765
10.1.2 運用コマンドおよびユティリティの一覧 765
10.1.3 OS やほかのプログラムが提供しているコマンド名と HiRDB/SD のコマンド名が同じ場合 773
10.2 データベース初期設定ユティリティ（pdinit） 775
10.2.1 create rdarea 文 775
10.2.2 HiRDB ファイルのセグメント数の指定 776
10.3 データベース構成変更ユティリティ（pdmod） 780
10.3.1 RD エリアの追加 780
10.3.2 RD エリアの再初期化 782
10.3.3 RD エリアの削除 784
10.3.4 RD エリアの属性変更 784
10.3.5 RD エリアの移動 784
10.4 データベース再編成ユティリティ（pdrorg） 785
10.4.1 ディクショナリ表の再編成 785
10.5 グローバルバッファ常駐化ユティリティ（pdpgbfon） 788
10.5.1 オプションの説明 788
10.5.2 ユティリティを実行できるユーザ 788
10.6 統計解析ユティリティ（pdstedit） 789
10.6.1 対象となる統計情報 789
10.6.2 UAP に関する統計情報（標準出力） 789
10.6.3 UAP に関する統計情報（DAT 形式ファイル） 794
10.7 データベース状態解析ユティリティ（pddbst） 796
10.7.1 実行できる機能 796
10.7.2 コマンドの形式およびオプションの説明 796
10.7.3 RD エリア単位の状態解析（論理的解析） 797
10.7.4 RD エリア単位の状態解析（物理的解析） 837
10.8 JP1/NETM/Audit 用監査ログ出力ユティリティ（pdaudput） 838

10.8.1 JP1/NETM/Audit 用監査ログ出力ファイルの内容 838
10.9 運用コマンド 840
10.9.1 pdsdbarc（SDB ディレクトリ情報の常駐化および最終更新日時のチェック） 840
10.9.2 pdbkupls（バックアップファイルの情報表示） 848
10.9.3 pddbls（RD エリアの状態表示） 849
10.9.4 pdrdrefls（関連 RD エリアの表示） 850
10.10 運用コマンド（更新可能なオンライン再編成で使用するコマンド）【4V FMB】 855
10.10.1 pdsdborcrt（追い付き反映キー対応表の操作） 855
10.10.2 pdorbegin（オンライン再編成のデータベース静止化） 858
10.10.3 pdorcheck（オンライン再編成の適用条件チェック） 859
10.10.4 pdorend（オンライン再編成の追い付き反映） 861

11 HiRDB/SD 定義ユティリティ（pdsdbdef） 868
11.1 機能 869
11.1.1 ディクショナリ操作機能 870
11.1.2 ディレクトリ操作機能 873
11.2 コマンドの形式 876
11.3 SDB 制御文 877
11.3.1 SDB 制御文の記述規則 877
11.3.2 pdsdbdef コマンドに指定する定義ファイルの関係 878
11.3.3 source 文（SDB 定義文ファイルの指定） 879
11.3.4 environment 文（コマンド実行時の動作環境の指定） 880
11.3.5 dirinf 文（SDB ディレクトリ情報ファイルの作成および配布） 883
11.4 SDB 定義文 888
11.4.1 SDB 定義文の記述規則 888
11.4.2 *ENTRY DICTIONARY 文（SDB ディクショナリ情報の追加） 891
11.4.3 *ALTER DICTIONARY 文（SDB ディクショナリ情報の変更）【4V FMB，4V AFM】 892
11.4.4 *DELETE DICTIONARY 文（SDB ディクショナリ情報の削除） 894
11.4.5 *CHECK DICTIONARY 文（データベース定義のチェック） 895
11.4.6 *ENTRY DIRECTORY 文（SDB ディレクトリ情報の追加） 895
11.4.7 *ALTER DIRECTORY 文（SDB ディレクトリ情報の変更）【4V FMB，4V AFM】 896
11.4.8 *DELETE DIRECTORY 文（SDB ディレクトリ情報の削除） 897
11.4.9 SDB 定義文の同期点処理と推奨する SDB 定義文の指定方法 898
11.4.10 エラー発生時の再実行について 899
11.5 SDB データベース定義，SDB データベース格納定義【4V FMB】 902
11.5.1 SDB データベース定義【4V FMB】 902
11.5.2 SDB データベース格納定義【4V FMB】 925
11.6 SDB データベース定義，SDB データベース格納定義【4V AFM】 952
11.6.1 SDB データベース定義【4V AFM】 952

11.6.2 SDB データベース格納定義【4V AFM】 977
11.7 SDB データベース定義，SDB データベース格納定義【SD FMB】 999
11.7.1 SDB データベース定義【SD FMB】 999
11.7.2 SDB データベース格納定義【SD FMB】 1013
11.8 規則および注意事項 1033
11.8.1 規則 1033
11.8.2 TP1/FSP の SDB ハンドラ機能使用時の注意事項【4V FMB，4V AFM】 1035
11.9 リターンコード 1037
11.10 実行結果ファイルの出力形式 1038
11.11 使用例 1039
11.11.1 SDB データベースの定義例（4V FMB の場合） 1039
11.11.2 SDB データベースの定義例（4V DAM の場合） 1044
11.11.3 SDB データベースの定義例（SD FMB の場合） 1050

12 HiRDB/SD データベース作成ユティリティ（pdsdblod） 1056
12.1 機能 1057
12.1.1 データロード 1057
12.1.2 フォーマットライト【4V DAM】 1061
12.1.3 インデクスの再作成 1065
12.2 コマンドの形式 1068
12.3 pdsdblod 制御文 1069
12.3.1 environment 文 1069
12.3.2 load 文 1077
12.3.3 dbinf 文【4V FMB，4V AFM】 1082
12.3.4 index 文 1084
12.3.5 idxload 文 1085
12.3.6 oreload 文【4V FMB】 1087
12.3.7 pdsdblod 制御文の各オペランドの指定可否 1088
12.3.8 pdsdblod 制御文の記述規則 1091
12.4 規則および注意事項 1093
12.4.1 規則 1093
12.4.2 注意事項 1095
12.4.3 メインフレームで使用していたデータをデータロードする場合の注意事項 1097
12.5 リターンコード 1099
12.6 入力データファイルの形式 1100
12.6.1 レコードのデータ形式 1100
12.6.2 レコードの格納順序 1104
12.7 実行結果ファイルの出力形式 1107
12.8 入力データの論理エラーが発生したときの対処方法 1112

12.8.1 対処の流れ　1112
12.8.2 論理エラー情報ファイルの出力形式　1112
12.8.3 論理エラーの詳細　1113
12.9 トラブルシューティング　1116
12.9.1 pdsdblod コマンドが異常終了した場合　1116
12.9.2 pdsdblod コマンドが無応答状態になった場合　1117
12.9.3 RD エリアの容量不足が発生した場合　1118
12.9.4 タイムアウトが発生した場合　1119
12.10 使用例　1120
12.10.1 初期データロードの例（4V FMB の場合）　1120
12.10.2 初期データロードの例（4V FMB の SDB データベースに二次インデクスを定義している場合）1123
12.10.3 初期データロードの例（4V MAM の場合）　1126
12.10.4 初期データロードの例（SD FMB の場合）　1129
12.10.5 追加データロードの例（4V FMB の場合）　1131
12.10.6 追加データロードの例（SD FMB の場合）　1135
12.10.7 フォーマットライトの例（4V DAM の場合）　1137
12.10.8 インデクスの再作成の例（4V FMB の場合）　1140
12.10.9 インデクスの再作成の例（SD FMB の場合）　1143

13 HiRDB/SD データベース再編成ユティリティ（pdsdbrog）　1146
13.1 機能　1147
13.1.1 レコードのアンロード　1147
13.2 コマンドの形式　1152
13.3 pdsdbrog 制御文　1153
13.3.1 environment 文　1153
13.3.2 unload 文　1155
13.3.3 dbinf 文【4V FMB】　1160
13.3.4 pdsdbrog 制御文の各オペランドの指定可否　1162
13.3.5 pdsdbrog 制御文の記述規則　1163
13.4 規則および注意事項　1164
13.4.1 規則　1164
13.4.2 注意事項　1165
13.5 リターンコード　1167
13.6 アンロードデータファイルの出力形式　1168
13.6.1 レコードのデータ形式　1168
13.6.2 レコードの格納順序　1168
13.6.3 プリフィクス部の出力形式（unldkind オペランドに lod_4v を指定した場合）【4V FMB，4V AFM】　1170
13.7 アンロードの実行結果ファイルの出力形式　1175
13.8 ページ切り替えフラグの扱い【4V FMB，4V AFM】　1177

13.9 トラブルシューティング 1178
13.9.1 pdsdbrog コマンドが異常終了した場合 1178
13.9.2 pdsdbrog コマンドが無応答状態になった場合 1178
13.9.3 タイムアウトが発生した場合 1179
13.10 使用例 1181
13.10.1 4V FMB の SDB データベースのアンロードおよびデータロード 1181
13.10.2 4V FMB の SDB データベースのアンロードおよびデータロード（インナレプリカ機能を使用したオンライン再編成の場合） 1185
13.10.3 4V MAM の SDB データベースのアンロードおよびデータロード 1190
13.10.4 SD FMB の SDB データベースのアンロードおよびデータロード 1194
13.10.5 4V FMB の SDB データベースのアンロード（アンロードデータファイルを UAP の入力情報として使用する場合） 1197
13.10.6 4V MAM の SDB データベースのアンロード（アンロードデータファイルを UAP の入力情報として使用する場合） 1199

14 HiRDB/SD データベースアクセスユティリティ（pdsdbexe） 1202
14.1 機能 1203
14.2 コマンドの形式 1205
14.3 pdsdbexe コマンドの実行手順 1207
14.3.1 4V FMB の SDB データベースのレコードを更新する 1207
14.3.2 4V FMB の SDB データベースのレコードを削除する 1211
14.3.3 4V AFM の SDB データベースのレコードを更新する 1214
14.3.4 4V AFM の SDB データベースのレコードを削除する 1217
14.3.5 SD FMB の SDB データベースのレコードを検索する 1219
14.3.6 SD FMB の SDB データベースのレコードを更新する 1222
14.3.7 SD FMB の SDB データベースのレコードを削除する 1225
14.3.8 ファイルに記述した DML コマンドを実行する場合 1228
14.4 DML コマンド【4V FMB，4V AFM】 1232
14.4.1 CLEAR（レコードの一括削除） 1232
14.4.2 ERASE（レコードの削除） 1236
14.4.3 FETCH（レコードの検索） 1239
14.4.4 FETCHDB_ALL（複数レコードの検索） 1246
14.4.5 FINISH（個別終了） 1251
14.4.6 MODIFY（レコードの更新） 1252
14.4.7 START（個別開始） 1256
14.4.8 STORE（レコードの格納） 1264
14.4.9 DML コマンド共通の規則 1270
14.5 DML コマンド【SD FMB】 1271
14.5.1 ERASE（レコードの削除） 1271
14.5.2 FETCH（レコードの検索） 1273

14.5.3 FINISH（個別終了） 1275
14.5.4 MODIFY（レコードの更新） 1276
14.5.5 START（個別開始） 1279
14.5.6 STORE（レコードの格納） 1281
14.6 DML コマンド共通の規則 1285
14.7 SQL コマンド 1286
14.7.1 COMMIT（トランザクションの正常終了） 1286
14.7.2 CONNECT（HiRDB への接続） 1286
14.7.3 DISCONNECT（HiRDB からの切り離し） 1289
14.7.4 ROLLBACK（トランザクションの取り消し） 1290
14.8 pdsdbexe サブコマンド 1291
14.8.1 #EXIT（pdsdbexe コマンドの終了） 1291
14.8.2 #HELP（ヘルプ情報の出力） 1291
14.8.3 #USAGE（DML コマンドのひな形の出力） 1294
14.9 規則および留意事項 1299
14.9.1 規則 1299
14.9.2 留意事項 1301
14.10 リターンコード 1303
14.11 トラブルシューティング 1304
14.11.1 障害が発生したときの対処方法 1304

15 DML プリプロセサ（pdsdbcbl）【SD FMB】 1306
15.1 機能 1307

16 HiRDB クライアントの環境設定 1308
16.1 概要 1309
16.2 UAP の設計 1310
16.2.1 性能向上，操作性向上に関する UAP の設計【4V FMB，4V AFM】 1310
16.3 HiRDB クライアントの環境設定 1312
16.3.1 クライアント環境定義（環境変数の設定） 1313
16.4 UAP 実行前の準備 1325
16.4.1 コンパイルとリンケージ 1325
16.5 UAP の障害対策 1326
16.5.1 トラブルシュート 1326
16.5.2 SQL トレース機能 1326
16.5.3 クライアントエラーログ機能 1339
16.5.4 拡張 SQL エラー情報出力機能 1340
16.5.5 UAP 統計レポート機能 1350
16.5.6 トラブルシュート機能に関する注意事項 1362

17 DML リファレンス【SD FMB】 1363
17.1 DML の一覧 1364
17.2 基本項目 1365
17.2.1 DML の記述形式 1365
17.2.2 DML のデータ型 1366
17.2.3 埋込み変数 1368
17.3 構成要素の詳細 1370
17.3.1 探索条件の指定 1370
17.3.2 インデクス選択規則 1375
17.3.3 検索の順序，開始位置および方向 1376
17.4 操作系 DML 1381
17.4.1 ERASE 文 1381
17.4.2 FETCH 文 1382
17.4.3 FIND 文 1386
17.4.4 GET 文 1390
17.4.5 MODIFY 文 1391
17.4.6 STORE 文 1393
17.5 埋込み言語文法 1398
17.5.1 DML 先頭子 1398
17.5.2 DML 終了子 1398

18 メッセージ 1400
18.1 HiRDB が出力するメッセージ 1401

付録 1402
付録 A 未サポート機能の一覧 1403
付録 B ディクショナリ表 1408
付録 B.1 ディクショナリ表の一覧 1408
付録 B.2 ディクショナリ表の詳細 1408
付録 C ユティリティの最大同時実行数およびコマンドの同時接続数 1422
付録 C.1 ユティリティの最大同時実行数 1422
付録 C.2 コマンドの同時接続数 1423
付録 D HiRDB/SD の最大値・最小値 1424
付録 E HiRDB/SD で起動するプロセス 1427
付録 F 単調増加ファイル 1428
付録 F.1 簡易ダンプ 1428
付録 F.2 トラブルシュート情報 1430
付録 F.3 ユティリティおよびコマンド実行時の出力ファイル 1431
付録 G ユティリティが出力するファイル 1434

付録 H　ユティリティの排他制御モード　1436
付録 H.1　HiRDB/SD 定義ユティリティ（pdsdbdef）の排他制御モード　1436
付録 H.2　HiRDB/SD データベース作成ユティリティ（pdsdblod）の排他制御モード　1437
付録 H.3　HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の排他制御モード　1441
付録 I　ユティリティ実行時の留意事項　1443
付録 I.1　RD エリアの状態によるユティリティの実行可否　1443
付録 I.2　ユティリティ実行中の割り込みによる強制終了　1445
付録 I.3　LANG 環境変数の設定　1445
付録 I.4　ユティリティ実行時に取得される統計情報　1446
付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】　1447
付録 J.1　HiRDB/SD 定義ユティリティ（pdsdbdef）を実行するための条件　1447
付録 J.2　RD エリアの状態ごとの HiRDB/SD 定義ユティリティ（pdsdbdef）の実行可否　1448
付録 J.3　インナレプリカ機能を使用中に HiRDB/SD 定義ユティリティ（pdsdbdef）を実行する場合の注意事項　1451
付録 K　検索範囲決定の契機，横分割時の検索範囲，および入力情報とキーの定義の関係　1453
付録 K.1　検索範囲決定の契機　1453
付録 K.2　横分割時の検索範囲　1456
付録 K.3　入力情報とキーの定義の関係【4V FMB，4V AFM】　1467
付録 L　SDB データベースを操作する API または DML と，データページの排他解除の有無の関係　1476
付録 M　複数レコードの検索時の，データ格納エリアに対するレコードの格納形式【4V FMB】　1481
付録 N　障害調査のために必要な情報　1483
付録 O　予約語の一覧　1484
付録 O.1　DML の予約語　1484
付録 O.2　pdsdbcbl コマンドの予約語　1485
付録 P　4V FMB または 4V AFM の SDB データベースと SD FMB の SDB データベースの機能差一覧 1486
付録 Q　用語解説　1491

索引　1501

1 HiRDB/SD の概要

この章では，HiRDB/SD の概要について説明します。

1.1 メインフレームのデータベース資産をオープン化する HiRDB/SD

HiRDB/SD は，構造型 DB を管理します。構造型 DB とは，データ間の関連を表形式ではなく，階層構造で管理するデータベースです。

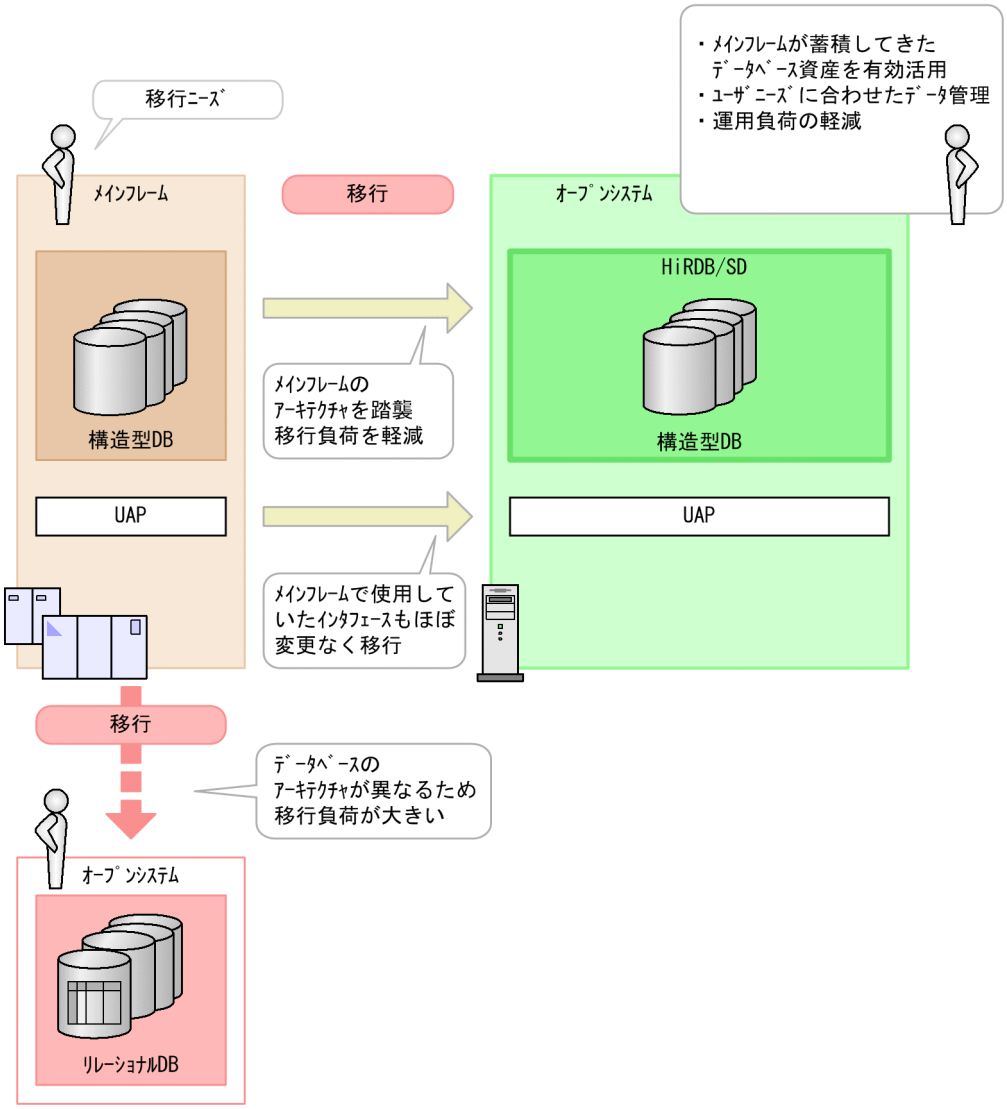
近年，メインフレームに対する性能や信頼性が再認識される一方で，メインフレーム上で管理される構造型 DB に対する，オープンシステムへの移行のニーズも高まっています。しかし，メインフレームからオープンシステムへの移行は進んでいません。その理由の 1 つとして，オープンシステムで管理されるデータベースの主流はリレーショナル DB であり，構造型 DB とはデータベースのアーキテクチャが大きく異なっていることが挙げられます。

メインフレーム上の構造型 DB をオープンシステム上のリレーショナル DB に移行することもできますが，移行のための費用や時間が多く掛かってしまいます。

HiRDB/SD は，メインフレーム上で稼働している構造型 DB のアーキテクチャを踏襲しています。また，UAP のインタフェースの差異もほぼ変更しないでオープンシステムに移行できます。そのため，メインフレーム上の構造型 DB をオープンシステム上に移行する際の負担を大きく軽減できます。

構造型 DB のオープンシステムへの移行と HiRDB/SD の位置づけを次の図に示します。

図 1-1　構造型 DB のオープンシステムへの移行と HiRDB/SD の位置づけ



メインフレームが蓄積してきたデータベース資産には，有効活用できるデータや今後も継続的な管理が必要なデータが数多く存在します。HiRDB/SD を使用して，メインフレームのデータベース資産をオープン化することによって，データベース資産のさらなる有効活用やユーザニーズに合わせたデータ管理，運用負荷の低減が期待できます。

1.2 HiRDB/SD の特長

HiRDB/SD の特長として，次の項目について説明します。

- オープンシステムで稼働する構造型 DB
- メインフレーム上の構造型 DB や UAP の移行負担を軽減できる
- 同一トランザクション内で構造型 DB とリレーショナル DB の両方にアクセスできる

1.2.1 オープンシステムで稼働する構造型 DB

HiRDB/SD は，オープンシステムで稼働する構造型 DB を提供します。

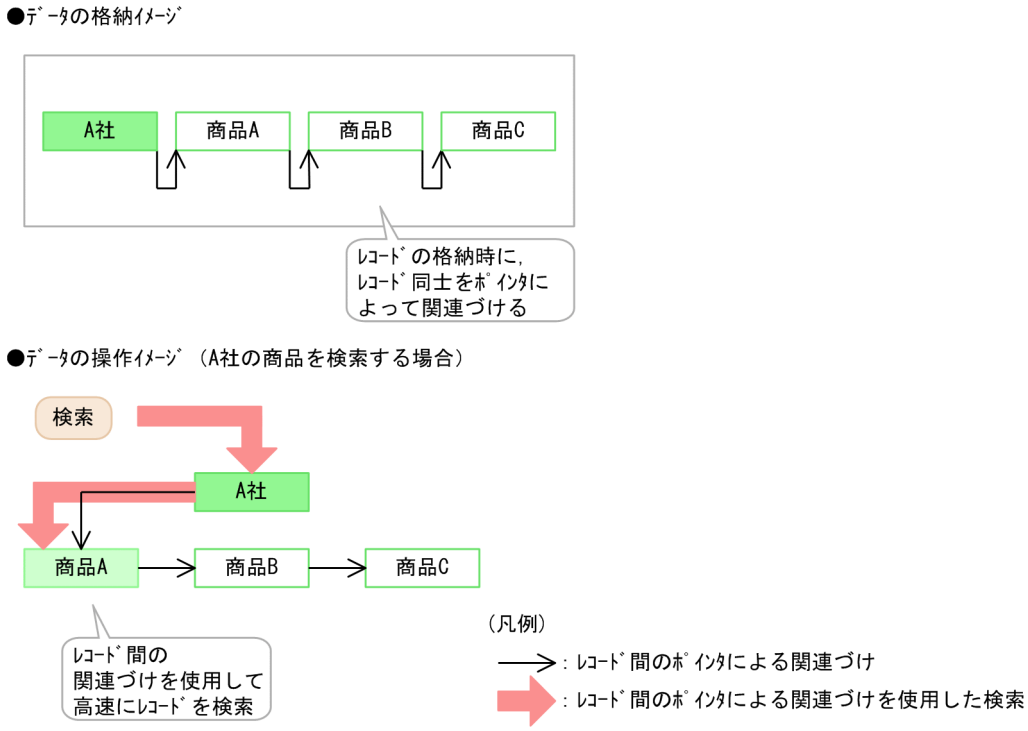
構造型 DB では，レコードという単位でデータを管理します。レコード間の関連は階層構造（親子関係）で管理します。

一般的に，構造型 DB を前提に設計された UAP では，親となるレコード（親レコード）を検索したあとに，そのレコードに関連づけられたレコード（子レコード）を検索するというように，データベースの構造に基づいてアクセス順序を決定します。このようなデータベースアクセス要求に対応するため，HiRDB/SD には次のような特長があります。

(1) レコード同士を関連づけるデータ構造

レコードの順序性を意識したデータベースアクセスを実現するため，レコード同士をポインタによって関連づけるデータ構造になっています。

図 1-2　レコード同士をポインタによって関連づけるデータ構造

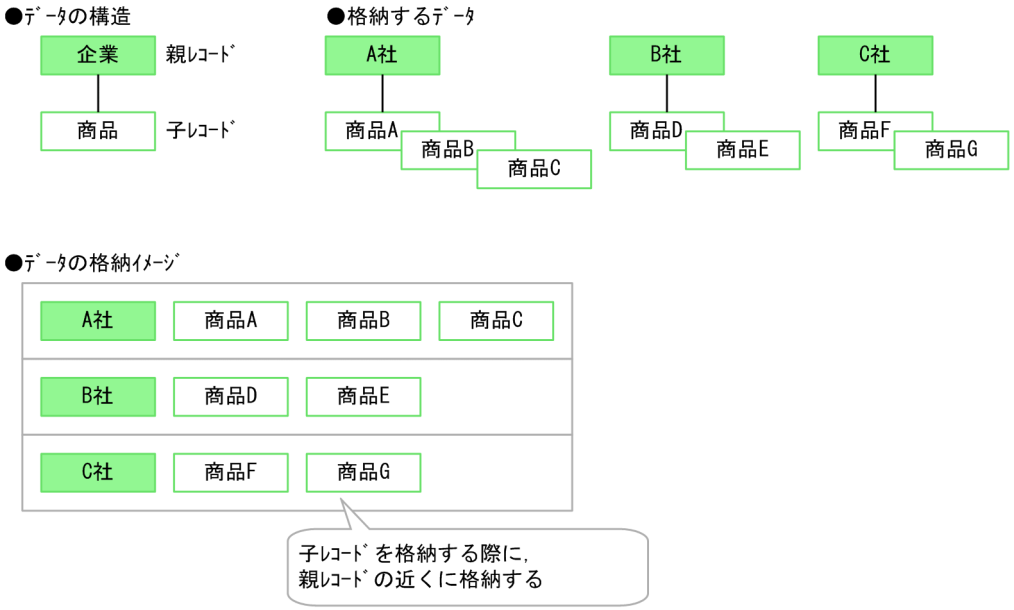


(2)　関連するレコードを近くに配置できる（近傍配置）

HiRDB/SD では，同じルートレコード（レコードの階層構造で最上位のレコード）に連なるレコード群を，基準となるレコード（基準レコード）が格納されているページ（基準ページまたは基準サブページ）およびそのページの近傍に格納します。これを近傍配置といいます。近傍配置をすることで，親子関係にあるレコードが近くに格納されるようになります。これによって，アクセス時間が短縮され，データベースへのアクセス性能が向上します。

基準レコード，基準ページついては，「2.6.1(1)　基準レコード，基準ページ」を参照してください。

図 1-3　レコードの近傍配置の概要



1.2.2　メインフレーム上の構造型 DB や UAP の移行負担を軽減できる

HiRDB/SD が提供する構造型 DB は，メインフレーム上で稼働している構造型 DB のアーキテクチャを踏襲しています。そのため，基本的にデータベース構造を変更することなく，メインフレームからオープンシステムに移行できます。これによって，移行に掛かる時間やコストを削減できます。

メインフレームの TMS-4V/SP の構造型 DB にアクセスするインタフェース（API）については，メインフレームとの仕様差を HiRDB/SD および前提製品である uCosminexus TP1/Financial Service Platform が吸収します。また，メインフレームの XDM/SD の階層型モデルのデータベースにアクセスするインタフェースを踏襲した DML を HiRDB/SD が提供します。これによって，メインフレーム上で使用していた UAP をオープンシステムに容易に移行できます。

1.2.3　同一トランザクション内で構造型 DB とリレーショナル DB の両方にアクセスできる

リレーショナル DB にアクセスするときは HiRDB が提供する SQL を使用します。また，構造型 DB にアクセスするときは uCosminexus TP1/Financial Service Platform が提供するインタフェース（API），または HiRDB/SD が提供するインタフェース（DML）を使用します。

上記のインタフェースを使用することで，同一トランザクション内で構造型 DB とリレーショナル DB の両方にアクセスできます。データベースへの接続や切断，トランザクションに関連するインタフェースについては，構造型 DB とリレーショナル DB で共通のインタフェースを使用します。

1.3　HiRDB/SD を使用する際のシステム構成

HiRDB/SD が管理する構造型 DB には，次のどちらかのインタフェースを使用してアクセスします。

- uCosminexus TP1/Financial Service Platform が提供するインタフェース（API）
- HiRDB/SD が提供するインタフェース（DML）

使用するインタフェースによってシステム構成が異なります。それぞれのインタフェースを使用したときのシステム構成例の図を示します。

図 1-4　HiRDB/SD を使用する際のシステム構成例（uCosminexus TP1/Financial Service Platform が提供するインタフェース（API）を使用する場合）

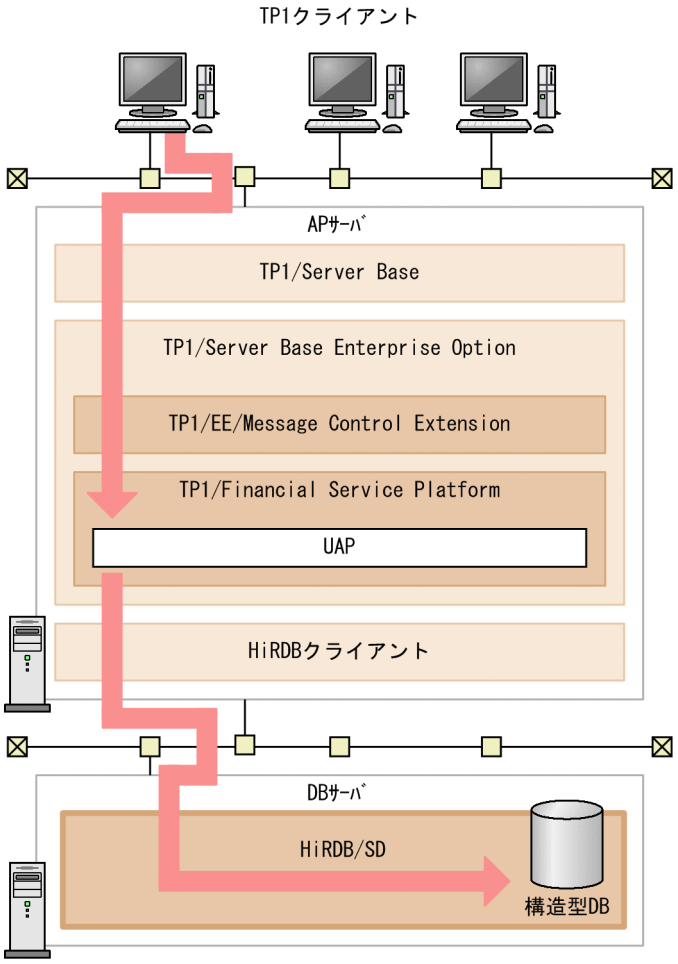
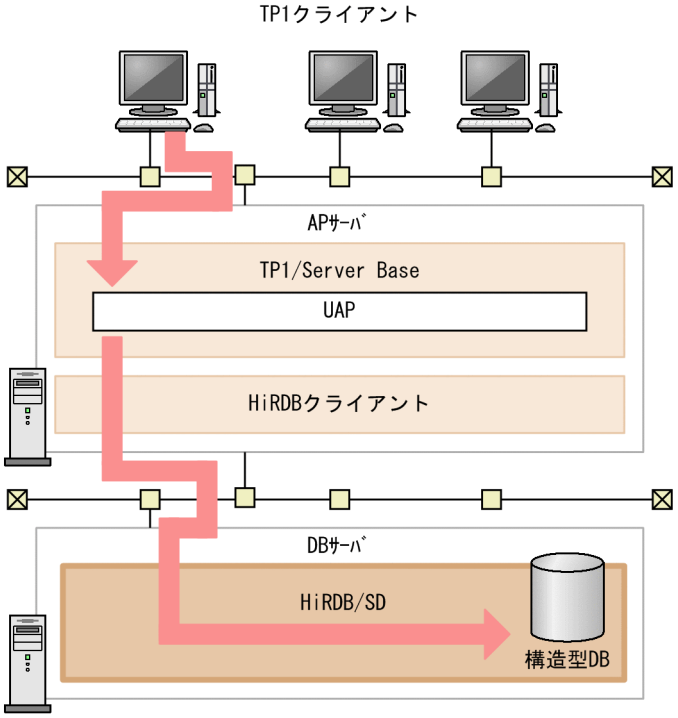


図 1-5　HiRDB/SD を使用する際のシステム構成例（HiRDB/SD が提供するインタフェース（DML）を使用する場合）



1.4　動作条件

HiRDB/SD を使用する際の動作条件について説明します。

1.4.1　使用できるプラットフォーム

HiRDB/SD を使用できるプラットフォームを次に示します。

- Red Hat Enterprise Linux Server 7（64-bit x86_64）
- Red Hat Enterprise Linux Server 8（64-bit x86_64）
- Red Hat Enterprise Linux Server 9（64-bit x86_64）

1.4.2　使用できる文字コード

HiRDB/SD で使用できる文字コードについて，次の点に注意してください。

- HiRDB/SD で使用できる文字コードは，シフト JIS 漢字コードまたは Unicode（UTF-8）です。
- HiRDB/SD にアクセスする UAP から渡されるデータに対して，HiRDB/SD では文字コード変換を行いません。このため，UAP が扱うデータの文字コードと HiRDB/SD が使用する文字コードは同じにしてください。
- ユティリティ実行時に使用するデータの文字コードと HiRDB/SD で使用する文字コードは同じにしてください。

1.5 HiRDB/SD の関連プログラムプロダクト

HiRDB/SD の関連プログラムプロダクトを次に示します。詳細については，リリースノートを参照してください。

- uCosminexus TP1/Financial Service Platform（TP1/FSP）
  HiRDB/SD が管理する構造型 DB にアクセスするインタフェースを提供します。
- COBOL2002 Net Server Suite(64)
  DML を記述した UAP のコンパイルおよびリンケージをするために使用します。
- COBOL2002 Net Server Runtime(64)
  DML を記述した UAP を実行するために使用します。
- OpenTP1/Server Base
  DML を記述した UAP を OLTP 環境下で実行する場合，UAP のトランザクション制御をするために使用します。
- HA モニタ
  系切り替え機能を実現するためのクラスタソフトウェアです。現用系サーバに障害が発生した場合，待機系サーバにサービスを引き継ぎます。

1.6 HiRDB/SD に関連する付加 PP

ここでは，次に示す HiRDB/SD に関連する付加 PP で実現できる機能および操作について説明します。

- HiRDB Staticizer Option
- HiRDB Advanced High Availability
- HiRDB Structured Data Access Facility Extension for XDM/SD type

1.6.1 HiRDB Staticizer Option

HiRDB Staticizer Option を導入すると，インナレプリカ機能が使用できるようになります。

インナレプリカ機能については，マニュアル「HiRDB 解説」の「HiRDB Staticizer Option【UNIX 版限定】」を参照してください。

1.6.2 HiRDB Advanced High Availability

HiRDB Advanced High Availability を導入すると，HiRDB/SD でも分割格納条件の変更機能が使用できるようになります。

分割格納条件の変更機能とは，SDB データベース格納定義で定義済みの横分割したレコード型の格納 RD エリア数を，HiRDB/SD 定義ユティリティ（pdsdbdef コマンド）で変更できる機能です。

分割格納条件の変更機能を使うことで，追加，削除，変更対象外の RD エリアに格納したデータに影響を及ぼすことなく，横分割する RD エリア数の変更ができます。

1.6.3 HiRDB Structured Data Access Facility Extension for XDM/SD type

HiRDB Structured Data Access Facility Extension for XDM/SD type を導入すると，SD FMB の SDB データベースに二次インデクスを定義できるようになります。

1.7　HiRDB Structured Data Access Facility（HiRDB/SD）の位置づけ

ここでは，HiRDB Structured Data Access Facility（HiRDB/SD）と HiRDB の関係，およびこのマニュアルの記載範囲と HiRDB マニュアルとの関係について説明します。

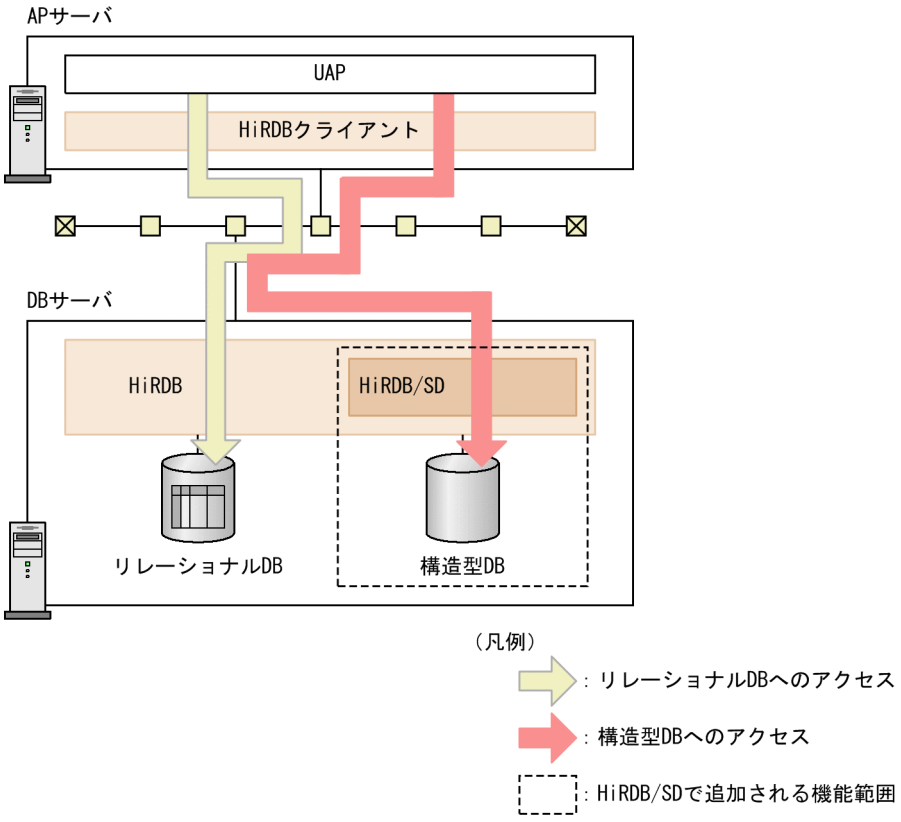
1.7.1　HiRDB/SD を使用することで追加される機能範囲

HiRDB Structured Data Access Facility（HiRDB/SD）は，リレーショナル DB に加えて，HiRDB で構造型 DB を管理するためのオプション製品です。

HiRDB/SD を使用する際の基本的な運用は HiRDB と同じです。コマンドも，構造型 DB 機能固有のコマンド以外は，HiRDB のコマンドを使用します。このため，マニュアルも HiRDB のマニュアルをご使用いただきます。

HiRDB/SD を使用することで追加される機能範囲を次の図に示します。

図 1-6　HiRDB/SD を使用することで追加される機能範囲



1.7.2　このマニュアルの記載範囲と HiRDB マニュアルとの関係

このマニュアルの記載範囲と HiRDB マニュアルとの関係について説明します。

(1)　このマニュアルの記載範囲

このマニュアルでは，HiRDB/SD を使用する際に必要となる，HiRDB/SD 固有の機能や運用方法についてだけ記載しています。リレーショナル DB に関する機能や HiRDB の運用方法については，HiRDB のマニュアル（「はじめに」の「関連マニュアル」に記載している HiRDB マニュアル）をご使用いただきます。

(2)　HiRDB のマニュアルをご使用いただく際の留意事項

構造型 DB 機能を使用する目的で，HiRDB のマニュアルをご使用いただく際の留意事項を次に示します。

- HiRDB/パラレルサーバを HiRDB/SD に読み替えてください。
- 表はレコード型に読み替えてください。
- SQL は次のように読み替えてください。
  - uCosminexus TP1/Financial Service Platform が提供するインタフェース（API）を使用する場合：SDB データベースを操作する API
  - HiRDB/SD が提供するインタフェース（DML）を使用する場合：DML

その他，マニュアルの本文中に機能の読み替えの指示がある場合は，その指示に従ってください。

(3)　構造型 DB 機能を使用する際の制限事項

構造型 DB 機能を使用する場合，HiRDB のマニュアルに記載してある機能が使用できなかったり，使用する際に制限があったりすることがあります。詳細については，「付録 A　未サポート機能の一覧」を参照してください。

2 HiRDB/SD のアーキテクチャ

この章では，HiRDB/SD のデータベースの論理構造，物理構造，および処理方式について説明します。

2.1 HiRDB/SD のアーキテクチャ

HiRDB/SD は，ネットワークで接続された複数のサーバマシンで構築された HiRDB 上で使用できます。

HiRDB のアーキテクチャについては，マニュアル「HiRDB 解説」を参照してください。

2.1.1 HiRDB/SD を使用する HiRDB のシステム構成

HiRDB/SD は，次に示すサーバから構成されたシステムで使用できます。

- システムマネジャ（MGR）
- ディクショナリサーバ（DS）
- フロントエンドサーバ（FES）
- バックエンドサーバ（BES）

サーバを管理する単位としてユニットという概念を使用します。

HiRDB のシステム構成とそれぞれのサーバの役割については，マニュアル「HiRDB 解説」の「HiRDB/パラレルサーバの構成」を参照してください。

2.2　データベースの格納構造

ここでは，HiRDB/SD が管理するデータベースが，物理的および論理的にどのように格納されるのかについて説明します。

2.2.1　HiRDB ファイルと HiRDB ファイルシステム領域

HiRDB ファイルと HiRDB ファイルシステム領域については，マニュアル「HiRDB 解説」の「HiRDB ファイルシステム領域」を参照してください。

HiRDB ファイルシステム領域の作成方法については，「4.4.2　HiRDB ファイルシステム領域の作成」を参照してください。

2.2.2　RD エリア

RD エリアとは，構造型 DB のデータ（レコードおよびインデクス）を格納する論理的な領域のことです。RD エリアは，HiRDB ファイル上に作成します。

ここでは，構造型 DB 機能で利用できる RD エリアの種類，および HiRDB ファイルと RD エリアの関係について説明します。

(1)　RD エリアの種類

構造型 DB 機能で利用できる RD エリアの種類を次の表に示します。それぞれの RD エリアについては，以降で説明します。

表 2-1　RD エリアの種類

項番	RD エリアの種類	
1	システム用 RD エリア	マスタディレクトリ用 RD エリア
2		データディクショナリ用 RD エリア
3		データディレクトリ用 RD エリア
4	ユーザ用 RD エリア	

(2)　マスタディレクトリ用 RD エリア

マスタディレクトリ用 RD エリアには，システムの内部情報を格納します。

(3)　データディクショナリ用 RD エリア

データディクショナリ用 RD エリアには，ディクショナリ表およびディクショナリ表のインデクスを格納します。ディクショナリ表は，HiRDB の利用者が検索できます。ディクショナリ表については，「付録 B　ディクショナリ表」を参照してください。

(4)　データディレクトリ用 RD エリア

データディレクトリ用 RD エリアには，システムの内部情報を格納します。

(5)　ユーザ用 RD エリア

ユーザ用 RD エリアには SDB データベースを格納します。

ユーザ用 RD エリアに SDB データベースを格納する際，レコードとインデクスは別のユーザ用 RD エリアに格納します。

1 つのユーザ用 RD エリアには，同一の SDB データベースに属するレコードだけが格納できます。ただし，インデクスの場合は，異なる SDB データベースのインデクスを 1 つのユーザ用 RD エリアに格納できます。

なお，RD エリアに対するアクセス管理によって次のように分類されます。

- 公用 RD エリア（HiRDB に登録されているすべてのユーザが使用できる）
- 私用 RD エリア（権限があるユーザだけが使用できる）

SDB データベースを格納するユーザ用 RD エリアは，公用 RD エリアとしてください。SDB データベースについては，「2.3.3　SDB データベース」を参照してください。

また，ユーザ用 RD エリアには，更新可能なオンライン再編成，またはセキュリティ監査機能で使用する次の表およびインデクスも格納します。

- 追い付き反映キー対応表（更新可能なオンライン再編成）
- 追い付き反映キー対応表のインデクス（更新可能なオンライン再編成）
- 追い付き状態管理表（更新可能なオンライン再編成）
- 監査証跡表（セキュリティ監査機能）

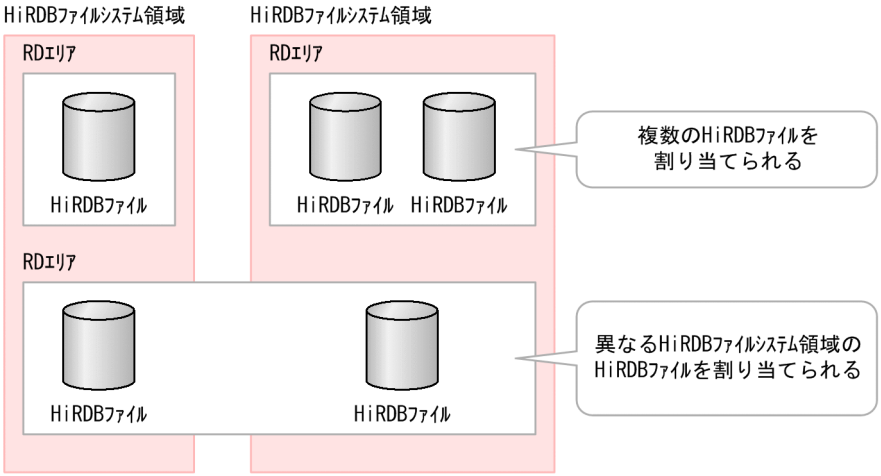
更新可能なオンライン再編成機能で使用する表およびインデクスを格納するユーザ用 RD エリアは，公用 RD エリアとして定義してください。

監査証跡表を格納するユーザ用 RD エリアについては，マニュアル「HiRDB システム運用ガイド」の「監査人の登録，監査証跡表を格納する RD エリアの作成，および監査証跡表の作成」の「監査証跡表を格納する RD エリアの作成」の「RD エリアの作成」を参照して，格納できる RD エリアの種類を確認してください。

(6) HiRDB ファイルと RD エリアの関係

RD エリアは，HiRDB ファイル上に作成します。1 つの RD エリアに対して複数の HiRDB ファイルを割り当てることができます。その際，異なる HiRDB ファイルシステム領域の HiRDB ファイルを割り当てることもできます。HiRDB ファイルと RD エリアの関係を次の図に示します。

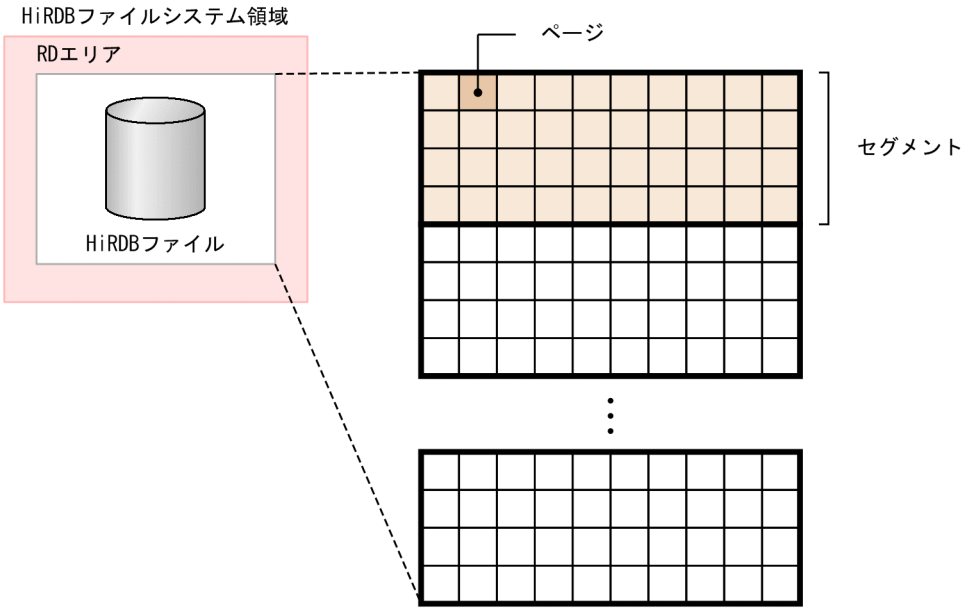
図 2-1　HiRDB ファイルと RD エリアの関係



2.2.3　RD エリアの構造

RD エリアの構造を次の図に示します。

図 2-2　RD エリアの構造



(1)　セグメント

セグメントは，レコードおよびインデクスのデータを RD エリアに格納する最小単位です。セグメントは連続した複数ページから構成されています。

1 セグメントには，同一の SDB データベースのレコードまたは 1 つのインデクスのデータだけが格納されます。SDB データベースについては，「2.3.3　SDB データベース」を参照してください。

(2)　ページ

ページは，データベースの入出力処理の最小単位です。ページ長を大きくすると，関連するレコードを同一ページ内に格納できるため，データを連続して処理する場合に入出力回数を削減できます。逆に，ページを小さくすると，1 ページ当たりに格納できるレコードの件数が少なくなり，データベースアクセスの同時実行性を高めることができます。また，データを格納する際にむだな領域が発生するのを抑えることができます。

ページには次の表に示す種類があります。

表 2-2　ページの種類

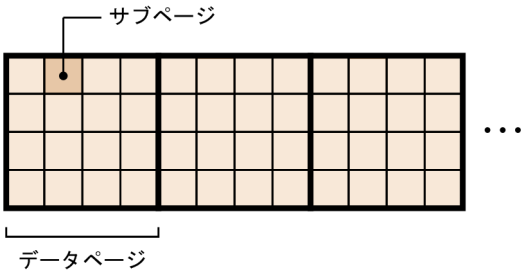
ページの種類	説明
データページ	レコードを格納するページです。
インデクスページ	インデクスのキー値を格納するページです。
ディレクトリページ	RD エリアの管理情報を格納するページです。

(3)　サブページ

ページのうち，データページ（レコードを格納するページ）は，さらに複数の領域に分割できます。この分割した 1 つの領域をサブページといい，ページを分割することをサブページ分割といいます。

サブページの構造を次の図に示します。

図 2-3　サブページの構造



[説明]

1 データページは，2～16 のサブページに分割できます。

HiRDB/SD では，データページにレコードを格納する際，ルートレコードが異なるレコードは別々のデータページに格納されます。そのため，ルートレコード下のレコード数が少ない場合など，レコードのサイズやレコードの配置制御によっては，レコードの格納効率が悪くなることがあります。このような場合，データページをサブページ分割すると，ルートレコードが異なるレコードを同じデータページ内の別々のサブページに格納できるため，レコードの格納効率が良くなります。

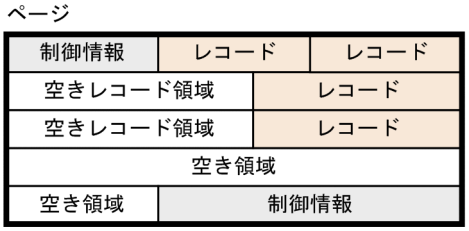
SDB データベース格納定義の STORAGE RECORD 句の SUBPAGE NUMBER でサブページ分割数を指定すると，データページがサブページ分割されます。

(4) ページの格納構造

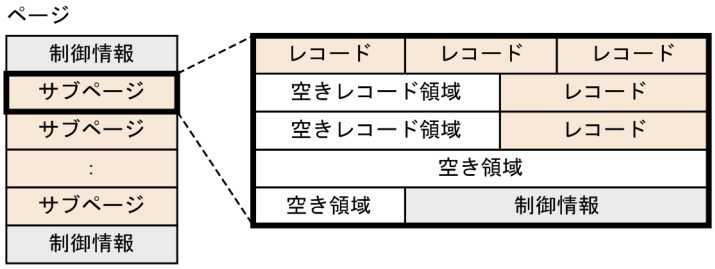
ページの格納構造を，サブページ分割していない場合とサブページ分割している場合について，次の図に示します。

図 2-4 ページの格納構造

●サブページ分割していない場合のページの格納構造



●サブページ分割している場合のページの格納構造



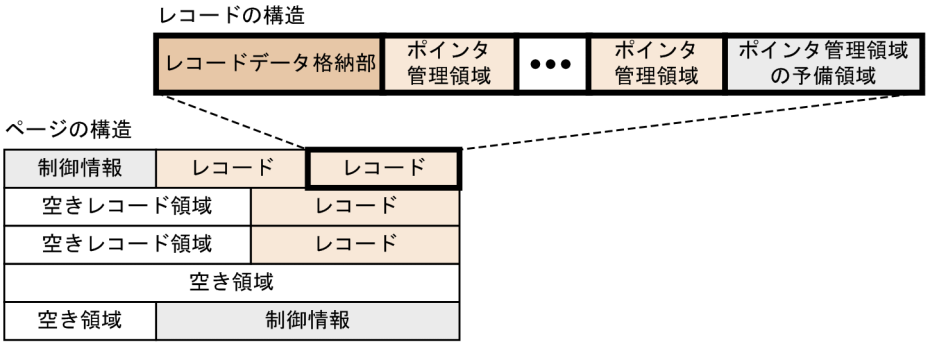
[説明]

- レコード
  レコードのデータを格納する領域です。
- 空きレコード領域
  レコードの削除などによって，ページおよびサブページ内の使用領域の途中にできた未使用領域です。
- 空き領域
  ページおよびサブページ中に格納されている最終レコード以降の未使用領域です。
- 制御情報
  ページおよびサブページを管理するための情報を格納する領域です。

(5)　レコードの構造

レコードの構造を次の図に示します。

図 2-5　レコードの構造



[説明]

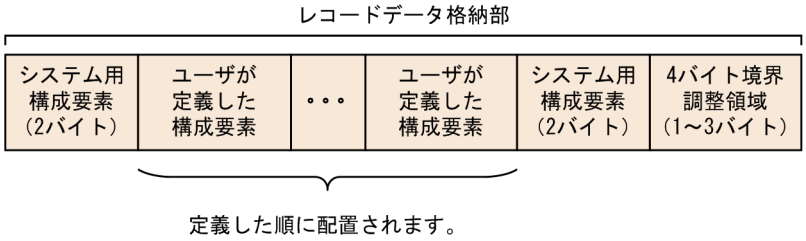
- レコードデータ格納部

  実際のデータを格納する領域です。

  レコードデータ格納部の形式は，仮想ルートレコードとそれ以外のレコードで次のように異なります。

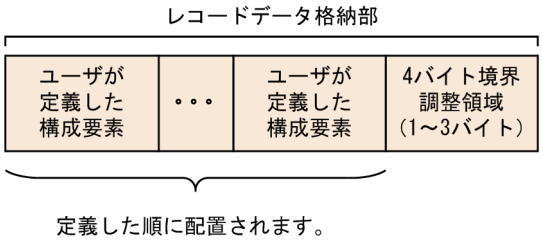
  ●仮想ルートレコードの場合

  SDB データベースの定義時にユーザが指定した構成要素が，定義した順に配置されます。それ以外に，HiRDB/SD が内部的に使用するシステム用構成要素が配置されます。レコードデータ格納部の長さが 4 の倍数でなければ，末尾に境界調整用の領域が付加されます。



  ●仮想ルートレコード以外のレコードの場合

  SDB データベースの定義時にユーザが指定した構成要素が，定義した順に配置されます。レコードデータ格納部の長さが 4 の倍数でなければ，末尾に境界調整用の領域が付加されます。



- ポインタ管理領域

  レコード間をつなぐポインタを管理する領域です。

- ポインタ管理領域の予備領域

将来の親子集合の拡張に備えて，レコード内に設定しておく領域です。ポインタ管理領域の予備領域のサイズは，SDB データベース格納定義の POINTER AREA SIZE 句で指定できます（省略時に仮定される値は 0 です）。

2.3　構造型 DB の仕組み

構造型 DB では，レコードという単位でデータを管理します。また，レコード間の関連は階層構造（親子関係）で管理します。

2.3.1　HiRDB/SD がサポートしている構造型 DB の種類

HiRDB/SD がサポートしている構造型 DB には，階層構造を持つデータベースと，階層構造を持たないデータベースの 2 種類があります。それぞれのデータベースについて説明します。

■階層構造を持つデータベース（FMB のデータベース）

階層構造を持つデータベースを，FMB のデータベースといいます。

FMB のデータベースの場合，レコード間の関連が親子関係で管理されます。下位のレコード（子レコード）にアクセスする場合は，上位のレコード（親レコード）から階層順にアクセスします。FMB のデータベースでは，1 つのレコード型は，複数の構成要素から構成されます。レコード型および構成要素については，「2.3.4　レコード」を参照してください。

参考

FMB のデータベースは，次のデータベースに相当します。

- メインフレームの TMS-4V/SP がサポートしている FMB のデータベース
- メインフレームの XDM/SD がサポートしている階層型モデルのデータベース

■階層構造を持たないデータベース（AFM のデータベース）

階層構造を持たないデータベースを，AFM のデータベースといいます。

AFM のデータベースの場合，1 つのレコード型は，1 つまたは複数の構成要素から構成されます。レコード型および構成要素については，「2.3.4　レコード」を参照してください。

参考

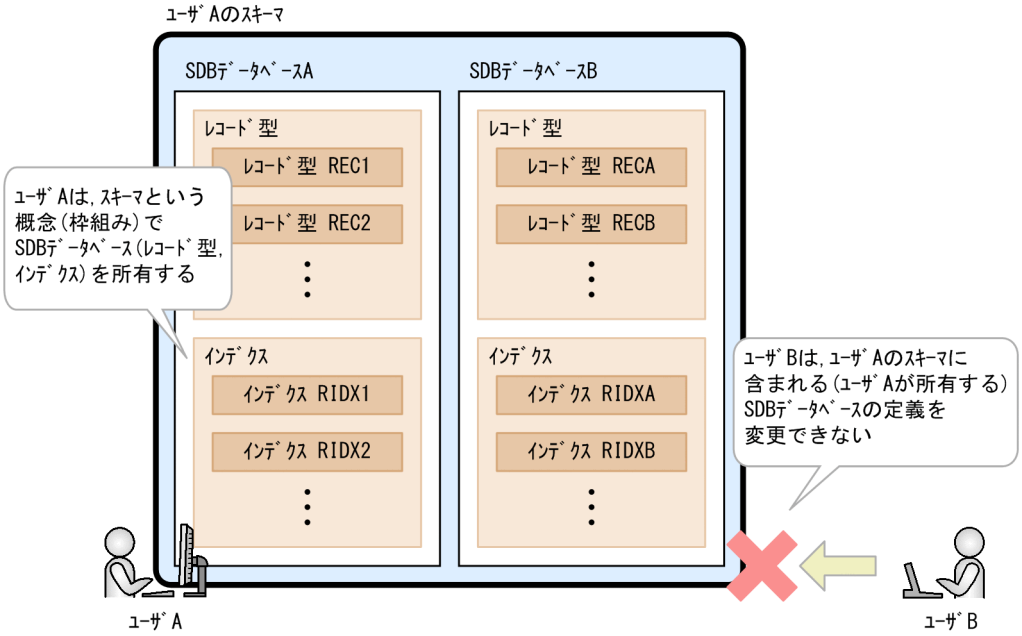
AFM のデータベースは，メインフレームの TMS-4V/SP がサポートしている DAM，MAM，TAM，または SAM のデータベースに相当します。

2.3.2　スキーマ

HiRDB/SD にはスキーマという概念があります。スキーマとは，SDB データベース，および SDB データベースを構成するレコード型およびインデクスなど，ユーザが所有するリソースを包括した概念です。

ユーザは，SDB データベースを定義する前にスキーマを定義しておく必要があります。スキーマは定義系 SQL の CREATE SCHEMA で定義します。1 ユーザには 1 スキーマを定義できます。スキーマの概念を次の図に示します。

図 2-6　スキーマの概念



例えば，ユーザ A のスキーマに含まれる SDB データベース A は，ユーザ A の所有となります。そのため，ユーザ B はユーザ A の SDB データベース A の定義内容を変更することはできません。

2.3.3　SDB データベース

SDB データベースとは，レコード，インデクス，親子集合など構造型 DB を構成するデータ群の総称です。HiRDB/SD では SDB データベース単位にレコードやインデクスを定義したり削除したりします。

SDB データベースは，HiRDB/SD 定義ユティリティ（pdsdbdef）で定義します。HiRDB/SD 定義ユティリティ（pdsdbdef）については，「11.　HiRDB/SD 定義ユティリティ（pdsdbdef）」を参照してください。また，SDB データベースの定義の概要については，「2.4　SDB データベースの定義」を参照してください。

(1)　SDB データベース種別

SDB データベースには，次の表に示す種類（SDB データベース種別）があります。

表 2-3　SDB データベースの種類（SDB データベース種別）

SDB データベース種別	説明
4V FMB	階層構造を持つデータベースで，メインフレームの TMS-4V/SP がサポートしている FMB のデータベースに相当します。

SDB データベース種別	説明
4V AFM	階層構造を持たないデータベースで，メインフレームの TMS-4V/SP がサポートしている DAM，MAM，TAM，または SAM のデータベースに相当します。 4V AFM の SDB データベースは，さらに次の SDB データベース種別に分けられます（4V AFM は次の SDB データベース種別の総称です）。 • 4V DAM • 4V MAM • 4V TAM • 4V SAM
SD FMB	階層構造を持つデータベースで，メインフレームの XDM/SD がサポートしている階層型モデルのデータベースに相当します。

SDB データベースの定義時に SDB データベース種別を指定します。

なお，SDB データベース種別によって，HiRDB/SD の機能のうち，使用できる機能が異なります。詳細については，「付録 P　4V FMB または 4V AFM の SDB データベースと SD FMB の SDB データベースの機能差一覧」を参照してください。

注意事項

SDB データベース種別が 4V（4V FMB または 4V AFM）の SDB データベースと SD FMB の SDB データベースをシステム内で混在して定義することはできません。

4V FMB と 4V AFM の SDB データベースをシステム内で混在して定義することはできますが，4V FMB と SD FMB（または 4V AFM と SD FMB）の SDB データベースをシステム内で混在して定義することはできません。

(2)　SDB データベース種別の表記に関する留意事項

このマニュアルでは，基本的に「表 2-3　SDB データベースの種類（SDB データベース種別）」に示すすべての種類の SDB データベースを対象に説明しています。ただし，使用する SDB データベースの種類によっては，使用できない機能があります。その場合には，マニュアル中の章，節，項などのタイトルに，その機能の対象となる SDB データベース種別を示す表記をしています。

（例）

「2.3.10　二次インデクス【4V FMB，4V AFM】」

上記のように表記されている場合，4V FMB または 4V AFM の SDB データベースを使用しているときに，該当する記述であることを意味しています。そのため，SD FMB の SDB データベースを使用している場合は，「2.3.10　二次インデクス」の説明をお読みいただく必要はありません。

SDB データベース種別を示す表記については，「はじめに」の「このマニュアルをお読みいただく際の留意事項」を参照してください。

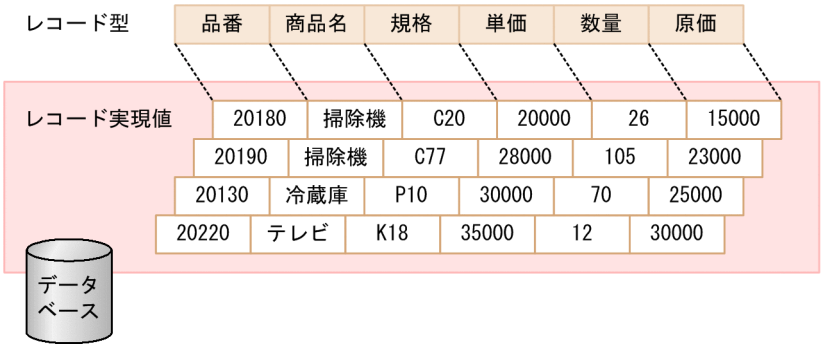
2.3.4　レコード

構造型 DB では，データベース中に存在する情報の基本単位をレコードといいます。レコードは情報の論理的な単位であると同時に，データを検索したり，更新したりするときの単位でもあります。

構造型 DB では，あるレコードをモデル化したものをレコード型といいます。レコード型は SDB データベース定義で定義します。

レコード型に対応してデータベース中で，実際のデータ値を持って存在するものをレコード実現値といいます。レコード型とレコード実現値の関係を次の図に示します。

図 2-7　レコード型とレコード実現値の関係



データベース中の「品番」，「商品名」，「規格」などの各情報をこの順番に持つとき，これをモデル化したものがレコード型であり，実際にデータ値を持つ個々のものをレコード実現値といいます。

なお，このマニュアルでは，特に誤解を与えない場合にはレコード実現値を単にレコードといいます。

レコードは次の要素から構成されます。

(1)　データ項目

データ項目は，データベース中で取り扱える情報の最小単位です。1 つのデータ項目はただ 1 つの値を持ちます。データ項目は格納するデータ種別ごとに属性を持ちます。これをデータ型といいます。

HiRDB/SD で指定できるデータ型を次の表に示します。

表 2-4　HiRDB/SD で指定できるデータ型

データ種別	データ型
文字列データ	CHARACTER
16 進データ	XCHARACTER
符号付きパック形式 10 進データ※1	PACKED DECIMAL FIXED
整数データ	INTEGER
	SMALLINT※2

注※1

符号付きパック形式の符号部の仕様を次に示します。

符号部	説明	
	正規化しない場合	正規化する場合（DECIMAL 型の符号正規化機能を使用する場合）
X'A'	エラーになります。	正の値を意味する X'C'に変換されます。
X'B'	エラーになります。	負の値を意味する X'D'に変換されます。
X'C'	正の値を意味します。	
X'D'	負の値を意味します。	
X'E'	エラーになります。	正の値を意味する X'C'に変換されます。
X'F'	正の値を意味します。	
X'0'～X'9'	エラーになります。	

DECIMAL 型の符号正規化機能を使用する場合は，システム共通定義の pd_dec_sign_normalize オペランドに Y を指定します。詳細については，マニュアル「HiRDB システム運用ガイド」の「DECIMAL 型の符号部を変換する方法」を参照してください。

注※2

SMALLINT のデータ型は，SD FMB の SDB データベースの場合に使用できます。

(2) 構成要素

データ項目を論理的な単位にまとめたものを構成要素といいます。構成要素には基本項目と集団項目の 2 種類があります。

(a) 基本項目

1 つのデータ項目から成る構成要素です。

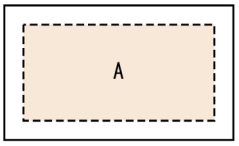
(b) 集団項目

階層構造を持ち，基本項目を含む構成要素です。関連する 2 つ以上のデータ項目をまとめて 1 つの構成要素となります。

(c) 構成要素の例

構成要素の例を次の図に示します。

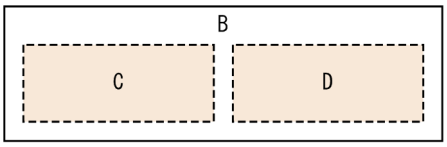
図 2-8 1 つのデータ項目から成る構成要素



[説明]

1 つのデータ項目から成る構成要素の例です。A は基本項目となります。

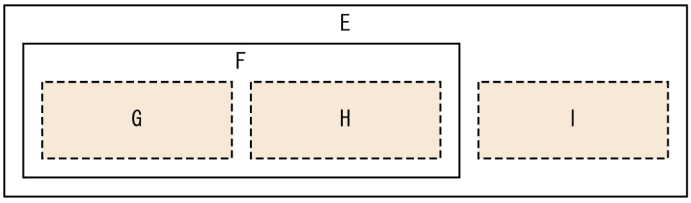
図 2-9　複数のデータ項目から成る構成要素（集団項目が集団項目を含まない場合）



[説明]

複数のデータ項目から成る構成要素の例です。B は集団項目であり，C および D はそれぞれ基本項目となります。

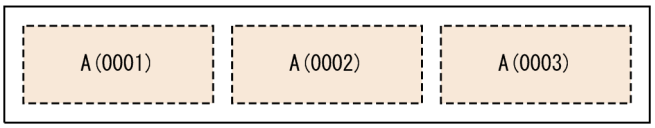
図 2-10　複数のデータ項目から成る構成要素（集団項目が集団項目を含む場合）



[説明]

複数のデータ項目から成る構成要素の例です。E および F は集団項目であり，G，H，および I はそれぞれ基本項目となります。

図 2-11　繰り返し指定をしたデータ項目から成る構成要素（OCCURS 句に繰り返し回数 3 を指定している場合）



[説明]

繰り返し指定をしたデータ項目から成る構成要素の例です。A(0001)，A(0002)，および A(0003)は，それぞれ基本項目となります。この場合，構成要素数は，SDB データベース定義の OCCURS 句で指定した繰り返し回数と同じになります。

(d)　基本項目の構成要素のデータ種別

基本項目の構成要素は，用途によって 3 種類に分けられます。基本項目の構成要素のデータ種別を次の表に示します。

表 2-5　基本項目の構成要素のデータ種別

データ種別	説明
データベースキー	HiRDB/SD が，SDB データベース内でレコードを一意に識別するために使用する構成要素です。

データ種別	説明
ユーザデータ	ユーザが自由に設定できる構成要素です。
ユーザキー	ユーザデータのうち，レコードを一意に識別するために使用する構成要素です。

(3) キー

SDB データベースでは，SDB データベース内でレコードを一意に識別するために，レコードごとにキーが設定されます。キーには，HiRDB/SD が管理するキー（データベースキー）とユーザが管理するキー（ユーザキー）があります。

- **データベースキー**

  HiRDB/SD が管理するキーです。キー値にはユーザが明示的に指定するキーと，HiRDB/SD が自動的に採番するキー（一連番号）があります。

- **ユーザキー**

  ユーザが管理するキーです。キー値はレコードの追加時にユーザが指定します。ユーザキーはレコード型ごとに 1 つ定義できます。

  ユーザキーは，4V FMB または 4V AFM の SDB データベースの場合に定義できます。

- **論理キー**

  データベースキーを構成する個々の構成要素に対して，キーの属性を設定できます。これを論理キーと呼びます。論理キーの種類を次の表に示します。

表 2-6　論理キーの種類

論理キーの種類	説明
レコード型の RD エリア分割キー	レコード型の RD エリア分割に使用するキーです。
レコード分割キー	レコード分割に使用するキーです。
レコード型の RD エリア分割キーでありレコード分割キーであるキー	レコード型の RD エリア分割およびレコード分割に使用するキーです。
上記以外のキー	ユーザが任意に使用するキーです。
一連番号	HiRDB/SD がレコードの発生順に，自動的に採番して管理する番号（キー）です。

2.3.5　親子集合

構造型 DB では，各レコードは互いに関連を持ちます。レコード間の論理的な関連は親子の関係で表現されます。これを親子集合といいます。

ルートレコードを除く各レコードは，親レコードを 1 つだけ持ち，子レコードを複数個持つことができます。

(1)　親レコード

親子集合では，親に当たるレコードを親レコードといいます。親レコードのレコード型を親レコード型といいます。

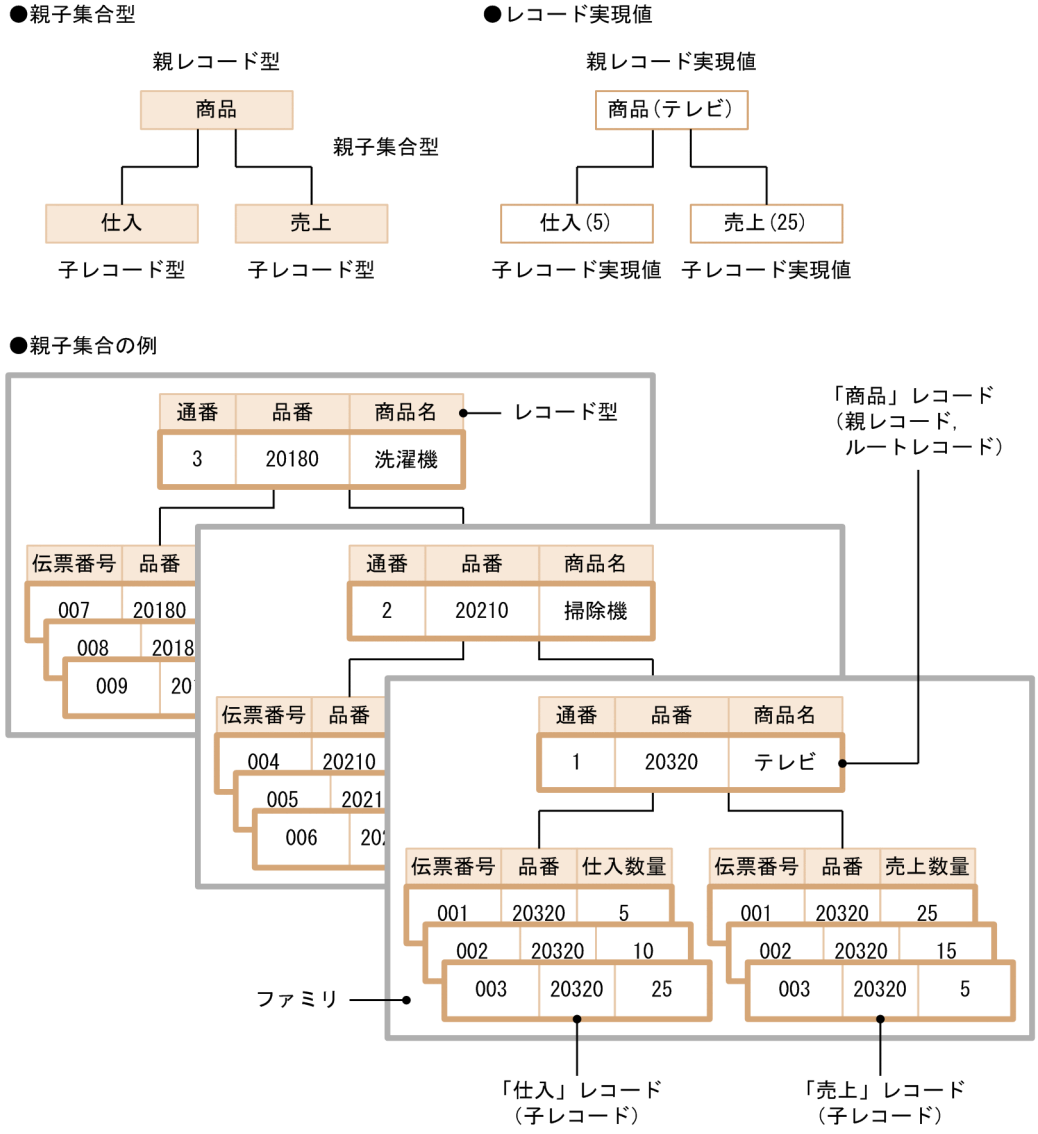
(2)　子レコード

親子集合では，子に当たるレコードを子レコードといいます。子レコードのレコード型を子レコード型といいます。

(3)　親子集合型

親レコード型と子レコード型をモデル化したものを親子集合型といいます。親子集合型は SDB データベース定義で定義します。親子集合型の例を次の図に示します。

図 2-12　親子集合型の例



この例の場合，「商品」レコードの商品（品番）に対応して，「売上」レコードと「仕入」レコードが存在します。そのため，「商品」レコードを親レコード，「売上」レコード，「仕入」レコードをそれぞれ子レコードとする，親子集合ができます。

(4)　ルートレコード

階層構造の最上位のレコードをルートレコードといいます。ルートレコードのレコード型をルートレコード型といいます。

ルートレコードは，親レコードを持つことはできません。子レコードを複数個持つことができます。

(5)　ファミリ

ルートレコード，およびそのルートレコードを親レコードとする子レコード（さらにその子レコードを親レコードとする子レコード）群をまとめてファミリといいます。

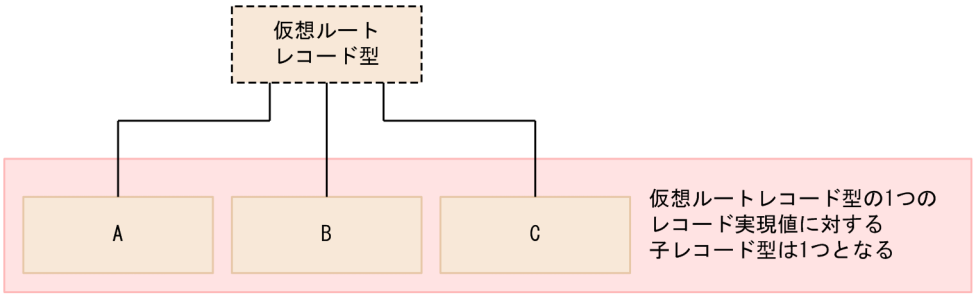
(6)　仮想ルートレコードと仮想親子集合【4V AFM】

親子集合を構成しない（階層構造を持たない）データベースを構築する場合，HiRDB/SD では仮想のルートレコード（仮想ルートレコード）を親レコードとして設定し，仮想的な親子集合を構成します。これを仮想親子集合といいます。

TMS-4V/SP の AFM のデータベースは親子集合を持ちません。そのため，HiRDB/SD の 4V AFM の SDB データベースに移行する場合，仮想親子集合を構成します。

仮想親子集合の例を次の図に示します。

図 2-13　仮想親子集合の例



仮想ルートレコード型のレコード実現値は，通常のレコード実現値とは異なり，1 つの子レコード型とだけ親子集合を構成します。つまり，仮想ルートレコード型の 1 つのレコード実現値に対する子レコード実現値は A，B，C のどれかの子レコード型にしか存在しません。

2.3.6　ポインタによる親子集合の実現

親子集合で表現したレコード間の論理的な親子関係を，データベース中で関係づけます。これをポインタといいます。ポインタは SDB データベース格納定義で定義できます。

次に示すポインタを親子集合を構成するレコード間に設定できます。ポインタの種類を次の表に示します。

表 2-7　ポインタの種類

ポインタの場所	種類	説明
親レコード	FIRST ポインタ	親レコードが子レコードを持つ場合，先頭の子レコードの位置情報を親レコードが管理するために FIRST ポインタを持ちます。 FIRST ポインタは，親レコードに関係づけられた子レコードのレコード型ごとに設定されます。
	LAST ポインタ	親レコードが子レコードを持つ場合，最終の子レコードの位置情報を親レコードが管理するために LAST ポインタを持ちます。 LAST ポインタは，親レコードに関係づけられた子レコードのレコード型ごとに設定されます。
	USER ポインタ	親レコードが子レコードを持つ場合，任意の子レコードの位置情報を子レコードごとに 1 つだけ持つことができます。これを USER ポインタといいます。 USER ポインタは，UAP の指示によって，設定・解除できます。USER ポインタが設定されている場合は，UAP の指示によって，直接該当するレコードにアクセスできます。 4V FMB の SDB データベースの場合に，USER ポインタを設定できます。
子レコード	NEXT ポインタ	同一レコード型の子レコードは，1 つの親レコードに関連づけられたレコードの範囲内で，次のレコード（次の一連番号またはユーザキーを持つレコード）の位置情報を管理するために，NEXT ポインタを持ちます。 最終レコードの NEXT ポインタは親レコードを指します。
	PRIOR ポインタ	同一レコード型の子レコードは，1 つの親レコードに関連づけられたレコードの範囲内で，直前のレコード（1 つ前の一連番号またはユーザキーを持つレコード）の位置情報を管理するために，PRIOR ポインタを持ちます。 先頭レコードの PRIOR ポインタは親レコードを指します。
	OWNER ポインタ	子レコードに親レコードの位置情報を保持する場合，OWNER ポインタを SDB データベース格納定義で定義します。OWNER ポインタは SD FMB の SDB データベースの場合に定義できます。 子レコードに二次インデクスを定義するときは，OWNER ポインタの定義が必要です。 OWNER ポインタは子レコードごとに 1 つだけ持つことができます。

レコードとポインタの関係（OWNER ポインタ以外の場合）を「図 2-14　レコードとポインタの関係（OWNER ポインタ以外の場合）」に，レコードとポインタの関係（OWNER ポインタの場合）を「図 2-15　レコードとポインタの関係（OWNER ポインタの場合）」に示します。

図 2-14　レコードとポインタの関係（OWNER ポインタ以外の場合）

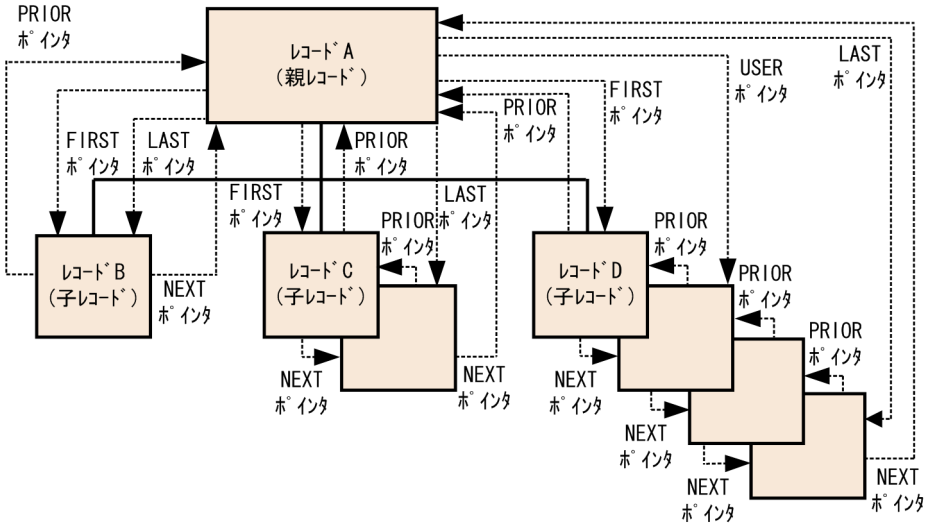
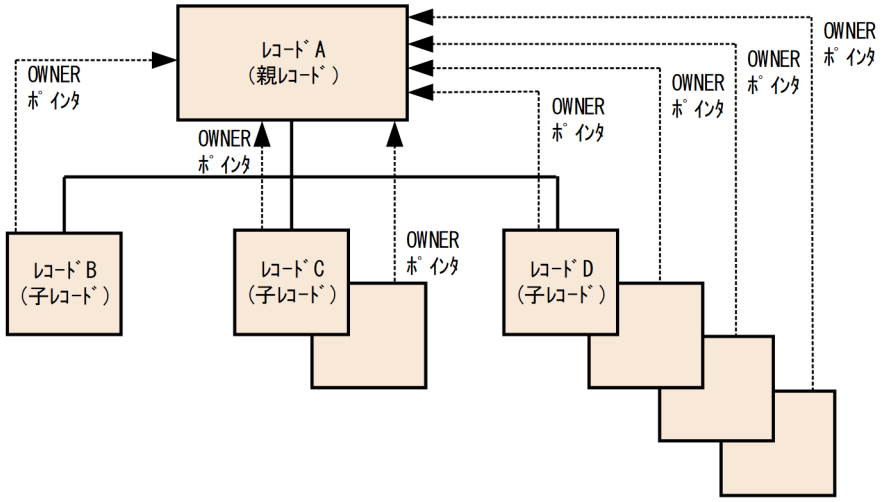


図 2-15　レコードとポインタの関係（OWNER ポインタの場合）



2.3.7　SDB データベースの横分割

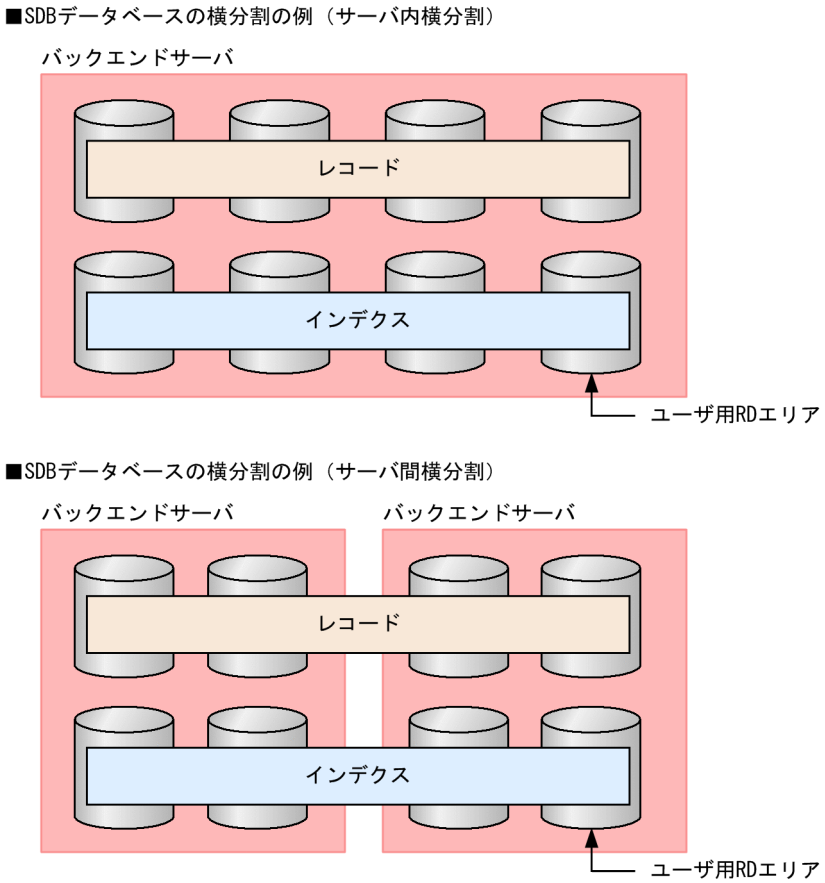
SDB データベースを複数の RD エリアに分割して格納することを SDB データベースの横分割といいます。SDB データベースの横分割をすると，ユーザ用 RD エリア単位に，データロード，再編成，バックアップの取得などの運用ができます。

例えば，SDB データベースを横分割して RD エリアに格納すると，バックアップの取得時にはバックアップ対象 RD エリアにアクセスする UAP だけを停止すればよく，運用の操作性が向上します。

なお，横分割したレコードとインデクスを格納する RD エリアを同じバックエンドサーバに配置する必要があります。

SDB データベースの横分割は，SDB データベース格納定義で定義できます。SDB データベースの横分割の例を次の図に示します。

図 2-16　SDB データベースの横分割の例



[説明]

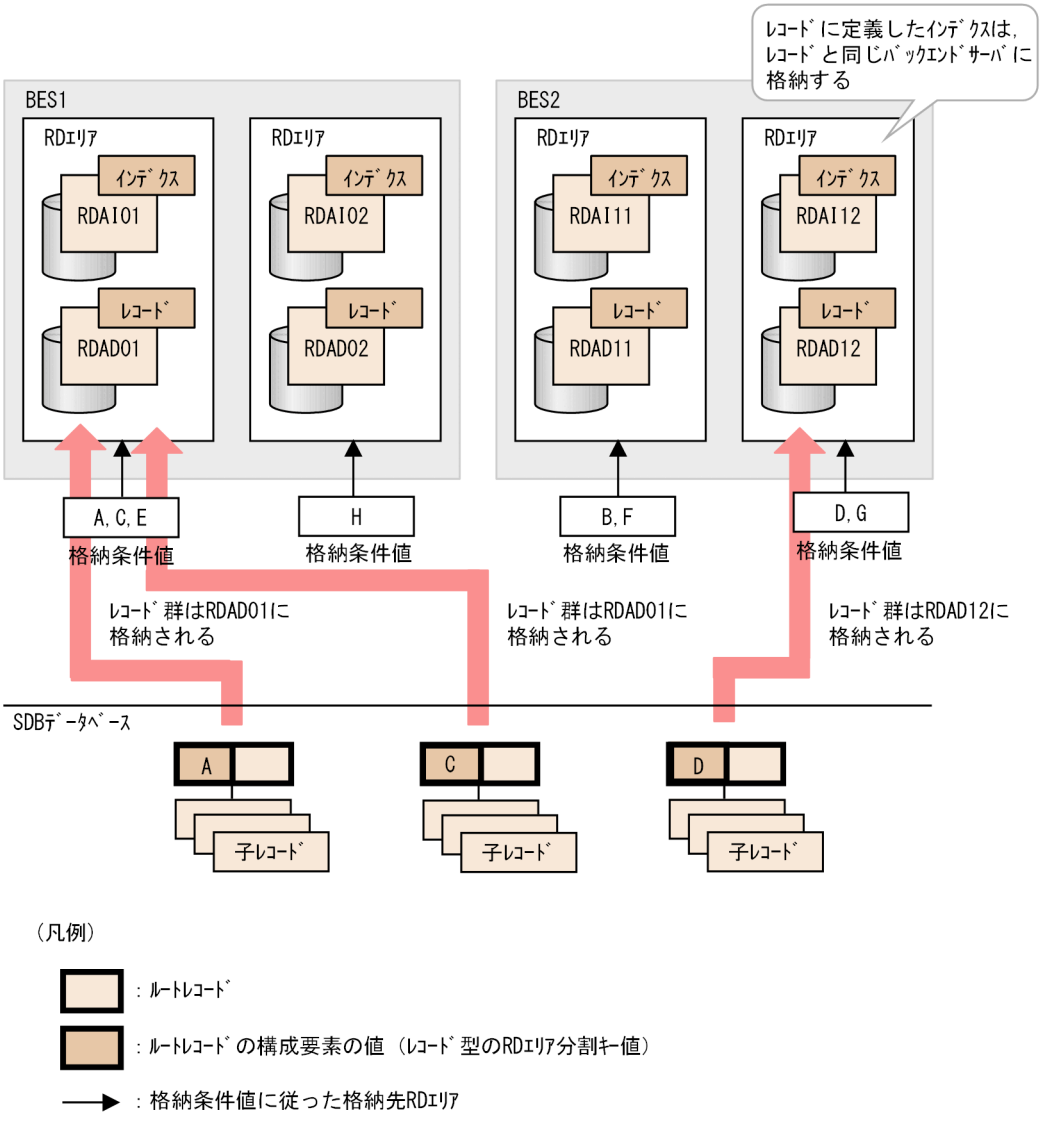
- レコードとインデクスの横分割数は同じにする必要があります。
- 1 つのバックエンドサーバ内で SDB データベースを横分割する形態をサーバ内横分割といい，複数のバックエンドサーバにわたって SDB データベースを横分割する形態をサーバ間横分割といいます。

SDB データベースを横分割する方法には，格納条件を指定して横分割する方法と境界値を指定して横分割する方法があります。それぞれの方法について説明します。

(1)　格納条件指定による横分割

RD エリアにデータを格納する際，ルートレコードの構成要素の値を格納条件（レコード型の RD エリア分割キー値）として指定します。HiRDB/SD は，ルートレコードを格納するときに，指定された格納条件に従って，該当する RD エリアにレコードを格納します。ルートレコードを親レコードとする子レコード群は，対応するルートレコードと同じ RD エリアに格納されます。

図 2-17　格納条件指定の例

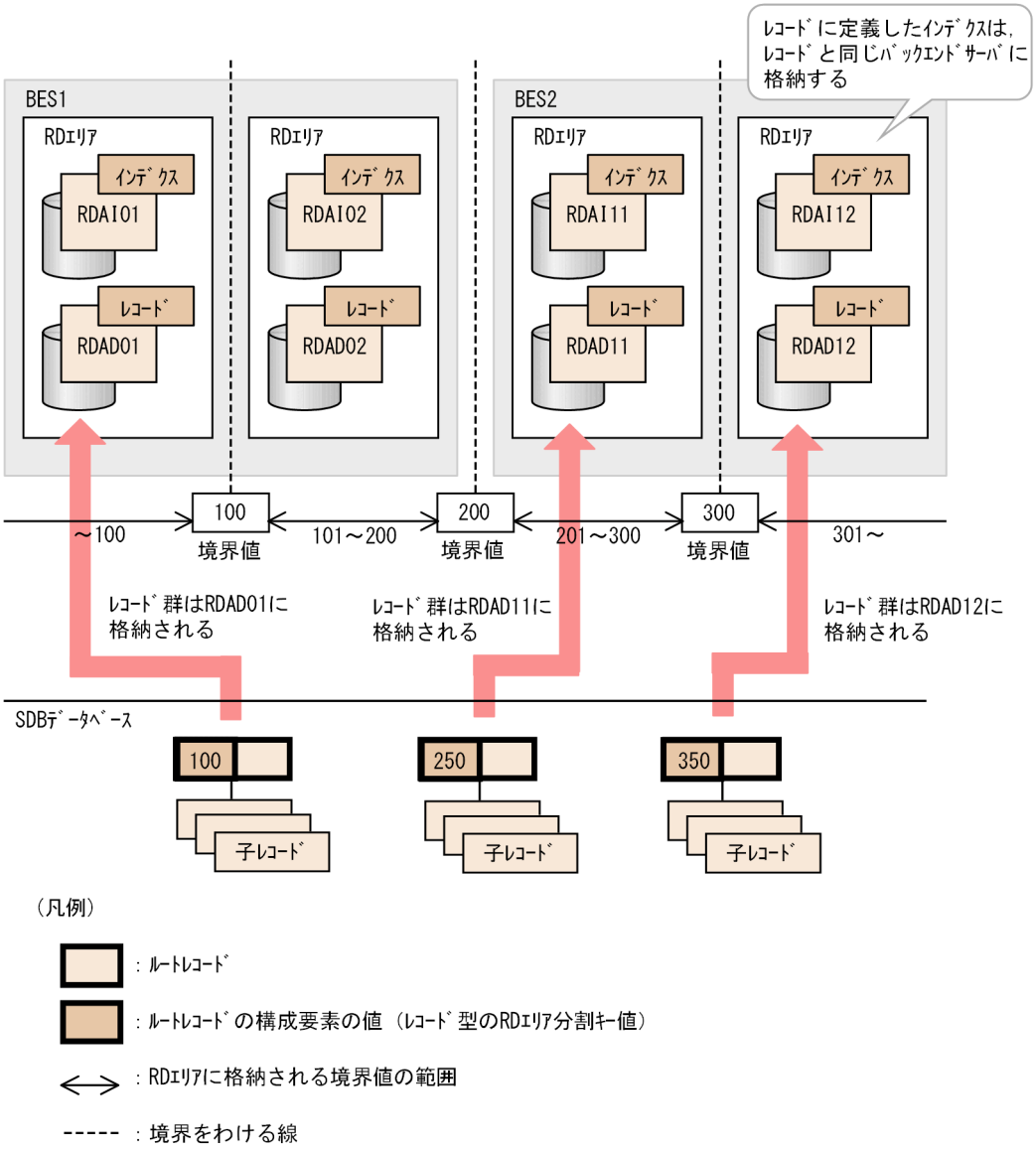


(2)　境界値指定による横分割【4V FMB，SD FMB】

RD エリアにデータを格納する際，ルートレコードの構成要素の値に対して境界値となる値（レコード型の RD エリア分割キー値）を指定します。HiRDB/SD は，ルートレコードを格納するときに，指定された境界値に従って，該当する RD エリアにレコードを格納します。ルートレコードを親レコードとする子レコード群は，対応するルートレコードと同じ RD エリアに格納されます。

境界値指定による横分割は，4V FMB または SD FMB の SDB データベースの場合に使用できます。

図 2-18　境界値指定の例



(3)　分割格納条件の変更機能【4V FMB，4V AFM】

SDB データベースの横分割数の変更が必要となった場合は，分割格納条件の変更機能で横分割数を変更できます。

分割格納条件を変更するには，HiRDB/SD 定義ユティリティ（pdsdbdef）で*ALTER DICTIONARY 文および*ALTER DIRECTORY 文を実行します。

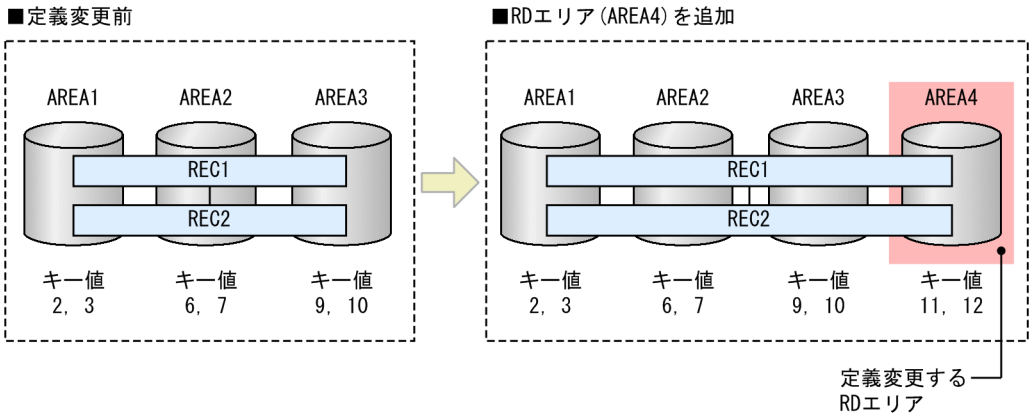
分割格納条件の変更の種類は，追加，分割，削除，統合の 4 種類があります。それぞれについて説明します。

なお，SD FMB の SDB データベースの場合，*ALTER DICTIONARY 文および*ALTER DIRECTORY 文を実行できないため，分割格納条件の変更機能で横分割数を変更することはできません。

(a) 追加

新たな分割キー値を格納条件とするRDエリアを追加することです。

■格納条件指定



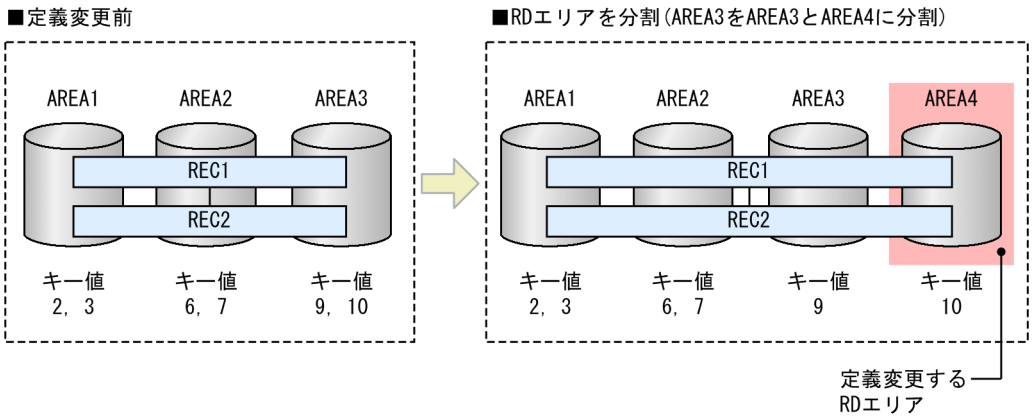
■境界値指定

該当しません。

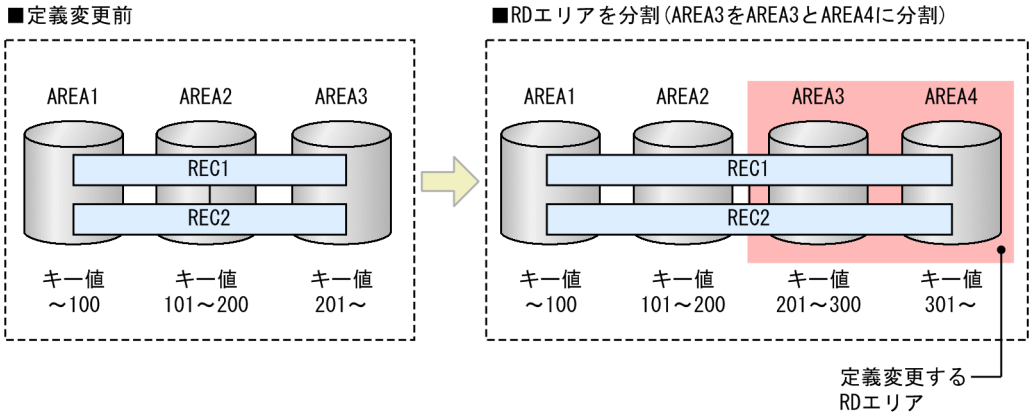
(b) 分割

使用中の分割キー値を，既存のRDエリアと追加するRDエリアの両方で格納条件として使用する格納条件の変更です。

■格納条件指定



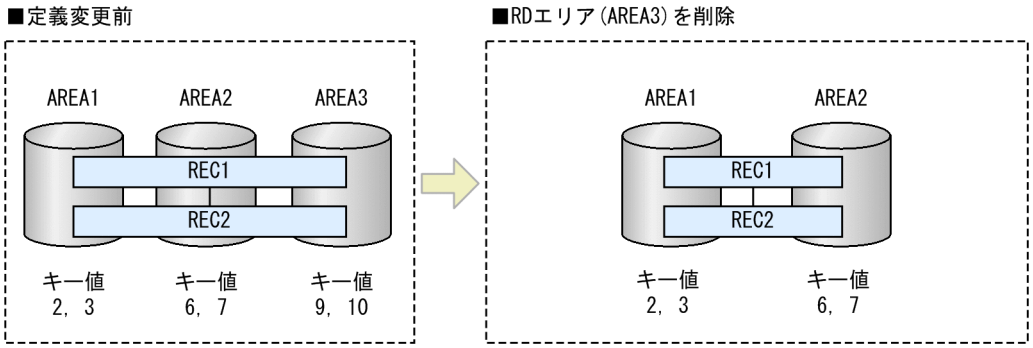
■境界値指定



(c) 削除

使用中の分割キー値を削除し，さらに削除したキー値を格納する RD エリア自体も削除する格納条件の変更です。

■格納条件指定



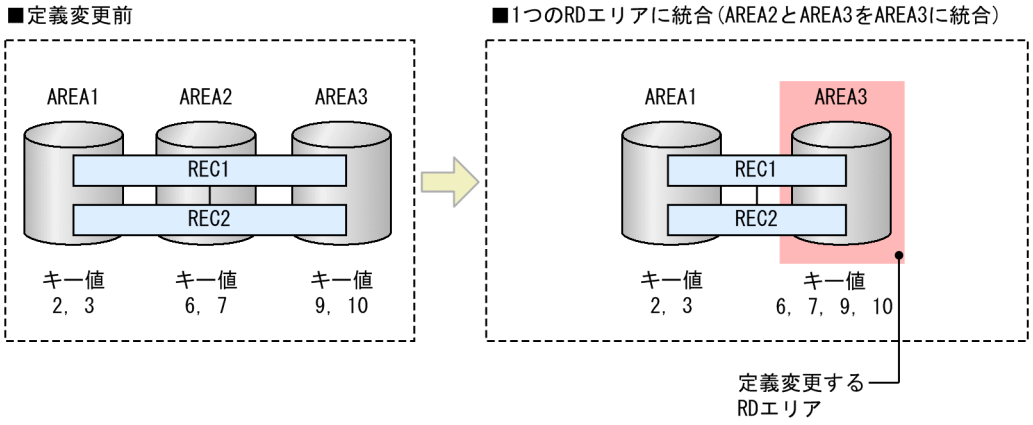
■境界値指定

該当しません。

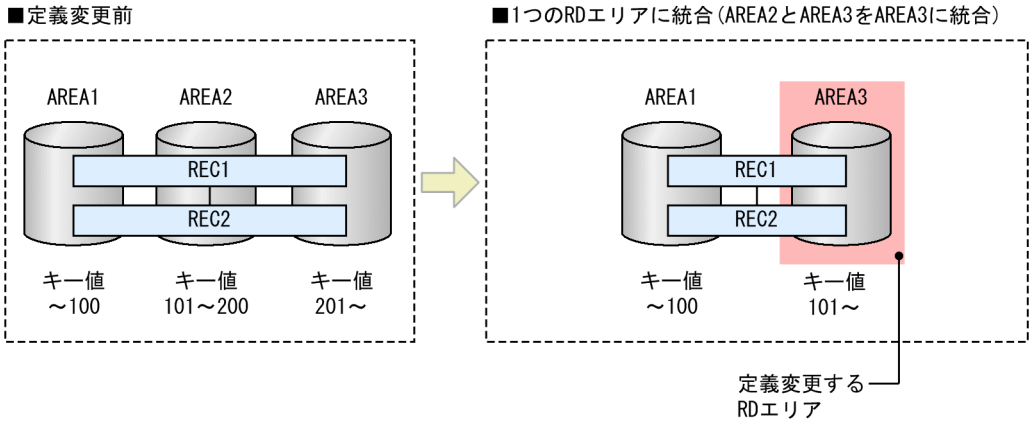
(d) 統合

使用中の分割キー値をほかの RD エリアの格納条件とし，それまでそのキー値を格納していた RD エリアを削除する格納条件の変更です。

■格納条件指定



■境界値指定



2.3.8　インデクスの概要

HiRDB/SD では，レコード実現値中の特定の構成要素の値に従ってデータを処理するために，該当する構成要素の値と，その値を持つレコード実現値の格納位置を管理します。この管理情報をインデクスと呼びます。インデクスを使用することで効率良くデータを検索できます。

HiRDB/SD が提供するインデクスには，シーケンシャルインデクスと二次インデクスがあります。それぞれのインデクスの概要を次の表に示します。

表 2-8　インデクスの概要

インデクス種別	用途	レコード型ごとの定義可否					
		4V FMB の場合		4V AFM の場合		SD FMB の場合	
		ルートレコード型	子レコード型	仮想ルートレコード型	子レコード型	ルートレコード型	子レコード型
シーケンシャルインデクス	ルートレコード，または仮想ルートレコードの検索※に使用します。	◎	×	◎	×	◎	×
二次インデクス（4V FMB または 4V AFM の SDB データベースの場合）	子レコードの検索や，子レコードを格納する際のレコード挿入位置の検索に使用します。	×	○	×	△	−	−
二次インデクス（SD FMB の SDB データベースの場合）	レコード型内の検索に使用します。	−	−	−	−	○	○

（凡例）

◎：必須で定義します。

○：任意で定義できます。

△：SDB データベース種別やオペランドの指定によって定義有無が異なります。詳細については，「表 11-62　二次インデクスの定義規則」を参照してください。

×：定義できません。

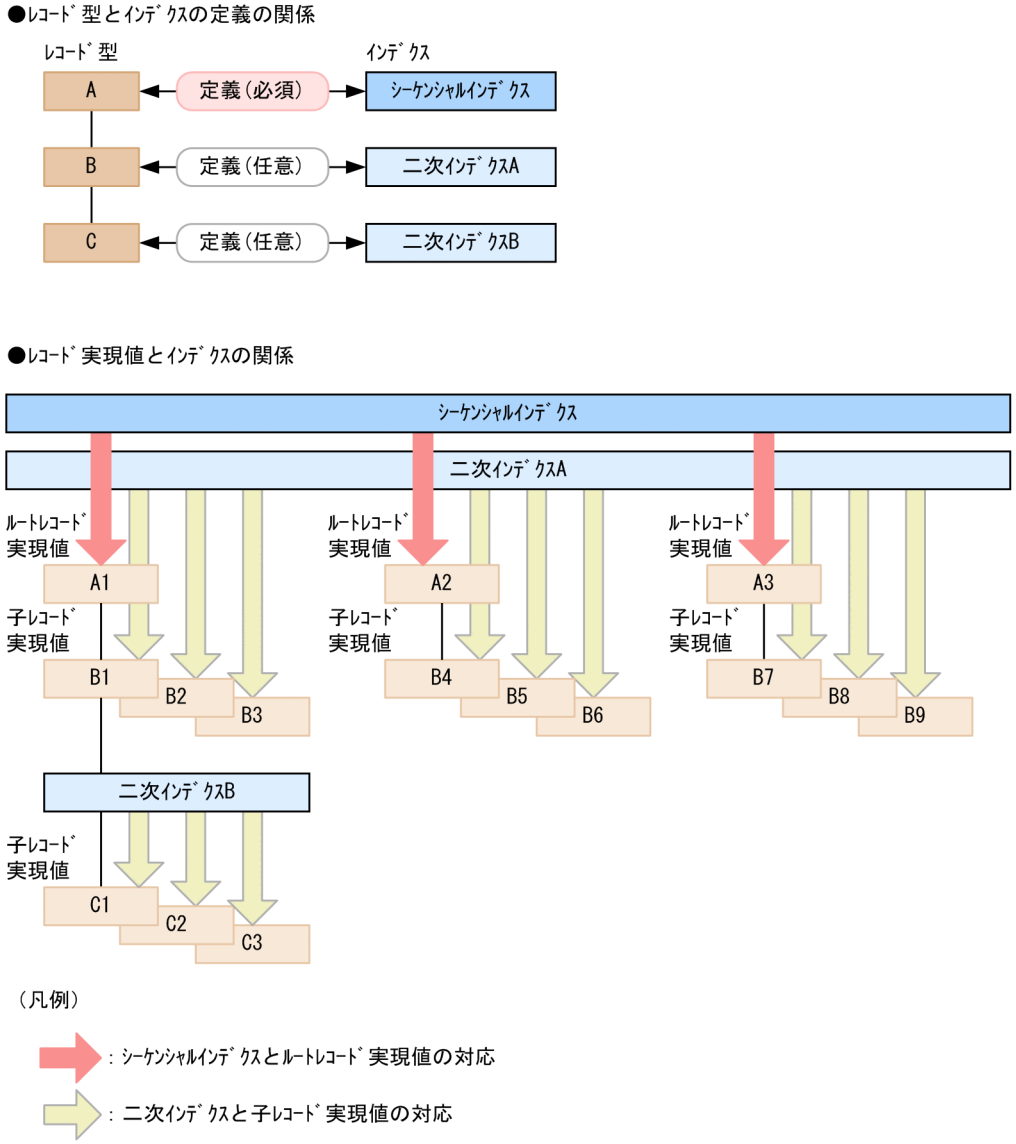
−：該当しません。

注※

4V AFM の SDB データベースを検索する場合，シーケンシャルインデクスを使用して，HiRDB/SD が内部的に仮想ルートレコードを検索します。

シーケンシャルインデクスおよび二次インデクスとレコード実現値の対応の概要を，4V FMB の SDB データベースの場合を例に次の図に示します。

図 2-19　シーケンシャルインデクスおよび二次インデクスとレコード実現値の対応の概要（4V FMB の SDB データベースの場合）



2.3.9　シーケンシャルインデクス

シーケンシャルインデクスは，ルートレコード，または仮想ルートレコードの検索に使用するインデクスです。次の構成要素に必ず定義します。

- 4V FMB または SD FMB の SDB データベースの場合
  ルートレコード型の先頭の集団項目に定義したすべての基本項目の構成要素
- 4V AFM の SDB データベースの場合
  仮想ルートレコード型に定義したすべての基本項目の構成要素

シーケンシャルインデクスは指定した構成要素の値をキーとして，キーの昇順に作成されます。シーケンシャルインデクスは SDB データベース格納定義の SEQUENTIAL 句で定義します。

なお，仮想ルートレコード型に定義したシーケンシャルインデクスの場合，ほかのインデクス（4V FMB または SD FMB の SDB データベースのルートレコード型に定義したシーケンシャルインデクス，または二次インデクス）とは次の点が異なります。

■仮想ルートレコード型に定義したシーケンシャルインデクスの場合

インデクスキーとレコード実現値の関係は，配下に有効な子レコード（一括削除後に無効な状態になる子レコードを除く）が存在する場合は 2 対 1 となり，配下に有効な子レコードが存在しない場合は 1 対 1 となります。

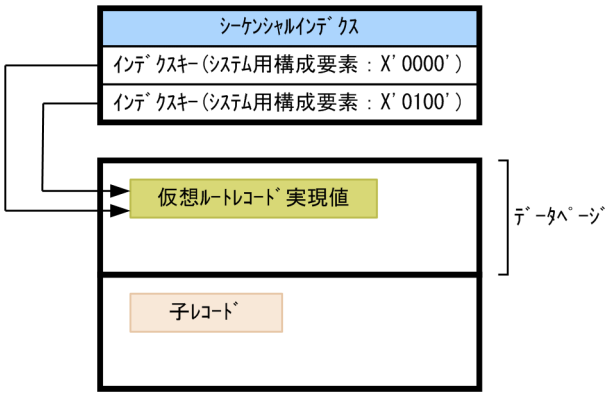
- **配下に有効な子レコードが存在する場合**

  2 つのインデクスキー（システム用構成要素が X'0000'のインデクスキー，およびシステム用構成要素が X'0100'のインデクスキー）から仮想ルートレコード実現値に直接アクセスできる構造になります。

  なお，システム用構成要素が X'0000'のインデクスキーは，ほかのインデクスと同様に，SDB データベースを操作する API の実行時に仮想ルートレコードを検索するために使用されます。これに対し，システム用構成要素が X'0100'のインデクスキーは，HiRDB/SD が仮想ルートレコードの存在有無を判別するために使用されます。

  仮想ルートレコード型に定義したシーケンシャルインデクスの物理構造（配下に有効な子レコードが存在する場合）を次の図に示します。

  図 2-20　仮想ルートレコード型に定義したシーケンシャルインデクスの物理構造（配下に有効な子レコードが存在する場合）

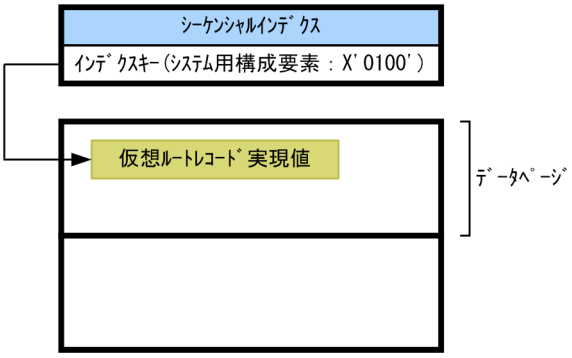


- **配下に有効な子レコードが存在しない場合**

  1 つのインデクスキー（システム用構成要素が X'0100'のインデクスキー）から，仮想ルートレコード実現値に直接アクセスできる構造になります。

  仮想ルートレコード型に定義したシーケンシャルインデクスの物理構造（配下に有効な子レコードが存在しない場合）を次の図に示します。

図 2-21　仮想ルートレコード型に定義したシーケンシャルインデクスの物理構造（配下に有効な子レコードが存在しない場合）

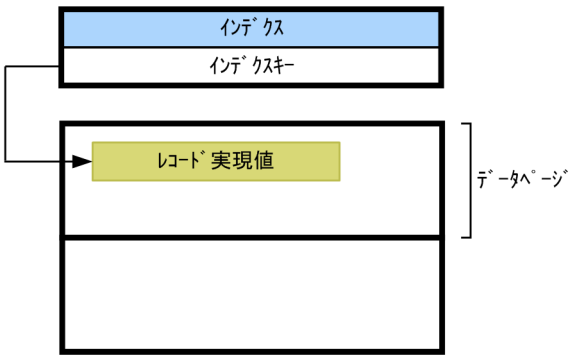


■仮想ルートレコード型に定義したシーケンシャルインデクス以外のインデクスの場合

インデクスキーとレコード実現値の関係は，1 対 1 固定となります。

仮想ルートレコード型に定義したシーケンシャルインデクス以外のインデクスの物理構造を次の図に示します。

図 2-22　仮想ルートレコード型に定義したシーケンシャルインデクス以外のインデクスの物理構造



2.3.10　二次インデクス【4V FMB，4V AFM】

4V FMB および 4V AFM の SDB データベースに定義する二次インデクス（親子集合型内二次インデクス）について説明します。

二次インデクスは，子レコードの検索や，子レコードを格納する際のレコード挿入位置の検索に使用するインデクスです。SDB データベースの子レコードの構成要素（一連番号，またはユーザキー）に定義します。

子レコードの数が多い場合，参照するレコード数が増えるため，NEXT ポインタまたは PRIOR ポインタに従って順アクセスすると検索性能が低下するおそれがあります。

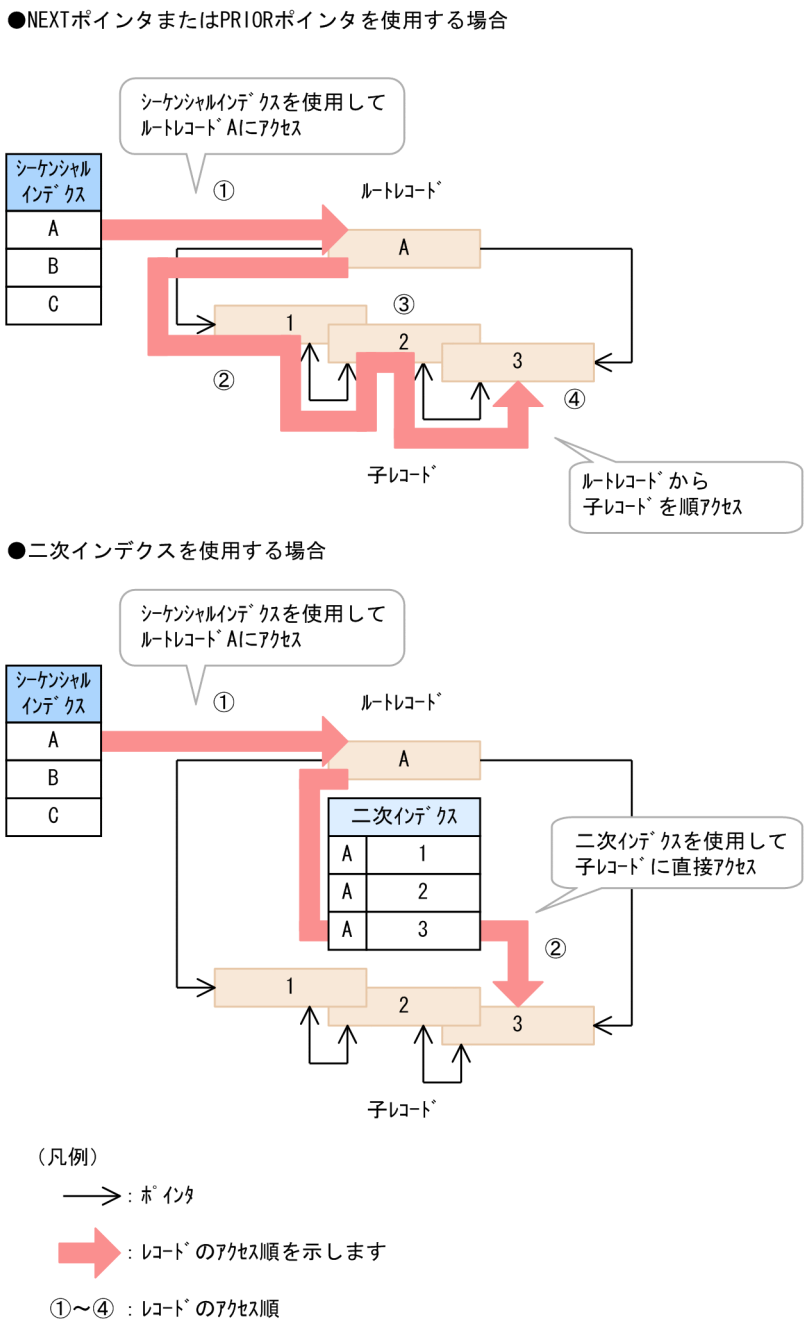
このように，NEXT ポインタまたは PRIOR ポインタを使用した順アクセスによる検索では検索性能に問題がある場合は，二次インデクスを使用すると，特定の子レコードに直接アクセスできるので，検索性能を向上できます。

二次インデクスは SDB データベース格納定義の SECONDARY INDEX 句で定義します。

二次インデクスを使用した場合の検索について，NEXT ポインタまたは PRIOR ポインタを使用して順アクセスで検索する場合との違いを次の図に示します。

なお，ポインタを使用した順アクセスによる検索をポインタ検索といいます。

図 2-23　二次インデクスを使用した検索（NEXT ポインタまたは PRIOR ポインタを使用して順アクセスで検索する場合との違い）



二次インデクスが定義された子レコードの検索時や格納時に，二次インデクスを使用するかどうかは，HiRDB/SD が決定します。二次インデクスが使用される検索条件については，「(2)　子レコード検索時の

二次インデクスの使用有無」および「(3)　子レコード格納時の二次インデクスの使用有無」を参照してください。

(1)　二次インデクスの効果が期待できるケース

二次インデクスの効果が期待できるケースについて，子レコードの検索時および子レコードの格納時について，それぞれ説明します。

(a)　子レコードの検索時

子レコードの検索時に二次インデクスの効果が期待できるケースについて，SDB データベース種別ごとに説明します。

4V FMB の SDB データベースの場合

子レコード実現値の数が多く，NEXT ポインタまたは PRIOR ポインタを使用した順アクセスによる検索では検索性能に問題がある場合は，二次インデクスを使用することで検索性能を向上できます。

4V AFM の SDB データベースの場合

- 子レコード実現値の数が多く，NEXT ポインタまたは PRIOR ポインタを使用した順アクセスによる検索では検索性能に問題がある場合は，二次インデクスを使用することで検索性能を向上できます。
- NEXT ポインタまたは PRIOR ポインタを使用して順アクセスで検索するときに，検索経路上にほかのトランザクションによって更新された子レコードを格納するページが存在すると，排他処理待ちが発生して同時実行性が低下することがあります。

  更新を含むトランザクションと参照だけのトランザクションを同時に実行するような場合，二次インデクスを使用すると，特定の子レコードに直接アクセスできるので，排他処理待ちの発生を軽減できます。

(b)　子レコードの格納時

4V FMB の SDB データベースの場合，子レコードの格納時にレコードの挿入位置を検索します。この検索に二次インデクスを使用することで，検索性能を向上できます。また，二次インデクスを使用すると，レコード数やレコードの格納位置に依存しないでレコード挿入位置を検索できるので，子レコード格納時の性能が一定になります。子レコード格納時のレコード挿入位置の検索については，「2.5.7(2)　レコードの格納順序」を参照してください。

なお，二次インデクスが定義されていると，子レコード格納時にインデクスも更新されます。

(2)　子レコード検索時の二次インデクスの使用有無

子レコード検索時に二次インデクスを使用するかどうかは，親子集合位置指示子が指すレコードの一連番号またはユーザキーと，検索条件から HiRDB/SD が決定します。

なお，4V FMB の SDB データベースでは，先頭のレコード（FIRST ポインタ）または最後のレコード（LAST ポインタ）を基点とする検索の場合（SDB データベースを操作する API の指示コードが'F'また

は'L'の場合）だけ二次インデクスが使用され，そのほかの検索では二次インデクスは使用されません（NEXT ポインタまたは PRIOR ポインタを使用した順アクセスで検索されます）。

また，4V AFM の SDB データベースでは，ユーザが指定した検索条件以外にも，検索範囲を絞り込める場合に，二次インデクスが使用されることがあります。

親子集合位置指示子については，「2.5.3　位置指示子」を参照してください。また，レコードの検索については，「2.5.4　レコードの検索【4V FMB，4V AFM】」を参照してください。

レコード検索時の二次インデクスの使用有無を次の表に示します。

表 2-9　レコード検索時の二次インデクスの使用有無

項番	SDB データベース種別	指示コード	検索条件※1	検索条件の有効/無効※3	二次インデクスの使用
1	4V FMB	F/L※2	あり	有効	○
2				無効	×
3			なし	−	
4		上記以外	−	−	
5	4V AFM	F/L/N/P	あり	有効	○
6				無効	×
7			なし	−	
8		上記以外	−	−	

（凡例）

F：先頭のレコード（親レコードの FIRST ポインタが指すレコード）を基点に，NEXT ポインタに従って検索します。

N：位置づけされているレコードの次のレコード（NEXT ポインタが指すレコード）を基点に，NEXT ポインタに従って検索します。

L：最後のレコード（親レコードの LAST ポインタが指すレコード）を基点に，PRIOR ポインタに従って検索します。

P：位置づけされているレコードの 1 つ前のレコード（PRIOR ポインタが指すレコード）を基点に，PRIOR ポインタに従って検索します。

○：二次インデクスが使用されます。

×：二次インデクスは使用されません。

−：該当しません。

注※1

二次インデクスの定義に指定した一連番号またはユーザキーを検索条件に指定している場合を「あり」，指定していない場合を「なし」とします。

注※2

親子集合位置指示子が位置づけられていない状態で，指示コード'N'，または'P'を指定して検索した場合も該当します。レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

注※3

指定した検索条件が次に示す（a）～（c）のどれかに該当する場合を「有効」，どれにも該当しない場合を「無効」とします。

次に示す（a）～（c）のどれかに該当する場合，二次インデクスが使用されます。

(a) '='条件が指定されている場合

(b) 検索範囲に親子集合位置指示子が指すレコードが含まれない場合

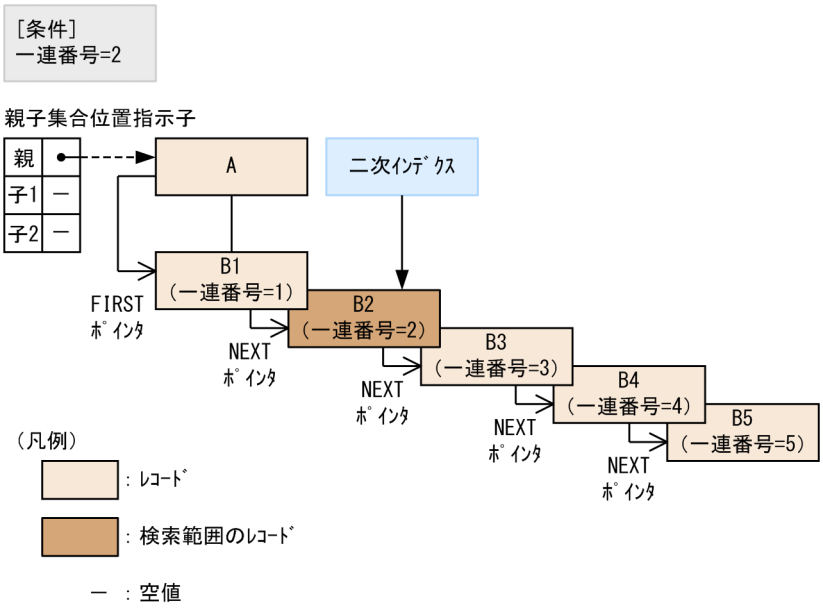
(c) 検索範囲の基点を絞り込む条件が存在する場合

それぞれの検索条件の詳細について次に示します。

(a) '='条件が指定されている場合

二次インデクスのキーを構成する一連番号またはユーザキーに対して，'='条件が指定されている場合，二次インデクスが使用されます。

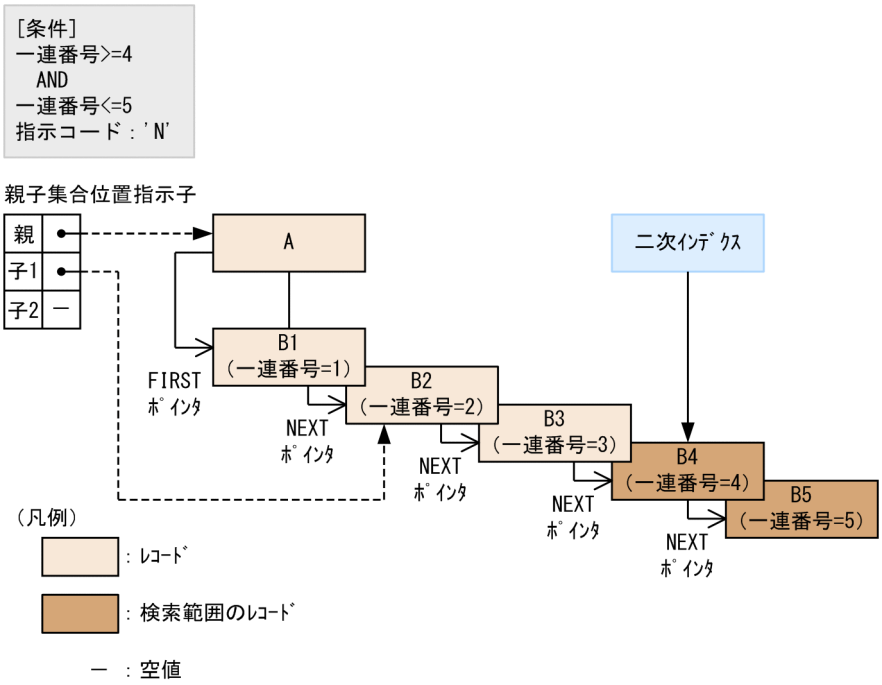
'='条件が指定されている場合に，二次インデクスが使用される例を次の図に示します。



(b) 検索範囲に親子集合位置指示子が指すレコードが含まれない場合

4V AFM の SDB データベースの場合，NEXT ポインタまたは PRIOR ポインタに従って順アクセスする際に（SDB データベースを操作する API の指示コードが'N'または'P'の場合に），二次インデクスの構成要素の一連番号またはユーザキーに対する条件によって決定される検索範囲に，親子集合位置指示子が指すレコードが含まれていなければ，二次インデクスが使用されます。

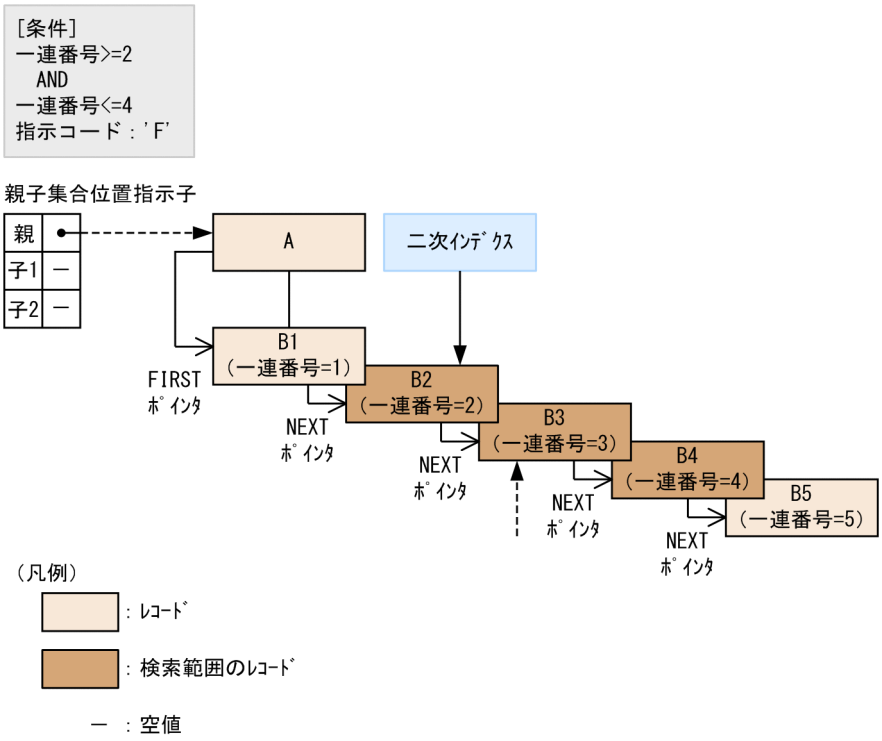
検索範囲に親子集合位置指示子が指すレコードが含まれないときに，二次インデクスが使用される例を次の図に示します。



(c)　検索範囲の基点を絞り込む条件が存在する場合

先頭のレコード（FIRST ポインタ）または最後のレコード（LAST ポインタ）を基点とする検索の場合（SDB データベースを操作する API の指示コードが'F'または'L'の場合），二次インデクスの構成要素の一連番号またはユーザキーに対する検索範囲の基点を絞り込む条件が存在すると，二次インデクスが使用されます。

二次インデクスの構成要素の一連番号またはユーザキーに対する検索範囲の基点を絞り込む条件が存在する場合に，二次インデクスが使用される例を次の図に示します。



(3) 子レコード格納時の二次インデクスの使用有無

次に示す条件をすべて満たしている場合，子レコード格納時のレコード挿入位置の検索に，二次インデクスが使用されます。

- レコード実現値を格納するレコード型にユーザキーが定義されている。
- レコード実現値を格納するレコード型に，上記のユーザキーを構成要素とする二次インデクスが定義されている。

二次インデクスを使用したレコード挿入位置の検索について，二次インデクスを使用しない場合と比較した例を次に示します。

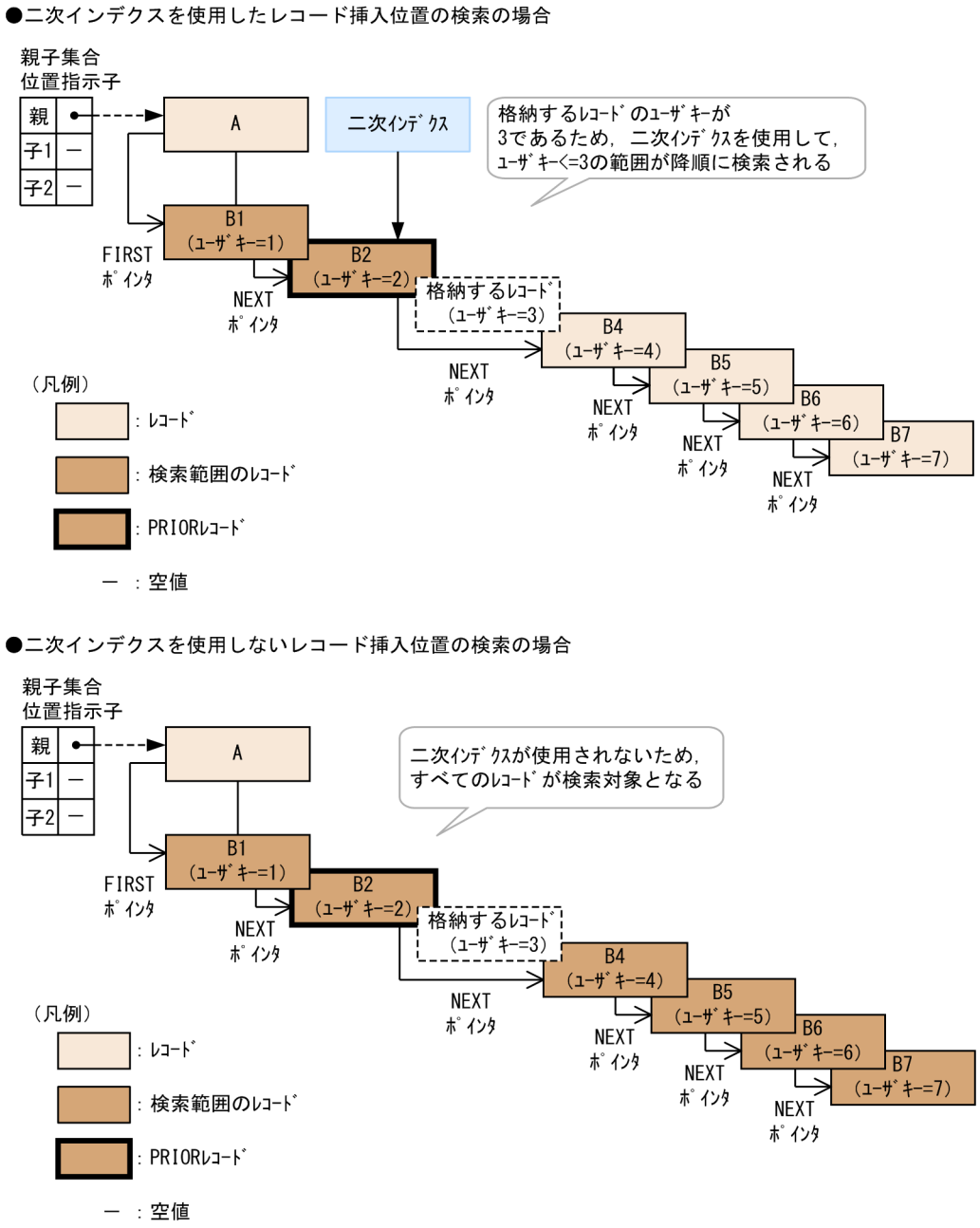
二次インデクスを使用したレコード挿入位置の検索の例

ユーザキーの値が 3 であるレコードを格納するとします。

上記の条件を満たしていれば，二次インデクスを使用してレコードの挿入位置（格納するレコードの PRIOR ポインタに接続されるレコード）が検索されます。二次インデクスを使用することで，検索範囲を絞り込むことができるため，効率的にレコードの挿入位置が検索されます。

なお，上記の条件を満たしていない場合は，挿入位置が見つかるまで NEXT ポインタまたは PRIOR ポインタを使用して順アクセスします。このため，レコードの挿入位置によって検索性能に差が出ます。

図 2-24　二次インデクスを使用したレコード挿入位置の検索の例



(4)　二次インデクスの使用有無の確認

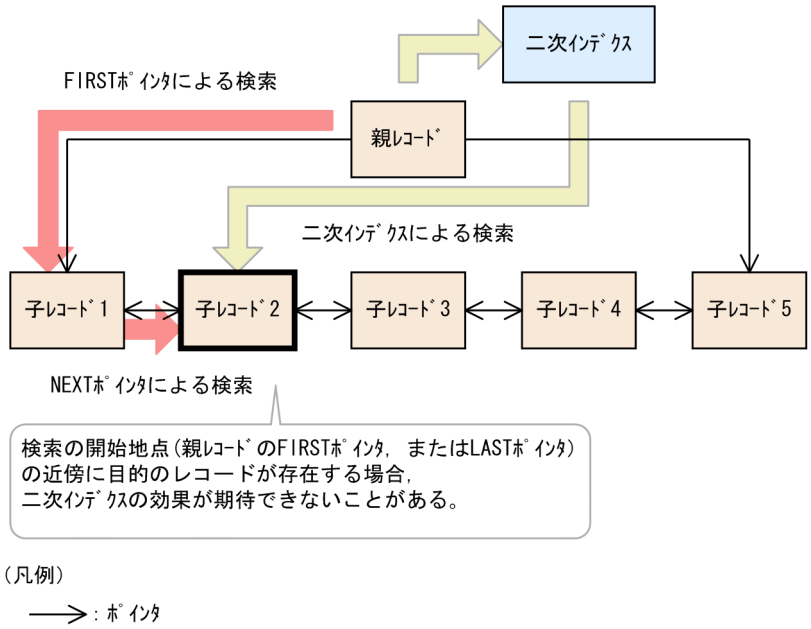
HiRDB/SD が二次インデクスを使用して検索したかどうかは，次の方法で確認できます。

- UAP 統計レポートの出力結果
- SDB データベースへのアクセス時に TP1/FSP で使用するインタフェースエリアの，エントリ部の二次インデクス使用フラグ

UAP 統計レポートについては，「16.5.5　UAP 統計レポート機能」を参照してください。また，SDB データベースへのアクセス時に使用するインタフェースエリアについては，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「インタフェースエリア構造」を参照してください。

(5)　二次インデクスに関する留意事項

- 一般的に，NEXT ポインタまたは PRIOR ポインタに従って順アクセスするよりも，二次インデクスを使用した方が検索性能は向上します。ただし，次のような場合には，二次インデクスを使用した方が検索性能が低下することがあります。
  - 検索の開始地点（親レコードの FIRST ポインタまたは LAST ポインタ）の近傍に目的のレコードが存在する場合



  - 二次インデクス中に無効なキーが多く存在する場合

    レコードを削除した場合，対応するインデクスキーは物理的には削除されないで，無効なキーとなります。そのため，無効となっているキーを含む範囲を検索すると，二次インデクスを使用した方が検索性能が低下することがあります。

- 二次インデクスが定義されている場合，レコードの格納や削除に伴ってインデクスが更新されます。そのため，レコードの格納や削除が頻繁に繰り返されると，更新性能が低下することがあります。

2.3.11　二次インデクス【SD FMB】

SD FMB の SDB データベースに定義する二次インデクス（レコード型内二次インデクス）について説明します。

なお，SD FMB の SDB データベースに二次インデクスを定義する場合は，HiRDB Structured Data Access Facility Extension for XDM/SD type をインストールしておく必要があります。

(1)　レコード型内二次インデクス

レコード型内のレコードの検索に使用する二次インデクスを，レコード型内二次インデクスといいます。

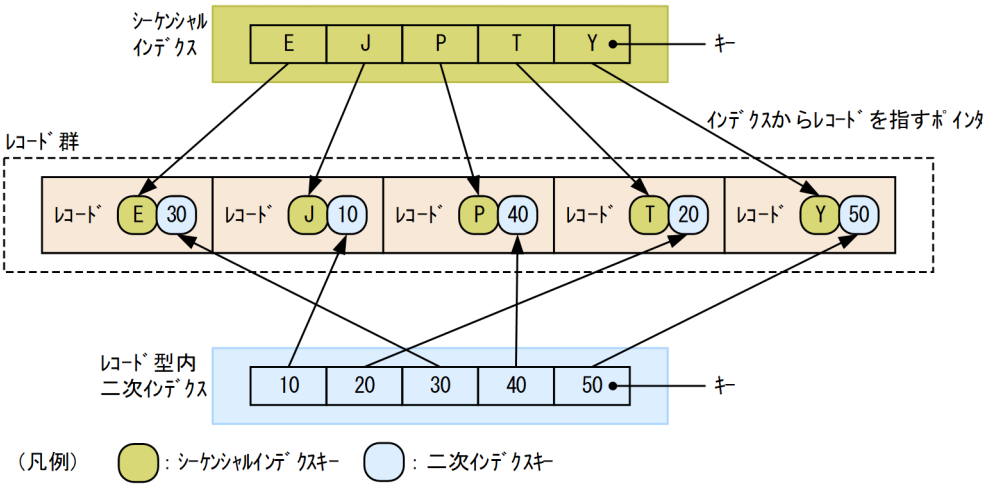
(a) レコード型内二次インデクスを作成する目的

ルートレコード型の場合

シーケンシャルインデクスのキー以外の構成要素をキーとして，ルートレコード型内のレコードを検索したい場合，レコード型内二次インデクスを定義します。

レコード型内二次インデクスは，指定した構成要素の値をキーとして，キーの昇順に作成されます。シーケンシャルインデクスとレコード型内二次インデクスのレコードを検索する順序の違いを次の図に示します。

図 2-25 シーケンシャルインデクスとレコード型内二次インデクスのレコードを検索する順序の違い



子レコード型の場合

子レコード型内のレコードを検索したい場合，レコード型内二次インデクスを定義します。

HiRDB/SD では，レコードを物理的な順序で検索することはできません。このため，子レコード型内のレコードを検索する場合は，レコード型内二次インデクスを定義する必要があります。レコード型内二次インデクスは，指定した構成要素の値をキーとして，キーの昇順に作成されます。

(b) レコード型内二次インデクスを作成するときの留意事項

- レコードの格納，またはレコードの削除を実行すると，レコード型内二次インデクスが更新されます。また，レコードの更新を実行し，二次インデクスを定義した構成要素を更新した場合も，レコード型内二次インデクスが更新されます。レコード型内二次インデクスの数に比例して，処理のオーバヘッドが増加するため，必要となるレコード型内二次インデクスだけを定義してください。
- HiRDB/SD では，インデクスキー値は一意である必要があります。子レコードの順序に関係なく検索する場合は，一連番号を含めたデータベースキーに，レコード型内二次インデクスを定義してください。

2.4　SDB データベースの定義

ここでは，SDB データベースの定義と，SDB データベースの定義を基に作成される SDB ディクショナリ情報および SDB ディレクトリ情報について説明します。

2.4.1　SDB データベースの定義の種類

SDB データベースの定義には，SDB データベース定義と SDB データベース格納定義の 2 つの定義があります。それぞれの定義内容を次の表に示します。

表 2-10　SDB データベースの定義の種類と定義内容

SDB データベースの定義の種類	定義内容
SDB データベース定義	SDB データベース定義には，SDB データベースの論理構造を定義します。データ操作の対象となるレコードの構成やレコード間の親子関係などを定義します。 SDB データベース定義の詳細については，「11．HiRDB/SD 定義ユティリティ（pdsdbdef）」を参照してください。
SDB データベース格納定義	SDB データベース格納定義には，SDB データベースの格納構造を定義します。インデクスの構成要素や親子関係を実現するためのポインタなどを定義します。 SDB データベース格納定義の詳細については，「11．HiRDB/SD 定義ユティリティ（pdsdbdef）」を参照してください。

上記の表に示す 2 つの定義を作成して HiRDB/SD 定義ユティリティ（pdsdbdef）を実行すると，SDB データベースが定義されます。このとき，次の表に示す 2 つの情報が作成され，ディクショナリ表に格納されます。HiRDB/SD はこれら 2 つの情報で SDB データベースの定義情報を管理します。

表 2-11　HiRDB/SD 定義ユティリティ（pdsdbdef）実行時に出力される情報

定義情報の種類	内容
SDB ディクショナリ情報	ユーザが定義した SDB データベースの定義情報を，ユーザが記述した形式に近い状態で保持したものです。
SDB ディレクトリ情報	SDB データベースの SDB ディクショナリ情報を，HiRDB/SD が解釈しやすい形式に変換したものです。

ディクショナリ表に格納される情報の詳細については，「付録 B　ディクショナリ表」を参照してください。

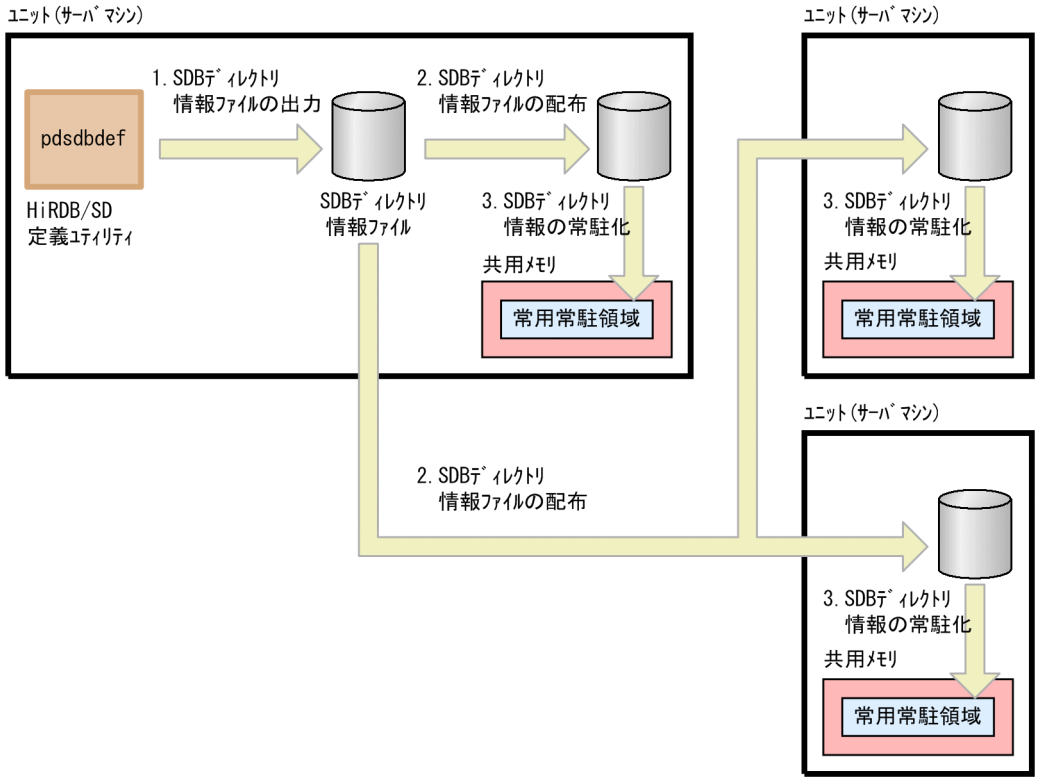
2.4.2　SDB ディレクトリ情報の常駐化

HiRDB/SD では，ディクショナリ表に格納された SDB ディレクトリ情報を，SDB ディレクトリ情報ファイルに出力して使用します。SDB ディレクトリ情報ファイルとは，SDB データベースへのアクセスに必要な情報を出力したファイルです。

出力した SDB ディレクトリ情報ファイルを全ユニットに配布し，ユニットの開始時に SDB ディレクトリ情報を共用メモリ上に常駐化します。UAP および一部のコマンドは，共用メモリ上に常駐化された SDB ディレクトリ情報を使用して SDB データベースにアクセスします。

SDB ディレクトリ情報の共用メモリ上への常駐化（ユニットの開始時）を次の図に示します。

図 2-26　SDB ディレクトリ情報の共用メモリ上への常駐化（ユニットの開始時）



[説明]

1. HiRDB/SD 定義ユティリティ（pdsdbdef）を使用して，SDB ディレクトリ情報を SDB ディレクトリ情報ファイルに出力します。
2. 1.で出力した SDB ディレクトリ情報ファイルを全ユニットに配布します。
3. ユニットの開始時（ユニットの再開始時を含む），SDB ディレクトリ情報が共用メモリ上に常駐化されます。SDB ディレクトリ情報を格納する領域を常用常駐領域といいます。

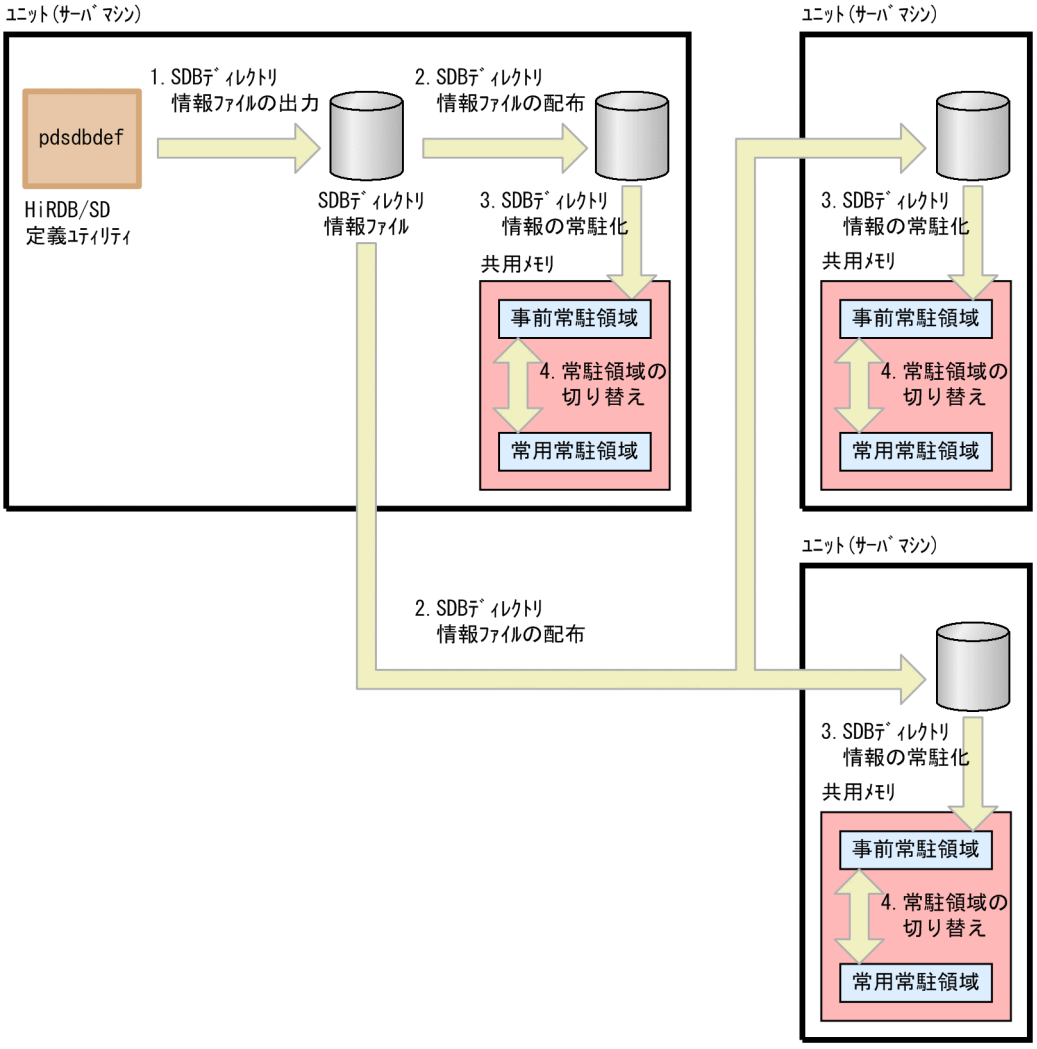
上記の流れで，SDB ディレクトリ情報が共用メモリ上に常駐化されます。

なお，1.と 2.は，HiRDB/SD 定義ユティリティ（pdsdbdef）の 1 回の操作で実行できます。

また，HiRDB の稼働中に，運用コマンドを使用して共用メモリ上の SDB ディレクトリ情報を切り替えることもできます。これによって，HiRDB の稼働中に SDB データベースの定義を追加したり，変更したりできます。

SDB ディレクトリ情報の常駐化コマンド（pdsdbarc コマンド）による SDB ディレクトリ情報の共用メモリ上への常駐化を次の図に示します。

図 2-27　SDB ディレクトリ情報の常駐化コマンド（pdsdbarc コマンド）による SDB ディレクトリ情報の共用メモリ上への常駐化



[説明]

1. HiRDB/SD 定義ユティリティ（pdsdbdef）を使用して，SDB ディレクトリ情報を SDB ディレクトリ情報ファイルに出力します。
2. 1.で出力した SDB ディレクトリ情報ファイルを全ユニットに配布します。
3. SDB ディレクトリ情報の常駐化コマンド（pdsdbarc -e コマンド）を実行すると，2.で配布した SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報が，共用メモリ上の事前常駐領域に格納されます。
4. SDB ディレクトリ情報の常駐化コマンド（pdsdbarc -w -q コマンド）を実行すると，事前常駐領域を常用常駐領域に切り替えられます。

上記の流れで，SDB ディレクトリ情報が共用メモリ上に常駐化されます。

なお，1.と 2.は，HiRDB/SD 定義ユティリティ（pdsdbdef）の 1 回の操作で実行できます。

2.5　SDB データベースの操作

ここでは，SDB データベースを操作する際の基本的な考え方や仕組みについて説明します。

4V FMB または 4V AFM の SDB データベースを操作するには，TP1/FSP が提供するライブラリ関数を使用するか，または HiRDB/SD データベースアクセスユティリティ（pdsdbexe）を使用します。

SD FMB の SDB データベースを操作するには，埋込み型 UAP に記述した DML を使用するか，または HiRDB/SD データベースアクセスユティリティ（pdsdbexe）を使用します。

TP1/FSP が提供するライブラリ関数を使用した SDB データベースの具体的な操作方法や，TP1/FSP が提供するライブラリ関数については，次のマニュアルを参照してください。

- 「TP1/Financial Service Platform 使用の手引」
- 「TP1/Financial Service Platform プログラム作成の手引」
- 「TP1/Server Base Enterprise Option　プログラム作成の手引」

HiRDB/SD データベースアクセスユティリティ（pdsdbexe）を使用した SDB データベースの具体的な操作方法については，「14.　HiRDB/SD データベースアクセスユティリティ（pdsdbexe）」を参照してください。

DML については，「17.　DML リファレンス【SD FMB】」を，埋込み型 UAP の作成方法については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」を参照してください。

2.5.1　SDB データベースの操作の概要（4V FMB または 4V AFM の場合）

(1)　SDB データベースに対する操作（API）

4V FMB または 4V AFM の SDB データベースは，次の表に示す API を使用して操作します。これらの API を「SDB データベースを操作する API」と表記します。

表 2-12　SDB データベースに対する操作（API）

項目	説明
個別開始	SDB データベースに対する操作を開始します。 SDB データベースに対する操作の開始時に，位置指示子の生成を行います。 位置指示子については，「2.5.3　位置指示子」を参照してください。
個別終了	SDB データベースに対する操作を終了します。 SDB データベースに対する操作の終了時に，位置指示子の解放を行います。
レコードの検索（FETCH）	操作対象にしたいレコードを検索し，TP1/FSP が用意するデータ格納エリアにレコードを取り出します。

項目	説明
複数レコードの検索（FETCHDB ALL）	ルートレコード以下のレコードを一括して検索し，TP1/FSP が用意するデータ格納エリアに，レコードを取り出します。 複数レコードの検索は，ファミリ単位で行われます。 1 ファミリの検索中であっても，データ格納エリアが満杯になると検索を中断します。この場合は，複数レコードの検索を複数回実行することで 1 ファミリのレコードを取得できます。 4V FMB の SDB データベースの場合に，複数レコードの検索ができます。
レコードの格納（STORE）	1 件のレコードを格納します。
レコードの更新（MODIFY）	1 件のレコードの内容を更新します。
レコードの削除（ERASE）	1 件以上のレコードを削除します。 削除するレコードが最下位のレコードである場合，そのレコードだけを削除します。 削除するレコードに子レコードが存在する場合，子レコードも同時に削除します。 4V FMB の SDB データベースで，削除対象のレコードが最下位のレコードであれば，API の一括オプションに'O'を指定することで，当該親レコード下のすべての子レコードを一括で削除できます。
レコードの一括削除	指定したキー（一連番号を除く）に対応したレコードを一括削除します。 4V DAM または 4V SAM の SDB データベースの場合に，レコードの一括削除ができます。
構成情報取得	指定したキーに対応したレコードの容量情報（割り当て済み一連番号）を取得します。 構成情報取得は，個別開始から SDB データベースを操作する前までに実行してください。SDB データベース操作後（レコードの検索など）に，構成情報取得はできません。 4V AFM の SDB データベースの場合に，構成情報取得を実行できます。

(2) データ格納エリア

データ格納エリアは，SDB データベースに対する操作を実行する際，検索したレコードを取り出したり，格納するレコードを設定したりするために使用する領域です。

データ格納エリアは TP1/FSP が用意します。データ格納エリアのサイズはユーザが指定します。

通常，レコードの検索時は，データ格納エリアには 1 件のレコードが格納されますが，複数レコードの検索をする場合は，複数レコードを格納するため，レコードの格納形式が異なります。

複数レコードの検索時の，データ格納エリアに対するレコードの格納形式については，「付録 M　複数レコードの検索時の，データ格納エリアに対するレコードの格納形式【4V FMB】」を参照してください。

2.5.2 SDB データベースの操作の概要（SD FMB の場合）

(1) SDB データベースに対する操作（DML）

SD FMB の SDB データベースは，次の表に示す DML を使用して操作します。

表 2-13　SDB データベースに対する操作（DML）

項目	説明
個別開始	SDB データベースに対する操作を開始します。 SDB データベースに対する操作の開始時に，位置指示子の生成を行います。位置指示子については，「2.5.3　位置指示子」を参照してください。 埋込み型 UAP で SDB データベースを操作する場合，個別開始は DML で要求するのではなく，各 SDB データベースに対する最初の操作要求時にシステムが自動的に実行します。 HiRDB/SD データベースアクセスユティリティ（pdsdbexe）で SDB データベースを操作する場合には，DML コマンドの START で個別開始を実行する必要があります。
個別終了	SDB データベースに対する操作を終了します。 SDB データベースに対する操作の終了時に，位置指示子の解放を行います。 埋込み型 UAP で SDB データベースを操作する場合，個別終了は DML で要求するのではなく，トランザクションの終了時にシステムが自動的に実行します。 HiRDB/SD データベースアクセスユティリティ（pdsdbexe）で SDB データベースを操作する場合には，DML コマンドの FINISH で個別終了を実行する必要があります。
レコードの検索（FETCH）	1 件のレコードを検索します。
位置指示子の位置づけ（FIND）	1 件のレコードに対して位置指示子の位置づけを行います。
レコードの格納（STORE）	1 件のレコードを格納します。
レコードの更新（MODIFY）	1 件のレコードの内容を更新します。
レコードの削除（ERASE）	1 件以上のレコードを削除します。 削除するレコードが最下位のレコードの場合，そのレコードだけを削除します。 削除するレコードに子レコードが存在する場合，子レコードも同時に削除します。

(2)　埋込み変数

埋込み変数は，埋込み型 UAP で SDB データベースに対する操作をする際，検索したレコードを取り出したり，格納するレコードを設定したりするために使用する領域です。

埋込み変数は，埋込み型 UAP のソースプログラム中で宣言します。埋込み変数の宣言方法や使い方については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「埋込み変数の宣言 」を参照してください。

2.5.3　位置指示子

HiRDB/SD は，操作中のレコードや親子集合を識別するための情報を保持します。この情報を位置指示子といいます。HiRDB/SD では，位置指示子を使用してレコードや親子集合に対する操作を行います。

(1) レコードの位置づけ

レコードの検索や格納などを行うと，そのレコードの位置情報が位置指示子に保持されます。これをレコードの位置づけといいます。レコードの位置づけは，ルートレコードから開始して，操作対象とするレコードまで位置づけます。

検索結果が NOT FOUND（SQLCODE=100）となった場合，検索対象のレコード位置指示子は空値になります。

SDB データベースにアクセスする場合は，操作の基点となるレコードを検索し，レコードに位置指示子を位置づけます。操作の対象としたいレコードに位置指示子を位置づけることで，以後の操作で，そのレコードを基点に SDB データベースを操作できます。

レコードの更新や削除をする場合は，直前に位置づけられたレコードを対象とします。そのため，レコードの位置づけを行わないで，レコードの更新やレコードの削除はできません。

また，4V FMB の SDB データベースのルートレコードを削除する場合は，構成要素名に DBKEY を指定した＝条件の検索で位置づけを行う必要があります。

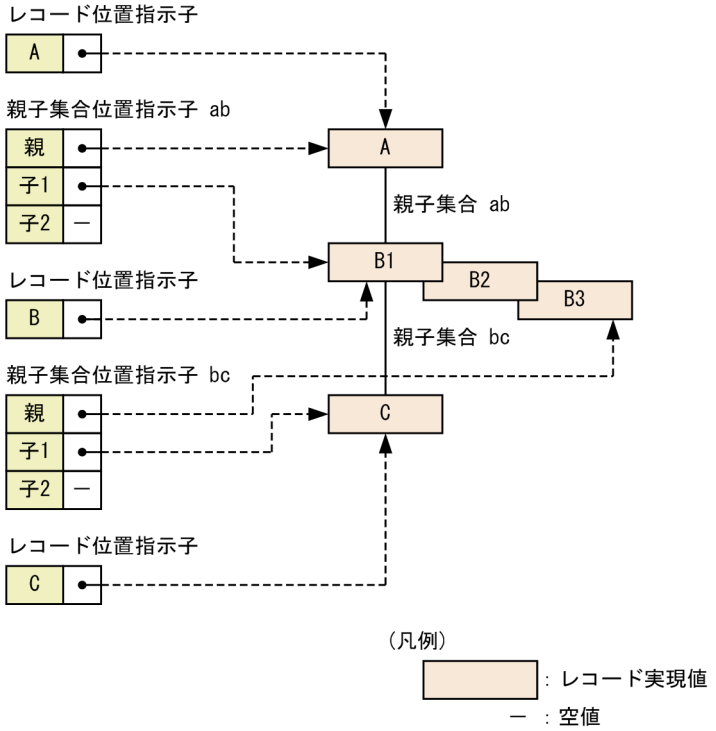
参考

レコードの一括削除や複数レコードの検索では，レコードの位置づけは不要です。

(2) 位置指示子の種類

位置指示子にはレコード位置指示子と親子集合位置指示子があります。

図 2-28　レコード位置指示子と親子集合位置指示子



(a)　レコード位置指示子

最後に操作したレコードの位置を示す位置指示子をレコード位置指示子といいます。

レコード位置指示子は，レコード型ごとに生成されます。

(b)　親子集合位置指示子

最後に操作した親レコード，および子レコードの位置を示す位置指示子を親子集合位置指示子といいます。

子レコードが削除された場合，削除されたレコードの，前後のレコードの位置情報を保持することがあります。そのため，親子集合位置指示子は，親レコードに対する位置指示子と，子レコードに対する 2 つ（子 1，子 2）の位置指示子で構成されます。

親子集合位置指示子は，親子集合型ごとに生成されます。

(3)　位置指示子の生成と消滅

位置指示子は個別開始時（SDB データベース単位の操作開始時）にレコード型，親子集合型ごとに生成され，個別終了時（SDB データベース単位の操作終了時）およびトランザクション終了時に消滅します。

(4)　位置指示子の状態

通常，位置指示子は操作中のレコードや親子集合を識別するための情報を保持します。ここでは位置指示子の特殊な状態について説明します。

(a) 空値

位置指示子がどのレコードも指していないか，または位置指示子に親子集合中のレコードを検索するために必要な情報が設定されていない状態を空値といいます。

個別開始時点の位置指示子は空値となります。個別開始後，SDB データベースを操作することによって，レコードを識別するために必要な情報が位置指示子に設定されます。

(b) 親子集合位置指示子に設定されるレコード

親子集合位置指示子に設定されるレコードは，最後に操作したレコードが親レコードか子レコードかによって異なります。

- 最後に操作したレコードが親レコードの場合

  親子集合位置指示子の親には操作したレコードが設定されます。親子集合位置指示子の子は空値となります。検索結果が NOT FOUND（SQLCODE=100）となった場合，親子集合位置指示子の親，および子の両方とも空値になります。
- 最後に操作したレコードが子レコードの場合

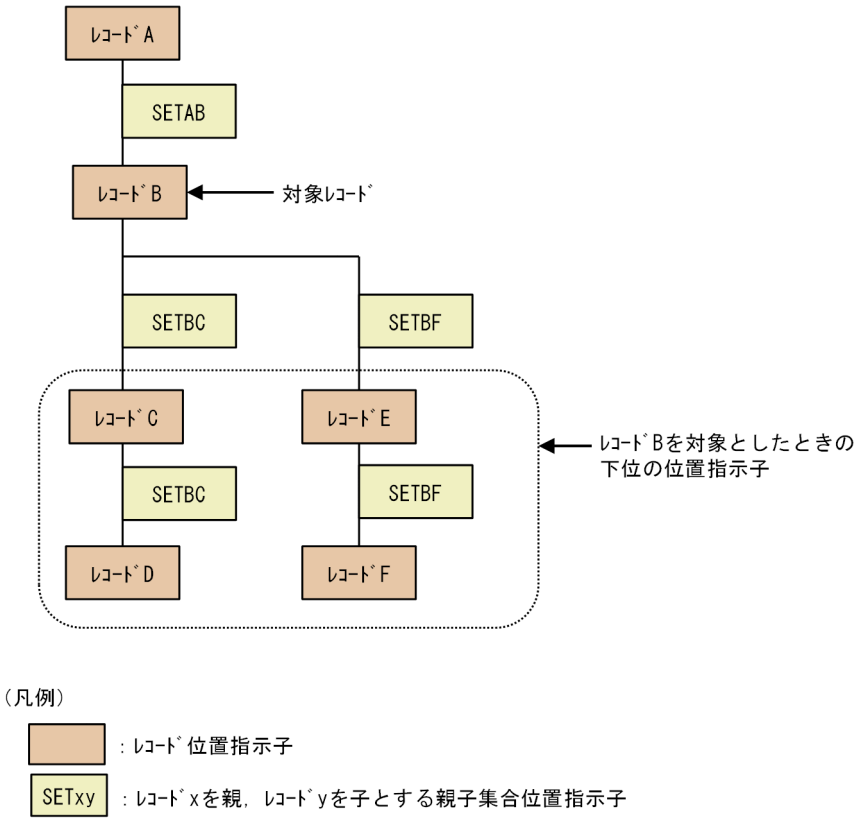
  親子集合位置指示子の子には操作したレコードが設定されます。親子集合位置指示子の親には操作したレコードの親レコードが設定されます。検索結果が NOT FOUND（SQLCODE=100）となった場合，親子集合位置指示子の子は空値になります。

(c) 下位の位置指示子の設定

下位の位置指示子とは，対象レコードの下位のレコード位置指示子，親子集合位置指示子のことです。

レコード B を対象としたときの下位の位置指示子の例を次に示します。図中の点線部分が下位の位置指示子になります。

図 2-29　下位の位置指示子



下位の位置指示子の設定内容はレコードの位置づけ方法によって異なります。

- レコード型内の検索，親子集合型内の検索，およびレコードの格納の場合
  下位の位置指示子はクリアされ空値となります。
- 位置指示子指定の検索の場合
  下位の位置指示子は変更されません。

(d)　子レコード削除後の親子集合位置指示子の状態（複合状態）

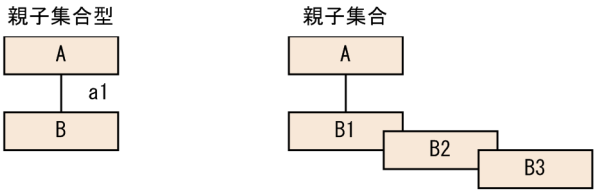
子レコードを削除した場合，削除した子レコードの位置によって，子レコードの削除後に親子集合位置指示子が保持する情報が異なります。このときの親子集合位置指示子の状態を複合状態といいます。

中間のレコードを削除した場合，親子集合位置指示子では，削除した子レコードの前後のレコードの位置情報を保持します。ただし，先頭の子レコードを削除した場合は，削除した子レコードの次のレコードの位置情報だけを保持します。また，最後尾の子レコードを削除した場合は，削除した子レコードの前のレコードの位置情報だけを保持します。

子レコードを削除した場合の親子集合位置指示子の状態の例を次の図に示します。

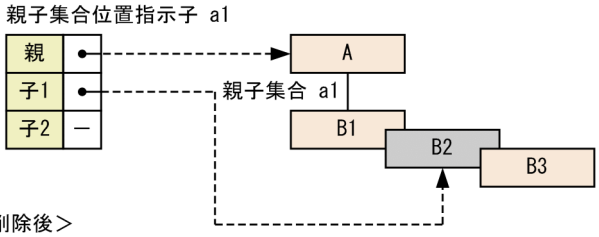
図 2-30　子レコードを削除した場合の親子集合位置指示子の状態の例（複合状態の例）

●親子集合型と親子集合



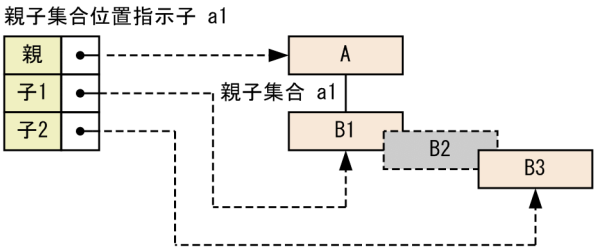
●中間の子レコードを削除した場合

＜削除前＞



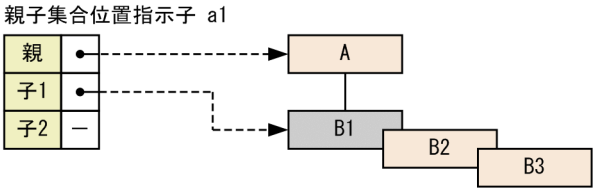
＜削除後＞

親子集合位置指示子a1は，削除したレコードの前後のレコードの位置情報を保持します。



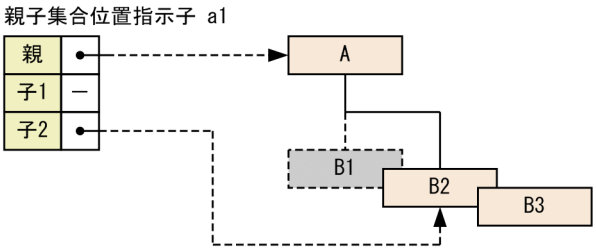
●先頭の子レコードを削除した場合

＜削除前＞



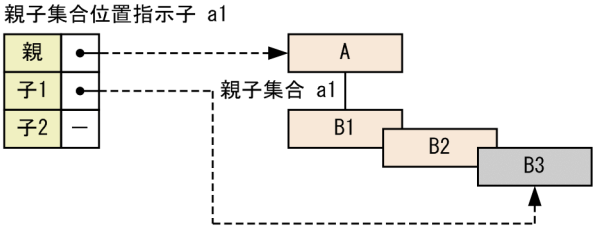
＜削除後＞

親子集合位置指示子a1は，削除したレコードの次のレコードの位置情報だけを保持します。



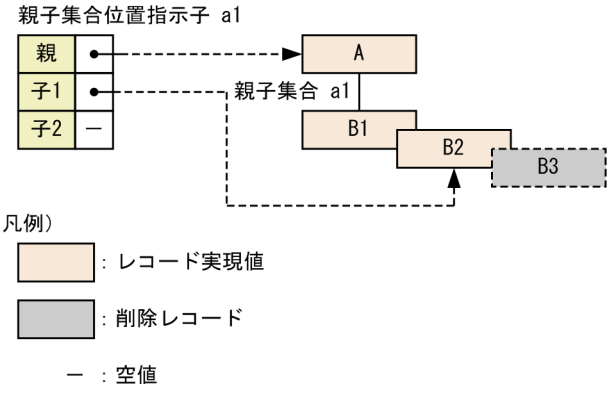
●最後尾の子レコードを削除した場合

＜削除前＞



＜削除後＞

親子集合位置指示子a1は，削除したレコードの前のレコードの位置情報だけを保持します。



(e) 子レコードのキー項目更新時の親子集合位置指示子の状態（複合状態）

子レコードのキー項目を更新した場合，レコード位置の移動に伴い親子集合位置指示子を更新します（レコード位置指示子は更新しません）。

- 親子集合位置指示子が複合状態ではない場合

  キー項目の更新に伴いレコード位置が移動した場合，親子集合位置指示子は複合状態となり，子 1 と子 2 に更新前のレコード位置の前後のレコードを設定します※。キー項目を更新してもレコード位置が移動しない場合は，親子集合位置指示子を更新しません。

注※

更新前のレコード位置が先頭のレコードの場合，子 1 は空値になります。更新前のレコード位置が最後尾のレコードの場合，子 2 は空値になります。

- 親子集合位置指示子が複合状態の場合

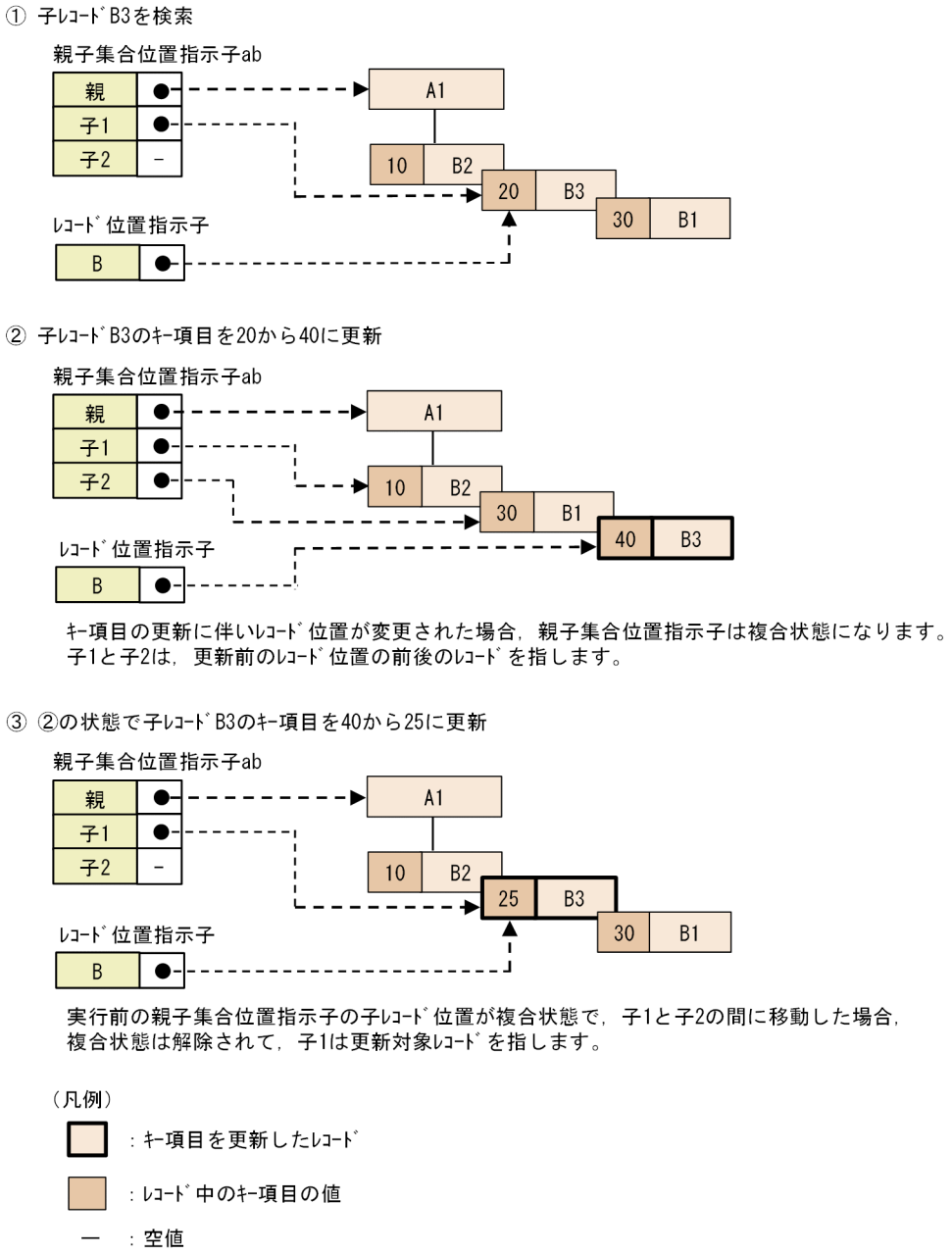
  キー項目の更新に伴いレコード位置が子 1 と子 2 の間※に移動した場合，親子集合位置指示子の複合状態が解除され，子 1 に更新レコードを設定します。子 1 と子 2 の間以外の場所に移動した場合，親子集合位置指示子を更新しません。

  注※

  子 1 が空値の場合は先頭に移動したときが該当します。子 2 が空値の場合は最後尾に移動したときが該当します。

キー項目を更新した場合の親子集合位置指示子の状態の例を次の図に示します。

図 2-31　キー項目を更新した場合の親子集合位置指示子の状態の例（複合状態の例）



(5)　エラー発生時の位置指示子の状態

SDB データベースの操作でエラーが発生した場合，エラーが発生したレコード型のレコード位置指示子は次のどちらかの状態になります。

- エラーが発生する前の状態
- 空値

また，親子集合位置指示子についても同様です。このため，エラー発生後も同一トランザクション内で SDB データベースの操作を続ける場合は，必要に応じて位置指示子を位置づけ直してから，SDB データベースの操作を行ってください。

2.5.4　レコードの検索【4V FMB，4V AFM】

SDB データベース中のレコードを検索し，データ格納エリアまたは埋込み変数にレコードを取り出します。検索されたレコードに対してはレコードの位置づけが行われます。

(1)　レコードのアクセス順序

レコードのアクセス順序について，SDB データベース種別ごとに説明します。

(a)　4V FMB の SDB データベースの場合

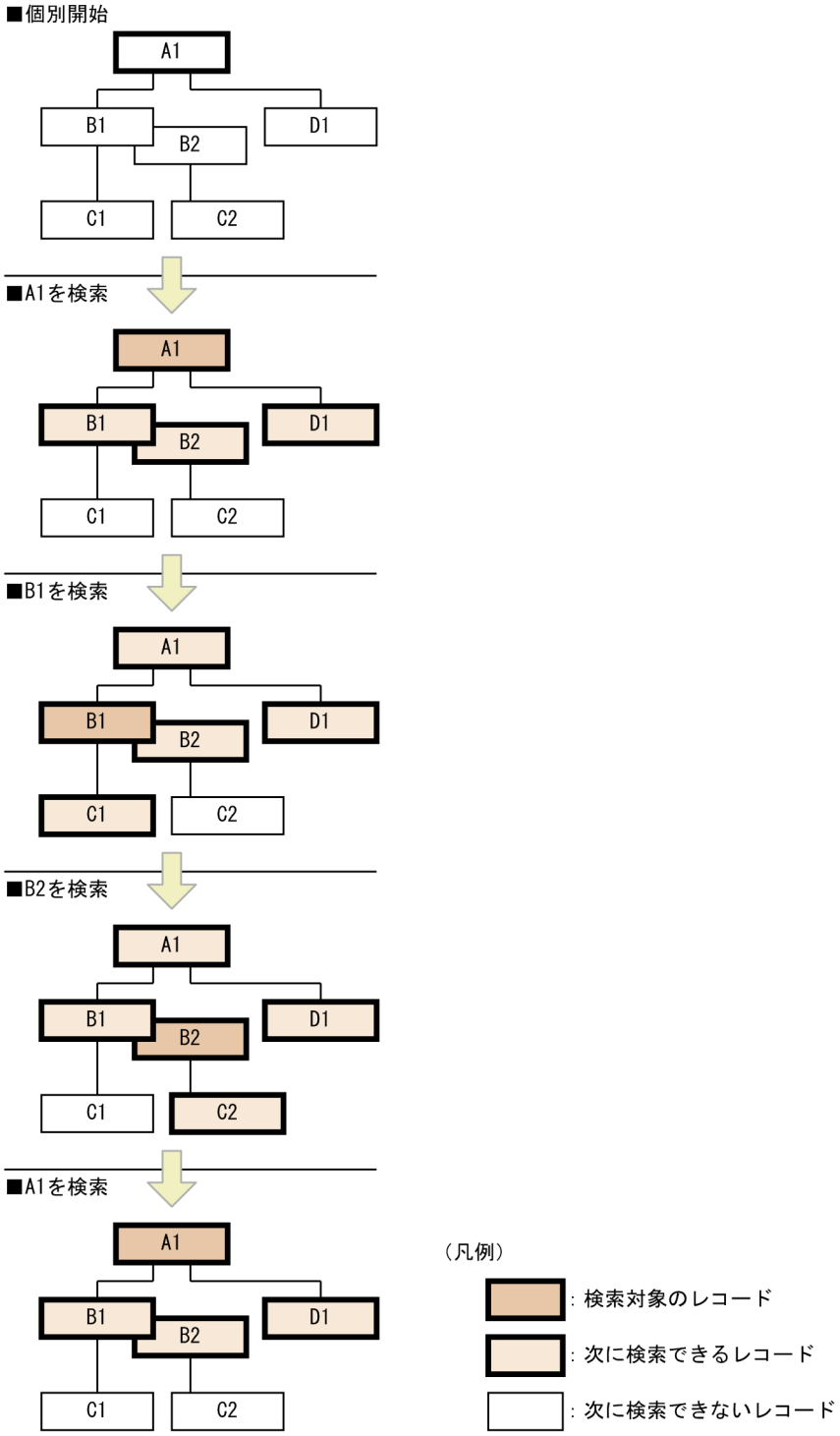
4V FMB の SDB データベースの場合，データベースが階層構造を持つため，レコードへのアクセスもルートレコードから階層順に行う必要があります（「図 2-32　4V FMB の SDB データベースのアクセス順序の概要」の場合，レコード A1→C1 の順にアクセスしたり，レコード C1 に直接アクセスしたりすることはできません）。

レコードの検索，格納，および位置指示子の位置づけを行うと，レコードの位置づけが行われ，そのレコードの子レコードに対してアクセスができるようになります。

同一レコード型の別レコードに対してレコードの位置づけを行うと，前回アクセスしたレコードの子レコードに対してはアクセスできなくなり，今回アクセスしたレコードの子レコードに対してアクセスできるようになります。

4V FMB の SDB データベースのアクセス順序の概要を次の図に示します。この図では，レコード A1→B1→B2→A1 の順序で検索します。

図 2-32　4V FMB の SDB データベースのアクセス順序の概要



ポイント

HiRDB/SD では，レコードの検索や格納などを行うと，そのレコードの位置情報を保持します。この情報を位置指示子といいます。HiRDB/SD は，位置指示子を使用して SDB データベースにアクセスします。位置指示子については，「2.5.3　位置指示子」を参照してください。

なお，検索結果が NOT FOUND（SQLCODE=100）となった場合，対象レコードのレコード位置指示子は空値になります。

(b)　4V AFM の SDB データベースの場合

4V AFM の SDB データベースの場合，データベースが階層構造を持たないため，指定したレコードに直接アクセスできます。

参考

4V AFM の SDB データベースでは仮想ルートレコード型を定義して，仮想親子集合を構成しますが，レコードへのアクセス時に仮想ルートレコードを意識する必要はありません。

(2)　検索の基点となるレコードと検索の方向

SDB データベースでは，レコードに設定されたポインタに従ってレコードの格納順にレコードを検索します。そのため，レコードを検索する場合は，検索の基点となるレコードと検索の方向を指定します。

(a)　検索の基点となるレコード

検索の基点となるレコードを API または DML の指示コードで指定します。指定できる指示コードと検索の基点となるレコードを次の表に示します。

表 2-14　指示コードと検索の基点となるレコード

指示コード		検索の基点となるレコード	説明
API の場合	DML の場合		
F	FIRST	先頭のレコード（FIRST ポインタ）	• ルートレコードの場合は，シーケンシャルインデクスのキー値の最小値を持つレコードが検索の基点となります。 • 子レコードの場合は，親レコードの FIRST ポインタが指すレコードが検索の基点となります。
L	LAST	最後のレコード（LAST ポインタ）	• 子レコードの場合は，親レコードの LAST ポインタが指すレコードが検索の基点となります。 • ルートレコードの場合は指定できません。
U	−	USER ポインタが指すレコード	• 子レコードの場合は，親レコードの USER ポインタが指すレコードが検索の基点となります。 • ルートレコードの場合は指定できません。
N	NEXT	位置づけされているレコードの次のレコード（NEXT ポインタ）	• ルートレコードの場合は，位置づけされたレコードの次のレコード（レコード位置指示子

指示コード		検索の基点となるレコード	説明
API の場合	DML の場合		
			が指すレコードの，シーケンシャルインデクスのキー値の次のキー値を持つレコード）が検索の基点となります。 • 子レコードの場合は，位置づけされているレコードの次のレコード（親子集合位置指示子の子レコードが指すレコードの，NEXT ポインタが指すレコード）が検索の基点となります。 なお，親子集合位置指示子が空値の場合は，親レコードの FIRST ポインタが指すレコードが検索の基点となります。
P	PRIOR	位置づけされているレコードの 1 つ前のレコード（PRIOR ポインタ）	• 子レコードの場合は，位置づけされているレコードの 1 つ前のレコード（親子集合位置指示子の子レコードが指すレコードの，PRIOR ポインタが指すレコード）が検索の基点となります。 なお，親子集合位置指示子が空値の場合は，親レコードの LAST ポインタが指すレコードが検索の基点となります。 • ルートレコードの場合は指定できません。

（凡例）

－：該当する指示コードはありません。

(b) 検索の方向

4V FMB の SDB データベースを操作する場合，USER ポインタを基点とした検索の方向を，API の検索コードで指定します。昇順・降順の順序性は，ユーザキーを持つ場合はユーザキー，持たない場合は一連番号に従います。

表 2-15 検索コードと検索の方向

検索コード	説明
N	NEXT ポインタに従って検索します。
P	PRIOR ポインタに従って検索します。

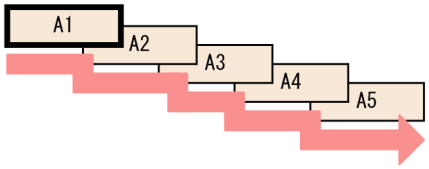
なお，4V AFM の SDB データベースを操作する場合は，検索コードは指定できません。

(3) レコードの検索範囲

検索の基点となるレコードと検索の方向の組み合わせによって，レコードの検索範囲が決定します。レコードと検索の方向の組み合わせを次の図に示します。

図 2-33　検索の基点となるレコードと検索の方向の組み合わせと検索範囲

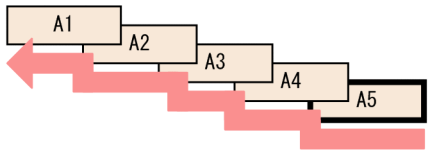
検索の基点	F	検索の方向	N



検索の基点から，NEXTﾎﾟｲﾝﾀに従って順検索します。
検索の基点は次のとおりです。

ﾙｰﾄﾚｺｰﾄﾞを検索する場合：
　ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽの最小のｷｰ値を持つﾚｺｰﾄﾞ
子ﾚｺｰﾄﾞを検索する場合：
　親ﾚｺｰﾄﾞのFIRSTﾎﾟｲﾝﾀが指すﾚｺｰﾄﾞ

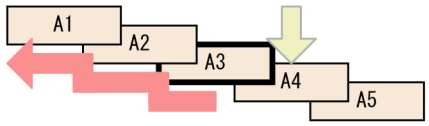
検索の基点	L	検索の方向	P



検索の基点から，PRIORﾎﾟｲﾝﾀに従って逆順検索します。
親ﾚｺｰﾄﾞのLASTﾎﾟｲﾝﾀが指すﾚｺｰﾄﾞが，検索の基点となります。
なお，ﾙｰﾄﾚｺｰﾄﾞの検索では，この指定はできません。

検索の基点	P	検索の方向	P

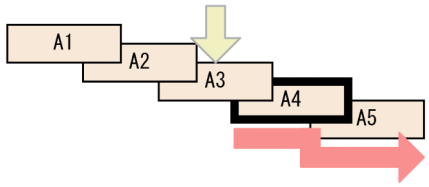
ﾚｺｰﾄﾞの現在位置



検索の基点から，PRIORﾎﾟｲﾝﾀに従って逆順検索します。
ﾚｺｰﾄﾞの現在位置の1つ前のﾚｺｰﾄﾞ（親子集合位置指示子が指す子ﾚｺｰﾄﾞの，PRIORﾎﾟｲﾝﾀが指すﾚｺｰﾄﾞ）が，検索の基点となります。
なお，ﾙｰﾄﾚｺｰﾄﾞの検索では，この指定はできません。

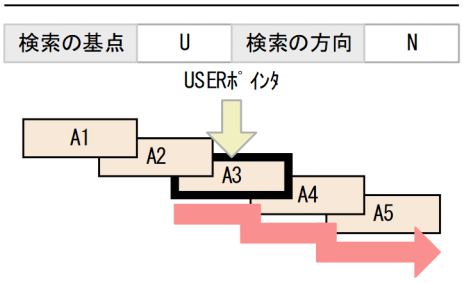
検索の基点	N	検索の方向	N

ﾚｺｰﾄﾞの現在位置

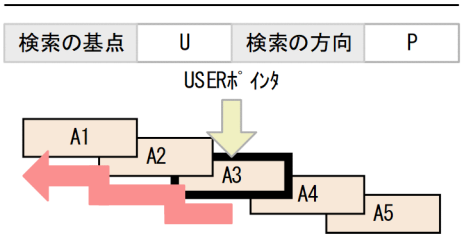


検索の基点から，NEXTﾎﾟｲﾝﾀに従って順検索します。
検索の基点は次のとおりです。

ﾙｰﾄﾚｺｰﾄﾞを検索する場合：
　ﾚｺｰﾄﾞの現在位置の次のﾚｺｰﾄﾞ
　（ﾚｺｰﾄﾞ位置指示子が指すﾚｺｰﾄﾞの，
　　ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽのｷｰ値の次のｷｰ値を持つﾚｺｰﾄﾞ）
子ﾚｺｰﾄﾞを検索する場合：
　ﾚｺｰﾄﾞの現在位置の次のﾚｺｰﾄﾞ
　（親子集合位置指示子が指す子ﾚｺｰﾄﾞの，
　　NEXTﾎﾟｲﾝﾀが指すﾚｺｰﾄﾞ）

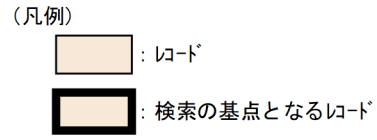


検索の基点から，NEXTﾎﾟｲﾝﾀに従って順検索します。
USERﾎﾟｲﾝﾀが指すﾚｺｰﾄﾞが検索の基点となります。
なお，ﾙｰﾄﾚｺｰﾄﾞの検索では，この指定はできません。
また，4V AFMのSDBデータベースの検索では，この指定はできません。



検索の基点から，PRIORﾎﾟｲﾝﾀに従って逆順検索します。
USERﾎﾟｲﾝﾀが指すﾚｺｰﾄﾞが検索の基点となります。
なお，ﾙｰﾄﾚｺｰﾄﾞの検索では，この指定はできません。
また，4V AFMのSDBデータベースの検索では，この指定はできません。

注　A1～A5はｷｰの昇順に並んでいるものとします。
ﾚｺｰﾄﾞの現在位置とは，位置指示子が位置づけられているﾚｺｰﾄﾞの位置を指します。



注意事項

SDB データベース種別やルートレコード・子レコードによって実行できない組み合わせがあります。詳細については，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「レコード検索範囲」を参照してください。

また，レコードの特定方法によっては，レコードの検索範囲は無視されます。

(4)　レコードの特定方法

レコードの検索では，次のどれかの方法でレコードを特定します。

- 特定位置のレコード（FIRST ポインタ，LAST ポインタ，または USER ポインタが指すレコード）を対象とする。
- 検索条件（DML の場合はキーの条件）に該当する最初のレコードを対象とする。
- 前回検索したレコードの前後のレコードを対象とする。

(5)　レコードの位置づけ

ルートレコードの検索（FETCH）時，または位置指示子の位置づけ（FIND）時は，レコードに位置指示子を位置づけておく必要はありません。

子レコードの検索時は，位置指示子を親レコードに位置づけておく必要があります。ただし，4V AFM の SDB データベースの場合は，位置指示子を親レコードに位置づけておく必要はありません（親レコードになる仮想ルートレコードの検索と位置づけを HiRDB/SD が内部的に行います）。

レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

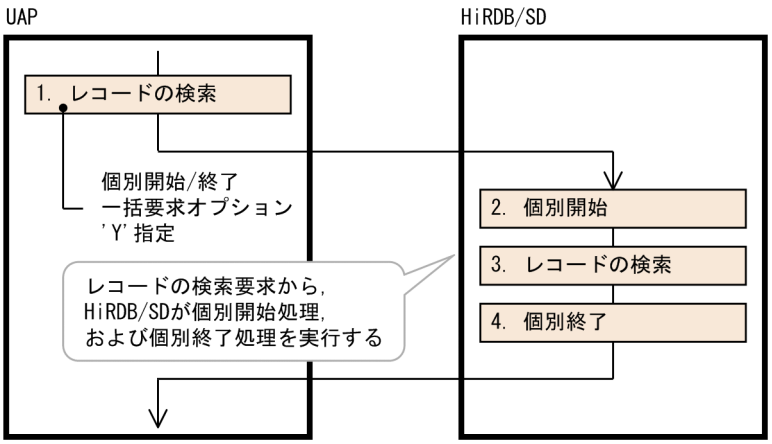
(6)　個別開始/終了一括要求オプションを指定したレコードの検索

SDB データベースを操作する API で個別開始/終了一括要求オプションを指定すると，UAP から個別開始および個別終了の要求を行わなくてもレコードを検索できます。個別開始/終了一括要求オプションの指定方法については，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「個別開始/終了一括要求」を参照してください。

(a)　レコードの検索処理の流れ

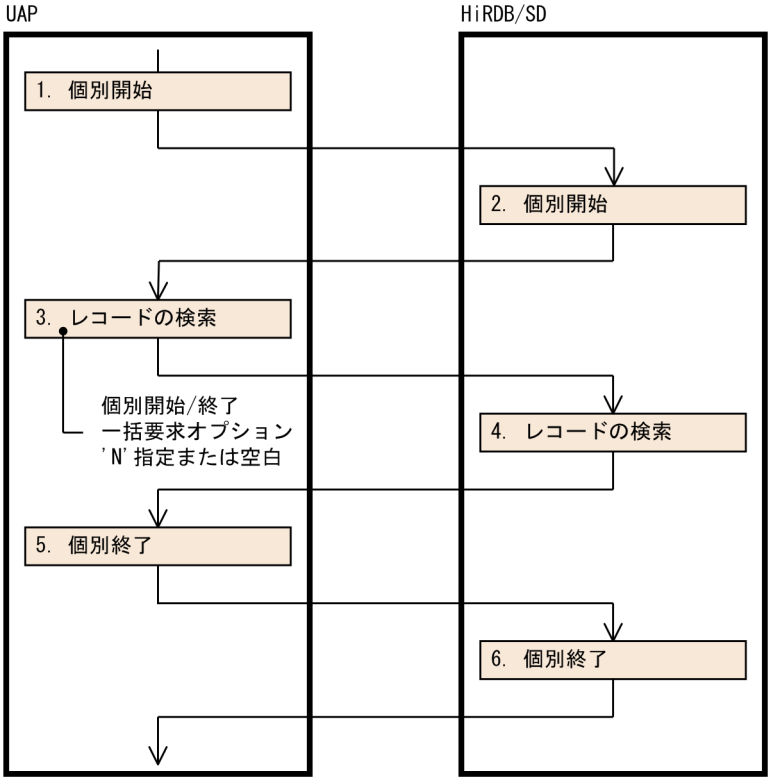
個別開始/終了一括要求オプションを指定した場合のレコードの検索処理の流れを次の図に示します。

図 2-34　レコードの検索処理の流れ（個別開始/終了一括要求オプションを指定した場合）



個別開始/終了一括要求オプションを指定しない場合のレコードの検索処理の流れを次の図に示します。

図 2-35　レコードの検索処理の流れ（個別開始/終了一括要求オプションを指定しない場合）



(b)　個別開始/終了一括要求オプションの適用基準

個別開始後，レコードの検索を 1 回だけ実行して個別終了を要求するような場合，個別開始/終了一括要求オプションを指定すると，UAP からの個別開始および個別終了の要求が不要になるため，個別開始と個別終了による UAP と HiRDB 間の処理のオーバーヘッドがなくなり，性能が向上します。

ただし，個別開始/終了一括要求オプションを指定した場合，レコードの検索後に個別終了が実行されるため，次のようなレコードの位置づけを目的としたレコード検索はできません。

- レコードの更新や削除をするためのレコードの検索
- 次のレコードの検索要求で，検索の基点とするためのレコードの検索

また，レコードの位置づけを目的としないレコードの検索の場合でも，同じ SDB データベースに対して複数回レコードの検索を行う際に，このオプションを指定すると，レコードの検索を行うごとに個別開始および個別終了処理が実行されるため，性能低下の原因になるおそれがあります。

(c)　留意事項

- 個別開始/終了一括要求オプションを指定した場合，UAP からの個別開始および個別終了の要求は不要になります。しかし，個別開始および個別終了の処理は実行されるため，個別開始および個別終了の要求を実行したものとしてメモリ所要量を見積もってください。
- 同一トランザクション内で，個別開始/終了一括要求オプションを指定したレコードの検索と，個別開始の要求を行う SDB データベースの操作を混在して実行できます。

- 個別開始/終了一括要求オプションを指定した場合，オプションを指定しない場合に比べて次の点が異なります。
  - 排他自動解除機能を使用している場合，検索したレコード実現値の格納ページに対する排他は解除されます。
  - 個別開始/終了一括要求オプションを指定して検索したレコード実現値を，同一トランザクション内で再度検索したり，一括削除したりしてもアクセス競合のエラーは発生しません。

2.5.5 レコードの検索【SD FMB】

SDB データベース中のレコードを検索し，データ格納エリアまたは埋込み変数にレコードを取り出します。検索されたレコードに対してはレコードの位置づけが行われます。

(1) 検索の種類

検索には，次の 2 つがあります。

- レコード型内の検索

  レコード位置指示子を使用した検索です。常にインデクスを使用した検索となります。ルートレコード，および二次インデクスを定義したレコードに対して実行できます。
- 親子集合型内の検索

  親子集合位置指示子を使用した検索です。特定の親レコード実現値配下の子レコードを検索します。常にポインタを使用した検索となります。子レコードに対して実行できます。

(2) SDB データベースのアクセス例

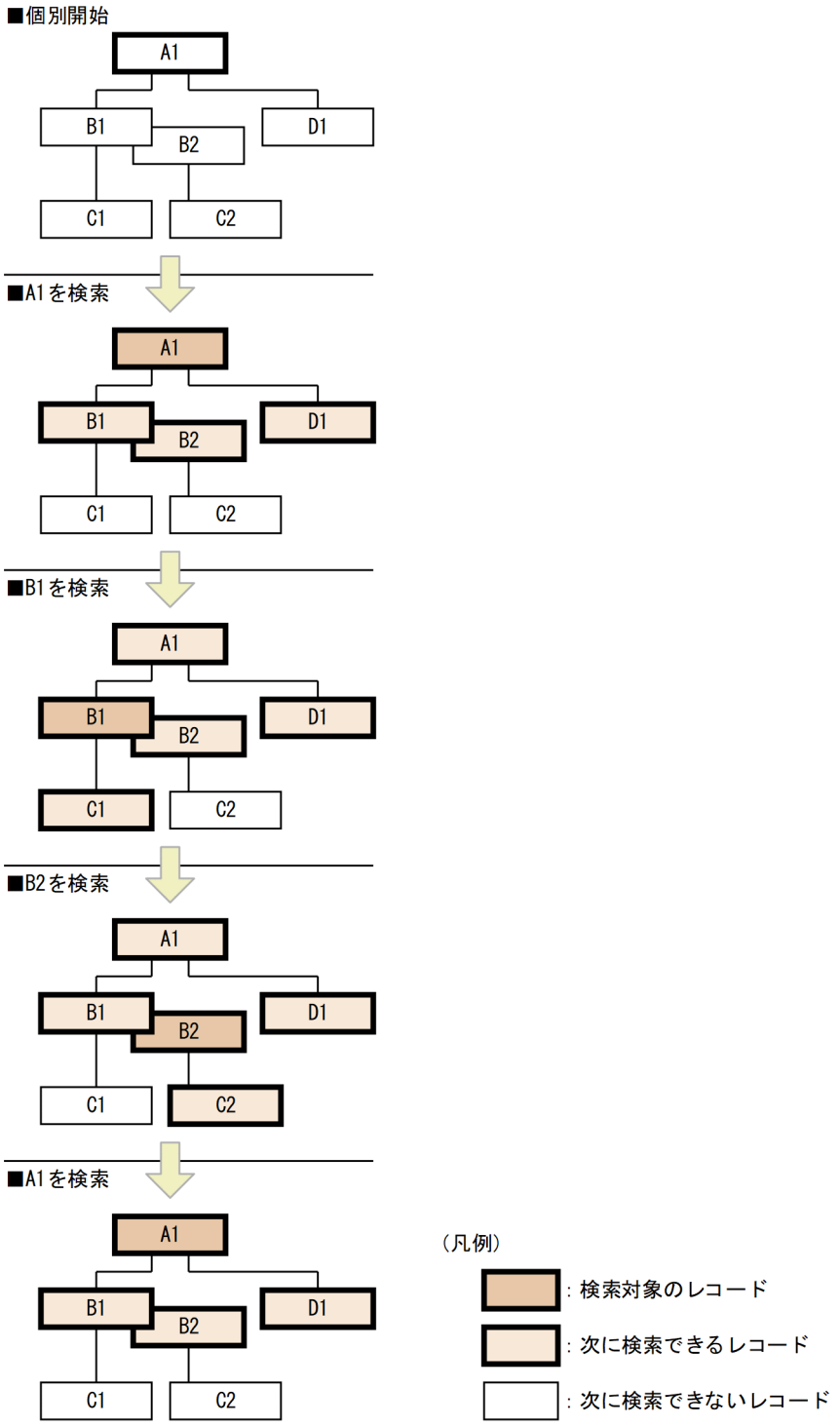
データベースを階層順に従ってアクセスする例を説明します。

レコードの検索，格納，および位置指示子の位置づけを行うと，レコードの位置づけが行われ，そのレコードの子レコードに対してアクセスができるようになります。

同一レコード型の別レコードに対してレコードの位置づけを行うと，前回アクセスしたレコードの子レコードに対してはアクセスできなくなり，今回アクセスしたレコードの子レコードに対してアクセスできるようになります。

SD FMB の SDB データベースのアクセス例を次の図に示します。この図では，レコード A1→B1→B2→A1 の順序で検索しています。

図 2-36　SD FMB の SDB データベースのアクセス例



注

COBOL の UAP から DML を実行した場合，上記の図中の個別開始はシステムが内部的に実行します。

ポイント

HiRDB/SD では，レコードの検索や格納などを行うと，そのレコードの位置情報を保持します。この情報を位置指示子といいます。HiRDB/SD は，位置指示子を使用して SDB データベースにアクセスします。位置指示子については，「2.5.3　位置指示子」を参照してください。

なお，検索結果が NOT FOUND（SQLCODE=100）となった場合，対象レコードのレコード位置指示子は空値になります。

(3)　検索の順序，開始位置および方向

検索は順序，開始位置および方向を決定してレコードを検索します。

順序，開始位置および方向については，「17.3.3　検索の順序，開始位置および方向」を参照してください。

(4)　レコードの特定方法

レコードの検索では，次の方法でレコードを特定します。

- DML の内容および，位置指示子の状態から検索の順序，開始位置および方向を決定します。順序，開始位置および方向については，「17.3.3　検索の順序，開始位置および方向」を参照してください。
- 条件が指定されていない場合は，開始位置のレコードを検索結果とします。条件が指定されている場合は，開始位置から順序に従いレコードデータを評価し，最初に条件が真となるレコードを検索結果とします。

(5)　レコードの位置づけ

レコード型内の検索時は，レコードに位置指示子を位置づけておく必要はありません。親子集合型内の検索時は，親子集合位置指示子の親レコードを位置づけておく必要があります。レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

2.5.6　位置指示子の位置づけ【SD FMB】

SD FMB の SDB データベース中のレコードを検索し，レコード実現値に対してレコードの位置づけを行います。

(1)　位置づけの種類

位置づけには，次の 3 つの検索があります。

- レコード型内の検索

  レコード位置指示子を使用した検索です。常にインデクスを使用した検索となります。ルートレコード，および二次インデクスを定義したレコードに対して実行できます。

- 親子集合型内の検索

  親子集合位置指示子を使用した検索です。特定の親レコード実現値配下の子レコードを検索します。常にポインタを使用した検索となります。子レコードに対して実行できます。

- 位置指示子指定の検索

  DML に指定した位置指示子が指すレコードを検索します。

(2)　位置づけの順序，開始位置および方向

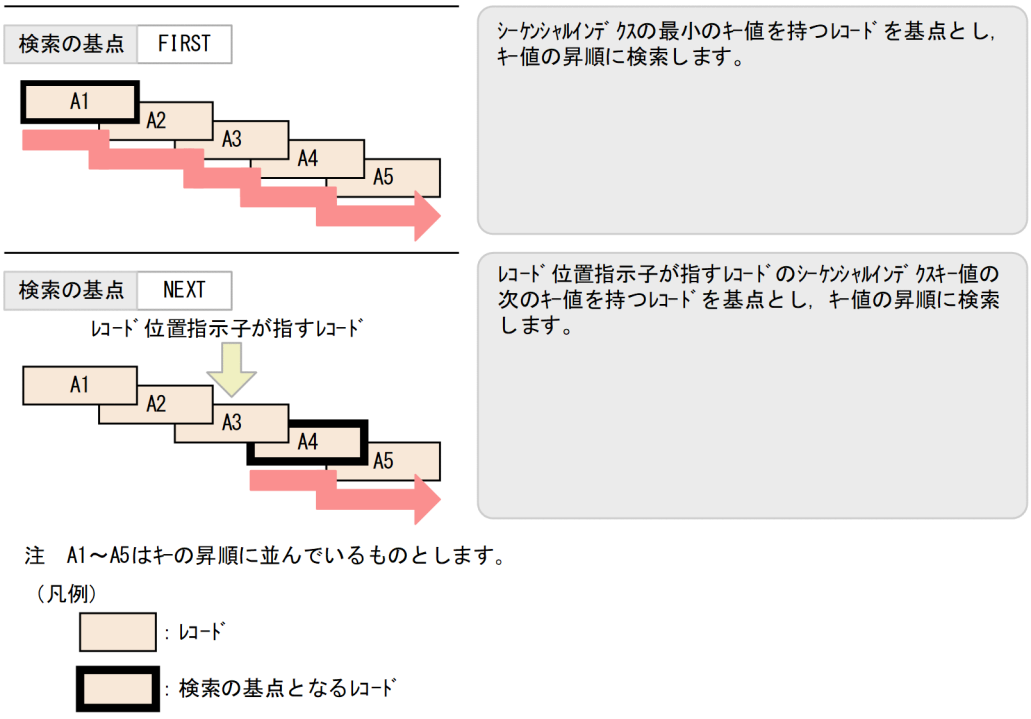
レコード型内の検索，および親子集合型内の検索では，順序，開始位置および方向を決定してレコードを検索します。

順序，開始位置および方向については，「17.3.3　検索の順序，開始位置および方向」を参照してください。

(3)　レコードの検索範囲の例

検索の基点となるレコードと検索の方向の例を次の図に示します。

図 2-37　検索の基点となるレコードと検索の方向と検索範囲



(4)　レコードの特定方法

レコード型内の検索，および親子集合型内の検索では，次の方法でレコードを特定します。

- DML の内容および，位置指示子の状態から検索の順序，開始位置および方向を決定します。順序，開始位置および方向については，「17.3.3　検索の順序，開始位置および方向」を参照してください。

- 条件が指定されていない場合は，開始位置のレコードを検索結果とします。条件が指定されている場合は，開始位置から順序に従いレコードデータを評価し，最初に条件が真となるレコードを検索結果とします。

位置指示子指定の検索の場合は，DML に指定した位置指示子が指すレコードを検索結果とします。

(5)　レコードの位置づけ

レコード型内の検索時は，レコードに位置指示子を位置づけておく必要はありません。親子集合型内の検索，および位置指示子指定の検索時は，親子集合位置指示子の親レコードを位置づけておく必要があります。レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

なお，検索結果が NOT FOUND（SQLCODE=100）となった場合，対象レコードのレコード位置指示子は空値になります。

2.5.7　レコードの格納

SDB データベースに，1 件のレコードを格納します。TP1/FSP が提供するライブラリ関数を使用する場合，格納するデータはデータ格納エリアに設定します。DML を使用する場合，格納するデータは埋込み変数に設定します。格納されたレコードに対しては，レコードの位置づけが行われます。

なお，ユーザキーが定義されている 4V MAM または 4V TAM の SDB データベースにレコードを格納する場合，格納するユーザキーには LAST ポインタが示すレコードのユーザキーよりも，大きな値を設定する必要があります。

(1)　USER ポインタの設定【4V FMB】

4V FMB の SDB データベースの子レコードを格納する場合，親レコード型に USER ポインタの定義があれば（SDB データベース格納定義の OWNER POINTER FOR 句に FIRST LAST USER を指定していれば），USER ポインタを設定できます。

(2)　レコードの格納順序

レコード格納時の格納順序は，レコードによって異なります。レコードの格納順序について，次に説明します。

なお，レコードの配置制御（具体的に，どのページにどのようにレコードを格納するか）については，次の項目を参照してください。

- 「2.6　レコードの配置制御（サブページ分割をしない場合）」
- 「2.7　レコードの配置制御（サブページ分割をする場合）」

(a) ルートレコードの格納順序

ルートレコードに格納順序はありません。

なお，ルートレコードの検索に使用するシーケンシャルインデクスは，データベースキーの昇順に作成されます。そのため，ルートレコードは，データベースキーの昇順に検索されます。

(b) 子レコードの格納順序

子レコードは次のどれかの方法で格納されます。子レコードの格納方法は，子レコード型の定義でユーザキーを定義しているかどうかで異なります。

- キー順での格納（SDB データベース定義の ORDER SORTED DUPLICATES 指定）

  子レコード型の定義でユーザキーを定義している場合，格納するレコード中のユーザキーの値に従って，HiRDB/SD がレコードの挿入位置を検索します。その検索結果に従って，キーの昇順または降順にレコードが格納されます。

  同一キー値のレコードに対する扱いは次のようになります。

  - ORDER SORTED DUPLICATES に PROHIBITED を指定した場合

    同一キー値のレコードは格納されません。

  - ORDER SORTED DUPLICATES に FIRST を指定した場合

    同一キー値のレコードは，同一キー値のレコード群の先頭に格納されます。

  - ORDER SORTED DUPLICATES に LAST を指定した場合

    同一キー値のレコードは，同一キー値のレコード群の最後尾に格納されます。

- 子レコード群の最後尾への格納（SDB データベース定義の ORDER LAST 指定）

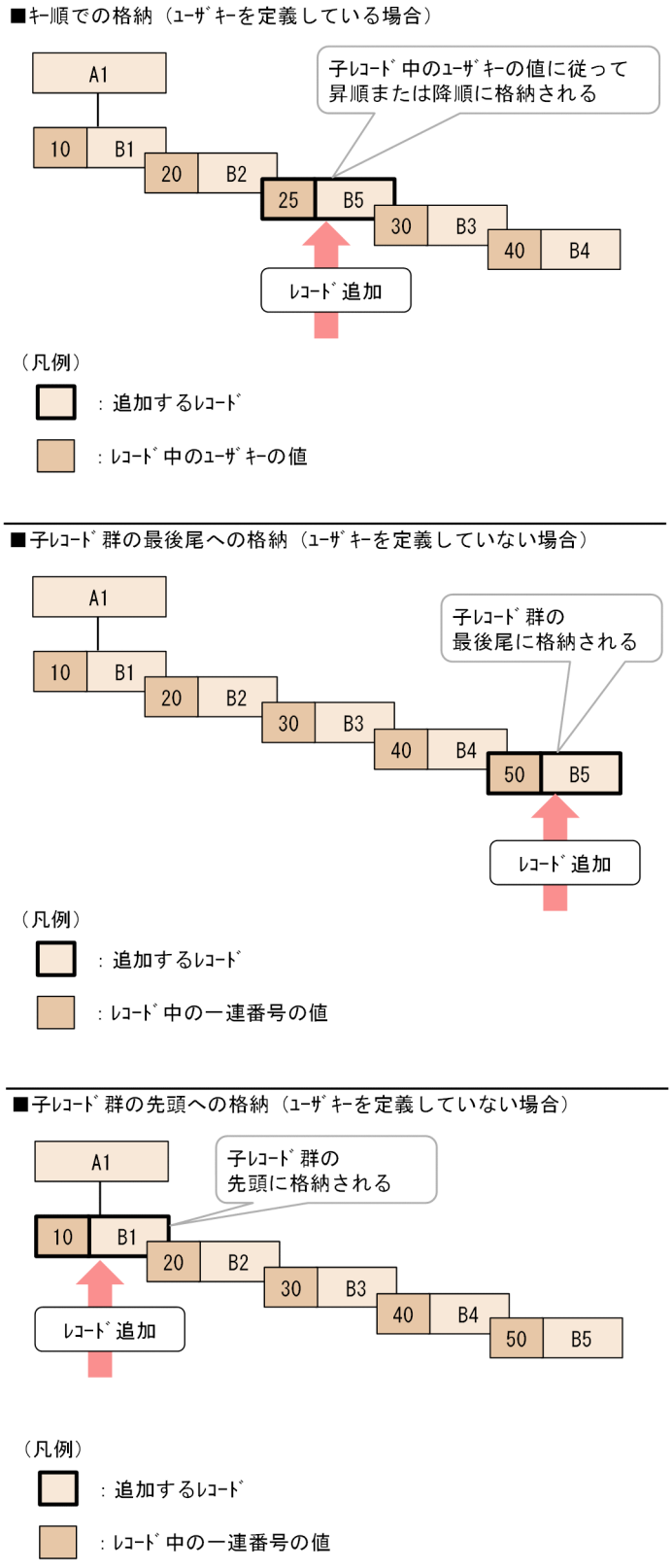
  子レコード型の定義でユーザキーを定義していない場合，HiRDB/SD がレコードの子レコード群の最後尾を検索します。その検索結果に従って，レコードが格納されます。

- 子レコード群の先頭への格納（SDB データベース定義の ORDER FIRST 指定）

  子レコード型の定義でユーザキーを定義していない場合，HiRDB/SD がレコードの子レコード群の先頭を検索します。その検索結果に従って，レコードが格納されます。

それぞれのレコードの格納方法を次の図に示します。

図 2-38　レコードの格納方法



(3)　レコードの位置づけ

ルートレコードの格納時は，レコードに位置指示子を位置づけておく必要はありません。

子レコードの格納時は，親レコードに対してレコードの検索，またはレコードの格納を実行して，位置指示子を親レコードに位置づけておく必要があります。

ただし，4V AFM の SDB データベースの場合は，位置指示子を親レコードに位置づけておく必要はありません（親レコードになる仮想ルートレコードの位置づけを HiRDB/SD が内部的に行います）。

レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

2.5.8　レコードの更新

SDB データベースの，1 件のレコードの内容を更新します。TP1/FSP が提供するライブラリ関数を使用する場合，更新するデータはデータ格納エリアに設定します。DML を使用する場合，更新するデータは埋込み変数に設定します。

(1)　USER ポインタの設定・解除【4V FMB】

4V FMB の SDB データベースの子レコードを更新する場合，親レコード型に USER ポインタの定義があれば（SDB データベース格納定義の OWNER POINTER FOR 句に FIRST LAST USER を指定していれば），USER ポインタの設定および解除もできます。このとき，レコード実現値を更新しないで，USER ポインタだけを変更することもできます。

(2)　レコードの位置づけ

レコードを更新する場合は，位置指示子を対象のレコードに位置づけておく必要があります。

レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

2.5.9　レコードの削除

SDB データベースから，1 件以上のレコードを削除します。

削除するレコードが最下位のレコードである場合，そのレコードだけを削除します。削除するレコードに子レコードが存在する場合，子レコードも同時に削除します。

4V FMB の SDB データベースの場合，SDB データベースを操作する API の一括オプションに'O'を指定することで，最下位のレコードをすべて削除することもできます。ただし，ルートレコードは最下層のレコードであっても，一括オプションに'O'を指定した削除はできません。ルートレコードを削除する場合には，一括オプションに'S'を指定してください。

(1)　USER ポインタの解除【4V FMB】

USER ポインタの指すレコードを削除した場合，USER ポインタは解除されます。

(2)　レコードの位置づけ

レコードを削除する場合は，削除するレコードに位置指示子を位置づけておく必要があります。

なお，4V FMB の SDB データベースで最下位のレコードをすべて削除する場合は，削除する最下位のレコードのどれかに位置指示子を位置づけておく必要があります。また，4V FMB の SDB データベースでルートレコードを削除する場合は，構成要素名に DBKEY を指定した＝条件の検索で，削除対象のレコードに位置指示子を位置づけておく必要があります。

レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

2.5.10　レコードの一括削除【4V DAM，4V SAM】

SDB データベースから，指定したデータベースキー（一連番号を除く）に対応したレコードを一括して削除します。

レコードの一括削除は，4V DAM または 4V SAM の SDB データベースの場合に実行できます。

> **参考**
>
> レコードの一括削除では，レコードの管理情報を更新して，レコードが削除された状態にします。実際のレコード自体は削除されません。

(1)　レコードの位置づけ

レコードの一括削除をする場合は，削除するレコードに位置指示子を位置づけておく必要はありません。

レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

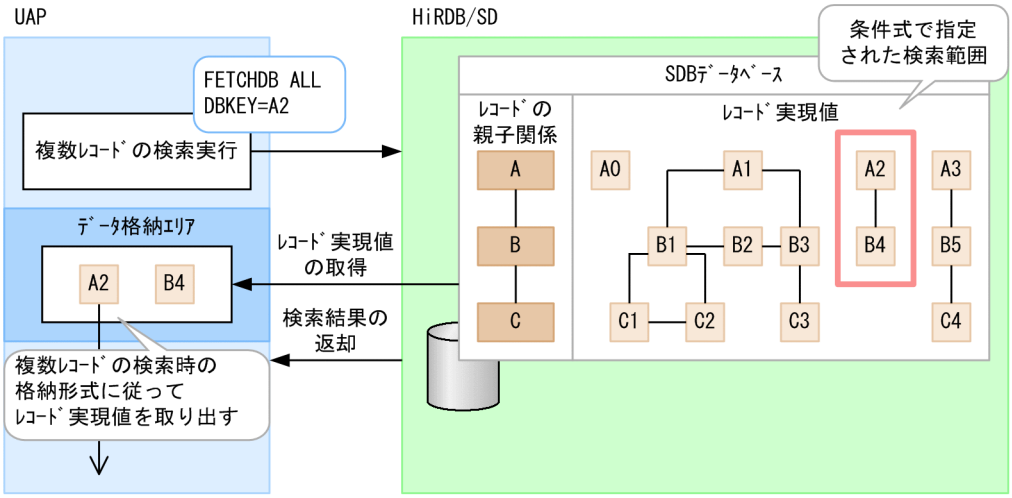
2.5.11　複数レコードの検索【4V FMB】

複数レコードの検索とは，ルートレコード以下のすべてのレコードを検索し，データ格納エリアに一括して取得する機能です。レコードを一括して検索することによって，レコードを 1 件ずつ検索するよりも，検索時のオーバヘッドを抑えることができます。

複数レコードの検索は，4V FMB の SDB データベースの場合に実行できます。

複数レコードの検索の概要を次の図に示します。

図 2-39　複数レコードの検索の概要



(1)　複数レコードの検索の詳細

複数レコードの検索の詳細を次に示します。

- 複数レコードの検索は，ルートレコードから開始し，レコード型を一筆書き順に検索します。レコード型の一筆書き順については，「11.5.1(3)　定義句の指定順序」の「図 11-11　RECORD 句および SET 句の指定順序（一筆書き順）の規則」を参照してください。
- 検索するルートレコードの条件を指定すると，検索する範囲を限定できます。
- 取得したレコードは，データ格納エリアにまとめて格納されます。
- 複数レコードの検索は，ファミリ単位で行われます。1 ファミリの検索が完了すると，データ格納エリアに未使用領域があっても，検索を継続しないで UAP にリターンします。
- 1 ファミリの検索中であっても，データ格納エリアが満杯になると検索を中断して，UAP にリターンします。
- 複数レコードの検索時は，データ格納エリアに対するレコードの格納形式が異なります。レコードを取り出す際には，複数レコードの検索時のレコードの格納形式に従って取り出します。複数レコードの検索時の，データ格納エリアに対するレコードの格納形式については，「付録 M　複数レコードの検索時の，データ格納エリアに対するレコードの格納形式【4V FMB】」を参照してください。

(2)　検索時の動作と指示コードの指定

複数レコードを検索する API では，検索時の動作を指示コードで指定できます。

複数レコードの検索で指定する指示コードと検索時の動作を次の表に示します。

表 2-16　複数レコードの検索で指定する指示コードと検索時の動作

項番	指示コード	検索時の動作
1	'F'	条件式で指定されたルートレコードから検索を開始します。

項番	指示コード	検索時の動作
2	'N'	前回，複数レコードの検索で検索したレコードの，次のレコードから検索を開始します。
3	'S'	前回，複数レコードの検索で検索したファミリの，次のファミリ（次のルートレコード）から検索を開始します。

(3)　複数レコードの検索の流れ

複数レコードの検索の流れを次に示します。実際の API の指定方法については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

1. 複数レコードの検索専用オプションを指定して，個別開始を実行します。
2. 指示コードに'F'を指定して，複数レコードの検索を実行します。
   検索するルートレコードの条件を指定すると，検索する範囲を限定できます。条件を指定しない場合はすべてのレコードが検索対象となります。
3. 必要に応じて，データ格納エリアからデータを取り出します。
4. 2.の複数レコードの検索の状態によって，次のどちらかの検索を実行します。

   1 ファミリのレコードをすべて検索した場合

   指示コードに'N'，または'S'を指定して，再度複数レコードの検索を実行し，次のルートレコードのファミリを検索します。

   1 ファミリの検索の途中でデータ格納エリアが満杯になった場合

   現在検索しているファミリを継続して検索する場合は，指示コードに'N'を指定して，再度複数レコードの検索を実行します。

   現在検索しているファミリの検索を中断して，次のファミリを検索する場合は，指示コードに'S'を指定して，再度複数レコードの検索を実行します。
5. 検索対象がなくなり，検索結果が NOT FOUND になるまで 4.を繰り返します。

(4)　レコードの位置づけ

複数レコードの検索時は，レコードに位置指示子を位置づけておく必要はありません。

レコードの位置づけについては，「2.5.3(1)　レコードの位置づけ」を参照してください。

(5)　複数レコード検索時の留意事項

複数レコード検索時の留意事項を次に示します。

(a)　データ格納エリアの領域長の指定

レコードを取得するデータ格納エリアのサイズは，SDB データベースを操作する API でユーザが指定します。データ格納エリアが小さ過ぎると，ファミリの検索中に領域が満杯になります。そのため，1 ファミリの取得に必要となる，複数レコード検索の実行回数が増加し，処理性能が低下するおそれがあります。

逆に，データ格納エリアが大き過ぎると，レコードの格納に使用されない余分な領域が確保されることによって，処理性能が低下するおそれがあります。

これらを回避するため，複数レコードの検索時のレコードの格納形式，およびUAP統計レポートに出力される「複数レコードの検索時のデータ格納エリアの余り領域長」の情報を参考にして，データ格納エリアのサイズに適切な領域長を指定してください。

複数レコードの検索時の，データ格納エリアに対するレコードの格納形式については，「付録M　複数レコードの検索時の，データ格納エリアに対するレコードの格納形式【4V FMB】」を参照してください。また，UAP統計レポートについては，「16.5.5　UAP統計レポート機能」を参照してください。

(b)　個別開始中の複数レコードの検索以外の，SDBデータベースの操作禁止

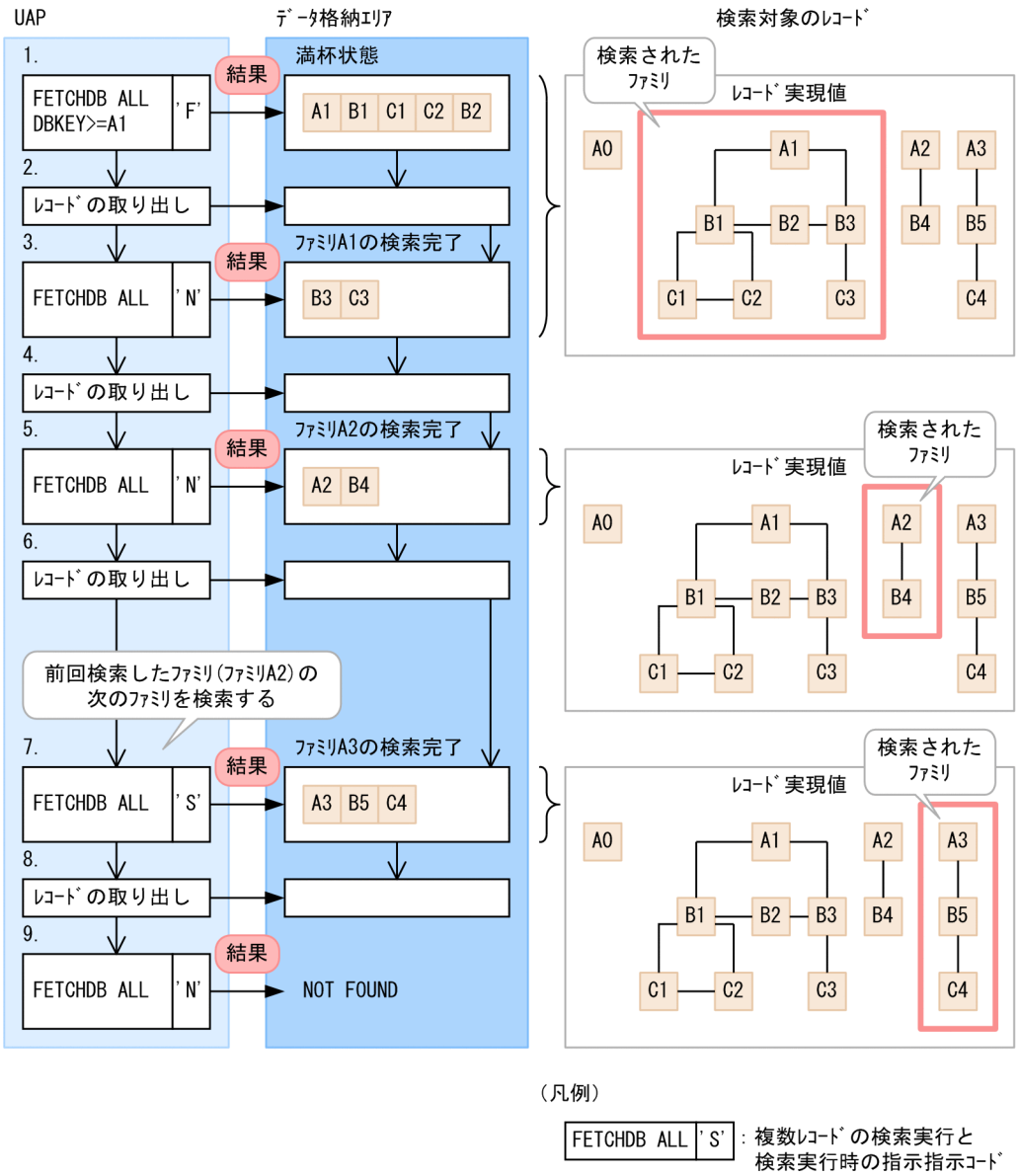
複数レコードの検索をする個別開始では，複数レコードの検索と個別終了しかできません（複数レコードの検索以外のSDBデータベースの操作はできません）。

(6)　複数レコードの検索例

複数レコードの検索例を次に示します。

なお，ここでは，ルートレコードA0のファミリをファミリA0と表記します。以降の例に使用するSDBデータベースには，ファミリA0，ファミリA1，ファミリA2，ファミリA3の4つのファミリがあるものとします。

(a) 検索例 1（検索範囲の指定と指示コード'F'および'N'の指定）



[説明]

1. 検索を開始するルートレコードを指定する条件を付加し，指示コードに'F'を指定して複数レコードの検索を実行します。

   ファミリ A1 はデータ格納エリアより大きなデータで構成されているため，この場合，ファミリ A1 の検索の途中でデータ格納エリアが満杯になります。

2. データ格納エリアからデータを取り出します。

3. ファミリ A1 の残りのレコードを取得するため，指示コードに'N'を指定して複数レコードの検索を実行します。

参考

ファミリの検索途中で指示コードに'F'を再度指定すると，検索途中であっても，それまでの検索過程は無効となり，指定された条件に従って再度ルートレコードから検索を開始します。

4. データ格納エリアからファミリ A1 の残りのデータを取り出します。
5. 次のファミリを検索するため，指示コードに'N'を指定して複数レコードの検索を実行します。

   ファミリ A2 はデータ格納エリアにすべて格納できるため，1 回の検索ですべてのレコードを取得できます。
6. データ格納エリアからデータを取り出します。
7. 指示コードに'S'を指定して複数レコードの検索を実行します。

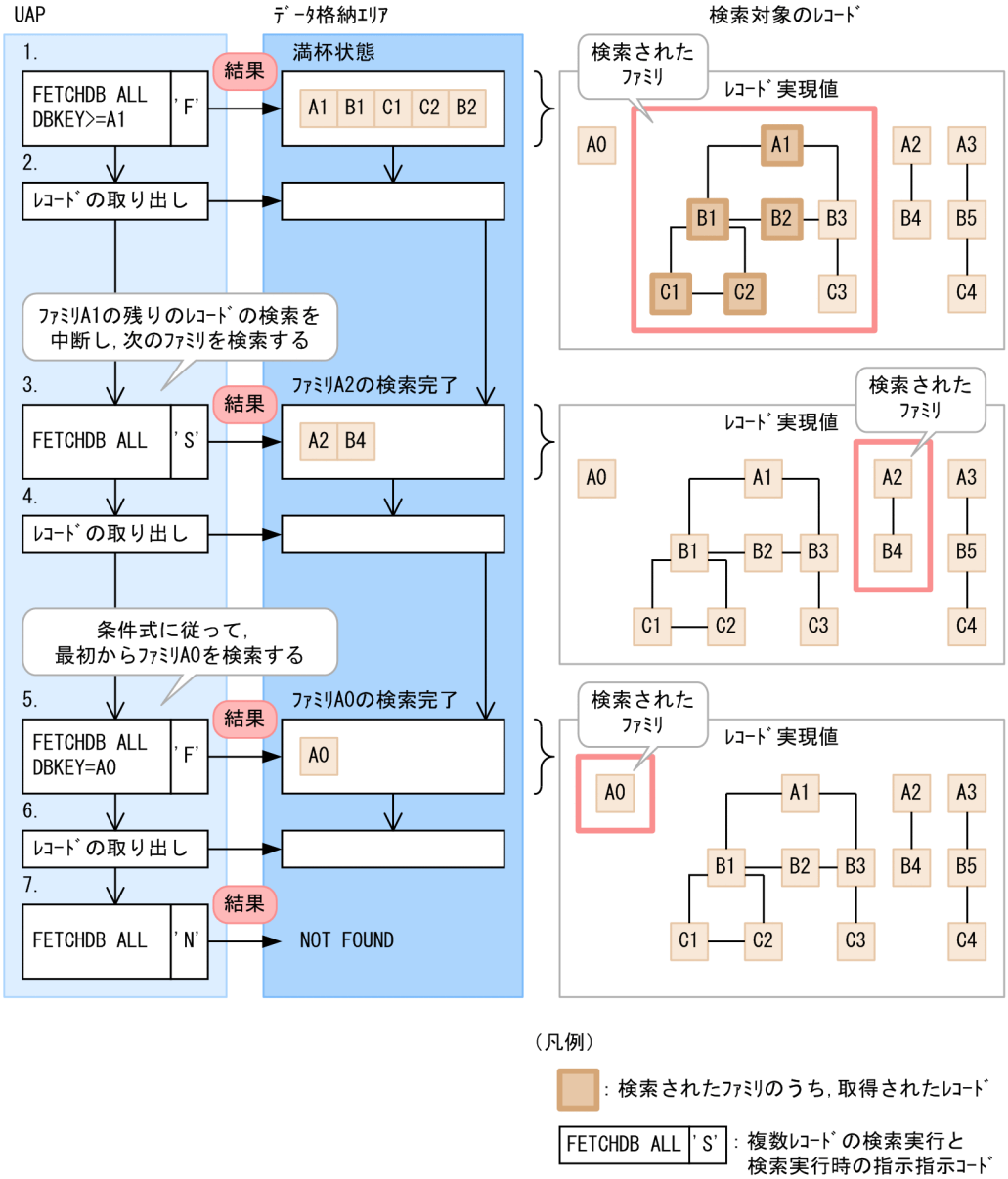
   1 つのファミリの検索が終了している状態で，指示コードに'S'を指定して複数レコードの検索を実行すると，次のファミリのルートレコードから検索を開始します。

   なお，指示コードに'S'を指定して複数レコードの検索を実行すると，ファミリの検索途中であっても，次のファミリのルートレコードから検索を開始します。

   ファミリ A3 はデータ格納エリアにすべて格納できるため，1 回の検索ですべてのレコードを取得できます。
8. データ格納エリアからデータを取り出します。
9. 次のファミリを検索するため，指示コードに'N'を指定して複数レコードの検索を実行します。

   次のファミリは存在しないので，検索結果として NOT FOUND が返却され，検索が終了します。

(b)　検索例 2（検索範囲の指定と指示コード'F'および'S'の指定）



[説明]

1. 検索を開始するルートレコードを指定する条件を付加し，指示コードに'F'を指定して複数レコードの検索を実行します。

   ファミリ A1 はデータ格納エリアより大きなデータで構成されているため，この場合，ファミリ A1 の検索の途中でデータ格納エリアが満杯になります。

2. データ格納エリアからデータを取り出します。

3. ファミリ A1 の残りのレコードの検索を中断して，次のファミリから検索を開始します（指示コードに'S'を指定して複数レコードの検索を実行します）。

   ファミリ A2 はデータ格納エリアにすべて格納できるため，1 回の検索ですべてのレコードを取得できます。

4. データ格納エリアからデータを取り出します。
5. 検索条件を変更し，指示コードに'F'を指定して複数レコードの検索を実行します。

   この場合，指定された条件に従って再度検索を開始するため，ルートレコードのレコード実現値がA0のファミリを検索します。

   ファミリA0はデータ格納エリアにすべて格納できるため，1回の検索ですべてのレコードを取得できます。
6. データ格納エリアからデータを取り出します。
7. 次のファミリを検索するため，指示コードに'N'を指定して複数レコードの検索を実行します。

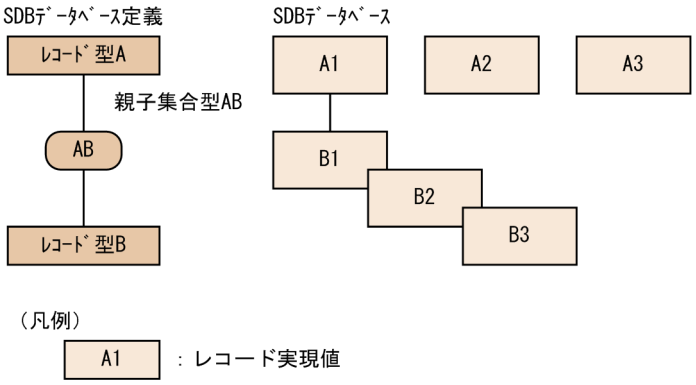
   検索条件に該当するファミリは存在しないので，検索結果としてNOT FOUNDが返却され，検索が終了します。

2.5.12 SDBデータベースの操作と位置指示子の関係

SDBデータベースの操作と位置指示子の関係を，それぞれの操作ごとに説明します。

以降の例で使用するSDBデータベースを次の図に示します。

図 2-40 SDBデータベースの操作と位置指示子の関係の説明で使用するSDBデータベース

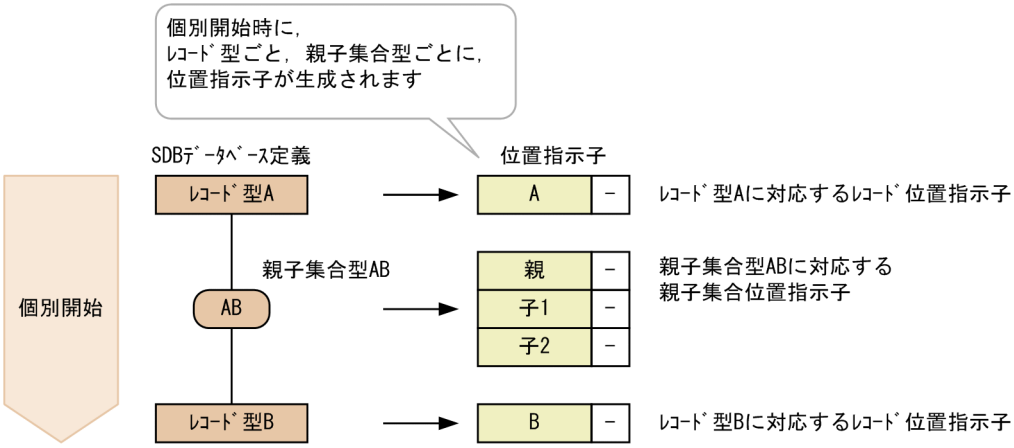


(1) 個別開始

個別開始時（SDBデータベース単位の操作開始時）に，レコード型，親子集合型ごとに位置指示子が生成され，空値が設定されます。

この例では，レコード型A，レコード型B，および親子集合型ABに対して，位置指示子が生成されます。

図 2-41　個別開始



(2)　ルートレコードの検索

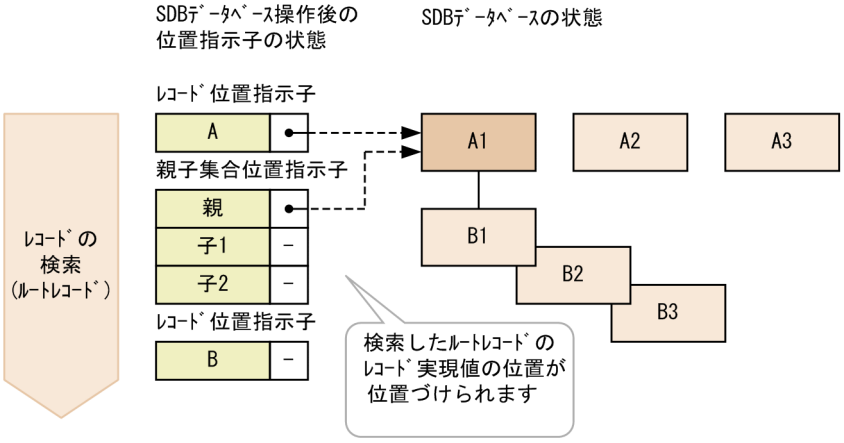
ルートレコードを検索する時点で，位置指示子にレコードが位置づけられている必要はありません。

ルートレコード A1 を検索すると，次の位置指示子にルートレコード A1 が位置づけられます（ルートレコード A1 の位置が位置指示子に保持されます）。

- レコード型 A のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

ルートレコード A1 を検索した場合の例を次に示します。

図 2-42　ルートレコードの検索



(3)　子レコードの検索（親子集合型内の検索）

子レコード B2 を検索する場合，次の位置指示子にルートレコード A1 が位置づけられている必要があります。

- 親子集合型 AB の親子集合位置指示子の親

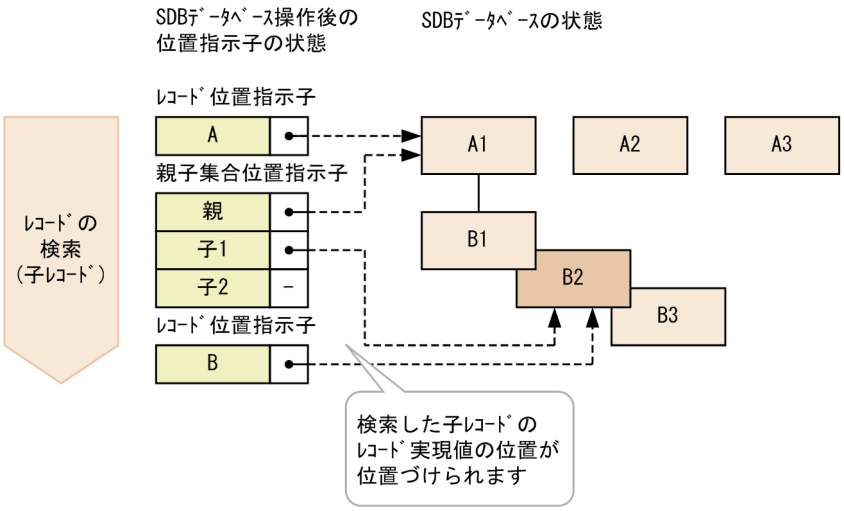
つまり，子レコード B2 を検索する前に，親レコードであるルートレコード A1 を検索して，レコードを位置づけておく必要があります。

子レコード B2 を検索すると，次の位置指示子に子レコード B2 が位置づけられます。

- レコード型 B のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子の子 1

子レコード B2 を検索した場合の例を次に示します。

図 2-43　子レコードの検索



(4)　子レコードの検索（レコード型内の検索）【SD FMB】

レコードを検索する時点で，位置指示子にレコードが位置づけられている必要はありません。

レコード B2 を検索すると，次の位置指示子にレコード B2 が位置づけられます（レコード B2 の位置が位置指示子に保持されます）。

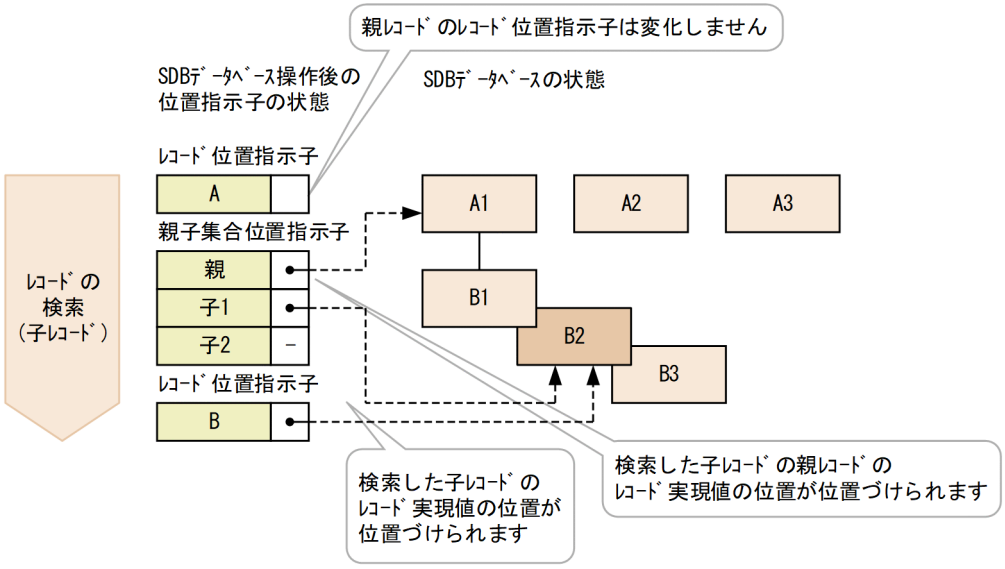
- レコード型 B のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子の子 1

また，親子集合型 AB の親子集合位置指示子の親にレコード B2 の親レコードの A1 が位置づけられます（レコード A1 の位置が位置指示子に保持されます）。

なお，レコード A のレコード位置指示子は変化しません。

子レコード B2 を検索した場合の例を次に示します。

図 2-44　子レコードの検索



(5)　位置指示子の位置づけ（レコード型内の検索）【SD FMB】

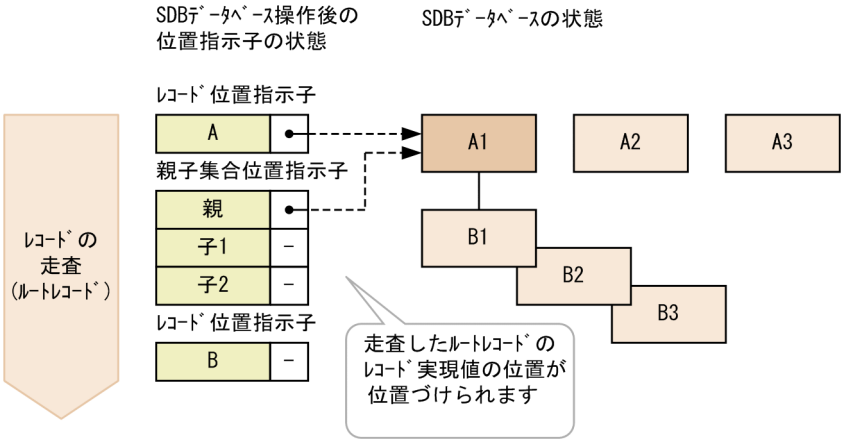
位置指示子にレコードが位置づけられている必要はありません。

ルートレコード A1 を走査し，レコードが特定されると，次の位置指示子にルートレコード A1 が位置づけられます（ルートレコード A1 の位置が位置指示子に保持されます）。

- レコード型 A のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

ルートレコード A1 を検索した場合の例を次に示します。

図 2-45　ルートレコードの走査



(6)　位置指示子の位置づけ（位置指示子指定の検索）【SD FMB】

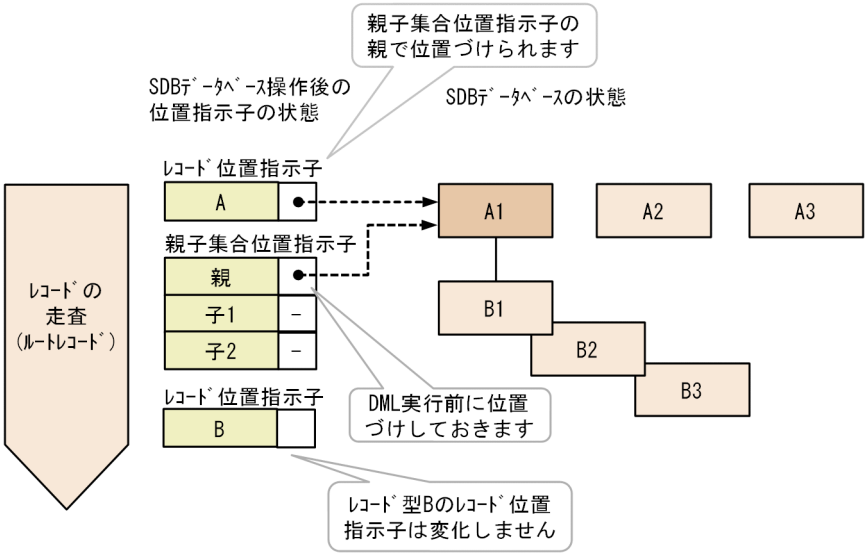
親子集合型 AB を指定した場合，親子集合型 AB の親子集合位置指示子の親が位置づけられている必要があります。

親子集合型 AB の親子集合位置指示子の親に A1 が位置づけられている場合の更新結果を次に示します。

- レコード型 A のレコード位置指示子
  A1 が位置づけられます。
- 親子集合型 AB の親子集合位置指示子
  親には A1 が位置づけられます（変化しません）。子は空値になります。

DML 実行後の状態を次に示します。

図 2-46　位置指示子の位置づけ（位置指示子指定の検索）



(7)　ルートレコードの格納（ルートレコードの追加）

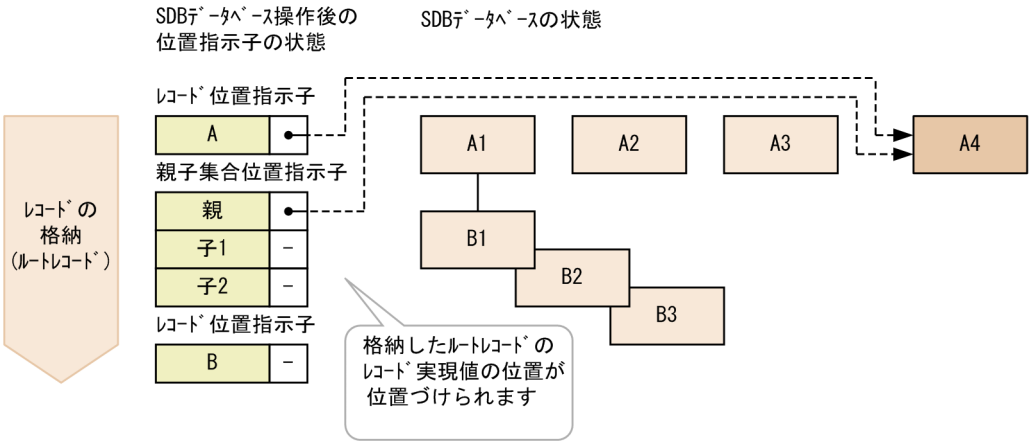
ルートレコードを格納する時点で，位置指示子にレコードが位置づけられている必要はありません。

ルートレコード A4 を格納すると，次の位置指示子にルートレコード A4 が位置づけられます。

- レコード型 A のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

ルートレコード A4 を格納した場合の例を次に示します。

図 2-47　ルートレコードの格納（ルートレコードの追加）



(8)　子レコードの格納（子レコードの追加）

子レコード B4 を格納する場合，次の位置指示子にルートレコード A1 が位置づけられている必要があります。

- 親レコードであるレコード型 A のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

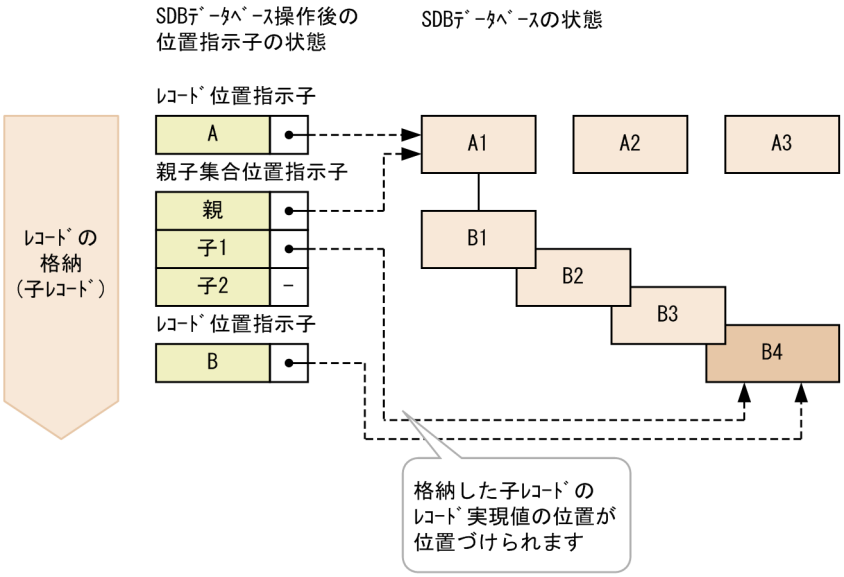
つまり，子レコード B4 を格納する前に，親レコードであるルートレコード A1 を検索して，レコードを位置づけておく必要があります。

子レコード B4 を格納すると，次の位置指示子に子レコード B4 が位置づけられます。

- レコード型 B のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

子レコード B4 を格納した場合の例を次に示します。

図 2-48　子レコードの格納（子レコードの追加）



(9)　子レコードの更新

子レコード B2 の値を更新する場合，次の位置指示子にルートレコード A1 が位置づけられている必要があります。

- 親レコードであるレコード型 A のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

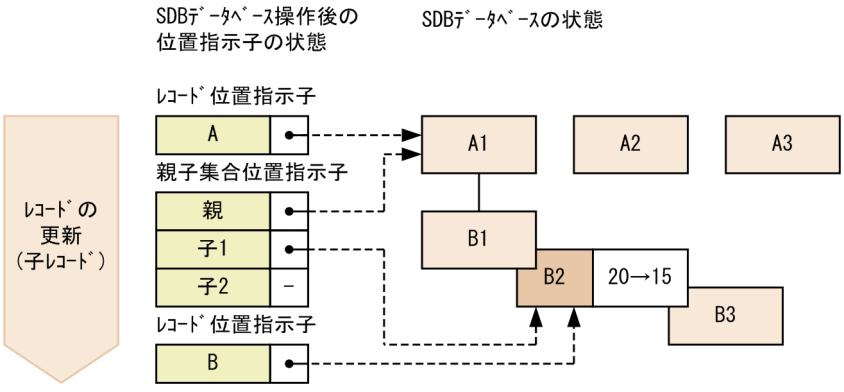
また，次の位置指示子に子レコード B2 が位置づけられている必要があります。

- レコード型 B のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

つまり，子レコード B2 の値を更新する前に，子レコード B2 を検索して，レコードを位置づけておく必要があります。

子レコード B2 のレコード実現値を更新した場合の例を次に示します。

図 2-49　子レコードの更新



(10)　ルートレコードの削除

ルートレコード A1 を削除する場合，次の位置指示子にルートレコード A1 が位置づけられている必要があります。

- レコード型 A のレコード位置指示子

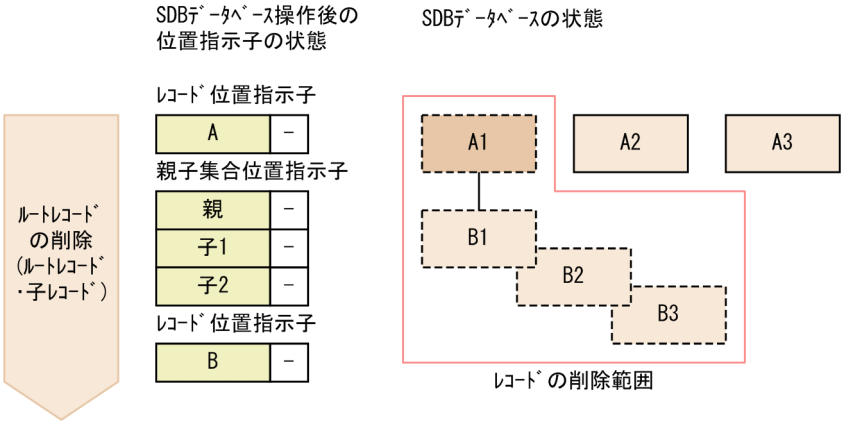
ルートレコード A1 の値を削除する前に，ルートレコード A1 を位置づけておく必要があります。

また，4V FMB の SDB データベースの場合，検索条件にルートレコード A1 の＝条件（構成要素名に DBKEY を指定した＝条件）を指定した検索で位置づけておく必要があります。

ルートレコード A1 を削除すると，レコード位置指示子 A は空値となります。また，ルートレコード型 A の下位のレコード型のレコード位置指示子，親子集合位置指示子も空値となります。

ルートレコード A1 を削除した場合の例を次に示します。

図 2-50　ルートレコード A1 を削除した場合の例



(11)　子レコードの削除

子レコード B2 を削除する場合，次の位置指示子にルートレコード A1 が位置づけられている必要があります。

- 親レコードであるレコード型 A のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

また，次の位置指示子に子レコード B2 が位置づけられている必要があります。

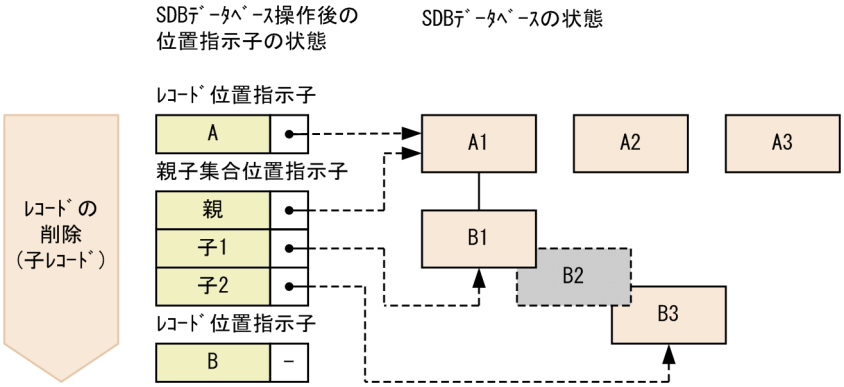
- レコード型 B のレコード位置指示子
- 親子集合型 AB の親子集合位置指示子

つまり，子レコード B2 の値を削除する前に，子レコード B2 を検索して，レコードを位置づけておく必要があります。

子レコードB2を削除すると，親子集合位置指示子には，削除した子レコードB2の前後のレコードが位置づけられ，親子集合位置指示子は複合状態になります。また，レコード型Bのレコード位置指示子は空値となります。

子レコードB2を削除した場合の例を次に示します。

図 2-51　子レコードの削除



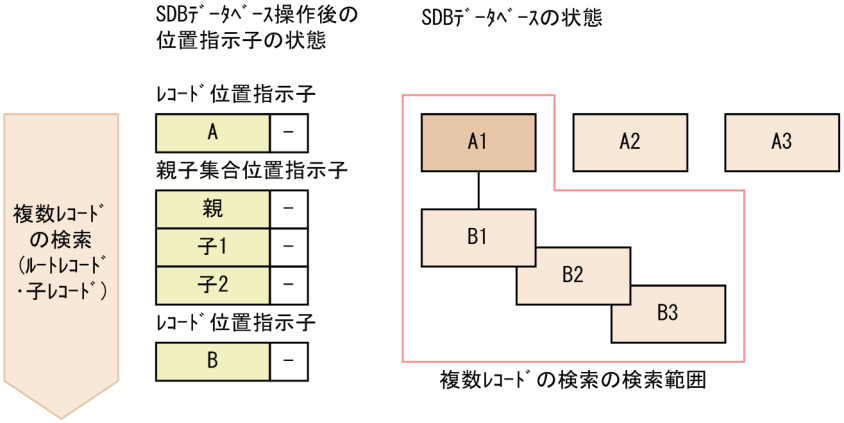
(12)　複数レコードの検索【4V FMB】

複数レコードの検索をする場合，検索時点で位置指示子にレコードが位置づけられている必要はありません。

指定したルートレコード以下のすべてのレコードを検索しても，位置指示子は空値のままとなります。

複数レコードの検索した場合の例を次に示します。

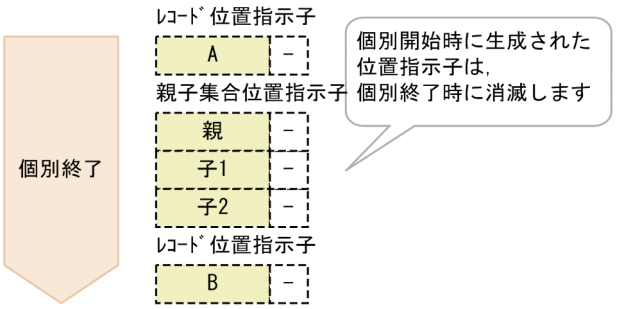
図 2-52　複数レコードの検索



(13)　個別終了およびトランザクションの終了

個別終了時（SDBデータベース単位の操作終了時）およびトランザクションの終了時に，位置指示子は消滅します。

図 2-53　個別終了およびトランザクションの終了



2.5.13　個別開始実行要求オプションを指定した SDB データベースの操作【4V FMB，4V AFM】

4V FMB または 4V AFM の SDB データベースを操作する API で個別開始実行要求オプションを指定すると，UAP から個別開始の要求を行わなくても SDB データベースの操作ができます。

個別開始実行要求オプションは次に示す SDB データベースの操作をする際に指定できます。

- レコードの検索（FETCH）
- 複数レコードの検索（FETCHDB ALL）
- レコードの格納（STORE）
- レコードの一括削除
- 構成情報取得

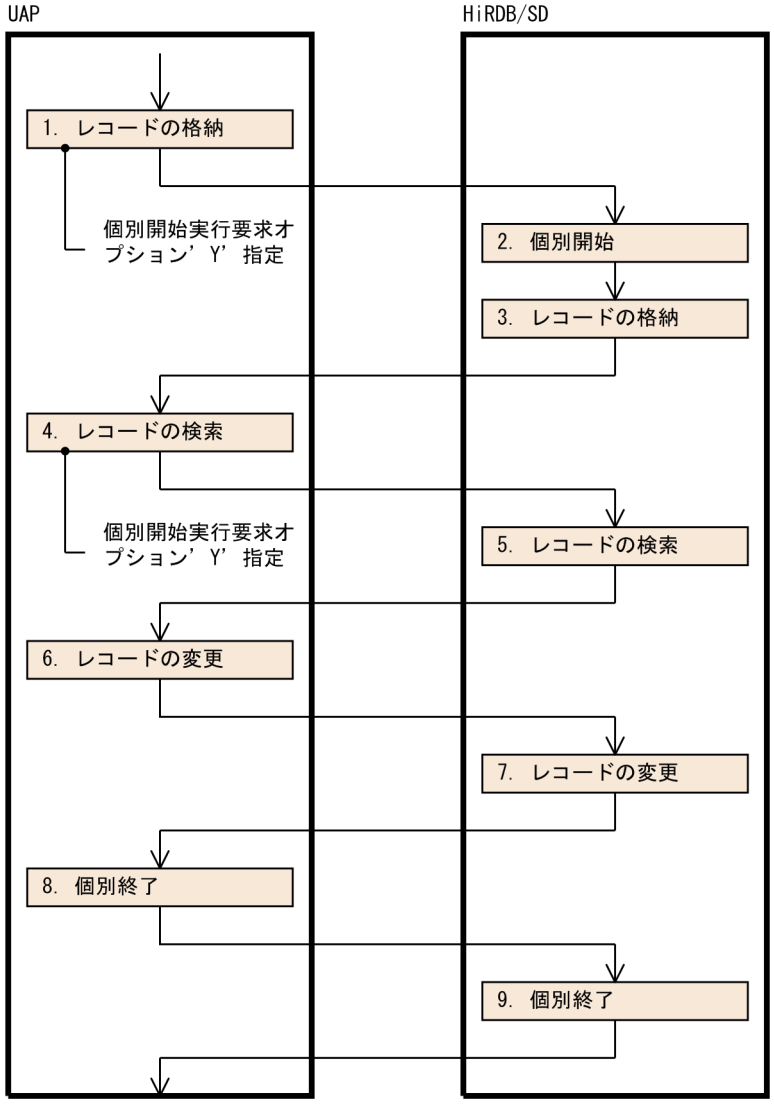
個別開始実行要求オプションの指定方法については，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「個別開始実行要求オプション」を参照してください。

SD FMB の SDB データベースを操作する DML では，個別開始実行要求オプションを指定できません。

(1)　個別開始実行要求オプション指定時の処理の流れ

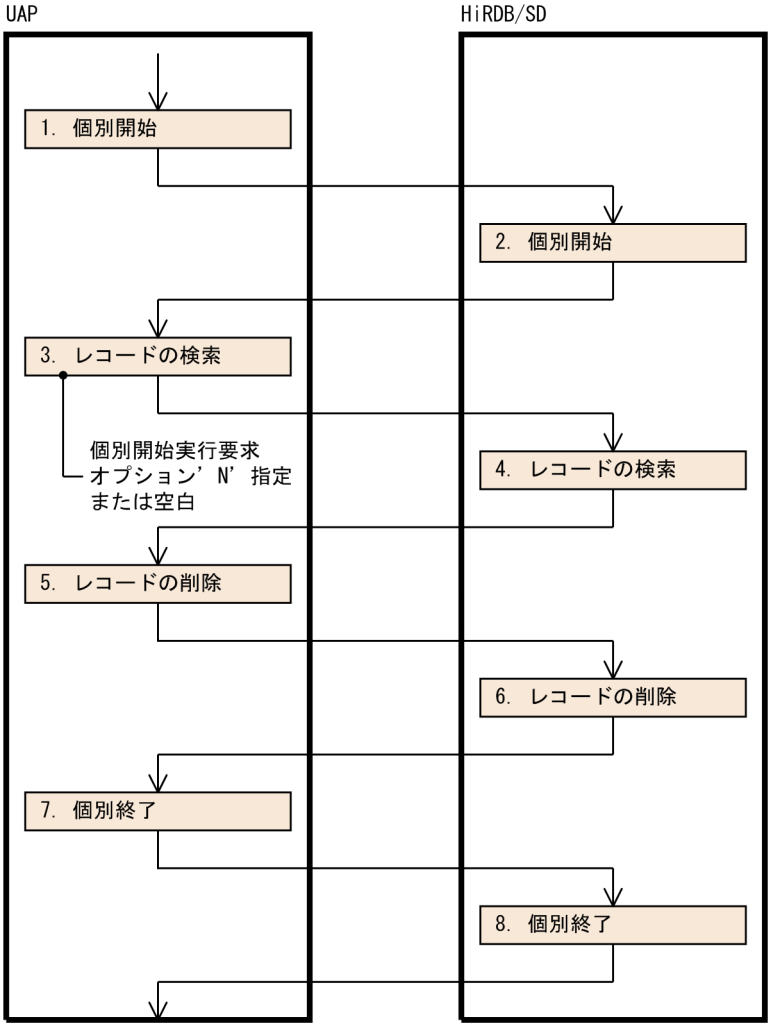
個別開始実行要求オプションを指定する場合の処理の流れを次の図に示します。

図 2-54　SDB データベースを操作する処理の流れ（個別開始実行要求オプションを指定する場合）



個別開始実行要求オプションを指定しない場合の処理の流れを次の図に示します。

図 2-55　SDB データベースを操作する処理の流れ（個別開始実行要求オプションを指定しない場合）



(2)　留意事項

- 個別開始実行要求オプションを指定した場合，UAP からの個別開始の要求は不要になります。ただし，個別開始の処理は実行されるため，個別開始の要求を実行したものとしてメモリ所要量を見積もってください。
- 同一トランザクション内で，個別開始実行要求オプションを指定した SDB データベースの操作と，個別開始の要求を行う SDB データベースの操作を混在して実行できます。
- トランザクション決着しない場合に要求する個別終了については，個別開始実行要求オプションの指定に関係なく要求できます。

2.6　レコードの配置制御（サブページ分割をしない場合）

HiRDB/SD のレコードの配置制御について説明します。HiRDB/SD はレコードを次の表に示すように格納します。

表 2-17　HiRDB/SD のレコードの配置制御（サブページ分割をしない場合）

レコードの配置制御	SDB データベース種別		
	4V FMB	4V AFM	SD FMB
• 同一のルートレコード下のレコードを近傍に配置して格納する。 • 異なるルートレコード下のレコードをページ内に混在させない。 • HiRDB/SD データベース作成ユティリティ（pdsdblod）を使用したデータロード時に，ページ内の未使用領域，セグメント内の空きページを設定してレコードを格納できる。	○	○	○
• レコード実現値ごとに格納するページを切り替え，空きページに格納できる。※ • 1 つのレコード実現値でページを占有できる（1 件/ページ）。※ • SDB データベースを操作する API によるレコード格納時に，ページ内の未使用領域の比率に従った格納ができる。※	○	○	×
ルートレコードの格納時に事前割り当てページをあらかじめ確保しておくことで，ルートレコード下のレコードの格納およびレコードの削除を繰り返し実行した場合でも，連続したデータページ内にレコードを配置できる（この機能を事前ページ割り当て機能といいます）。	○	×	×

（凡例）

○：レコードの配置制御が適用されます。

×：レコードの配置制御は適用されません。

注※

オプションの指定が必要になります。

> **参考**
>
> サブページ分割をする場合の配置制御については，「2.7　レコードの配置制御（サブページ分割をする場合）」を参照してください。

レコード格納時，およびデータロード時の空き領域の作成方法については，「3.2.5　レコード格納時の空き領域の作成（サブページ分割をしない場合）」を参照してください。

SD FMB の SDB データベースの場合，DML によるレコードの格納時に空き領域を作成することはできません。

2.6.1　親レコードと子レコードの近傍配置

ここでは親レコードと子レコードの近傍配置について説明します。

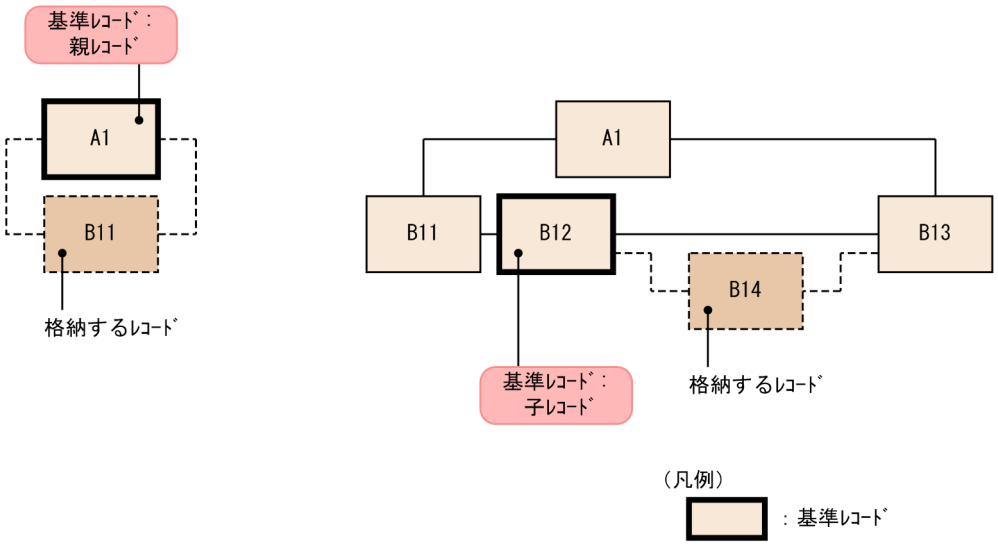
(1)　基準レコード，基準ページ

HiRDB/SD は，基準とするレコードから，レコードを格納する位置を決定します。基準とするレコードを基準レコードといいます。

基準レコードは，格納するレコードの PRIOR ポインタが指すレコードになります。親子集合の場合は，1 件目の子レコードに対する基準レコードは，親レコードとなります。なお，ルートレコードは基準レコードを持ちません。

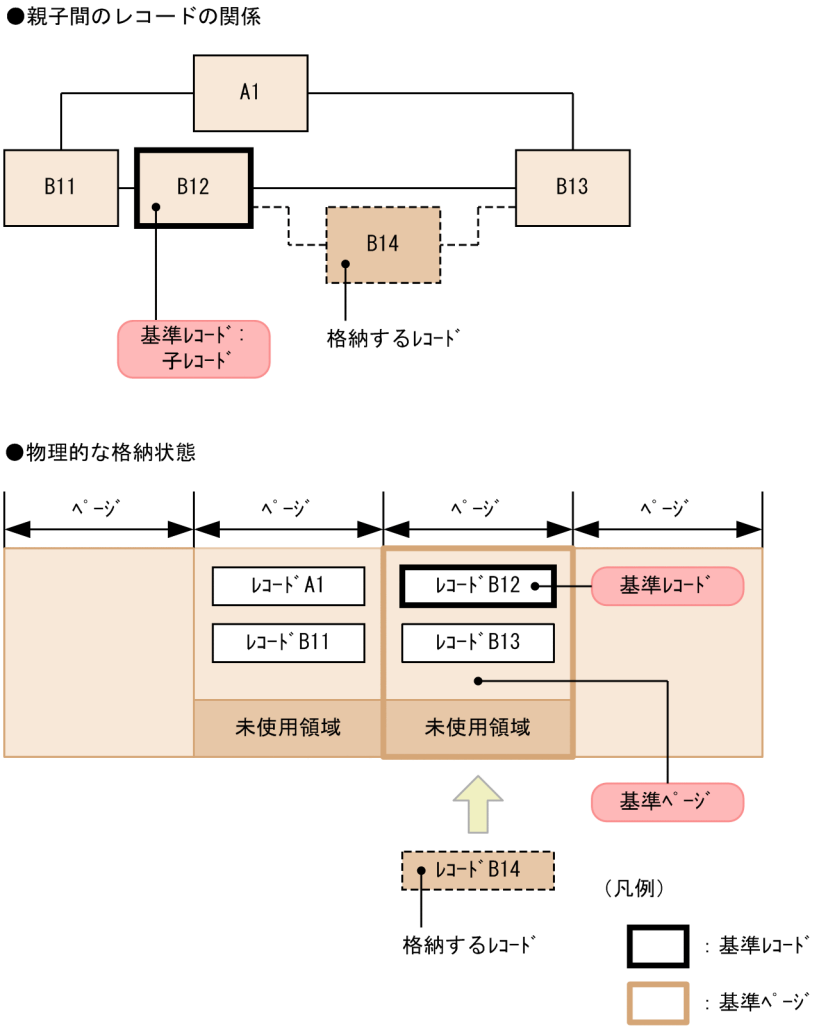
基準レコードの例を次の図に示します。

図 2-56　基準レコードの例



基準レコードが格納されているページを基準ページといいます。HiRDB/SD では，基準ページおよびその近傍のページにレコードを格納することで，親子集合内のレコード群を近傍に配置して格納できます。基準ページへの格納の例を次の図に示します。

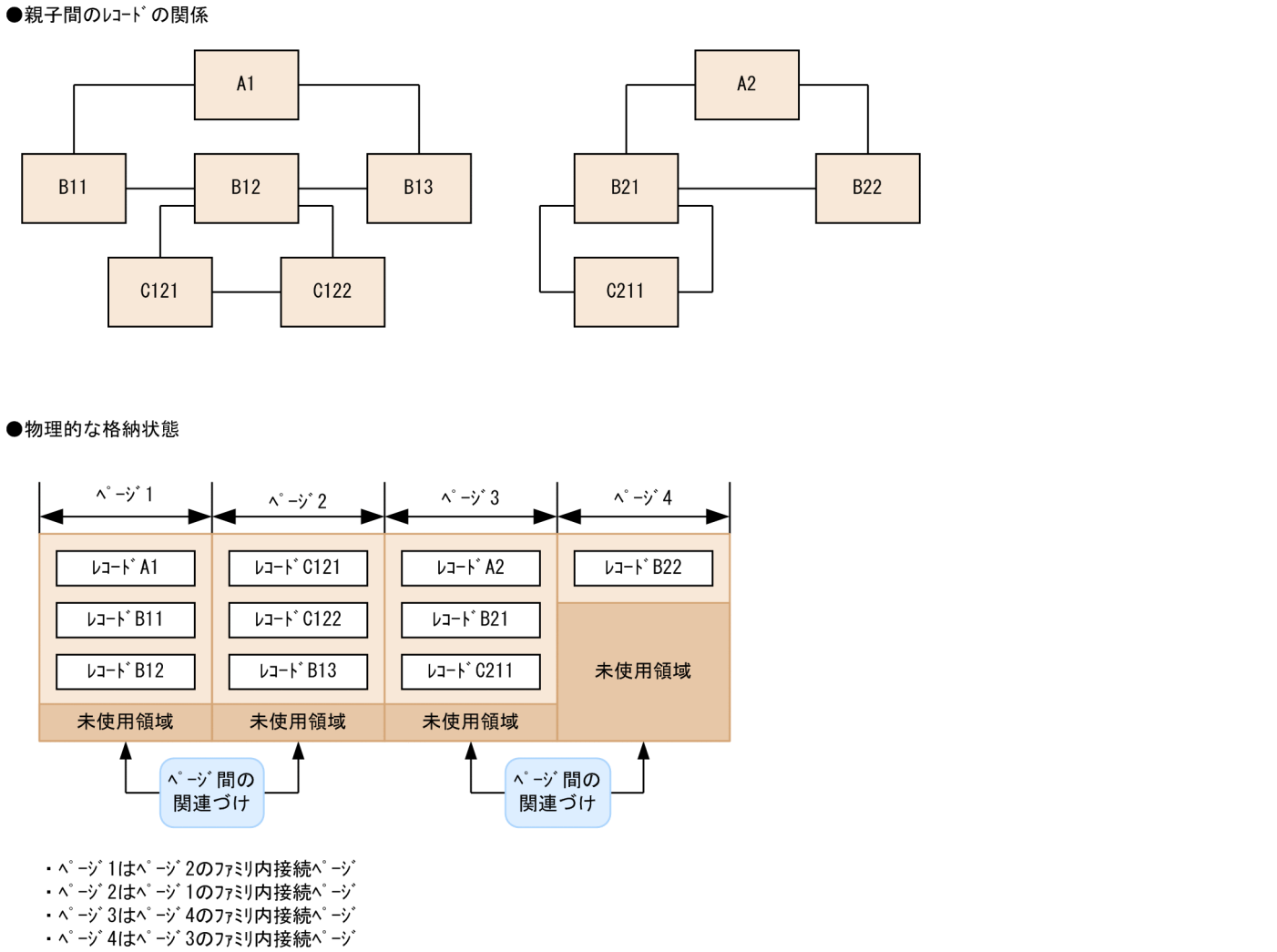
図 2-57　基準ページへの格納の例



(2)　ページ間の関連づけとファミリ内接続ページ

HiRDB/SD では，同一のルートレコード下のレコードをファミリといい，ファミリを格納するページを関連づけて管理します。関連づけられたページをファミリ内接続ページといいます。ページ間の関連づけとファミリ内接続ページの例を次の図に示します。

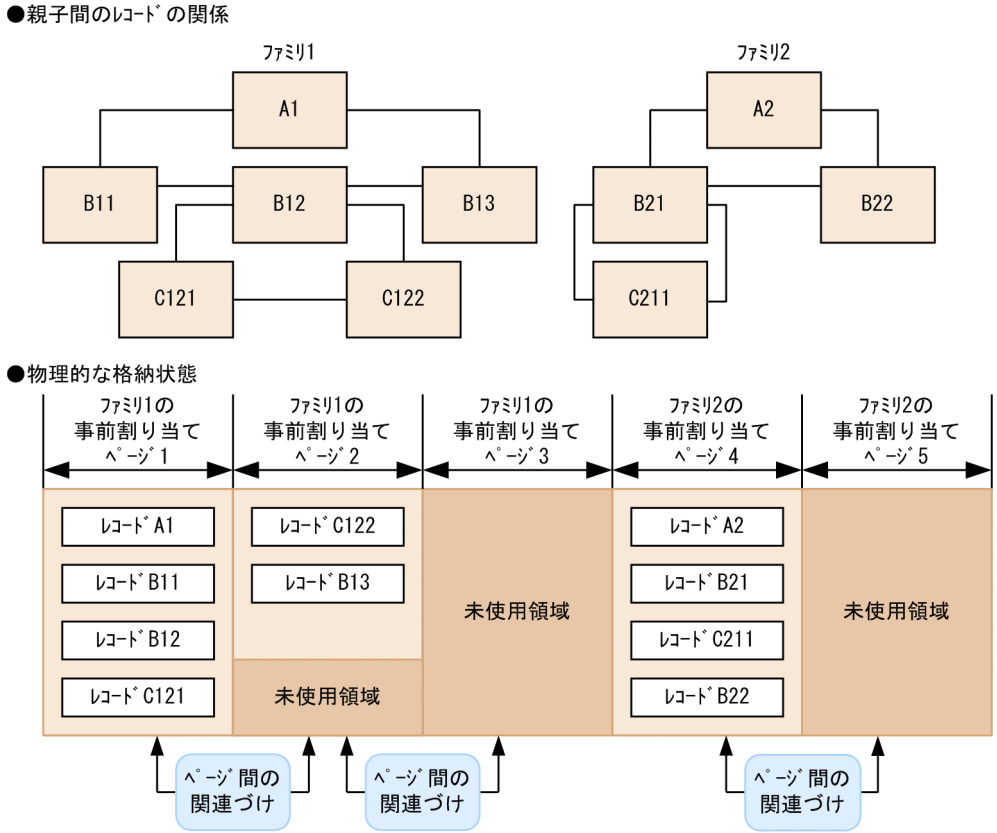
図 2-58　ページ間の関連づけとファミリ内接続ページの例



HiRDB/SD は，確保した空きページにレコードを格納する際に，ページ間を関連づけることによって，基準ページとレコードを格納するページをファミリ内接続ページとして組み入れます。

なお，ファミリ内の事前割り当てページは，すべてファミリ内接続ページ群として管理されます。事前ページ割り当て機能を適用している場合の，ページ間の関連づけとファミリ内接続ページの例を次の図に示します。

図 2-59　事前ページ割り当て機能を適用している場合の，ページ間の関連づけとファミリ内接続ページの例



2.6.2　レコードを格納するページの決定方法

レコードを格納するには，次の 2 つの方法があります。

- SDB データベースを操作する API または DML でレコードを格納する方法
- HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードを格納する方法

各方法でのレコードを格納するページの決定方法について説明します。

参考

- SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。
- DML については，「17. DML リファレンス【SD FMB】」を参照してください。

(1)　SDB データベースを操作する API または DML でレコードをページに格納する場合

SDB データベースを操作する API または DML でレコードを格納する場合，そのファミリに事前ページ割り当て機能を適用しているかどうかによって，格納するページの使用順序が異なります。事前ページ割り当て機能については，「(4)　事前ページ割り当て機能【4V FMB】」を参照してください。

(a)　事前ページ割り当て機能を適用していない場合

事前ページ割り当て機能を適用していない場合，次の表に示す優先順位に従って格納するページを決定します。優先順位 1～5 のページにレコードを格納できない場合，RD エリアのページ不足となります。

表 2-18　SDB データベースを操作する API または DML でレコードを格納する場合の優先順位（事前ページ割り当て機能を適用していない場合）

優先順位	レコード種別	
	ルートレコード	2 階層以上のレコード
1	空きページ	基準ページ※
2	－	基準ページの前後のファミリ内接続ページ※
3	－	基準ページのセグメント，およびその直前のセグメント内の空きページ
4	－	優先順位 3 以外の空きページ
5	－	優先順位 2 以外の全ファミリ内接続ページ※

（凡例）

－：該当しません。

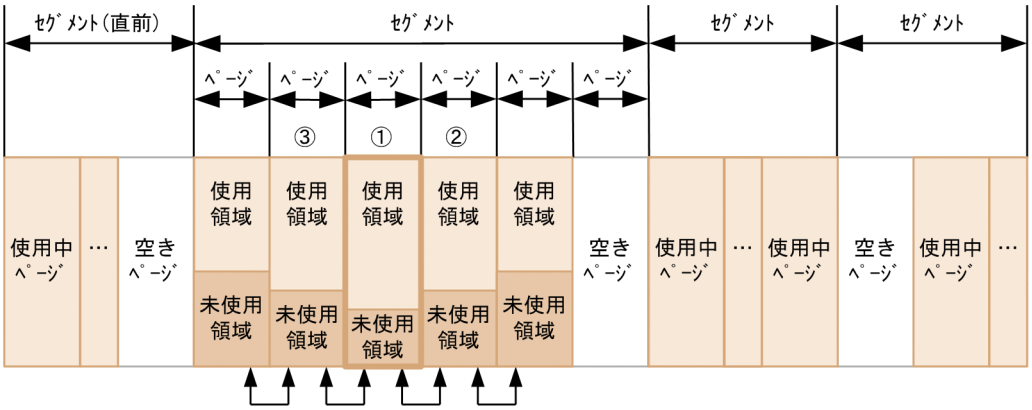
注※

ページ切り替えの指定がある場合，格納の対象になりません。ページ切り替えについては，「2.6.2(3)　レコード格納時のページ切り替え【4V FMB，4V AFM】」を参照してください。

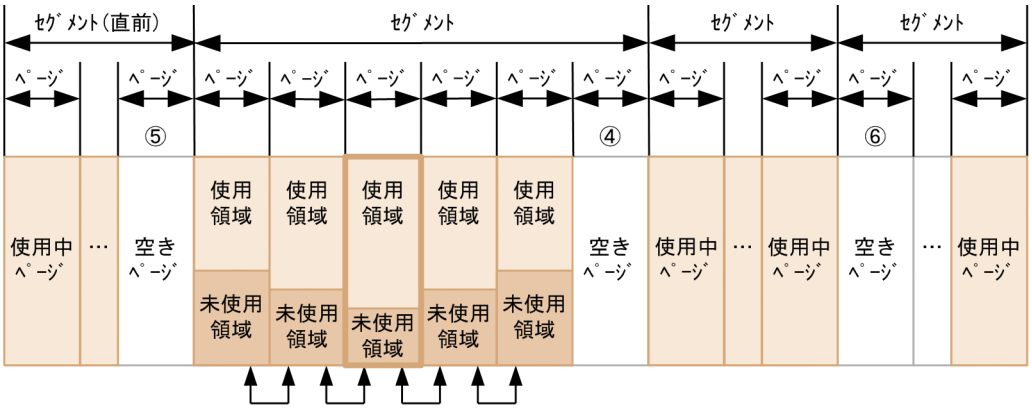
優先順位に従った 2 階層以上のレコードの格納時の動作を次の図に示します。

図 2-60　優先順位に従った 2 階層以上のレコードの格納時の動作（事前ページ割り当て機能を適用していない場合）

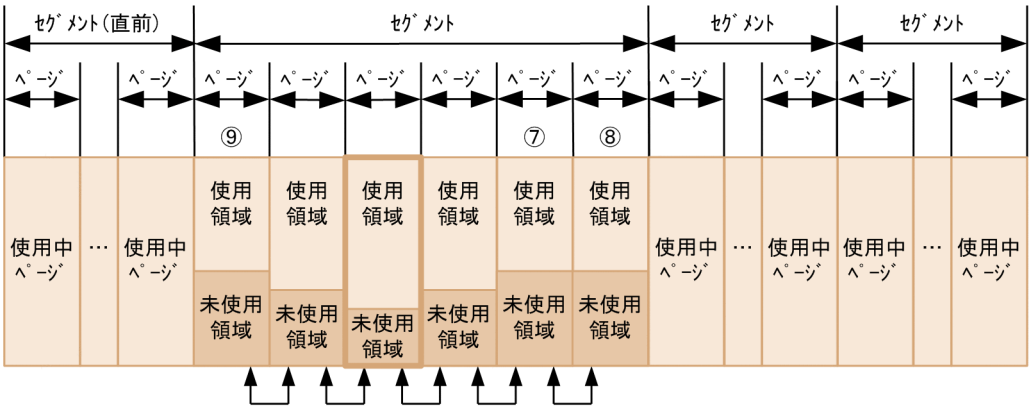
＜優先順位1＞
①：基準ﾍﾟｰｼﾞに格納を試みる。
＜優先順位2＞
②，③：基準ﾍﾟｰｼﾞ前後のﾌｧﾐﾘ内接続ﾍﾟｰｼﾞに格納を試みる。



＜優先順位3＞
④：基準ページのセグメントに空きページがあれば使用する。
⑤：④の空きページがない場合，基準ページのセグメントの直前のセグメントに空きページがあれば使用する。
＜優先順位4＞
⑥：④，⑤の空きページがない場合，それ以外の空きページがあれば使用する。



＜優先順位5＞
⑦，⑧，⑨：残りのファミリ内接続ページに格納を試みる。



（凡例）

：基準ページ

：ページ間の関連づけ

(b)　事前ページ割り当て機能を適用している場合【4V FMB】

4V FMB の SDB データベースのルートレコードを格納する場合，指定数分の事前割り当てページを確保し，最初に確保した事前割り当てページを使用します。また，2 階層以上のレコードを格納する場合，基本的には確保済みの事前割り当てページを使用します。ただし，確保済みの事前割り当てページ内にレコードを格納できる空き領域がない場合は，指定数分の事前割り当てページを追加で確保し，最初に確保した事前割り当てページを使用します。

事前ページ割り当て機能を適用している場合，次の表に示す優先順位に従って，格納するページを決定します。優先順位 1～5 のページにレコードを格納できない場合，RD エリアのページ不足となります。

表 2-19　SDB データベースを操作する API でレコードを格納する場合の優先順位（事前ページ割り当て機能を適用している場合）

優先順位	レコード種別	
	ルートレコード	2 階層以上のレコード
1	空きページ※1	基準ページ
2	－	優先ページ※2
3	－	基準ページと優先ページ以外の事前割り当てページ
4	－	基準ページのセグメント，およびその直前のセグメント内の空きページ※1
5	－	優先順位 4 以外の空きページ※1

（凡例）

－：該当しません。

注※1

事前割り当てページとして確保します。

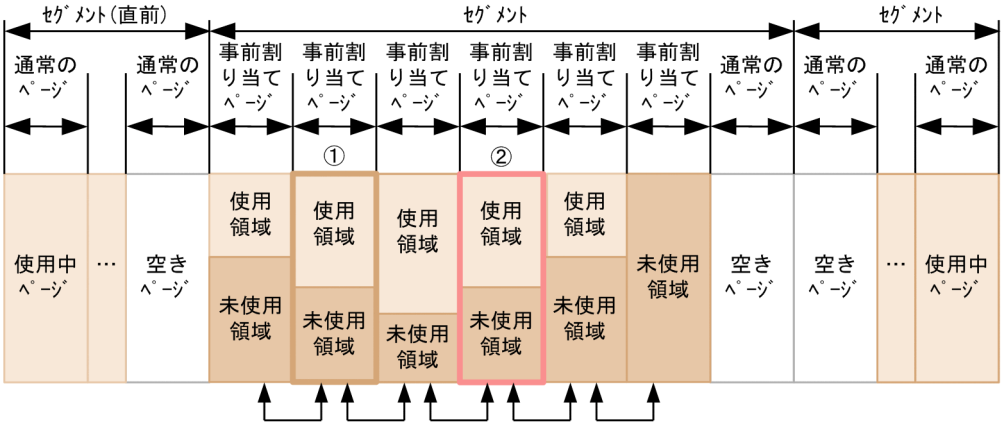
注※2

基準ページの次に HiRDB/SD が優先的に格納を試みる事前割り当てページです。

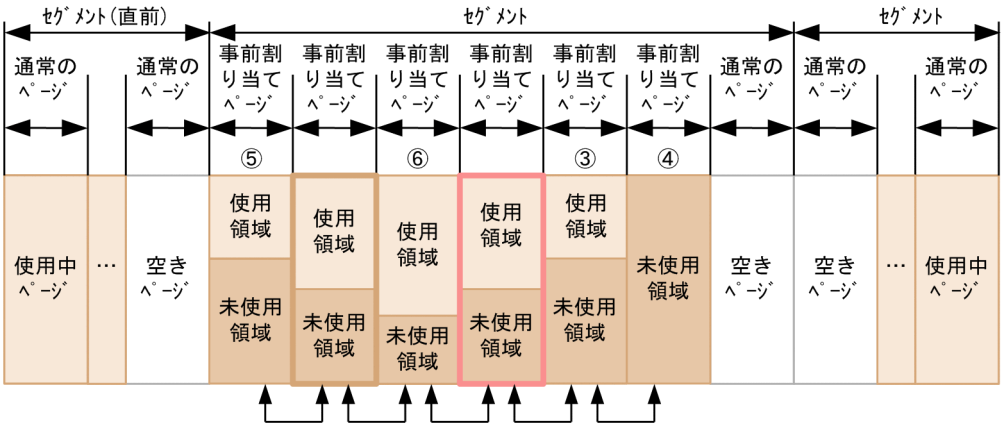
優先順位に従った 2 階層以上のレコードの格納時の動作を次の図に示します。

図 2-61　優先順位に従った 2 階層以上のレコードの格納時の動作（事前ページ割り当て機能を適用している場合）

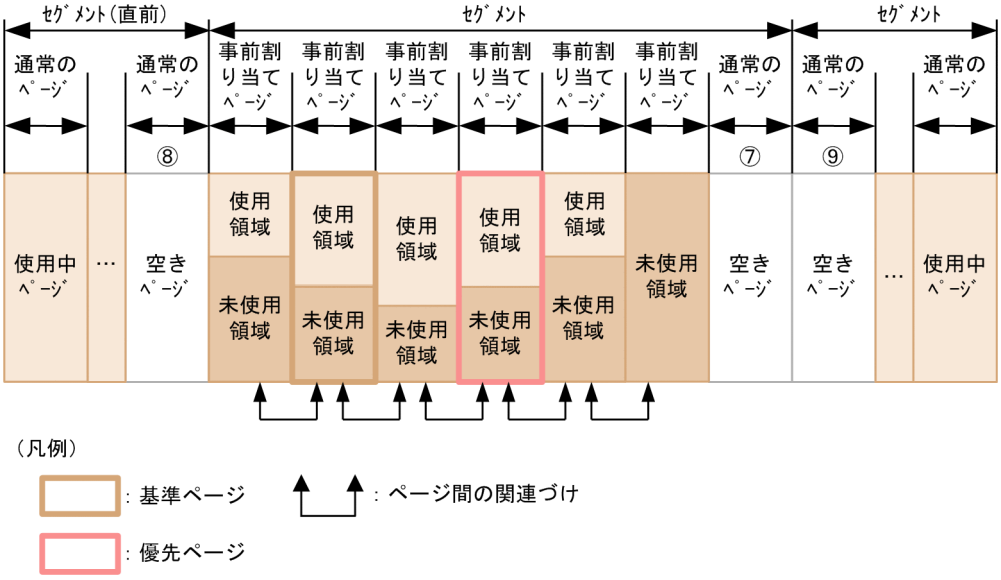
＜格納優先順位1＞
①：基準ﾍﾟｰｼﾞに格納を試みる。
＜格納優先順位2＞
②：優先ﾍﾟｰｼﾞに格納を試みる。



＜格納優先順位3＞
③, ④, ⑤, ⑥：基準ﾍﾟｰｼﾞと優先ﾍﾟｰｼﾞ以外の事前割り当てﾍﾟｰｼﾞに格納を試みる。



＜格納優先順位4＞
⑦：基準ﾍﾟｰｼﾞのｾｸﾞﾒﾝﾄに空きﾍﾟｰｼﾞがあれば使用する。
⑧：⑦の空きﾍﾟｰｼﾞがない場合，基準ﾍﾟｰｼﾞのｾｸﾞﾒﾝﾄの直前のｾｸﾞﾒﾝﾄに空きﾍﾟｰｼﾞがあれば使用する。
＜格納優先順位5＞
⑨：⑦，⑧の空きﾍﾟｰｼﾞがない場合，それ以外の空きﾍﾟｰｼﾞがあれば使用する。



(2)　HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードをページに格納する場合

HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードをページに格納する場合，次の表に示す優先順位に従って，格納するページを決定します。優先順位 1～2 のページにレコードを格納できない場合，RD エリアのページ不足となります。

表 2-20　HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードをページに格納する場合の優先順位

レコード種別	優先順位	
	1	2
ルートレコードの格納	空きページ	－
2 階層以上のレコードの格納	直近でレコードを格納したページ※	空きページ

（凡例）
－：該当しません。

注※
ページ切り替えの指定がある場合，格納の対象になりません。ページ切り替えについては，「(3)　レコード格納時のページ切り替え【4V FMB，4V AFM】」を参照してください。

(3)　レコード格納時のページ切り替え【4V FMB，4V AFM】

レコード格納時のページ切り替えについて説明します。ページ切り替えは，4V FMB または 4V AFM の SDB データベースの場合に指定できます。

(a) ページ切り替えの概要

4V FMB または 4V AFM の SDB データベースにレコードを格納する際，格納対象のページを切り替えて，空きページにレコードを格納できます。これをページ切り替えといいます。また，このとき，格納したレコードだけでそのページを占有し，ほかのレコードが格納されないようにすることもできます。格納対象のページを切り替えることによって，レコードに対するアクセスをページ単位に分散し，排他の競合を回避できます。

ページ切り替えの指定には，次の 2 つの指定があります。

- **切り替え指定**
  切り替えたページにほかのレコードも格納できる指定
- **占有指定**
  切り替えたページを格納するレコードで占有する指定（ほかのレコードは格納できない指定）

ページ切り替えの指定方法を次の表に示します。

表 2-21　ページ切り替えの指定方法

格納方法		指定方法
HiRDB/SD データベース作成ユティリティ（pdsdblod）	フォーマットライト機能使用時	dbinf 文の pagecflg でページ切り替えフラグを指定します。dbinf 文については，「12.3.3　dbinf 文【4V FMB，4V AFM】」を参照してください。
	上記以外	入力データファイルのプレフィクス部のページ切り替えフラグを指定します。詳細については，「12.3.3　dbinf 文【4V FMB，4V AFM】」を参照してください。
SDB データベースを操作する API によるレコード格納		SDB データベースを操作する API に，ページ切り替えオプションを指定することによって，レコードの格納時にページを切り替えることができます。 'C'指定（切り替え指定） 　レコード格納時にページを切り替え，空きページを確保します。 　確保したページにほかのレコードを格納できます。 'O'指定（占有指定） 　レコード格納時にページを切り替え，空きページを確保します。 　確保したページは格納したレコードで占有します。ほかのレコードは格納できません。 'N'または'△'（空白）指定 　ページの切り替えを指定しません。 SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

(b) ページ切り替えに関する留意事項

- 4V AFM の SDB データベースの場合，仮想ルートレコードの格納時に，SDB データベース格納定義の OCCURRENCE NUMBER 句の指定値が 1 以上のレコードでは，仮想ルートレコードにアクセスする際の排他と，同一仮想ルートレコード下の子レコードにアクセスする際の排他の競合を回避するた

め，ページ切り替えの指定が無視されます。このとき，'O'が指定されたと仮定してレコードが格納されます。

子レコードの格納時には，ページ切り替えの指定に従ってレコードが格納されます。

- レコードの一括削除後にレコードを格納する場合，一括削除時に存在していたレコードの格納領域が再利用されます。そのため，一括削除時に存在していたレコード件数分のレコードを格納するときには，ページ切り替え指定は無視されます。
- HiRDB/SD は，異なるルートレコード下のレコードをページ内に混在させないため，ルートレコードの格納時にページを切り替えます。そのため，ルートレコード格納時の'C'指定，または指定なしの場合については，動作に違いはありません。'O'指定の場合には，ルートレコードだけで格納ページを占有するため，ほかのレコードは格納されません。
- ページ切り替え指定に従ってレコード格納時にページを切り替えた場合，確保したページと基準ページとの間で，ページ間の関連づけはされません。そのため，ページ切り替え指定以前に作成されていたページは，確保したページのファミリ内接続ページとはなりません。

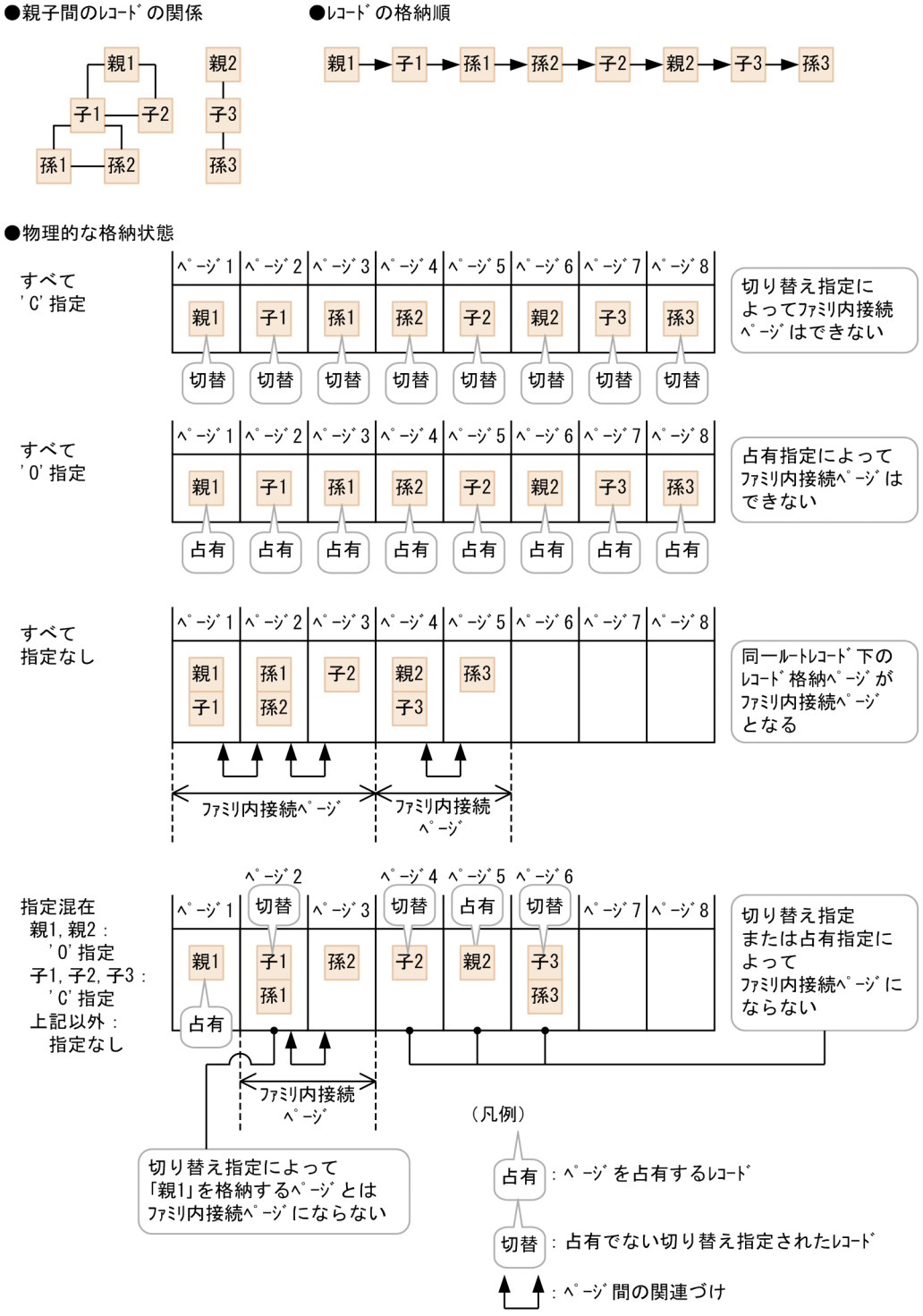
  同一ルートレコード下のレコード格納ページであっても，異なるファミリ内接続ページに分割された場合には，「表 2-18　SDB データベースを操作する API または DML でレコードを格納する場合の優先順位（事前ページ割り当て機能を適用していない場合）」の優先順位 2 および 5 の対象ページとはなりません。ただし，ページの切り替え後にレコードを格納することによって作成されたページ間は関連づけられるので，新たなファミリ内接続ページとなります。ページ間の関連づけ，ファミリ内接続ページについては，「2.6.1(2)　ページ間の関連づけとファミリ内接続ページ」を参照してください。
- 基準レコードが'O'指定でページを占有しているレコードの場合であっても，ページ切り替え指定なしで空きページを確保できているときには，確保したページと基準ページとが関連づけられます。
- 事前ページ割り当て機能を適用しているファミリのレコードを格納する場合，ページ切り替え指定でページを切り替えることはできません。ページ切り替えの指定があるとエラーになります。事前ページ割り当て機能については，「(4)　事前ページ割り当て機能【4V FMB】」を参照してください。

(c)　ページ切り替えオプションによるレコードの格納例

4V FMB の SDB データベースの場合

4V FMB の SDB データベースに対するレコードの格納時の，ページ切り替えオプションによるレコードの格納例を次の図に示します。

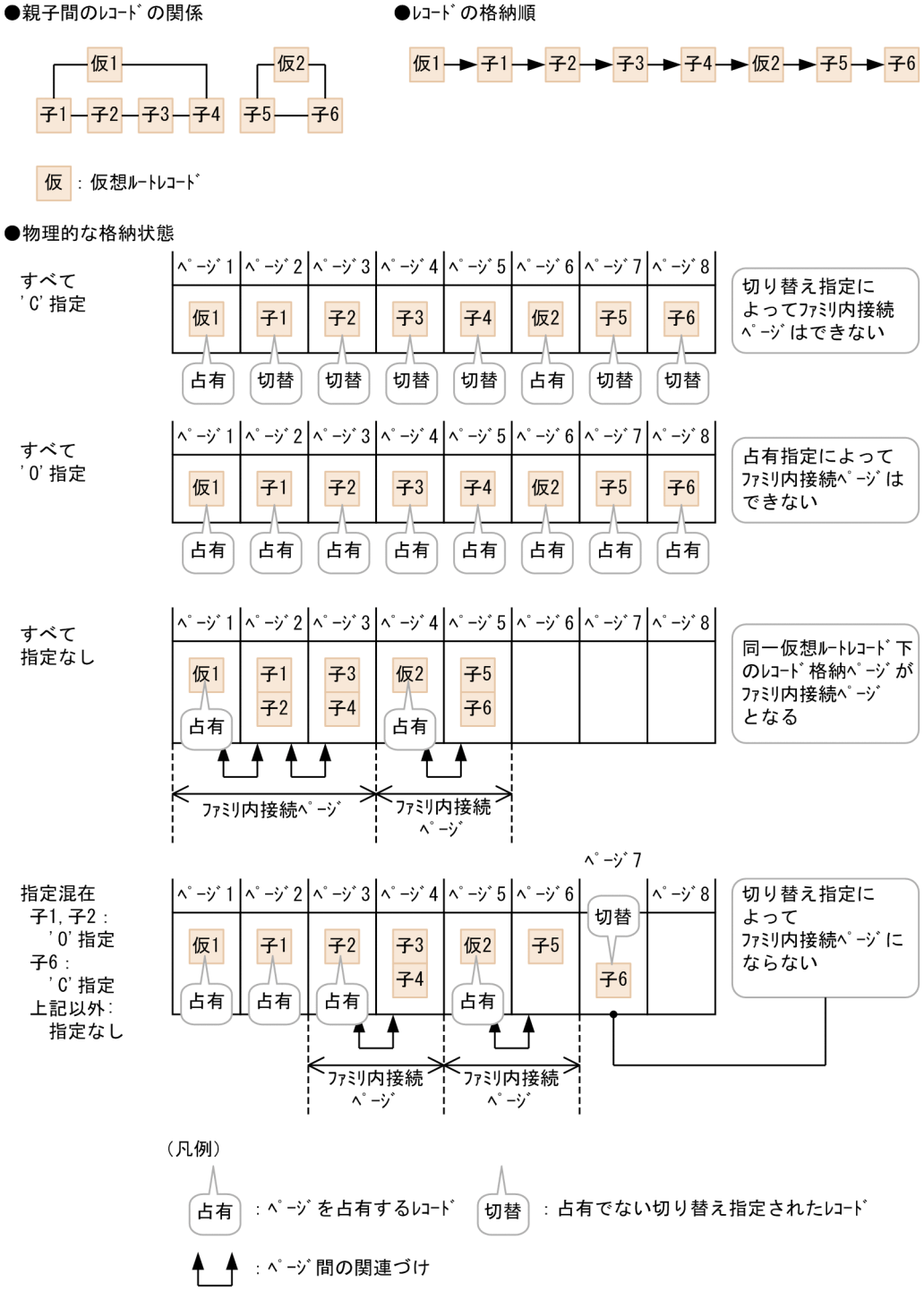
図 2-62　ページ切り替えオプションによるレコードの格納例（4V FMB の SDB データベースの場合）



OCCURRENCE NUMBER 句に 1 以上を指定している 4V AFM の SDB データベースの場合

SDB データベース格納定義の OCCURRENCE NUMBER 句に 1 以上を指定している 4V AFM の SDB データベースに対するレコードの格納時の，ページ切り替えオプションによるレコードの格納例を次の図に示します。

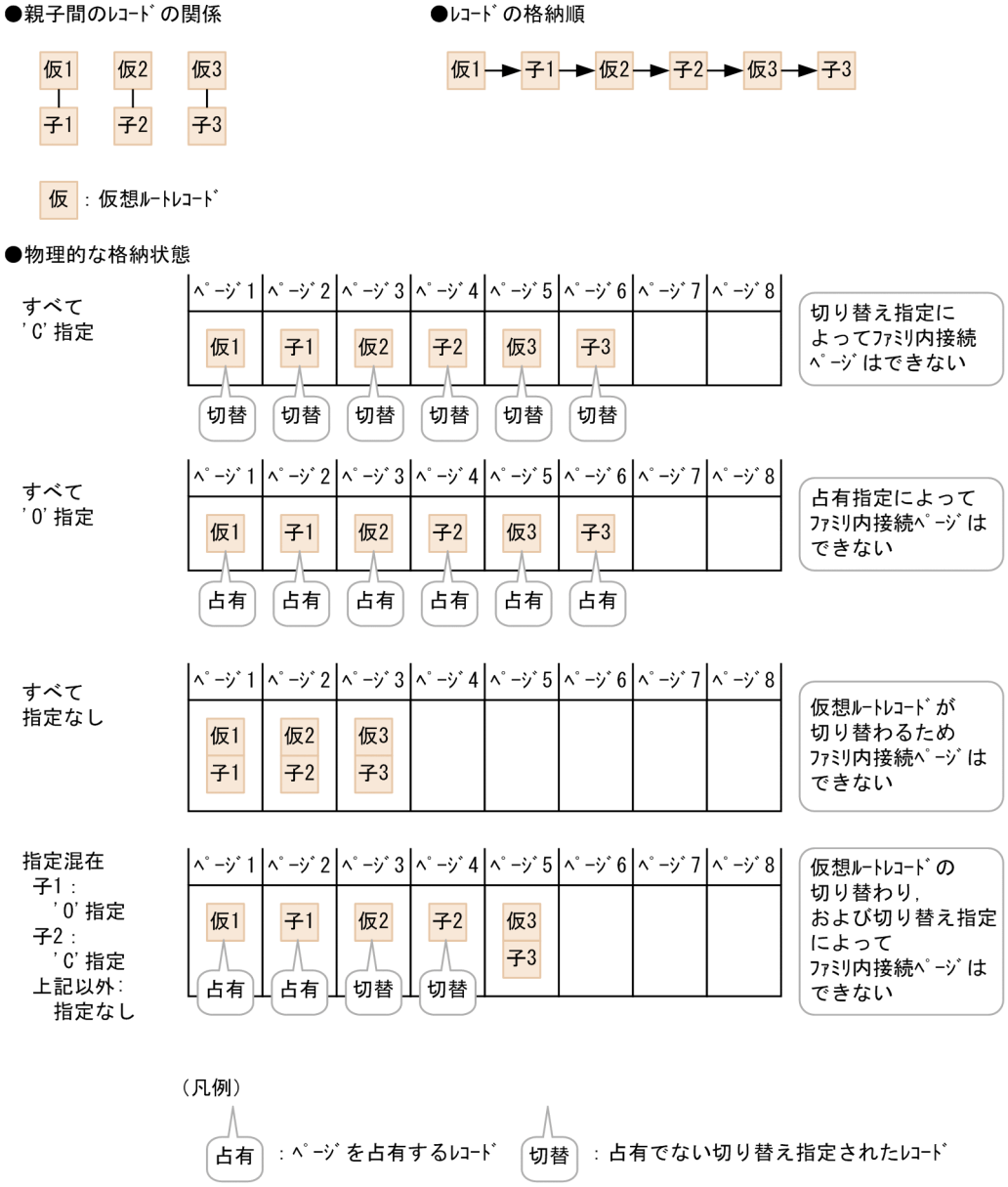
図 2-63　ページ切り替えオプションによるレコードの格納例（OCCURRENCE NUMBER 句に 1 以上を指定している 4V AFM の SDB データベースの場合）



OCCURRENCE NUMBER 句に 0 を指定している 4V AFM の SDB データベースの場合

SDB データベース格納定義の OCCURRENCE NUMBER 句に 0 を指定している 4V AFM の SDB データベースに対するレコードの格納時の，ページ切り替えオプションによるレコードの格納例を次の図に示します。

図 2-64　ページ切り替えオプションによるレコードの格納例（OCCURRENCE NUMBER 句に 0 を指定している 4V AFM の SDB データベースの場合）



(d)　適用基準

レコードに対するアクセス時の排他をページ単位で分散させたい場合，'O'指定または'C'指定を選択します。同時実行性は次の順に高くなります。

1. 'O'指定（占有指定）（同時実行性がいちばん高い）
2. 'C'指定（切り替え指定）
3. ページ切り替え指定なし

ただし，次の場合は指定による同時実行性の差異はありません。

- 4V FMB の SDB データベースの場合（ルートレコードの格納ページの排他によって，ルートレコード下のレコード群のアクセスがシリアライズされるため）
- 4V AFM の SDB データベースで，SDB データベース格納定義の OCCURRENCE NUMBER 句の指定が 0 の場合（仮想ルートレコード下に子レコードが最大 1 件しか存在しないため）

なお，ページ切り替えをすることによってデータベース容量が増大することがあるので注意してください。一般的に次の順でデータベース容量は増加します。

1. ページ切り替え指定なし（データベース容量がいちばん少ない）
2. 'C'指定（切り替え指定）
3. 'O'指定（占有指定）

SDB データベース種別ごとに推奨する指定を次の表に示します。作成するデータベースや格納するレコードの特性に合わせて指定を決定してください。

表 2-22　SDB データベース種別ごとに推奨する指定

SDB データベース種別		ページ切り替えで推奨する指定※1	備考
4V FMB		指定なし	ルートレコードの格納ページの排他によって，ルートレコード下のレコードのアクセスがシリアライズされるため，'O'指定または'C'指定に同時実行性を向上させる効果はありません。
4V DAM	OCCURRENCE NUMBER 句の指定が 0	指定なし	仮想ルートレコード下に子レコードが最大 1 件しか存在しないため，'O'指定または'C'指定に同時実行性を向上させる効果はありません。
	OCCURRENCE NUMBER 句の指定が 1 以上	指定なし または'O'指定（占有指定）※2	指定なしの場合，ページに格納するレコード件数を制限し，排他の影響範囲を局所化したいときには，SDB データベース格納定義の PCTFREE 句を指定します。 ページに 1 件だけレコードを格納する場合には'O'指定とします。
4V MAM		'O'指定（占有指定）※2	4V MAM の SDB データベースは一般的に同時実行性が求められます。
4V TAM		'O'指定（占有指定）※2	4V TAM の SDB データベースは一般的に同時実行性が求められます。
4V SAM		指定なし	4V SAM の SDB データベースは一般的に同時実行性が求められません。

注※1

データベース単位で，すべてのレコードに一律同じ指定をする場合の推奨値です。格納するレコードの特性やレコードの件数などがわかっている場合は，レコード単位に指定を切り替えることで，格納配置をより詳細に制御できます。レコード単位の指定の切り替えによるレコードの配置例については，「(e)　ページ切り替え指定によるレコードの配置例」を参照してください。

注※2

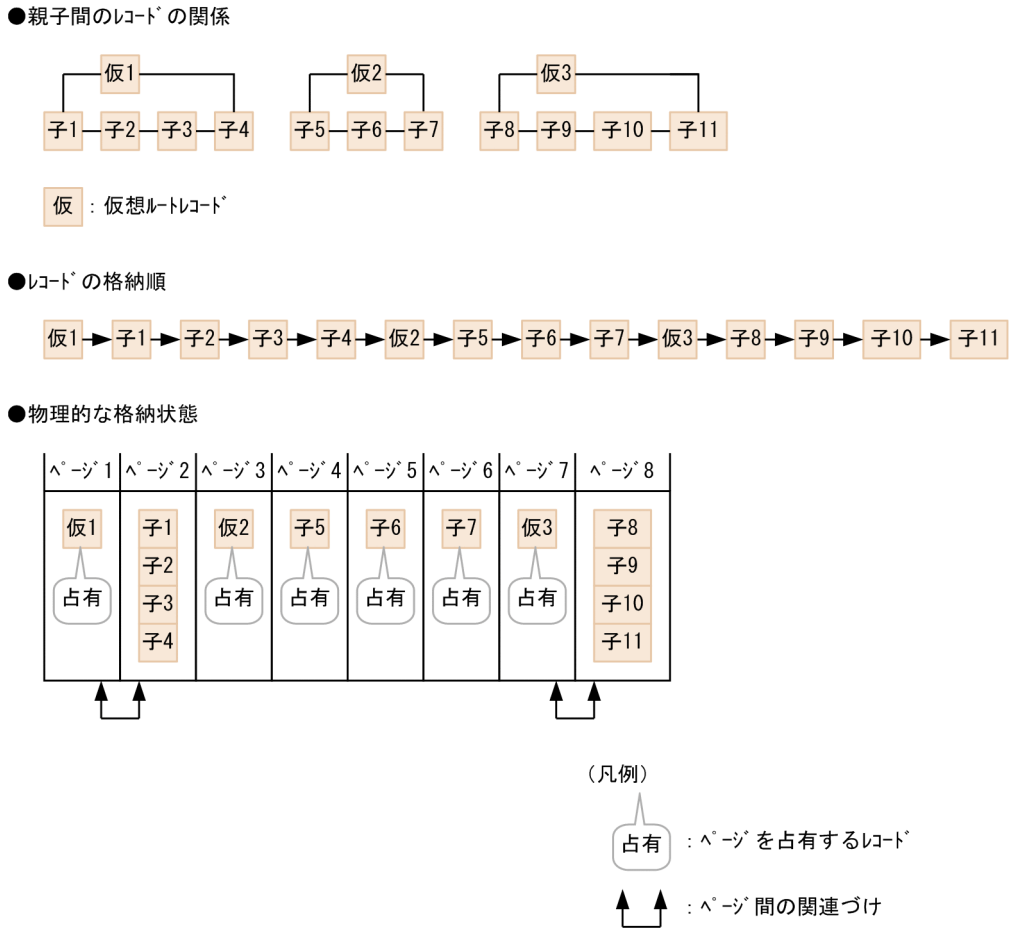
参照しかしないデータベースの場合や，ほかのトランザクションが存在しない状態で，更新処理を単独で実行するような場合には，ページ排他での排他待ちが発生しないため，「指定なし」を推奨します。この場合，ページに格納するレコード件数を制限し，排他の影響範囲を局所化する目的で PCTFREE 句を指定する必要はありません。

(e)　ページ切り替え指定によるレコードの配置例

アクセス頻度によってページ切り替え指定を使い分ける

4V DAM の SDB データベースなどで，アクセス頻度が高く同時実行性が要求されるレコードに対してだけ'O'指定をするというように指定を使い分けると，同時実行性を確保しながらデータベース容量を小さく抑えるといった，データベース設計ができます。アクセス頻度によってページ切り替え指定を使い分ける例を次の図に示します。この例では仮想ルートレコード 2 下のレコードだけ'O'指定をしています。

図 2-65　アクセス頻度によってページ切り替え指定を使い分ける例

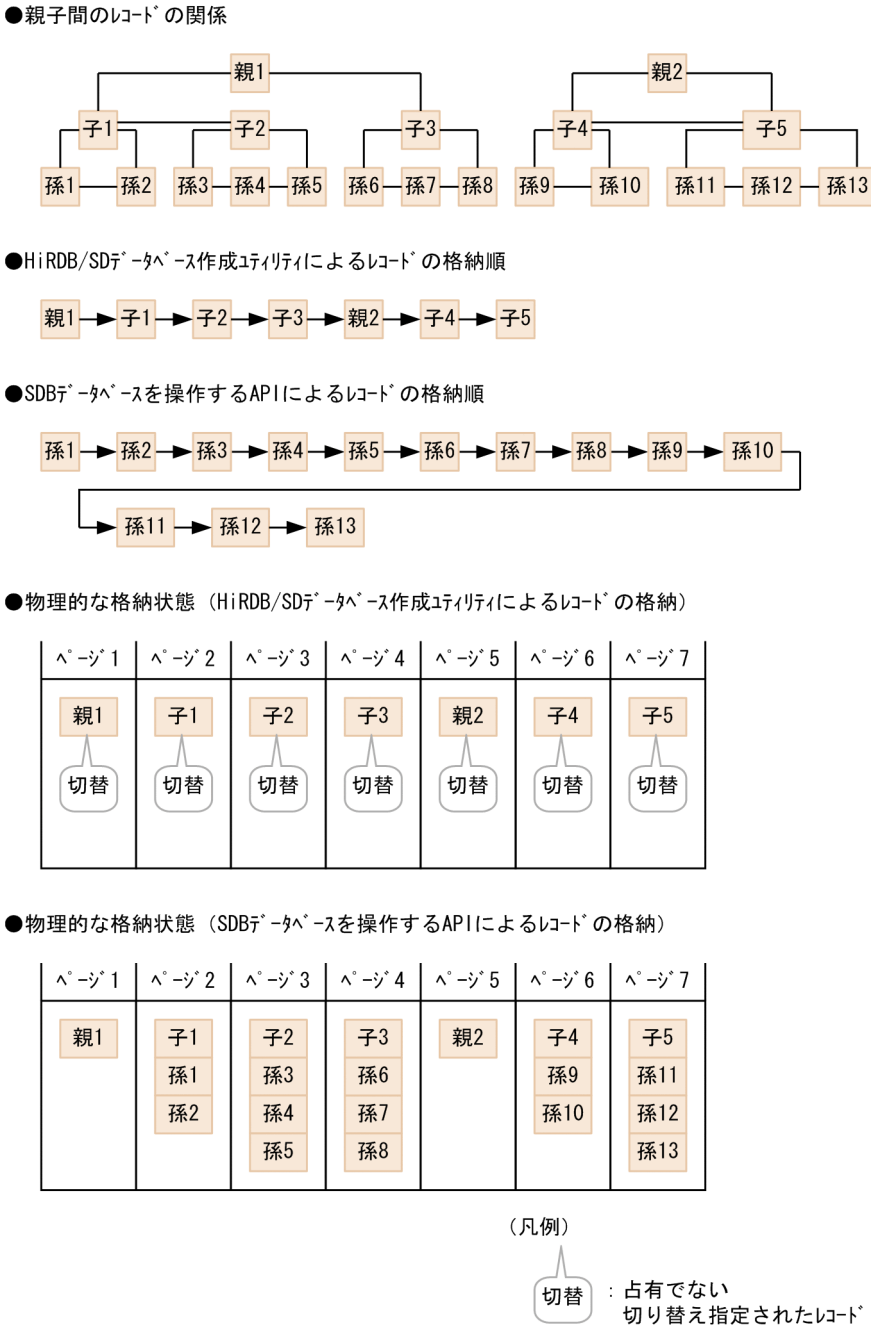


論理的に関連するレコードの格納位置をまとめる

4V FMB の SDB データベースで，データベース初期作成後のレコードの格納に備えて'C'指定をしておくことで，論理的に関連するレコードを同一ページに格納するといった，データベース設計ができます。

論理的に関連するレコードの格納位置をまとめる例を次の図に示します。この例では孫レコードを格納するための領域をページ切り替え指定によって確保しておくことで，同一子レコード下の孫レコードを同一ページに格納しています。

図 2-66　論理的に関連するレコードの格納位置をまとめる例



(4)　事前ページ割り当て機能【4V FMB】

事前ページ割り当て機能について説明します。事前ページ割り当て機能は，4V FMB の SDB データベースの場合に使用できます。

(a)　事前ページ割り当て機能の概要

HiRDB/SD データベース作成ユティリティ（pdsdblod），または SDB データベースを操作する API で 4V FMB の SDB データベースにルートレコードを格納する場合，事前割り当てページ数を指定すると，指定したファミリにあらかじめ指定数分の連続したデータページ（事前割り当てページ）が確保されます。

HiRDB/SD データベース作成ユティリティ（pdsdblod）での指定方法については，次の説明を参照してください。

- 「12.3.2　load 文」の prefix オペランドの説明
- 「12.3.3　dbinf 文【4V FMB，4V AFM】」の pagenum オペランドの説明
- 「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」の事前割り当てページ数の説明

SDB データベースを操作する API によるレコードの格納（STORE）での，事前割り当てページ数の指定方法については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

SDB データベースを操作する API の実行時には次の制御を行うため，2 階層以上のレコードの格納およびレコードの削除を繰り返し実行した場合でも，ファミリ単位に連続したデータページ内へのレコードの配置を維持できます。

- 2 階層以上のレコードの格納時に，事前割り当てページ内にレコードを格納できる空き領域がある場合，その空き領域にレコードが格納されます。
- 2 階層以上のレコードの削除によって，事前割り当てページ内のレコードがすべて削除された場合でも，事前割り当てページは解放されないで，2 階層以上のレコードの格納時に再利用されます。

この配置制御によって，特定のファミリに連続してアクセスする場合に I/O 回数の削減が見込まれるため，SDB データベースを操作する API を繰り返し実行しても，一定のアクセス性能を維持できます。

2 階層以上のレコードの格納時に，事前割り当てページ内にレコードを格納できる空き領域がない場合，追加で事前割り当てページが確保されます。このときに確保される事前割り当てページの数は，該当するレコードの格納時の事前割り当てページ数の指定に従います。事前割り当てページ不足が発生していない場合，または事前ページ割り当て機能を適用していない場合，その指定は無視されます。

なお，事前割り当てページ数に 0 を指定した場合は 1 を指定したときと同じく，格納対象のレコードを格納するページだけが確保されます。

(b)　事前割り当てページ数の引き継ぎ

確保済みの事前割り当てページの数は，次の表に示す機能で取得できます。

表 2-23　事前割り当てページ数を取得できる機能

事前割り当てページ数を取得できる機能	機能概要
HiRDB/SD データベース再編成ユティリティ（pdsdbrog）	アンロード時，ファミリ単位に確保済みの事前割り当てページの数がアンロードデータファイルのプリフィクス部に設定されます。

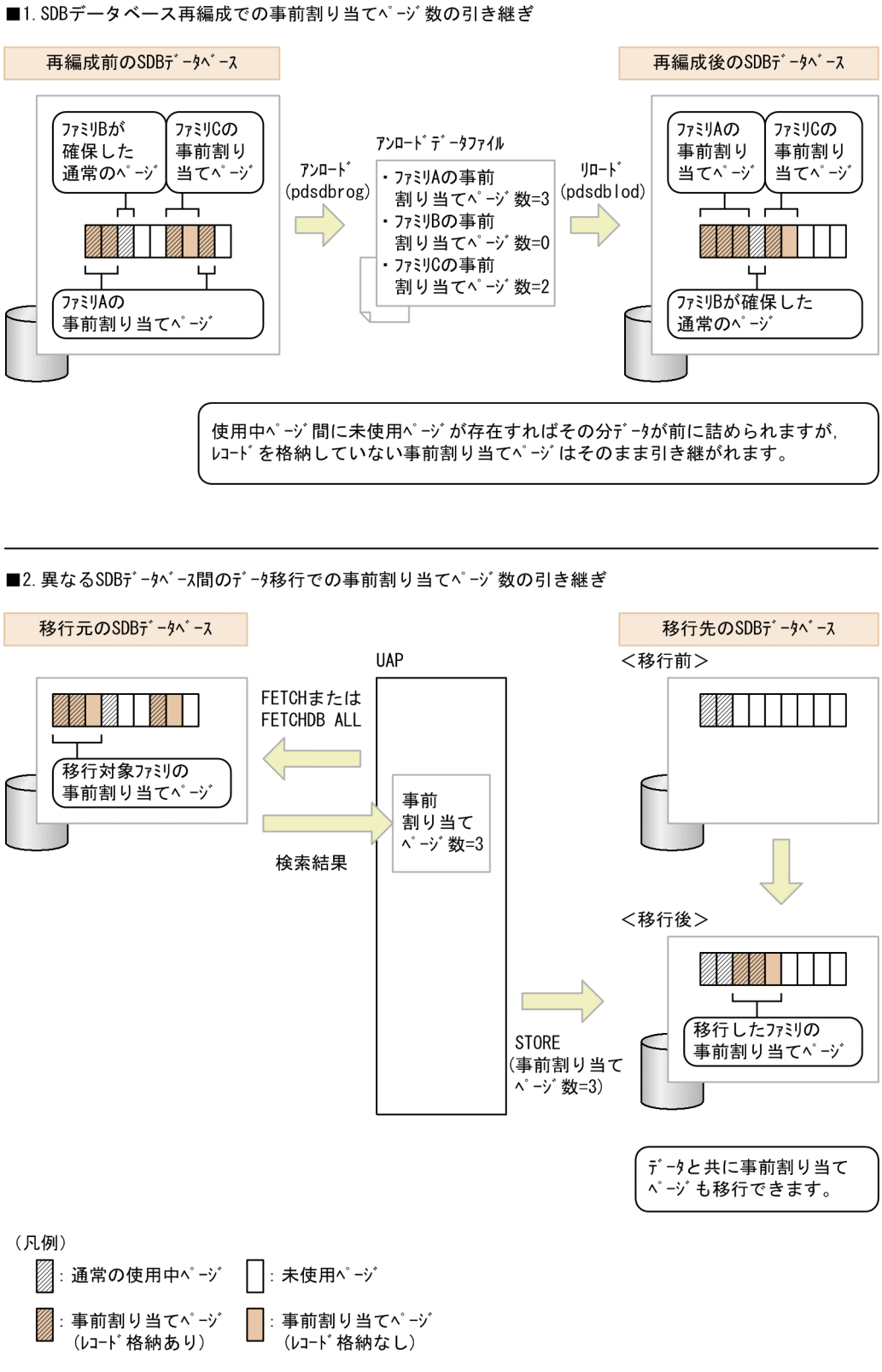
事前割り当てページ数を取得できる機能	機能概要
	事前割り当てページ数の取得方法については，次の説明を参照してください。 • 「13.3.2　unload 文」の prefix オペランドの説明 • 「13.3.3　dbinf 文【4V FMB】」の pagenum オペランドの説明 • 「13.6.1　レコードのデータ形式」のプリフィクス部の説明
レコードの検索（FETCH）	ルートレコードの検索時に確保済みの事前割り当てページの数が返却されます。2 階層以上のレコードだけを検索した場合は 0 が返却されます。また，事前ページ割り当て機能を適用していないファミリを検索した場合にも 0 が返却されます。 SDB データベースを操作する API での事前割り当てページ数の取得方法については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。
複数レコードの検索（FETCHDB ALL）	

これによって，次の場合に事前割り当てページ数を引き継ぐことができます。

1. SDB データベースの再編成時
2. 異なる SDB データベース間のデータ移行時

事前割り当てページ数の引き継ぎ例を次の図に示します。

図 2-67　事前割り当てページ数の引き継ぎ例



(c)　事前ページ割り当て機能の適用条件

事前ページ割り当て機能は，次の機能と同時に適用できません。

- ページ切り替え

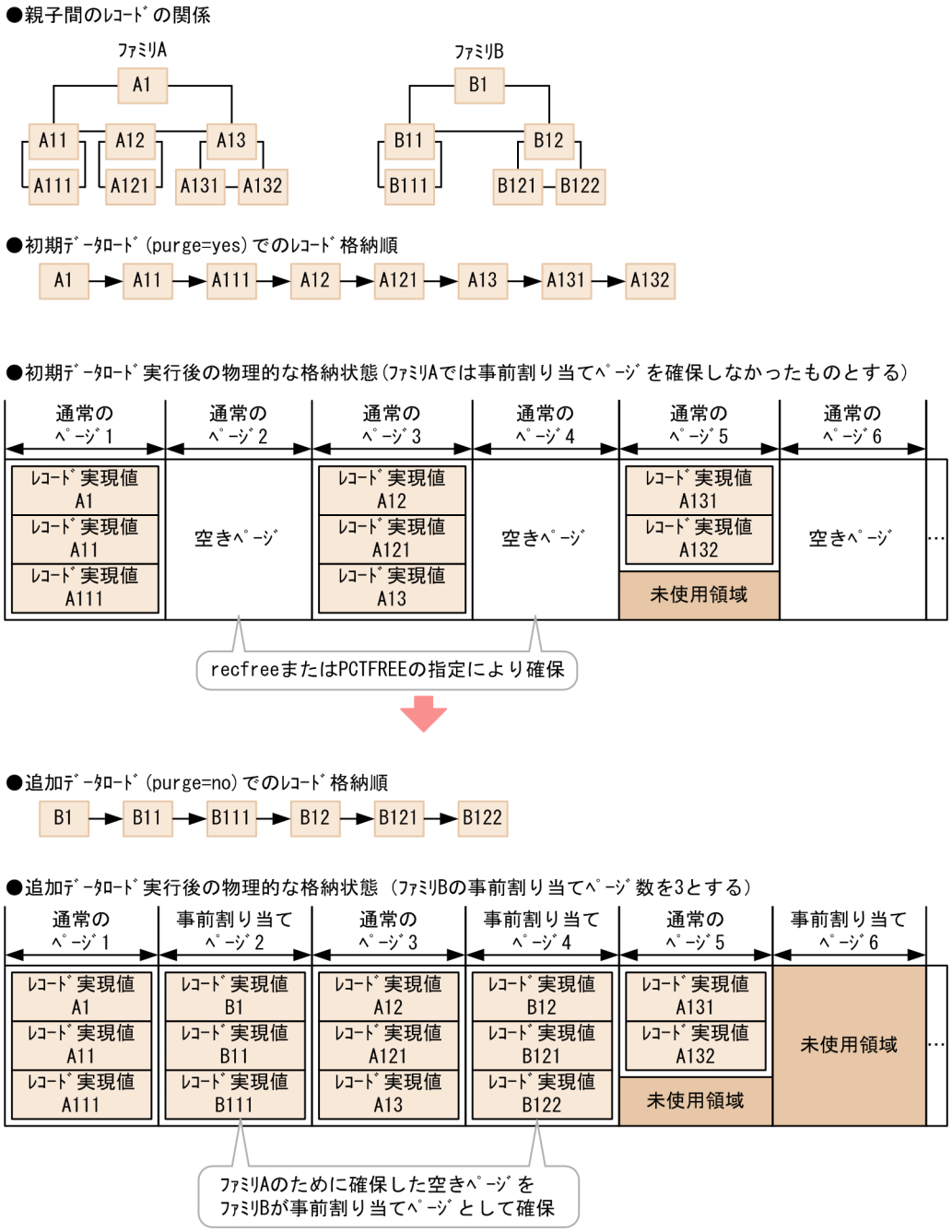
- SDB データベース格納定義の PCTFREE オペランド，または HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の recfree オペランドで指定する空き領域の作成

(d) 事前ページ割り当て機能に関する留意事項

- 確保できる事前割り当てページ数の上限値は，1 ファミリ当たり 2,147,483,647 です。
- HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードを格納する場合，次のように動作します。
  - 事前ページ割り当て機能を適用するファミリにページ切り替えの指定がある場合，論理エラーになります。
  - 事前割り当てページを確保する際，セグメント内の空きページの比率の指定は無視されるため，事前割り当てページ間に空きページは確保されません。通常のページを確保する際に空きページは確保されますが，同じセグメント内に事前割り当てページが含まれる場合，通常は指定したセグメント内の空きページの比率にはなりません。この場合のセグメント内の空きページの比率については，「3.2.5(4)(b) 未使用領域および空きページの作成に関する留意事項」を参照してください。
  - 事前割り当てページにレコードを格納する際，ページ内の未使用領域の比率は無視されるため，レコードが詰めて格納されます。
- SDB データベースを操作する API でレコードを格納する場合，次のように動作します。
  - 事前ページ割り当て機能を適用しているファミリのレコードを格納する際に，ページ切り替えの指定があるとエラーになります。
  - 事前ページ割り当て機能を適用しているファミリのレコードを格納する際に，PCTFREE 有効化オプションを適用するとエラーになります。
- データロード時にレコードを格納するために確保したページの数が，指定した事前割り当てページ数を超えた場合，該当するファミリを格納するために指定値を超えて確保したページも，事前割り当てページとして扱われます。
- 次のどれかの場合，事前ページ割り当て機能を適用しても非連続のデータページが確保されることがあります。
  1. 空きページが散在している SDB データベースに対して，追加データロードまたは SDB データベースを操作する API によるルートレコードの格納を実行し，事前割り当てページが確保される場合
  2. SDB データベースを操作する API による 2 階層以上のレコードの格納時に，事前割り当てページ不足が発生し，追加で事前割り当てページが確保される場合
  3. SDB データベースを操作する API によるレコードの格納時の事前割り当てページの確保中に，ほかのトランザクションによるページの確保が割り込む場合

     追加データロード時に，非連続のデータページが事前割り当てページとして確保される例（1.の例）を次の図に示します。この図では，初期データロード時に確保した空きページを事前割り当てページとして確保する例を示しています。

図 2-68　追加データロード時の非連続データページ（事前割り当てページ）の確保例

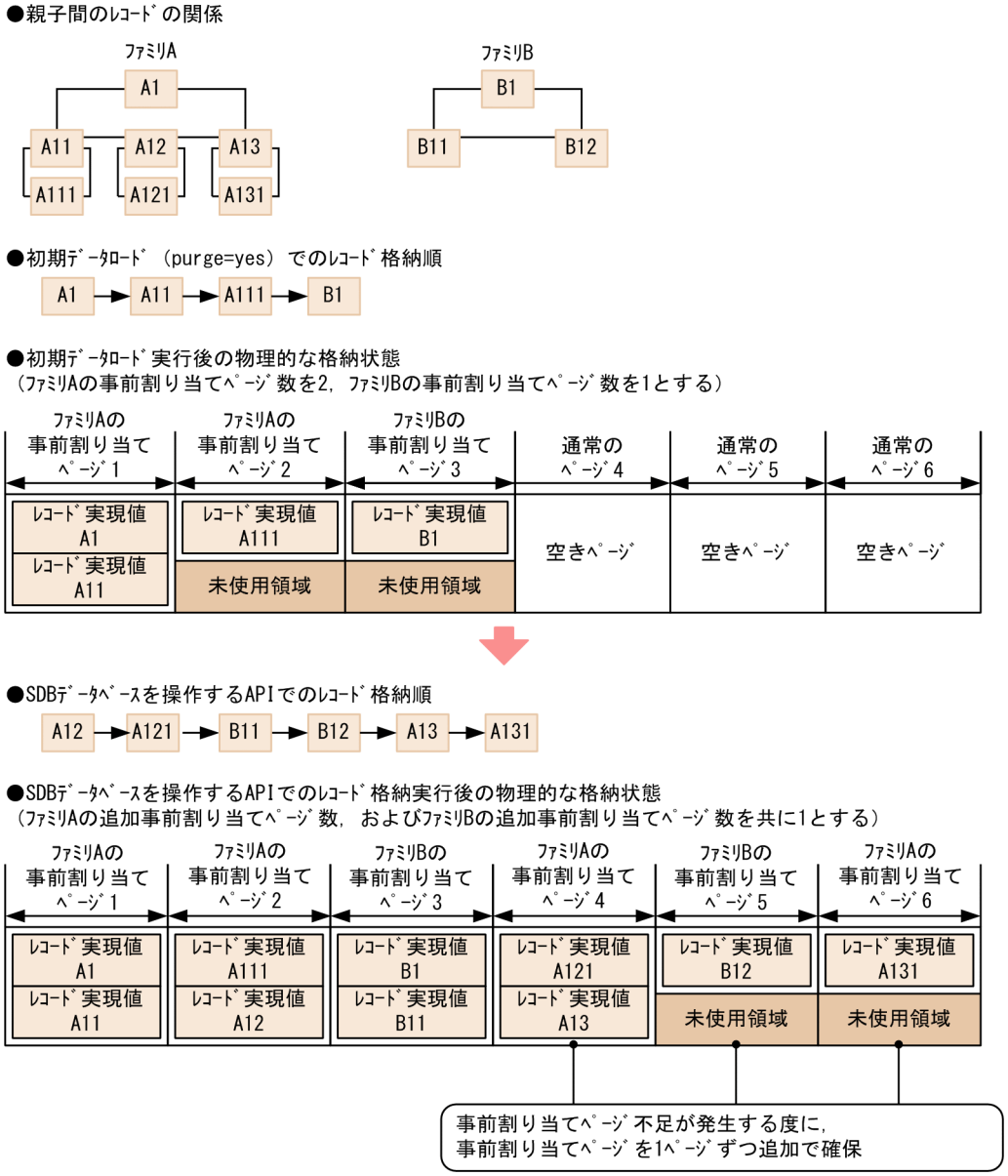


なお，SDB データベースを操作する API のレコードの削除によってページが解放され，未使用ページが存在する状態で追加データロードを実行した場合でも，そのページが再利用されることで，同様に非連続のデータページが確保されることがあります。ただし，次の場合，この状態は pdsdblod コマンドまたは SDB データベースを操作する API を実行する前の，使用中の最終セグメントでだけ発生します。

- 追加データロード時
- 新規にセグメントを確保できる状況での，SDB データベースを操作する API によるルートレコードの格納時

事前割り当てページ不足発生時に，非連続のデータページが事前割り当てページとして確保される例（2.の例）を次の図に示します。

図 2-69　事前割り当てページ不足発生時の非連続データページ（事前割り当てページ）の確保例



この図では，1 ページずつ追加しているため，データページの連続性が大きく損なわれています。この問題は，一定量の事前割り当てページを一括して確保することで緩和できます。

非連続のデータページが事前割り当てページとして確保されているかどうかは，データベース状態解析ユティリティ（pddbst）で確認できます。

この問題を解消するには，SDB データベースを再編成する必要があります。

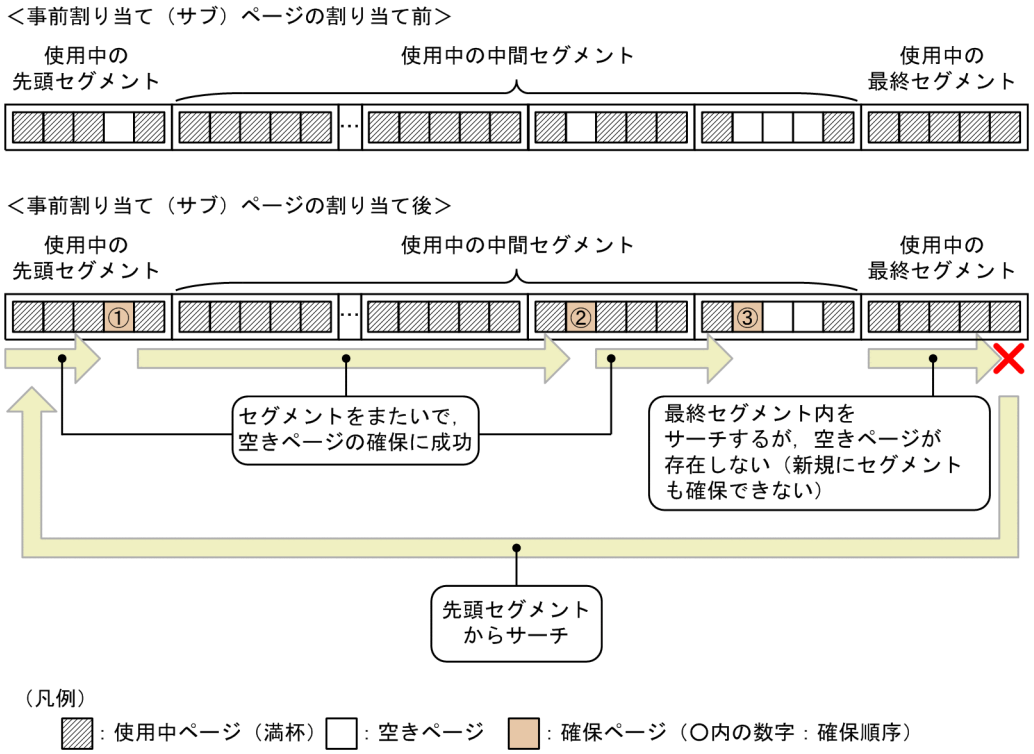
- 確保する事前割り当てページの数に比例して，レコードの格納性能は低下します。そのため，事前割り当てページの連続性とレコードの格納性能を考慮して，指定する事前割り当てページ数を決定する必要があります。ただし，ルートレコードを格納する際には，事前割り当てページの連続性を優先させ，ファミリ内のすべてのレコードを格納できるだけの事前割り当てページを一括して確保することを推奨します。
- 事前割り当てページを確保する際，基本的には使用中の最終セグメント，または新規に確保したセグメント内の空きページが使用されます。これらのセグメントから空きページを確保できなくなると，それ

以外の使用中のセグメントから空きページが確保されます。このとき，次のようなケースでは，レコードの格納性能が大きく低下します。

- 空きページが散在している。
- 空きページを持たないセグメントが多数存在する。

空きページのサーチ例を次の図に示します。

図 2-70　空きページのサーチ例



このような場合，次のどちらかの対応が必要です。

- HiRDB/SD データベース再編成ユティリティ（pdsdbrog）と HiRDB/SD データベース作成ユティリティ（pdsdblod）で，SDB データベースを再編成します。
- データベース構成変更ユティリティ（pdmod）で，レコード格納用 RD エリアを拡張します。

「図 2-70　空きページのサーチ例」に示した空きページのサーチが動作して，レコードの格納性能が低下する前に，上記の対応をすることを推奨します。そのためには，システム共通定義の pd_rdarea_warning_point オペランドを指定します。これによって，次のメッセージが出力されるため，RD エリアの使用状況を監視できます。

・RD エリアのセグメント使用率通知メッセージ（KFPH00211-I または KFPA12300-I）

・増分する HiRDB ファイルの領域使用率通知メッセージ（KFPH22037-W）

詳細については，マニュアル「HiRDB システム定義」の pd_rdarea_warning_point オペランドの説明を参照してください。

- SDB データベースを再編成する際に，事前割り当てページ数を変更したいときは，事前割り当てページ数を指定し直す必要があります。

- 更新可能なオンライン再編成時に，事前ページ割り当て機能の適用有無を変更した場合，ルートレコード下のレコードの格納（STORE）に対する更新追い付き反映処理は次のように動作します。

  適用ありから適用なしに変更した場合

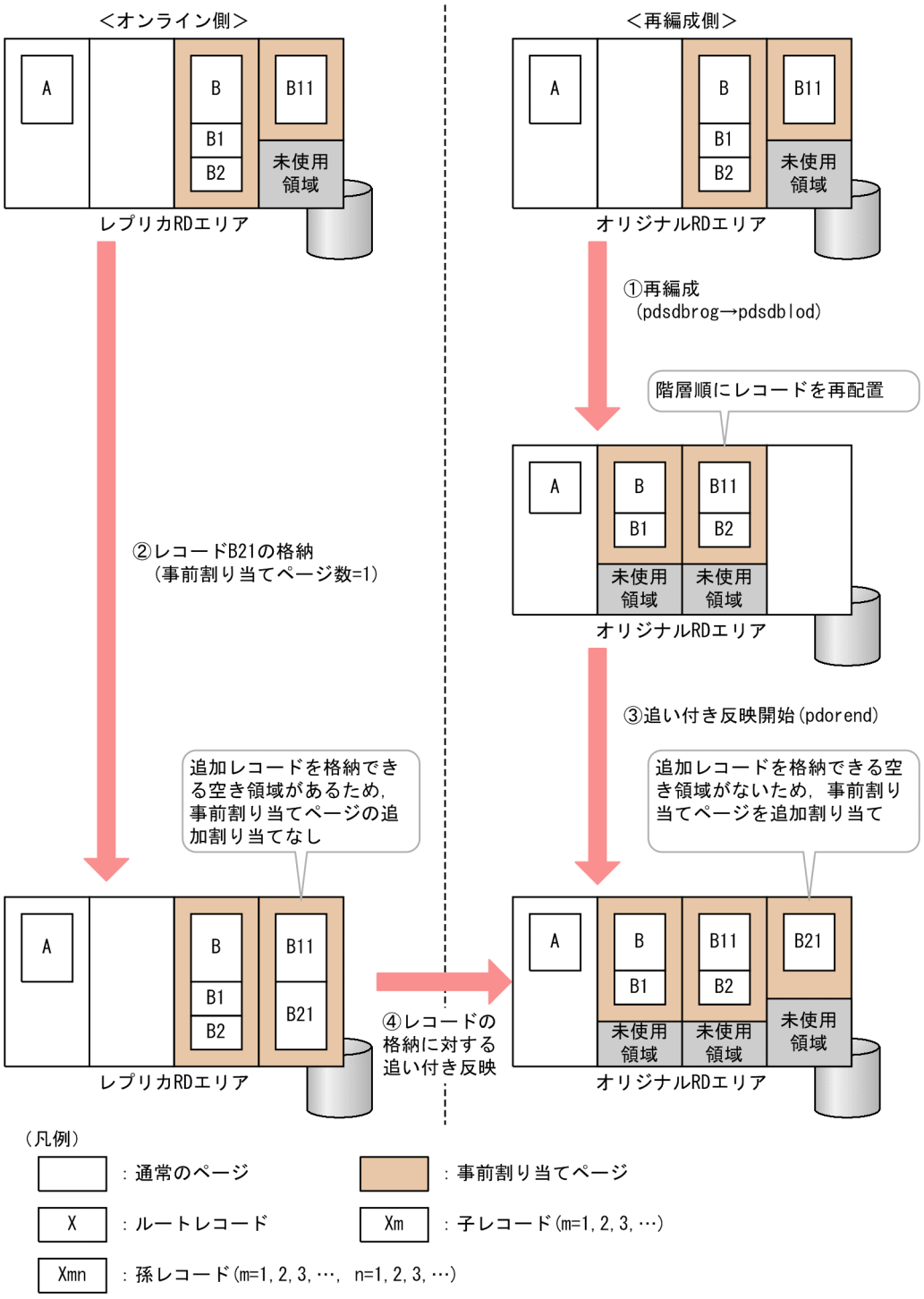
  事前割り当てページ数の指定は無視し，0 が指定されたものとして動作します。

  適用なしから適用ありに変更した場合

  - 事前割り当てページ不足発生時，事前割り当てページ数の指定値として 0 が指定されたものとして動作するため，1 ページだけ確保されます。
  - 次のオプションの指定は無視され，X'00'が指定されたものとして動作します。
    ・ページ切り替えオプション
    ・PCTFREE 有効化オプション

- 更新可能なオンライン再編成時の追い付き反映処理では，SDB データベースを操作する API によるレコードの格納時に指定した事前割り当てページ数の指定値を引き継ぎます。オンライン（レプリカ RD エリア）側と再編成（オリジナル RD エリア）側では，ページの使用状況が異なるため，追い付き反映処理後に確保済みの事前割り当てページの数が一致しないことがあります。更新可能なオンライン再編成の追い付き反映処理での事前割り当てページ数の引き継ぎ例を次の図に示します。

図 2-71　更新可能なオンライン再編成の追い付き反映処理での事前割り当てページ数の引き継ぎ例



なお，上記と同様にページの使用状況が変わるため，SDB データベースの再編成によって，確保済みの事前割り当てページの数が増加することがあります。

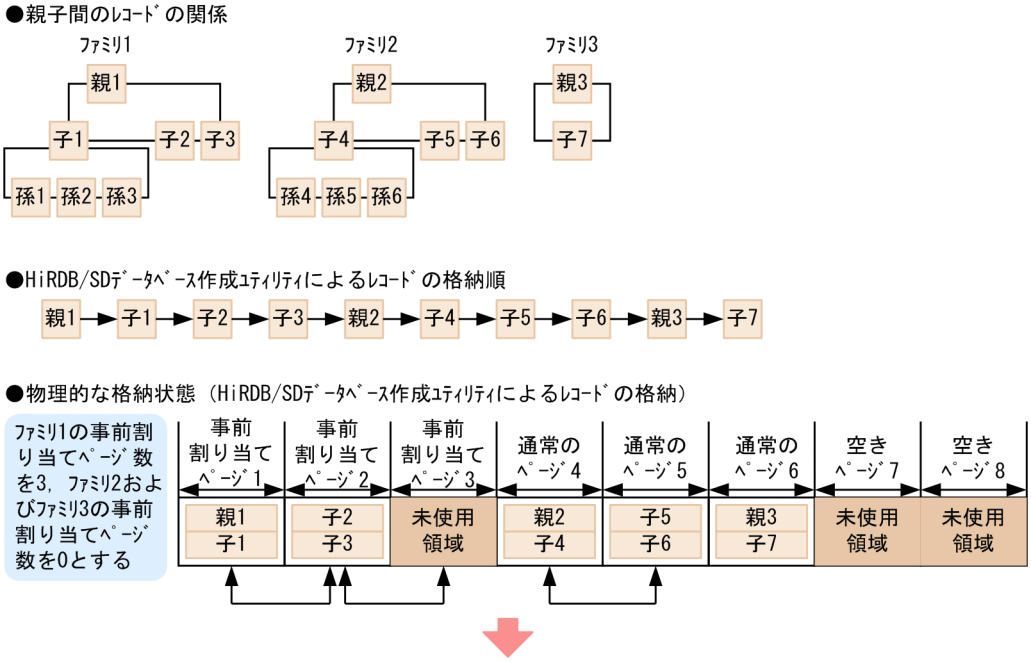
- 「図 2-71　更新可能なオンライン再編成の追い付き反映処理での事前割り当てページ数の引き継ぎ例」で示したように，SDB データベースを操作する API によるレコードの格納に対する追い付き反映処理によって，事前割り当てページが追加で確保された場合，オンライン（レプリカ RD エリア）側のトランザクションがロールバックしても，追加で確保された事前割り当てページは解放されません。なお，

SDB データベースを操作する API による，ルートレコードの格納時に確保された事前割り当てページの場合は解放されます。

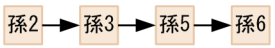
(e)　事前ページ割り当て機能を適用したファミリの格納例

データロード時に事前ページ割り当て機能を適用したファミリと適用していないファミリの格納例を次の図に示します。

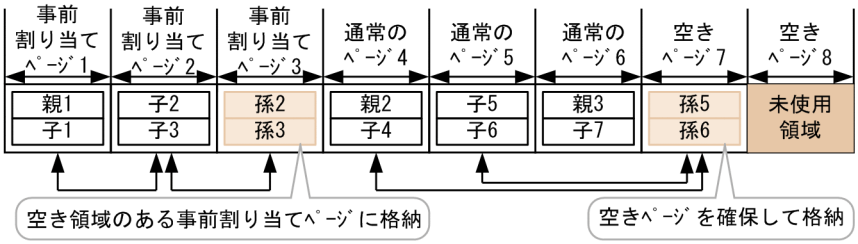
図 2-72　事前ページ割り当て機能の適用有無の違いによるファミリの格納例（サブページ分割をしない場合）



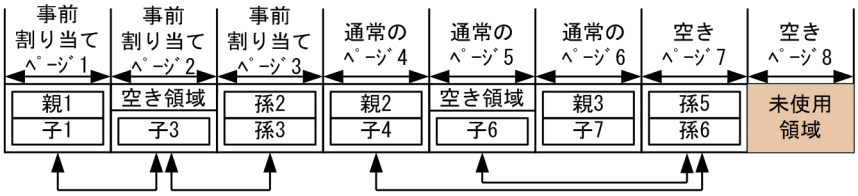
●SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの格納順



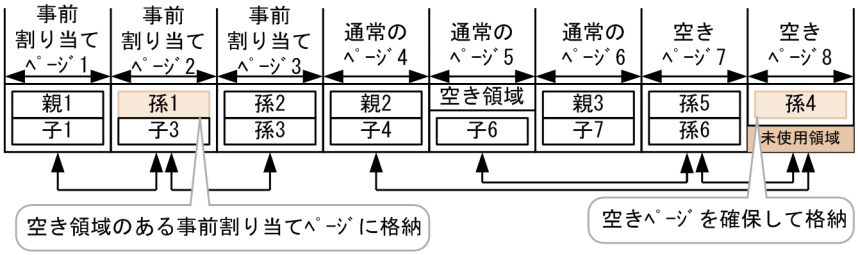
●物理的な格納状態（SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの格納）



●物理的な格納状態（SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの削除（子2，子5））



●物理的な格納状態（SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの格納（孫1，孫4））



（凡例）

：ページ間の関連づけ

このように，事前ページ割り当て機能を適用しているファミリの場合は，事前割り当てページ内にレコードを格納できる空き領域があるときは，その空き領域を再利用してレコードが格納されます。事前ページ割り当て機能を適用していないファミリの場合は，確保済みのページ内にレコードを格納できる空き領域があっても，その空き領域を再利用しないで，空きページを確保してレコードが格納されることがあります。

なお，事前割り当てページ不足が発生し，追加で事前割り当てページが確保される場合の格納例については，「図 2-69　事前割り当てページ不足発生時の非連続データページ（事前割り当てページ）の確保例」を参照してください。

(f)　適用基準

次のような場合に，事前ページ割り当て機能の適用を検討してください。

- 2 階層以上のレコードの格納およびレコードの削除を繰り返し実行するファミリへの連続したアクセスがあり，このとき一定のアクセス性能を維持したい場合（ページ切り替え指定や，PCTFREE オペランドまたは recfree オペランド指定で厳密な空き領域設計ができる場合を除く）

- ほかの平均的なサイズのファミリと比べて，サイズが極端に大きいファミリであり，個別に格納領域の設計を行う必要がある場合
- ページ切り替えを行うための厳密な見積もりが困難である場合

(g) 警告情報の出力

HiRDB/SD データベース作成ユティリティ（pdsdblod）での警告情報の出力

データロードによるレコードの格納時，レコードを格納するために確保したページ数が，指定した事前割り当てページ数を超えて事前ページを確保した場合，ファミリ単位に KFPB63518-I メッセージが出力されます。KFPB63518-I メッセージについては，「12.7　実行結果ファイルの出力形式」を参照してください。

HiRDB/SD データベース再編成ユティリティ（pdsdbrog）での警告情報の出力

アンロード時，事前ページ割り当て機能が適用されているファミリが存在するのに，アンロードデータファイル内にプリフィクス部が存在しないため，事前割り当てページ数が取得されなかった場合，KFPB63516-W メッセージが出力されます（事前割り当てページ数の引き継ぎが行われなかったことが通知されます）。

SDB データベースを操作する API によるレコードの格納（STORE）での警告情報の出力

SDB データベースを操作する API によるレコードの格納時に，事前割り当てページが不足して，事前割り当てページが追加で確保された場合，レコード格納用 RD エリア単位に KFPB32000-W～KFPB32003-W メッセージが出力されます（事前割り当てページ数の再見積もりの要否確認が必要であることが通知されます）。

なお，このメッセージは 1 回出力されると，次回以降の事前割り当てページの不足時に，事前割り当てページを追加で確保してもメッセージが出力されません。この場合，次のどれかの操作が行われるまでメッセージが出力されません。メッセージの出力抑止状態の解除条件，および出力対象となる RD エリアを次に示します。

出力抑止状態の解除条件

- クローズ状態の該当するレコード格納用 RD エリアを pdopen コマンド，または pdrels -o コマンドでオープンしたとき
- 該当するレコード格納用 RD エリアに対して初期データロードを実行したとき
- HiRDB を再起動したとき

出力対象となる RD エリア

- インナレプリカ機能を使用していないレコード格納用 RD エリアの場合，出力対象となります。
- レコード格納用 RD エリアにインナレプリカ機能を使用している場合，オリジナル RD エリアが出力対象となります。

(5)　更新可能なオンライン再編成の追い付き反映処理でレコードをページに格納する場合【4V FMB】

更新可能なオンライン再編成の追い付き反映処理時に有効となる格納配置制御のオプションを次の表に示します。

表 2-24　追い付き反映処理時に有効となる格納配置制御のオプション

追い付き反映処理の種類	格納配置制御のオプション		
	ページ切り替えオプション	PCTFREE 有効化オプション	事前割り当てページ数
レコードの格納（STORE）の更新追い付き反映処理	○	○	○
レコードの削除（ERASE）の回復追い付き反映処理	×	×	−

（凡例）

○：オンライン（レプリカ RD エリア）側で，レコードの格納（STORE）時に指定した格納配置制御のオプションの指定を適用します。SDB データベースの再編成時に，事前ページ割り当て機能の適用有無を変更した場合の動作については，「(4)(d)　事前ページ割り当て機能に関する留意事項」を参照してください。

×：格納配置制御のオプションの指定を適用しません。

- ページを切り替えません。
- SDB データベース格納定義の PCTFREE 句で指定したページ内の未使用領域の比率を適用しないで，レコード実現値を格納します。

−：該当しません。

(a)　レコードの格納（STORE）の更新追い付き反映処理時の格納配置制御

レコードの格納（STORE）の更新追い付き反映処理時の格納配置制御では，レコードを事前割り当てページに格納する場合を除いて，オンライン（レプリカ RD エリア）側で指定した格納配置制御のオプションの指定が適用されます。

ページ切り替えオプションの指定の違いによる格納配置制御については，「図 2-62　ページ切り替えオプションによるレコードの格納例（4V FMB の SDB データベースの場合）」を参照してください。

PCTFREE 有効化オプションを適用した場合の格納配置制御については，「図 3-10　ページに格納するレコード件数を制限する例」を参照してください。

事前ページ割り当て機能の適用有無の違いによる格納配置制御については，「図 2-72　事前ページ割り当て機能の適用有無の違いによるファミリの格納例（サブページ分割をしない場合）」を参照してください。なお，この図では，HiRDB/SD データベース作成ユティリティ（pdsdblod）を実行する場合に，事前割

り当てページ数を指定したときのファミリの格納例を示しています。SDB データベースを操作する API でレコードの格納を実行する場合に，事前割り当てページ数を指定したときも同様の格納配置となります。

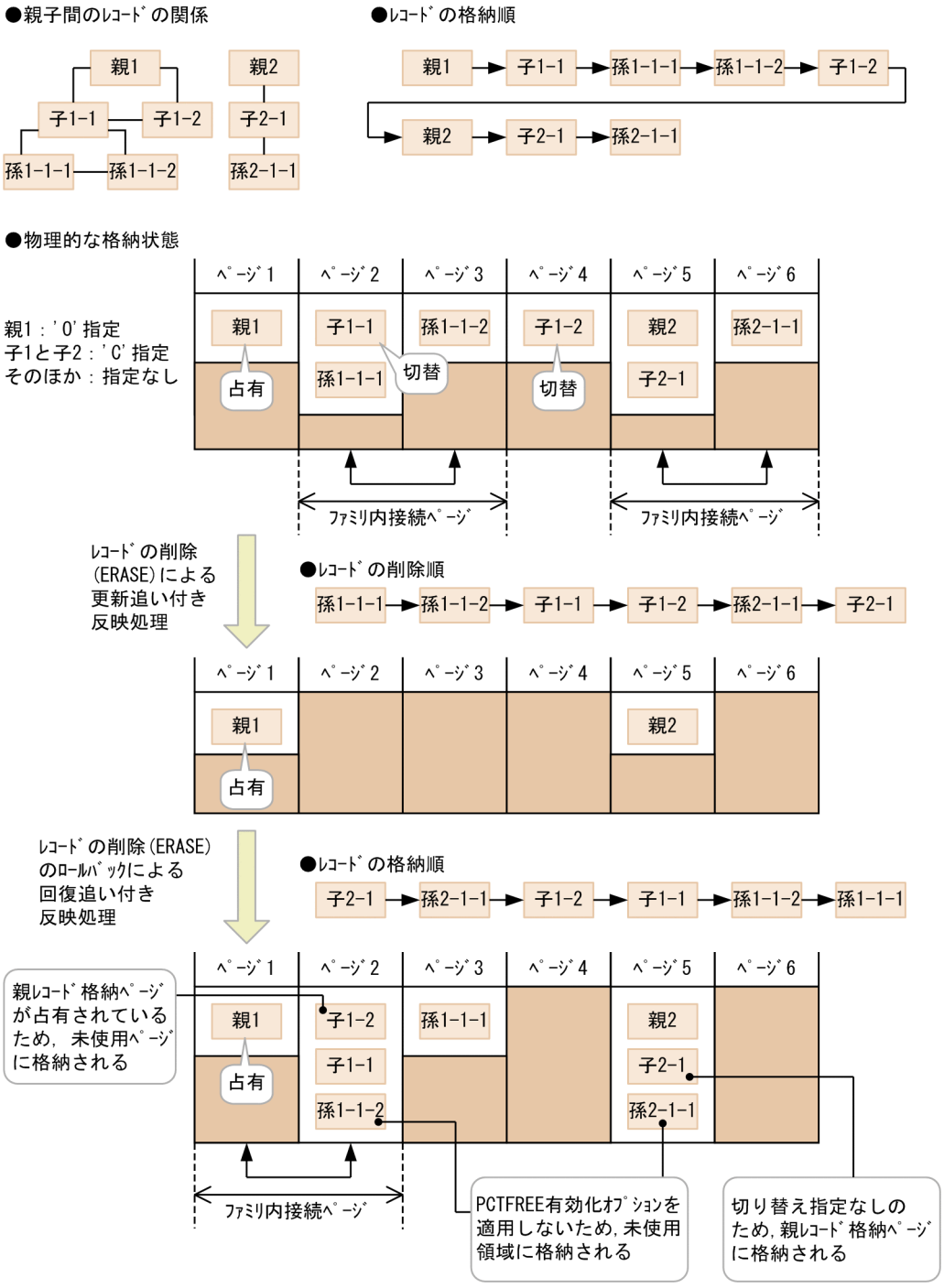
ルートレコード下のレコードの格納時に，事前割り当てページが不足した場合の格納配置制御については，「図 2-69　事前割り当てページ不足発生時の非連続データページ（事前割り当てページ）の確保例」を参照してください。

(b)　レコードの削除（ERASE）の回復追い付き反映処理時の格納配置制御

レコードの削除（ERASE）の回復追い付き反映処理時の格納配置制御では，オンライン（レプリカ RD エリア）側で指定した格納配置制御のオプションの指定が適用されません。

ページ切り替えオプションを指定し，PCTFREE 有効化オプションを適用したときの回復追い付き反映処理時の格納配置制御を次の図に示します。

図 2-73　レコードの削除（ERASE）の回復追い付き反映処理時の格納配置制御



2.7　レコードの配置制御（サブページ分割をする場合）

ここでは，サブページ分割をする場合のレコードの配置制御について説明します。サブページ分割をするときの配置制御の仕組みは，サブページ分割をしないときの配置制御の仕組みと基本的には同じになります。ページをサブページに置き換えて考えてください。

レコード長が短いレコードを格納する場合にサブページ分割をすると，むだな領域の発生を抑えることができ，レコードの格納効率が向上します。

サブページ分割をしている場合，HiRDB/SD はレコードを次の表に示すように格納します。

表 2-25　HiRDB/SD のレコードの配置制御（サブページ分割をする場合）

レコードの配置制御	SDB データベース種別		
	4V FMB	4V AFM	SD FMB
• 同一のルートレコード下のレコードを近傍に配置して格納する。 • 異なるルートレコード下のレコードをサブページ内に混在させない。 • HiRDB/SD データベース作成ユティリティ（pdsdblod）を使用したデータロード時に，サブページ内の未使用領域，セグメント内の空きページを設定してレコードを格納できる。	○	○	○
• レコード実現値ごとに格納するサブページを切り替え，未使用サブページに格納できる。※ • 1 つのレコード実現値でサブページを占有できる（1 件/サブページ）。※ • SDB データベースを操作する API によるレコード格納時に，サブページ内の未使用領域の比率に従った格納ができる。※	○	○	×
ルートレコードの格納時に事前割り当てサブページをあらかじめ確保しておくことで，ルートレコード下のレコードの格納およびレコードの削除を繰り返し実行した場合でも，連続したデータページ内にレコードを配置できる（この機能を事前ページ割り当て機能といいます）。	○	×	×

（凡例）

○：レコードの配置制御が適用されます。

×：レコードの配置制御は適用されません。

注※

オプションの指定が必要になります。

レコード格納時，およびデータロード時の空き領域の作成方法については，「3.2.6　レコード格納時の空き領域の作成（サブページ分割をする場合）」を参照してください。

SD FMB の SDB データベースの場合，DML によるレコードの格納時に空き領域を作成することはできません。

2.7.1　親レコードと子レコードの近傍配置

ここでは親レコードと子レコードの近傍配置について説明します。

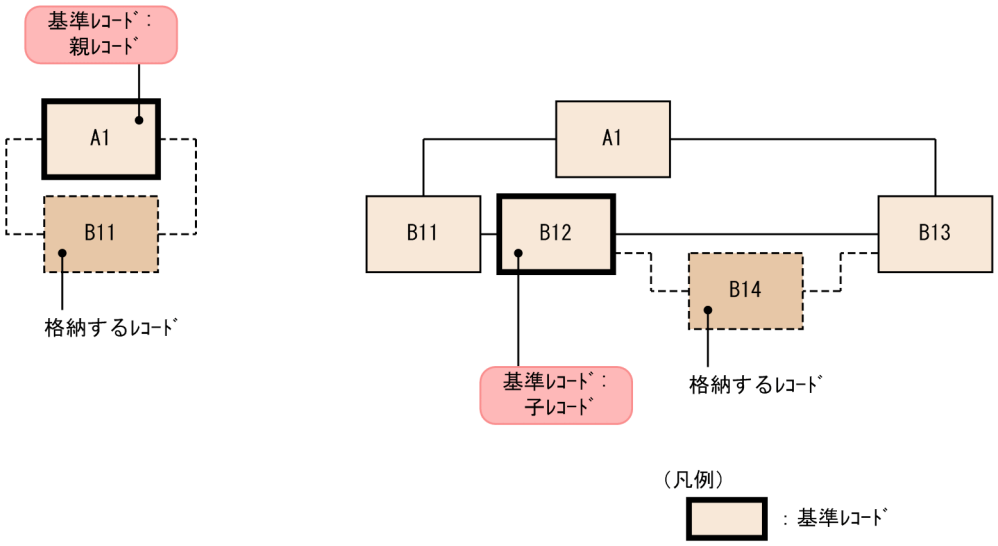
(1)　基準レコード，基準サブページ

HiRDB/SD は，基準とするレコードから，レコードを格納する位置を決定します。基準とするレコードを基準レコードといいます。

基準レコードは，格納するレコードの PRIOR ポインタが指すレコードになります。親子集合の場合は，1 件目の子レコードに対する基準レコードは，親レコードとなります。なお，ルートレコードは基準レコードを持ちません。

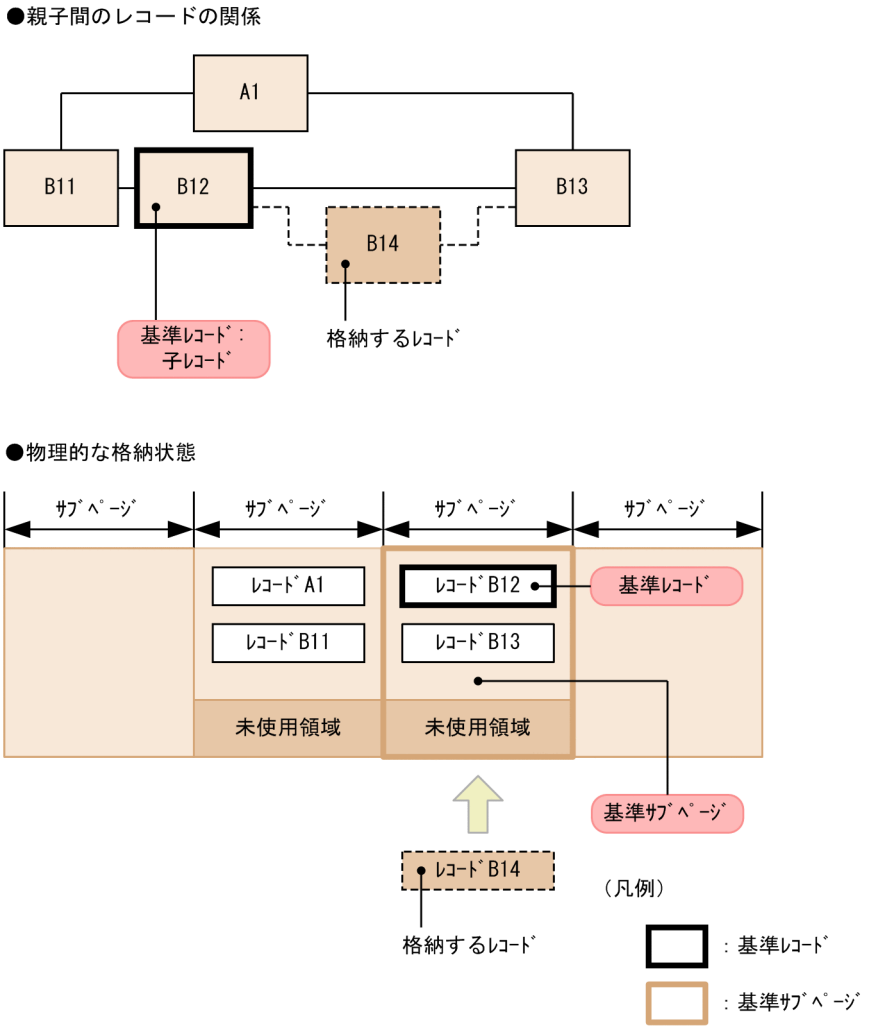
基準レコードの例を次の図に示します。

図 2-74　基準レコードの例



基準レコードが格納されているサブページを基準サブページといいます。HiRDB/SD では，基準サブページおよびその近傍のサブページにレコードを格納することで，親子集合内のレコード群を近傍に配置して格納できます。基準サブページへの格納の例を次の図に示します。

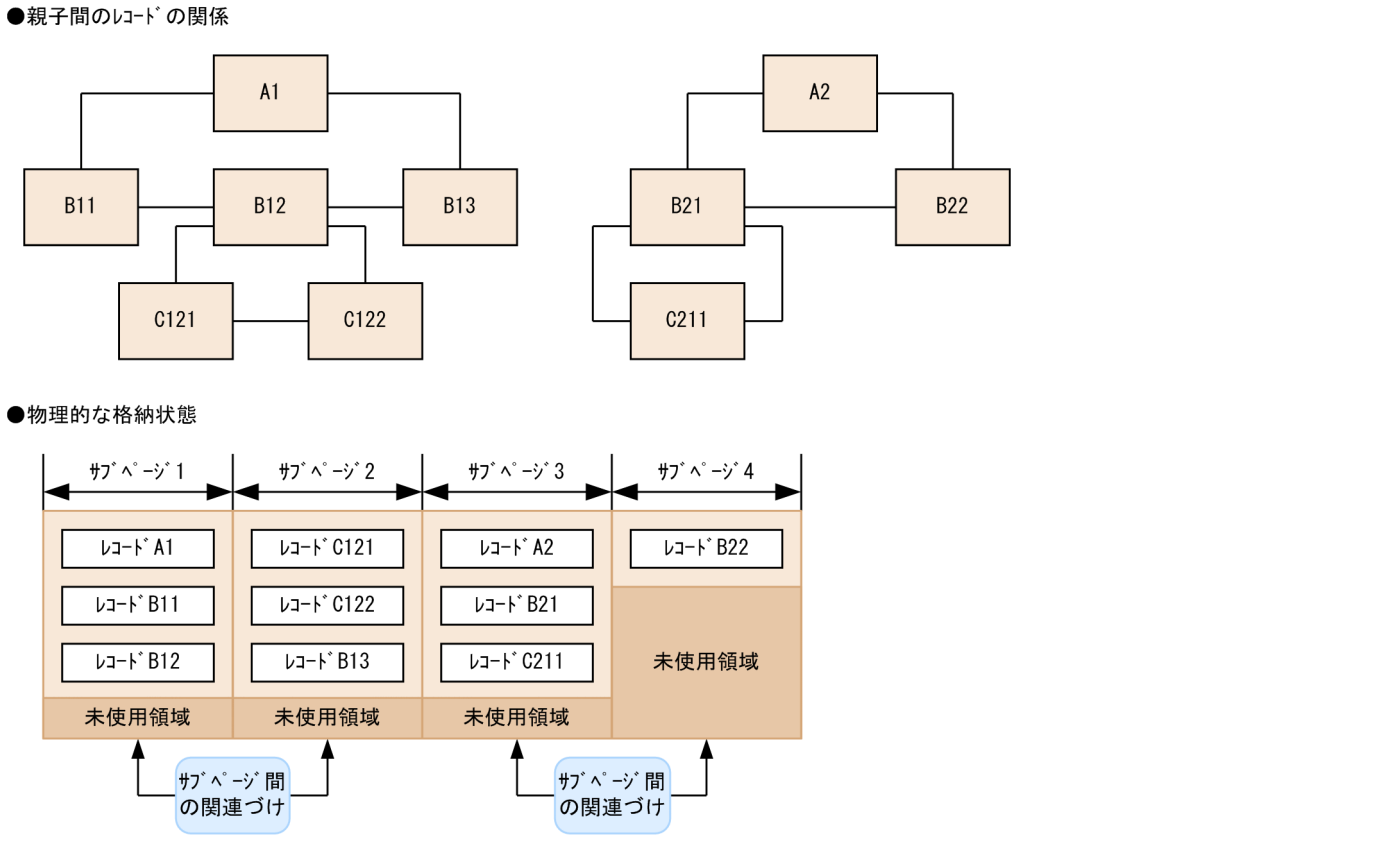
図 2-75　基準サブページへの格納の例



(2)　サブページ間の関連づけとファミリ内接続サブページ

HiRDB/SD では，同一のルートレコード下のレコードをファミリといい，ファミリを格納するサブページを関連づけて管理します。関連づけられたサブページをファミリ内接続サブページといいます。サブページ間の関連づけとファミリ内接続サブページの例を次の図に示します。

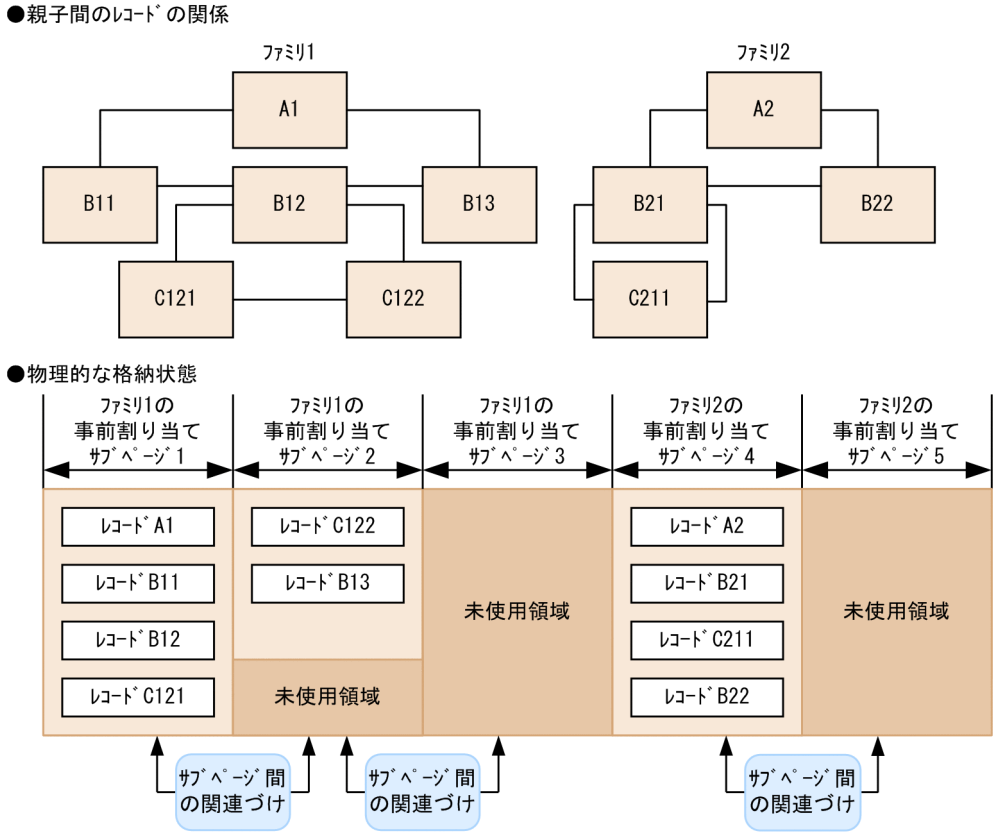
図 2-76　サブページ間の関連づけとファミリ内接続サブページの例



HiRDB/SD は，確保した未使用サブページにレコードを格納する際に，サブページ間を関連づけることによって，基準サブページとレコードを格納するサブページをファミリ内接続サブページとして組み入れます。

なお，ファミリ内の事前割り当てサブページは，すべてファミリ内接続サブページ群として管理されます。事前ページ割り当て機能を適用している場合の，サブページ間の関連づけとファミリ内接続サブページの例を次の図に示します。

図 2-77　事前ページ割り当て機能を適用している場合の，サブページ間の関連づけとファミリ内接続サブページの例



2.7.2　レコードを格納するサブページの決定方法

レコードを格納するには，次の 2 つの方法があります。

- SDB データベースを操作する API または DML でレコードを格納する方法
- HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードを格納する方法

各方法でのレコードを格納するサブページの決定方法について説明します。

参考

- SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。
- DML については，「17. DML リファレンス【SD FMB】」を参照してください。

(1)　SDB データベースを操作する API または DML でレコードをサブページに格納する場合

SDB データベースを操作する API または DML でレコードを格納する場合，そのファミリに事前ページ割り当て機能を適用しているかどうかによって，格納するサブページの使用順序が異なります。事前ページ割り当て機能については，「(4)　事前ページ割り当て機能【4V FMB】」を参照してください。

(a)　事前ページ割り当て機能を適用していない場合

事前ページ割り当て機能を適用していない場合，次の表に示す優先順位に従って格納するサブページを決定します。優先順位 1～6 のサブページにレコードを格納できない場合，RD エリアのページ不足となります。

表 2-26　SDB データベースを操作する API または DML でレコードを格納する場合の優先順位（事前ページ割り当て機能を適用していない場合）

優先順位	レコード種別	
	ルートレコード	2 階層以上のレコード
1	未使用サブページ	基準サブページ※
2	－	基準サブページが属するページ内のファミリ内接続サブページ※
3	－	基準サブページが属するページ内の未使用サブページ※
4	－	基準サブページが属するページのセグメントおよびその直前のセグメント内の未使用サブページ
5	－	優先順位 4 以外の未使用サブページ
6	－	優先順位 2 以外の全ファミリ内接続サブページ※

（凡例）

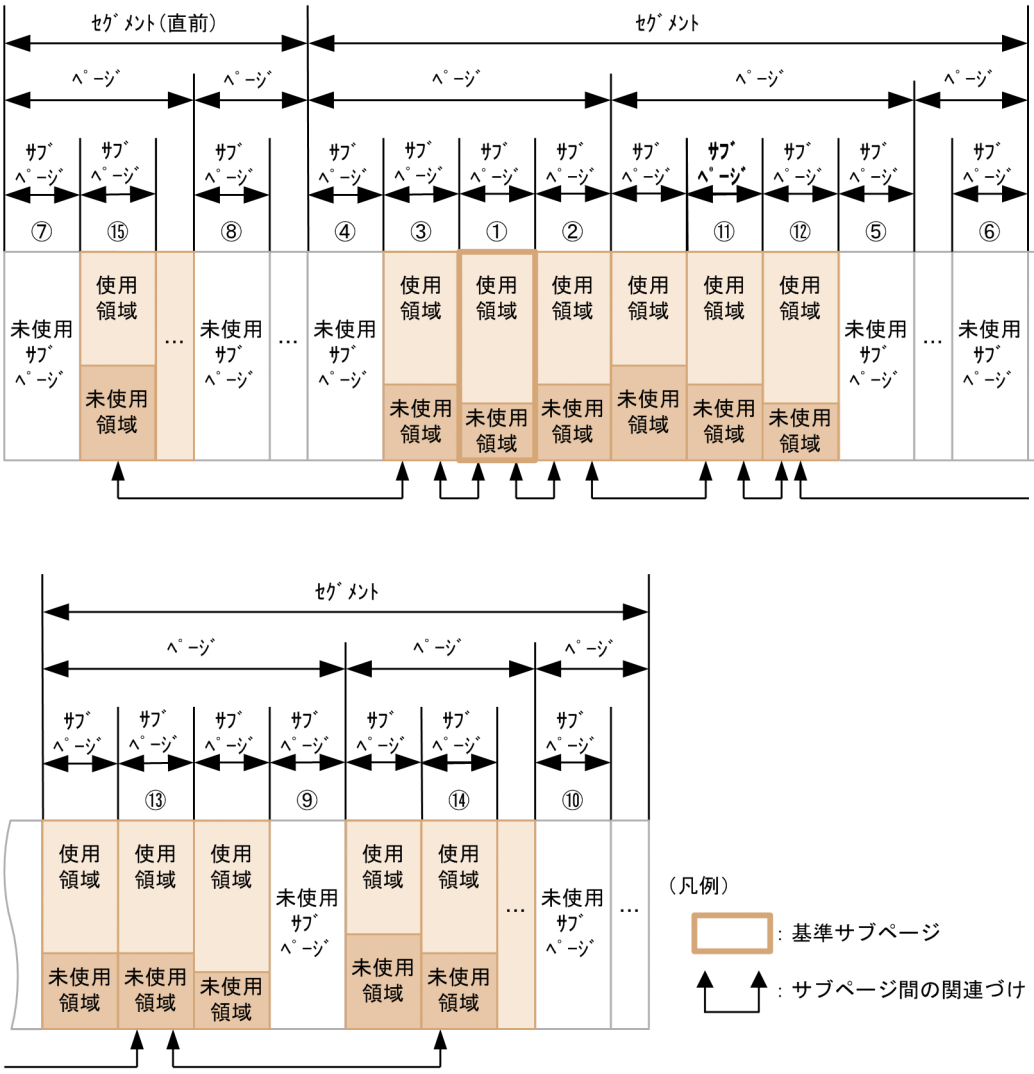
－：該当しません。

注※

サブページ切り替えの指定がある場合，格納の対象になりません。サブページ切り替えについては，「2.7.2(3)　レコード格納時のサブページ切り替え【4V FMB，4V AFM】」を参照してください。

優先順位に従った 2 階層以上のレコードの格納時の動作を次の図に示します。

図 2-78　優先順位に従った 2 階層以上のレコードの格納時の動作（事前ページ割り当て機能を適用していない場合）



＜優先順位1＞
①：基準サブページに格納を試みる。

＜優先順位2＞
②，③：基準サブページが属するページ内のファミリ内接続サブページに格納を試みる。

＜優先順位3＞
④：基準サブページが属するページ内の未使用サブページに格納を試みる。

＜優先順位4＞
⑤，⑥：基準サブページが属するページのセグメントに未使用サブページがあれば使用する。
⑦，⑧：⑤，⑥の未使用サブページがない場合，基準サブページが属するページのセグメントの直前のセグメントに未使用サブページがあれば使用する。

＜優先順位5＞
⑨，⑩：⑤～⑧の未使用サブページがない場合，それ以外の未使用サブページがあれば使用する。

＜優先順位6＞
⑪，⑫，⑬，⑭，⑮：残りのファミリ内接続サブページに格納を試みる。

(b) 事前ページ割り当て機能を適用している場合【4V FMB】

4V FMB の SDB データベースのルートレコードを格納する場合，指定数分の事前割り当てサブページを確保し，最初に確保した事前割り当てサブページを使用します。また，2 階層以上のレコードを格納する場合，基本的には確保済みの事前割り当てサブページを使用します。ただし，確保済みの事前割り当てサブページ内にレコードを格納できる空き領域がない場合は，指定数分の事前割り当てサブページを追加で確保し，最初に確保した事前割り当てサブページを使用します。

事前ページ割り当て機能を適用している場合，次の表に示す優先順位に従って，格納するサブページを決定します。優先順位 1～7 のサブページにレコードを格納できない場合，RD エリアのページ不足となります。

表 2-27 SDB データベースを操作する API でレコードを格納する場合の優先順位（事前ページ割り当て機能を適用している場合）

優先順位	レコード種別	
	ルートレコード	2 階層以上のレコード
1	未使用サブページ※1	基準サブページ
2	－	基準サブページから関連づけられている，同一ページ内のファミリ内接続サブページ
3	－	優先サブページ※2
4	－	優先サブページから関連づけられている，同一ページ内の事前割り当てサブページ
5	－	優先順位 1～4 以外の事前割り当てサブページ
6	－	最終事前割り当てサブページが属するページ内の未使用サブページ※1
7	－	基準サブページが属するページのセグメント，およびその直前のセグメント内の未使用サブページ※1
8	－	優先順位 6 および 7 以外の未使用サブページ※1

（凡例）

－：該当しません。

注※1

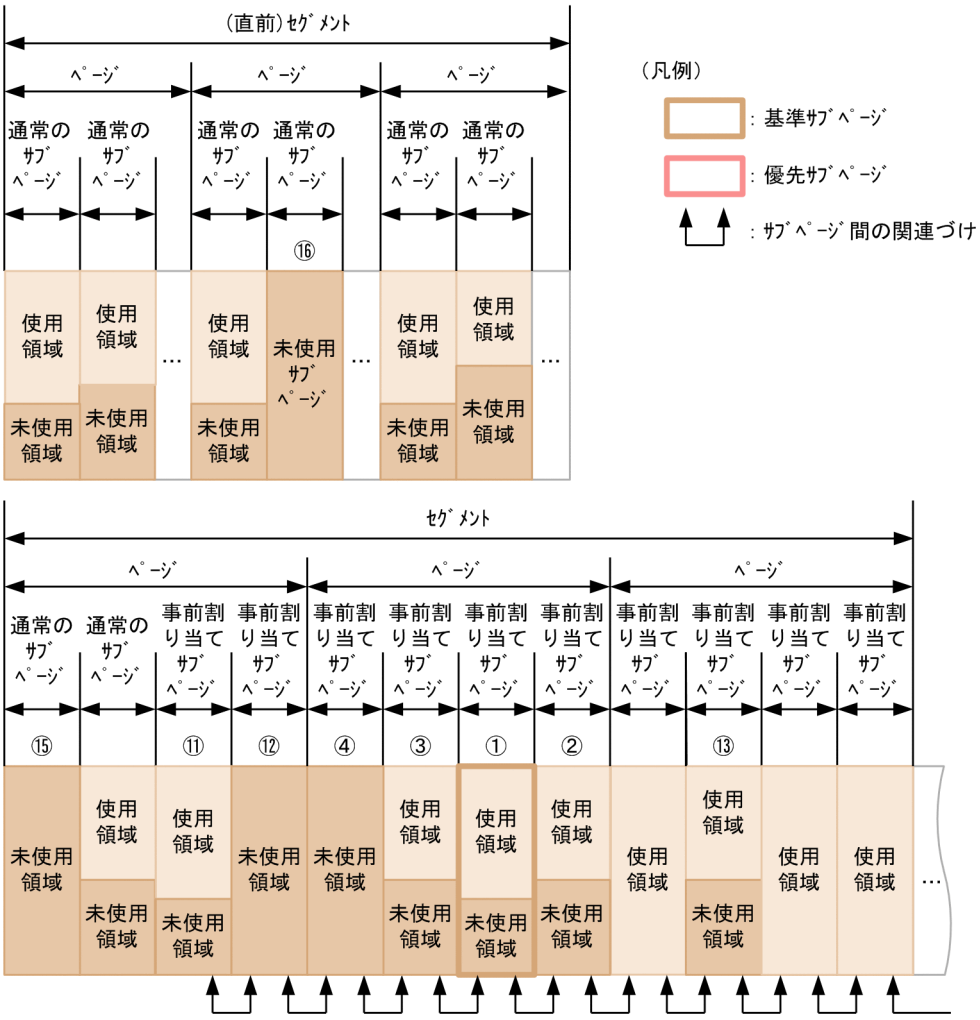
事前割り当てサブページとして確保します。

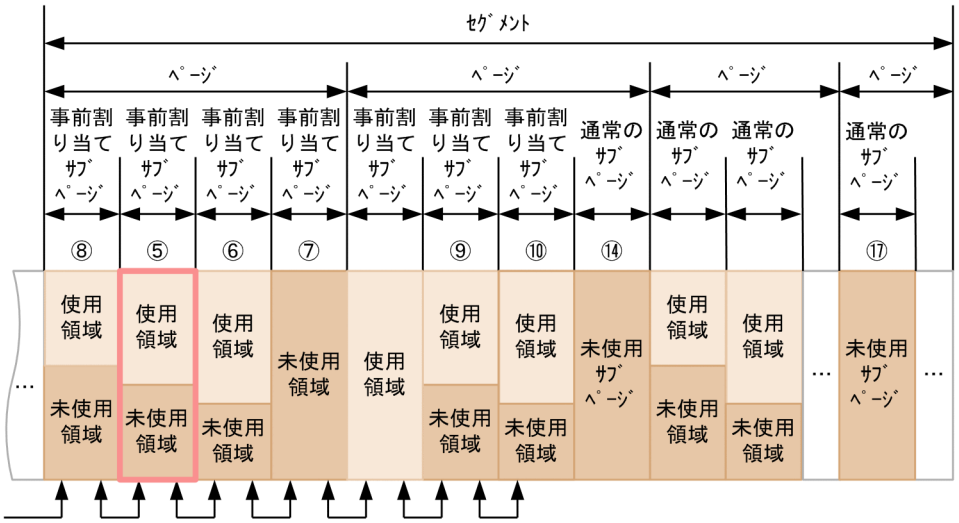
注※2

基準サブページが属するページの次に HiRDB/SD が優先的に格納を試みる事前割り当てサブページです。

優先順位に従った 2 階層以上のレコードの格納時の動作を次の図に示します。

図 2-79　優先順位に従った 2 階層以上のレコードの格納時の動作（事前ページ割り当て機能を適用している場合）





＜格納優先順位1＞
①：基準ｻﾌﾞﾍﾟｰｼﾞに格納を試みる。
＜格納優先順位2＞
②,③,④：基準ｻﾌﾞﾍﾟｰｼﾞから関連づけられている同一ﾍﾟｰｼﾞ内の事前割り当てｻﾌﾞﾍﾟｰｼﾞに格納を試みる。
＜格納優先順位3＞
⑤：優先ｻﾌﾞﾍﾟｰｼﾞに格納を試みる。
＜格納優先順位4＞
⑥,⑦,⑧：優先ｻﾌﾞﾍﾟｰｼﾞから関連づけられている同一ﾍﾟｰｼﾞ内の事前割り当てｻﾌﾞﾍﾟｰｼﾞに格納を試みる。
＜格納優先順位5＞
⑨,⑩,⑪,⑫,⑬：＜格納優先順位1～4＞以外の事前割り当てｻﾌﾞﾍﾟｰｼﾞに格納を試みる。
＜格納優先順位6＞
⑭：最終事前割り当てｻﾌﾞﾍﾟｰｼﾞが属するﾍﾟｰｼﾞ内に未使用ｻﾌﾞﾍﾟｰｼﾞがあれば使用する。
＜格納優先順位7＞
⑮：基準ｻﾌﾞﾍﾟｰｼﾞが属するﾍﾟｰｼﾞのｾｸﾞﾒﾝﾄに未使用ｻﾌﾞﾍﾟｰｼﾞがあれば使用する。
⑯：基準ｻﾌﾞﾍﾟｰｼﾞが属するﾍﾟｰｼﾞのｾｸﾞﾒﾝﾄの直前のｾｸﾞﾒﾝﾄに未使用ｻﾌﾞﾍﾟｰｼﾞがあれば使用する。
＜格納優先順位8＞
⑰：⑭⑮⑯の未使用ｻﾌﾞﾍﾟｰｼﾞがない場合，それ以外の未使用ｻﾌﾞﾍﾟｰｼﾞがあれば使用する。

(2) HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードをサブページに格納する場合

HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードをサブページに格納する場合，次の表に示す優先順位に従って，格納するサブページを決定します。優先順位 1～2 のサブページにレコードを格納できない場合，RD エリアのページ不足となります。

表 2-28 HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードをサブページに格納する場合の優先順位

レコード種別	優先順位	
	1	2
ルートレコードの格納	未使用サブページ	－
2 階層以上のレコードの格納	直近でレコードを格納したサブページ※	未使用サブページ

（凡例）
－：該当しません。

注※

サブページ切り替えの指定がある場合，格納の対象になりません。サブページ切り替えについては，「(3)　レコード格納時のサブページ切り替え【4V FMB，4V AFM】」を参照してください。

(3)　レコード格納時のサブページ切り替え【4V FMB，4V AFM】

レコード格納時のサブページ切り替えについて説明します。サブページ切り替えは，4V FMB または 4V AFM の SDB データベースの場合に指定できます。

(a)　サブページ切り替えの概要

4V FMB または 4V AFM の SDB データベースにレコードを格納する際，格納対象のサブページを切り替えて，未使用サブページにレコードを格納できます。これをサブページ切り替えといいます。また，このとき，格納したレコードだけでそのサブページを占有し，ほかのレコードが格納されないようにすることもできます。格納対象のサブページを切り替えることによって，レコードに対するアクセスをサブページ単位に分散し，排他の競合を回避できます。

サブページ切り替えの指定には，次の 2 つの指定があります。

- **切り替え指定**
  切り替えたサブページにほかのレコードも格納できる指定
- **占有指定**
  切り替えたサブページを格納するレコードで占有する指定（ほかのレコードは格納できない指定）

サブページ切り替えの指定方法を次の表に示します。

表 2-29　サブページ切り替えの指定方法

格納方法		指定方法
HiRDB/SD データベース作成ユティリティ（pdsdblod）	フォーマットライト機能使用時	dbinf 文の pagecflg でページ切り替えフラグを指定します。dbinf 文については，「12.3.3　dbinf 文【4V FMB，4V AFM】」を参照してください。
	上記以外	入力データファイルのプレフィクス部のページ切り替えフラグを指定します。詳細については，「12.3.3　dbinf 文【4V FMB，4V AFM】」を参照してください。
SDB データベースを操作する API によるレコード格納		SDB データベースを操作する API に，ページ切り替えオプションを指定することによって，レコードの格納時にサブページを切り替えることができます。 'C'指定（切り替え指定） レコード格納時にサブページを切り替え，未使用サブページを確保します。確保したサブページにほかのレコードを格納できます。 'O'指定（占有指定） レコード格納時にサブページを切り替え，未使用サブページを確保します。確保したサブページは格納したレコードで占有します。ほかのレコードは格納できません。 'N'または'△'（空白）指定 サブページの切り替えを指定しません。

格納方法	指定方法
	SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

(b)　サブページ切り替えに関する留意事項

- 4V AFM の SDB データベースの場合，仮想ルートレコードの格納時に，SDB データベース格納定義の OCCURRENCE NUMBER 句の指定値が 1 以上のレコードでは，仮想ルートレコードにアクセスする際の排他と，同一仮想ルートレコード下の子レコードにアクセスする際の排他の競合を回避するため，サブページ切り替えの指定が無視されます。このとき，'O'が指定されたと仮定してレコードが格納されます。

  子レコードの格納時には，サブページ切り替えの指定に従ってレコードが格納されます。

- レコードの一括削除後にレコードを格納する場合，一括削除時に存在していたレコードの格納領域が再利用されます。そのため，一括削除時に存在していたレコード件数分のレコードを格納するときには，ページ切り替え指定は無視されます。

- HiRDB/SD は，異なるルートレコード下のレコードをサブページ内に混在させないため，ルートレコードの格納時にサブページを切り替えます。そのため，ルートレコード格納時の'C'指定，または指定なしの場合については，動作に違いはありません。'O'指定の場合には，ルートレコードだけで格納サブページを占有するため，ほかのレコードは格納されません。

- ページ切り替え指定に従ってレコード格納時にサブページを切り替えた場合，確保したサブページと基準サブページとの間で，サブページ間の関連づけはされません。そのため，ページ切り替え指定以前に作成されていたサブページは，確保したサブページのファミリ内接続サブページとはなりません。

  同一ルートレコード下のレコード格納サブページであっても，異なるファミリ内接続サブページに分割された場合には，「表 2-26　SDB データベースを操作する API または DML でレコードを格納する場合の優先順位（事前ページ割り当て機能を適用していない場合）」の優先順位 2 および 6 の対象サブページとはなりません。ただし，サブページの切り替え後にレコードを格納することによって作成されたサブページ間は関連づけられるので，新たなファミリ内接続サブページとなります。サブページ間の関連づけ，ファミリ内接続サブページについては，「2.7.1(2)　サブページ間の関連づけとファミリ内接続サブページ」を参照してください。

- 基準レコードが'O'指定でサブページを占有しているレコードの場合であっても，ページ切り替え指定なしで未使用サブページを確保できているときには，確保したサブページと基準サブページとが関連づけられます。

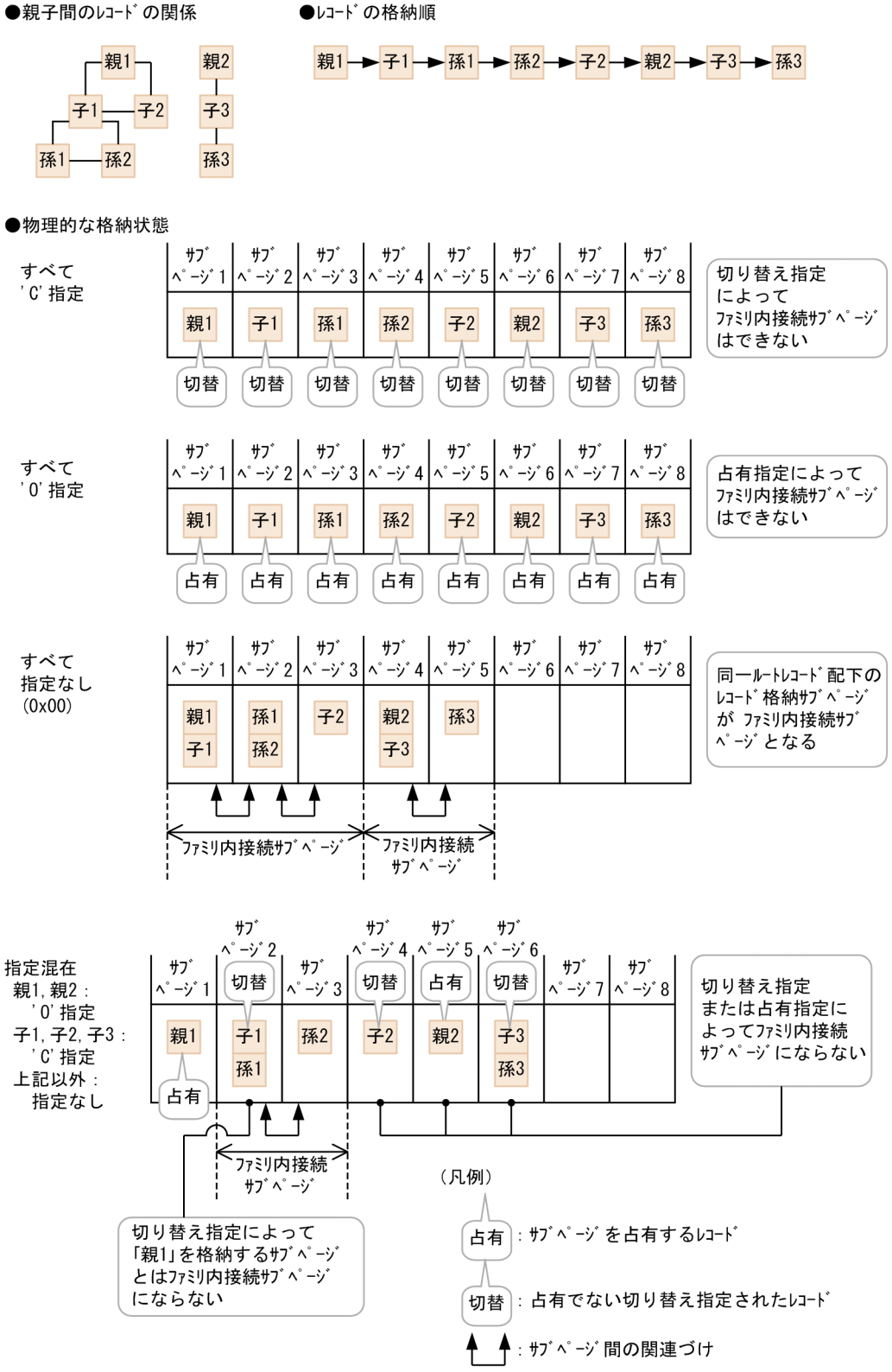
- 事前ページ割り当て機能を適用しているファミリのレコードを格納する場合，サブページ切り替え指定でサブページを切り替えることはできません。サブページ切り替えの指定があるとエラーになります。事前ページ割り当て機能については，「(4)　事前ページ割り当て機能【4V FMB】」を参照してください。

(c)　ページ切り替えオプションによるレコードの格納例

4V FMB の SDB データベースの場合

4V FMB の SDB データベースに対するレコードの格納時の，ページ切り替えオプションによるレコードの格納例を次の図に示します。

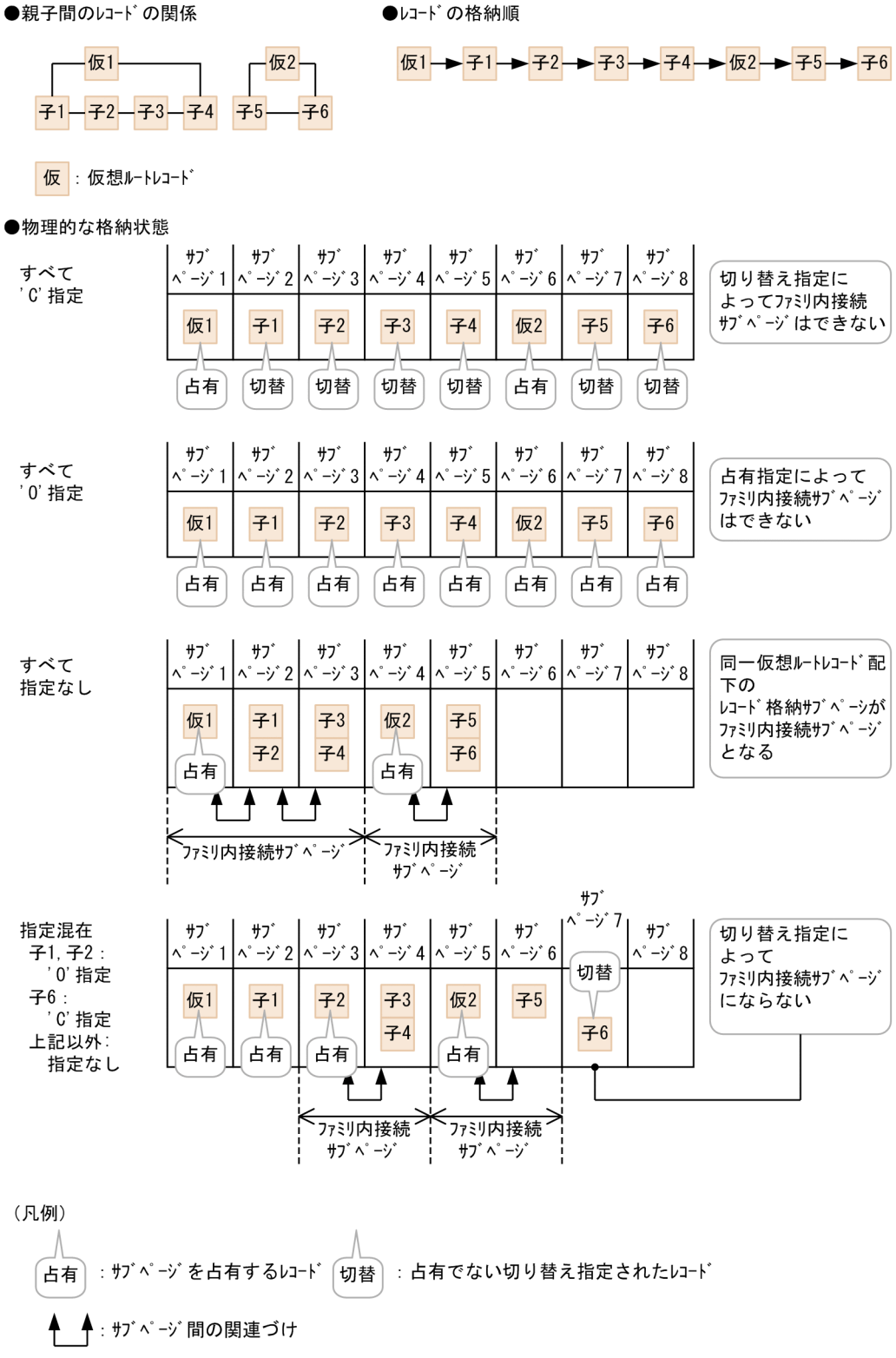
図 2-80　ページ切り替えオプションによるレコードの格納例（4V FMB の SDB データベースの場合）



OCCURRENCE NUMBER 句に 1 以上を指定している 4V AFM の SDB データベースの場合

SDB データベース格納定義の OCCURRENCE NUMBER 句に 1 以上を指定している 4V AFM の SDB データベースに対するレコードの格納時の，ページ切り替えオプションによるレコードの格納例を次の図に示します。

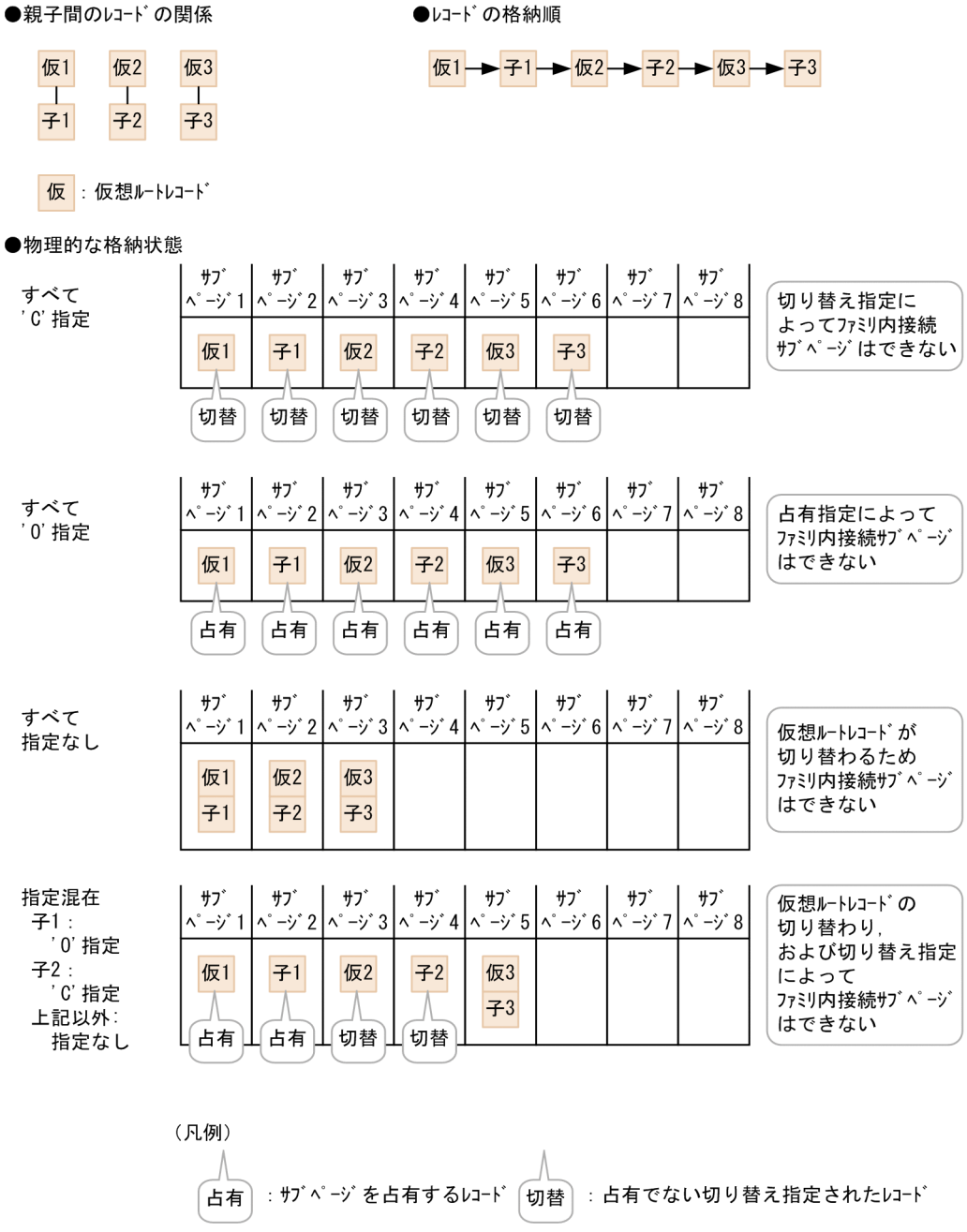
図 2-81　ページ切り替えオプションによるレコードの格納例（OCCURRENCE NUMBER 句に 1 以上を指定している 4V AFM の SDB データベースの場合）



OCCURRENCE NUMBER 句に 0 を指定している 4V AFM の SDB データベースの場合

SDB データベース格納定義の OCCURRENCE NUMBER 句に 0 を指定している 4V AFM の SDB データベースに対するレコードの格納時の，ページ切り替えオプションによるレコードの格納例を次の図に示します。

図 2-82　ページ切り替えオプションによるレコードの格納例（OCCURRENCE NUMBER 句に 0 を指定している 4V AFM の SDB データベースの場合）



(d)　適用基準

レコードに対するアクセス時の排他をサブページ単位で分散させたい場合，'O'指定または'C'指定を選択します。同時実行性は次の順に高くなります。

1. 'O'指定（占有指定）（同時実行性がいちばん高い）
2. 'C'指定（切り替え指定）
3. ページ切り替え指定なし

ただし，次の場合は指定による同時実行性の差異はありません。

- 4V FMB の SDB データベースの場合（ルートレコードの格納サブページの排他によって，ルートレコード下のレコード群のアクセスがシリアライズされるため）
- 4V AFM の SDB データベースで，SDB データベース格納定義の OCCURRENCE NUMBER 句の指定が 0 の場合（仮想ルートレコード下に子レコードが最大 1 件しか存在しないため）

なお，サブページ切り替えをすることによってデータベース容量が増大することがあるので注意してください。一般的に次の順でデータベース容量は増加します。

1. ページ切り替え指定なし（データベース容量がいちばん少ない）
2. 'C'指定（切り替え指定）
3. 'O'指定（占有指定）

SDB データベース種別ごとに推奨する指定を次の表に示します。作成するデータベースや格納するレコードの特性に合わせて指定を決定してください。

表 2-30　SDB データベース種別ごとに推奨する指定

SDB データベース種別		サブページ切り替えで推奨する指定※1	備考
4V FMB		指定なし	ルートレコードの格納サブページの排他によって，ルートレコード下のレコードのアクセスがシリアライズされるため，'O'指定または'C'指定に同時実行性を向上させる効果はありません。
4V DAM	OCCURRENCE NUMBER 句の指定が 0	指定なし	仮想ルートレコード下に子レコードが最大 1 件しか存在しないため，'O'指定または'C'指定に同時実行性を向上させる効果はありません。
	OCCURRENCE NUMBER 句の指定が 1 以上	指定なし または'O'指定（占有指定）※2	指定なしの場合，サブページに格納するレコード件数を制限し，排他の影響範囲を局所化したいときには，SDB データベース格納定義の PCTFREE 句を指定します。 サブページに 1 件だけレコードを格納する場合には'O'指定とします。
4V MAM		'O'指定（占有指定）※2	4V MAM の SDB データベースは一般的に同時実行性が求められます。
4V TAM		'O'指定（占有指定）※2	4V TAM の SDB データベースは一般的に同時実行性が求められます。

SDB データベース種別	サブページ切り替えで推奨する指定※1	備考
4V SAM	指定なし	4V SAM の SDB データベースは一般的に同時実行性が求められません。

注※1

データベース単位で，すべてのレコードに一律同じ指定をする場合の推奨値です。格納するレコードの特性やレコードの件数などがわかっている場合は，レコード単位に指定を切り替えることで，格納配置をより詳細に制御できます。レコード単位の指定の切り替えによるレコードの配置例については，「(e)　ページ切り替え指定によるレコードの配置例」を参照してください。

注※2

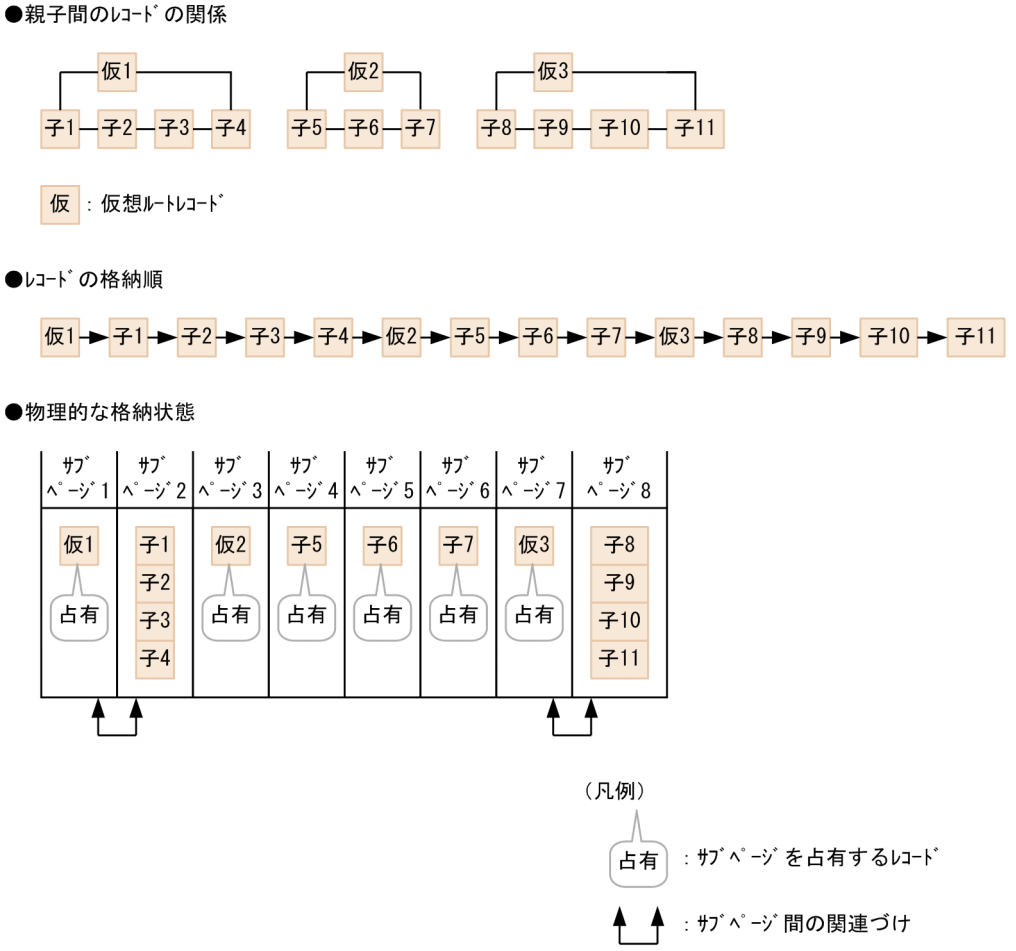
参照しかしないデータベースの場合や，ほかのトランザクションが存在しない状態で，更新処理を単独で実行するような場合には，サブページ排他での排他待ちが発生しないため，「指定なし」を推奨します。この場合，サブページに格納するレコード件数を制限し，排他の影響範囲を局所化する目的で PCTFREE 句を指定する必要はありません。

(e)　ページ切り替え指定によるレコードの配置例

アクセス頻度によってページ切り替え指定を使い分ける

4V DAM の SDB データベースなどで，アクセス頻度が高く同時実行性が要求されるレコードに対してだけ'O'指定をするというように指定を使い分けると，同時実行性を確保しながらデータベース容量を小さく抑えるといった，データベース設計ができます。アクセス頻度によってページ切り替え指定を使い分ける例を次の図に示します。この例では仮想ルートレコード 2 下のレコードだけ'O'指定をしています。

図 2-83　アクセス頻度によってページ切り替え指定を使い分ける例

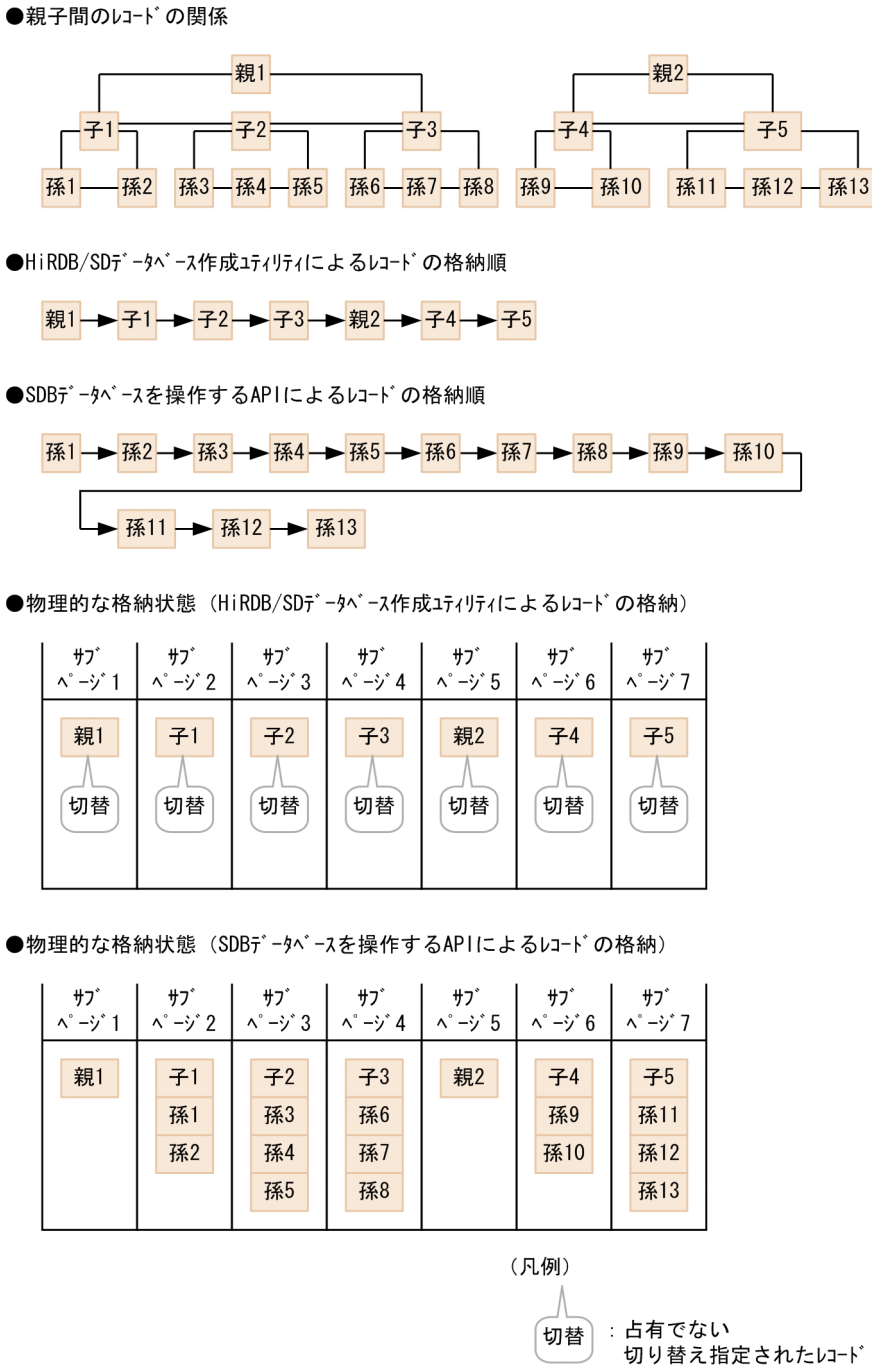


論理的に関連するレコードの格納位置をまとめる

4V FMB の SDB データベースで，データベース初期作成後のレコードの格納に備えて'C'指定をしておくことで，論理的に関連するレコードを同一サブページに格納するといった，データベース設計ができます。

論理的に関連するレコードの格納位置をまとめる例を次の図に示します。この例では孫レコードを格納するための領域をページ切り替え指定によって確保しておくことで，同一子レコード下の孫レコードを同一サブページに格納しています。

図 2-84　論理的に関連するレコードの格納位置をまとめる例



(4)　事前ページ割り当て機能【4V FMB】

サブページ分割をする 4V FMB の SDB データベースのファミリに，事前ページ割り当て機能を適用すると，サブページが事前割り当てサブページとして確保されます。

事前ページ割り当て機能は，4V FMB の SDB データベースの場合に使用できます。

(a)　事前ページ割り当て機能の概要

HiRDB/SD データベース作成ユティリティ（pdsdblod），または SDB データベースを操作する API で 4V FMB の SDB データベースにルートレコードを格納する場合，事前割り当てサブページ数を指定すると，指定したファミリにあらかじめ指定数分の連続したデータページ（事前割り当てサブページ）が確保されます。

HiRDB/SD データベース作成ユティリティ（pdsdblod）での指定方法については，次の説明を参照してください。

- 「12.3.2　load 文」の prefix オペランドの説明
- 「12.3.3　dbinf 文【4V FMB，4V AFM】」の pagenum オペランドの説明
- 「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」の事前割り当てページ数の説明

SDB データベースを操作する API によるレコードの格納（STORE）での，事前割り当てサブページ数の指定方法については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

SDB データベースを操作する API の実行時には次の制御を行うため，2 階層以上のレコードの格納およびレコードの削除を繰り返し実行した場合でも，ファミリ単位に連続したデータページ内へのレコードの配置を維持できます。

- 2 階層以上のレコードの格納時に，事前割り当てサブページ内にレコードを格納できる空き領域がある場合，その空き領域にレコードが格納されます。
- 2 階層以上のレコードの削除によって，事前割り当てサブページ内のレコードがすべて削除された場合でも，事前割り当てサブページは解放されないで，2 階層以上のレコードの格納時に再利用されます。

この配置制御によって，特定のファミリに連続してアクセスする場合に I/O 回数の削減が見込まれるため，SDB データベースを操作する API を繰り返し実行しても，一定のアクセス性能を維持できます。

2 階層以上のレコードの格納時に，事前割り当てサブページ内にレコードを格納できる空き領域がない場合，追加で事前割り当てサブページが確保されます。このときに確保される事前割り当てサブページの数は，該当するレコードの格納時の事前割り当てサブページ数の指定に従います。事前割り当てサブページ不足が発生していない場合，または事前ページ割り当て機能を適用していない場合，その指定は無視されます。

なお，事前割り当てサブページ数に 0 を指定した場合は 1 を指定したときと同じく，格納対象のレコードを格納するサブページだけが確保されます。

(b)　事前割り当てサブページ数の引き継ぎ

確保済みの事前割り当てサブページの数は，次の表に示す機能で取得できます。

表 2-31　事前割り当てサブページ数を取得できる機能

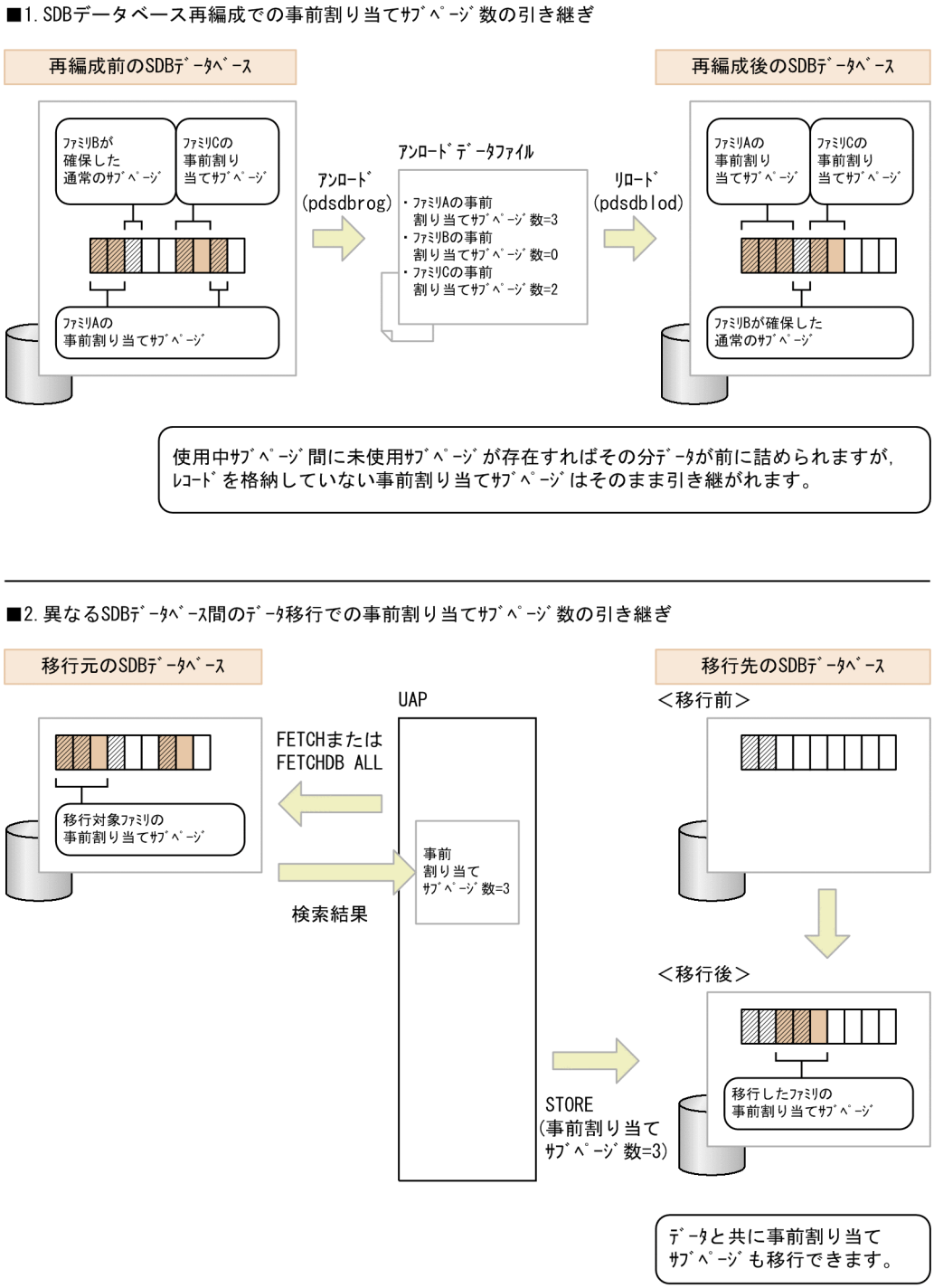
事前割り当てサブページ数を取得できる機能	機能概要
HiRDB/SD データベース再編成ユティリティ（pdsdbrog）	アンロード時，ファミリ単位に確保済みの事前割り当てサブページの数がアンロードデータファイルのプリフィクス部に設定されます。 事前割り当てページ数の取得方法については，次の説明を参照してください。 •「13.3.2　unload 文」の prefix オペランドの説明 •「13.3.3　dbinf 文【4V FMB】」の pagenum オペランドの説明 •「13.6.1　レコードのデータ形式」のプリフィクス部の説明
レコードの検索（FETCH）	ルートレコードの検索時に確保済みの事前割り当てサブページの数が返却されます。2 階層以上のレコードだけを検索した場合は 0 が返却されます。また，事前ページ割り当て機能を適用していないファミリを検索した場合にも 0 が返却されます。 SDB データベースを操作する API での事前割り当てページ数の取得方法については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。
複数レコードの検索（FETCHDB ALL）	

これによって，次の場合に事前割り当てサブページ数を引き継ぐことができます。

1. SDB データベースの再編成時
2. 異なる SDB データベース間のデータ移行時

事前割り当てサブページ数の引き継ぎ例を次の図に示します。

図 2-85　事前割り当てサブページ数の引き継ぎ例



(c)　事前ページ割り当て機能の適用条件

事前ページ割り当て機能は，次の機能と同時に適用できません。

- サブページ切り替え

- SDB データベース格納定義の PCTFREE オペランド，または HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の recfree オペランドで指定する空き領域の作成

(d) 事前ページ割り当て機能に関する留意事項

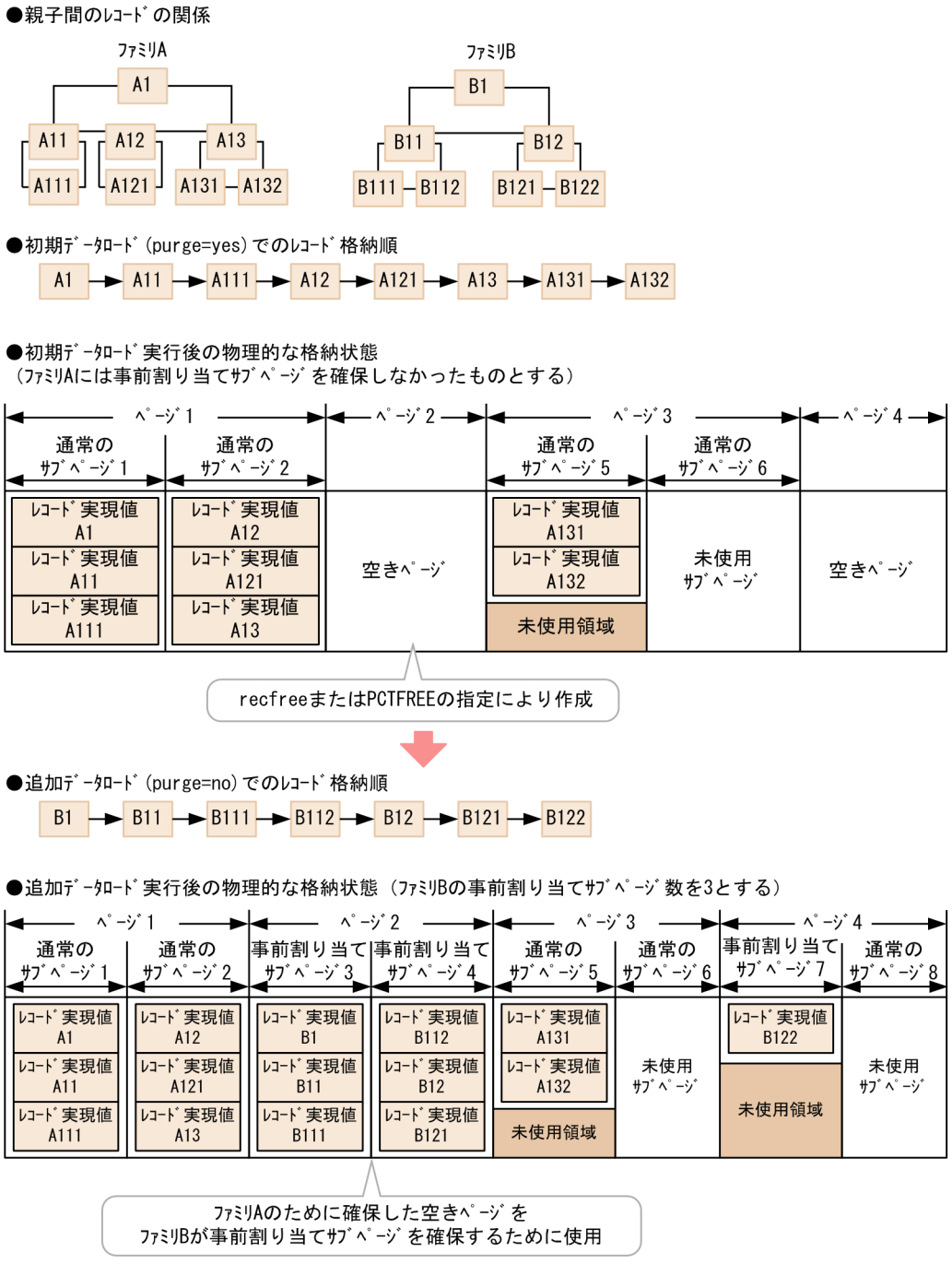
- 確保できる事前割り当てサブページ数の上限値は，1 ファミリ当たり 2,147,483,647 です。サブページ分割をしている場合，ファミリを格納するために必要な事前割り当てサブページ数がこの上限値を超える可能性があります。その場合はエラーになるため，次のどちらかの対応が必要です。
  - SDB データベース格納定義の STORAGE RECORD 句で指定している SUBPAGE NUMBER 句の指定値を小さくします。これによって，1 サブページ当たりに格納できるレコード件数を増やすことができます。その結果，該当するファミリを格納するために必要なサブページの数を上限値以下にできます。
  - SDB データベース格納定義の STORAGE RECORD 句で指定している SUBPAGE NUMBER 句の指定を省略することで，サブページ分割をしないようにします。
- HiRDB/SD データベース作成ユティリティ（pdsdblod）でレコードを格納する場合，次のように動作します。
  - 事前ページ割り当て機能を適用するファミリにサブページ切り替えの指定がある場合，論理エラーになります。
  - 事前割り当てサブページが属するページを確保する際，セグメント内の空きページの比率の指定は無視されるため，事前割り当てサブページが属するページ間に空きページは確保されません。通常のサブページが属するページを確保する際に空きページは確保されますが，同じセグメント内に事前割り当てサブページが属するページが含まれる場合，通常は指定したセグメント内の空きページの比率にはなりません。この場合のセグメント内の空きページの比率については，「3.2.6(4)(b) 未使用領域および空きページの作成に関する留意事項」を参照してください。
  - 事前割り当てサブページにレコードを格納する際，サブページ内の未使用領域の比率は無視されるため，レコードが詰めて格納されます。
- SDB データベースを操作する API でレコードを格納する場合，次のように動作します。
  - 事前ページ割り当て機能を適用しているファミリのレコードを格納する際に，サブページ切り替えの指定があるとエラーになります。
  - 事前ページ割り当て機能を適用しているファミリのレコードを格納する際に，PCTFREE 有効化オプションを適用するとエラーになります。
- データロード時にレコードを格納するために確保したサブページの数が，指定した事前割り当てサブページ数を超えた場合，該当するファミリを格納するために指定値を超えて確保したサブページも，事前割り当てサブページとして扱われます。
- 次のどれかの場合，事前ページ割り当て機能を適用しても非連続のデータページ，または非連続の事前割り当てサブページが確保されることがあります。
  1. 空きページが散在している SDB データベースに対して，追加データロードまたは SDB データベースを操作する API によるルートレコードの格納を実行し，事前割り当てサブページが確保される場合

2. SDB データベースを操作する API による 2 階層以上のレコードの格納時に，事前割り当てサブページ不足が発生し，追加で事前割り当てサブページが確保される場合

3. SDB データベースを操作する API によるレコードの格納時の事前割り当てサブページの確保中に，ほかのトランザクションによるサブページの確保が割り込む場合

追加データロード時に，非連続のデータページ内のサブページが事前割り当てサブページとして確保される例（1.の例）を次の図に示します。この図では，初期データロード時に確保した空きページを事前割り当てサブページが属するページとして確保する例を示しています。

図 2-86　追加データロード時の非連続データページ（事前割り当てサブページが属するページ）の確保例

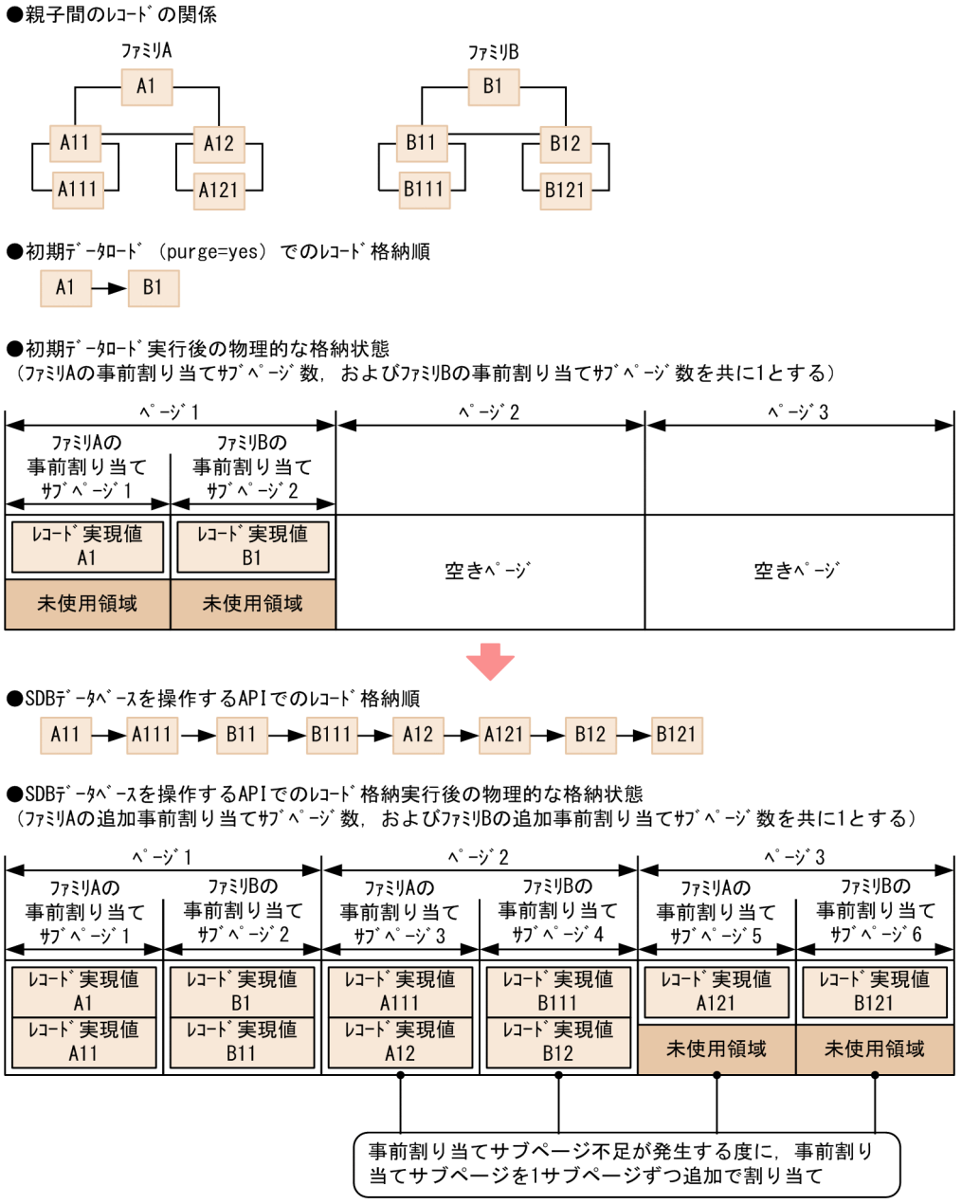


なお，SDB データベースを操作する API のレコードの削除によってページが解放され，未使用ページが存在する状態で追加データロードを実行した場合でも，そのページが再利用されることで，同様に非連続のデータページが確保されることがあります。ただし，次の場合，この状態は pdsdblod コマンドまたは SDB データベースを操作する API を実行する前の，使用中の最終セグメントでだけ発生します。

- 追加データロード時
- 新規にセグメントを確保できる状況での，SDB データベースを操作する API によるルートレコードの格納時

事前割り当てサブページ不足発生時に，非連続の事前割り当てサブページが確保される例（2.の例）を次の図に示します。

図 2-87　事前割り当てサブページ不足発生時の非連続の事前割り当てサブページの確保例



この図では，データページは連続していますが，1 サブページずつ追加しているため，本来は各ファミリが 2 ページずつに収まるところが，3 ページにまたがっています。このような状況でファミリ内のレコードすべてにアクセスする場合，I/O 回数が増加してしまいます。この問題は，一定量の事前割り当てサブページを一括して確保することで緩和できます。

非連続のデータページが事前割り当てサブページが属するページとして確保されているかどうかは，データベース状態解析ユティリティ（pddbst）で確認できます。

この問題を解消するには，SDB データベースを再編成する必要があります。

- 確保する事前割り当てサブページの数に比例して，レコードの格納性能は低下します。そのため，事前割り当てサブページの連続性とレコードの格納性能を考慮して，指定する事前割り当てサブページ数を決定する必要があります。ただし，ルートレコードを格納する際には，事前割り当てサブページの連続性を優先させ，ファミリ内のすべてのレコードを格納できるだけの事前割り当てサブページを一括して確保することを推奨します。
- 事前割り当てサブページを確保する際，基本的には使用中の最終セグメント，または新規に確保したセグメント内の空きページが使用されます。これらのセグメントから空きページを確保できなくなると，それ以外の使用中のセグメントから空きページが確保されます。このとき，次のようなケースでは，レコードの格納性能が大きく低下します。
  - 空きページが散在している。
  - 空きページを持たないセグメントが多数存在する。

  空きページのサーチ例については，「図 2-70　空きページのサーチ例」を参照してください。

  このような場合，次のどちらかの対応が必要です。
  - HiRDB/SD データベース再編成ユティリティ（pdsdbrog）と HiRDB/SD データベース作成ユティリティ（pdsdblod）で，SDB データベースを再編成します。
  - データベース構成変更ユティリティ（pdmod）で，レコード格納用 RD エリアを拡張します。

  「図 2-70　空きページのサーチ例」に示した空きページのサーチが動作して，レコードの格納性能が低下する前に，上記の対応をすることを推奨します。そのためには，システム共通定義の pd_rdarea_warning_point オペランドを指定します。これによって，次のメッセージが出力されるため，RD エリアの使用状況を監視できます。

  ・RD エリアのセグメント使用率通知メッセージ（KFPH00211-I または KFPA12300-I）

  ・増分する HiRDB ファイルの領域使用率通知メッセージ（KFPH22037-W）

  詳細については，マニュアル「HiRDB システム定義」の pd_rdarea_warning_point オペランドの説明を参照してください。
- SDB データベースを再編成するときに，事前割り当てサブページ数を変更したい場合は，事前割り当てサブページ数を指定し直す必要があります。
- 更新可能なオンライン再編成時に，事前ページ割り当て機能の適用有無を変更した場合，ルートレコード配下のレコードの格納（STORE）に対する更新追い付き反映処理は次のように動作します。

  適用ありから適用なしに変更した場合

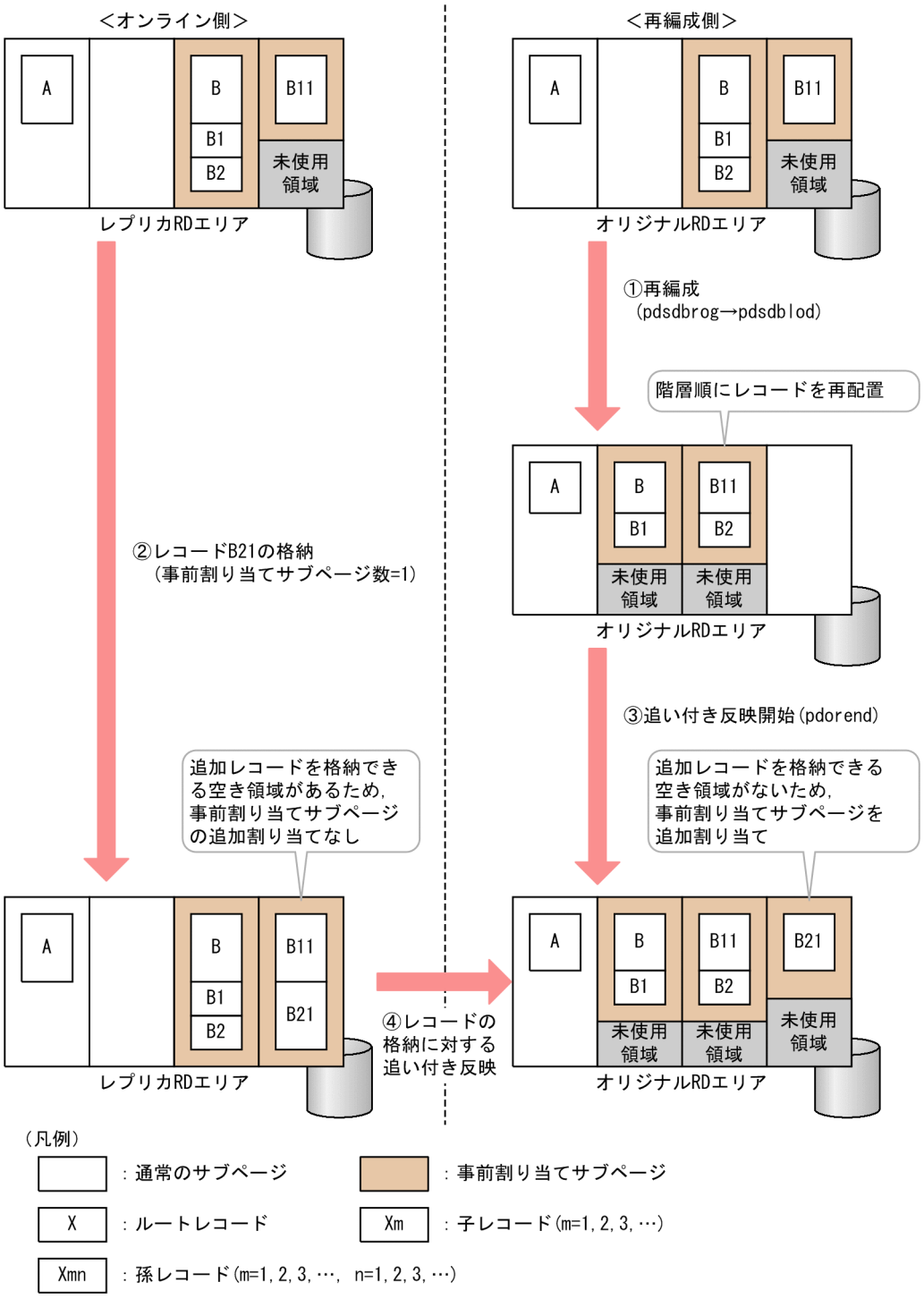
  事前割り当てサブページ数の指定は無視し，0 が指定されたものとして動作します。

適用なしから適用ありに変更した場合

- 事前割り当てサブページ不足発生時，事前割り当てサブページ数の指定値として 0 が指定されたものとして動作するため，1 サブページだけ確保されます。
- 次のオプションの指定は無視され，X'00'が指定されたものとして動作します。
  ・ページ切り替えオプション
  ・PCTFREE 有効化オプション

- 更新可能なオンライン再編成時の追い付き反映処理では，SDB データベースを操作する API によるレコードの格納時に指定した事前割り当てサブページ数の指定値を引き継ぎます。オンライン（レプリカ RD エリア）側と再編成（オリジナル RD エリア）側では，サブページの使用状況が異なるため，追い付き反映処理後に確保済みの事前割り当てサブページの数が一致しないことがあります。更新可能なオンライン再編成の追い付き反映処理での事前割り当てサブページ数の引き継ぎ例を次の図に示します。

図 2-88　更新可能なオンライン再編成の追い付き反映処理での事前割り当てサブページ数の引き継ぎ例



なお，上記と同様にサブページの使用状況が変わるため，SDB データベースの再編成によって，確保済みの事前割り当てサブページの数が増加することがあります。

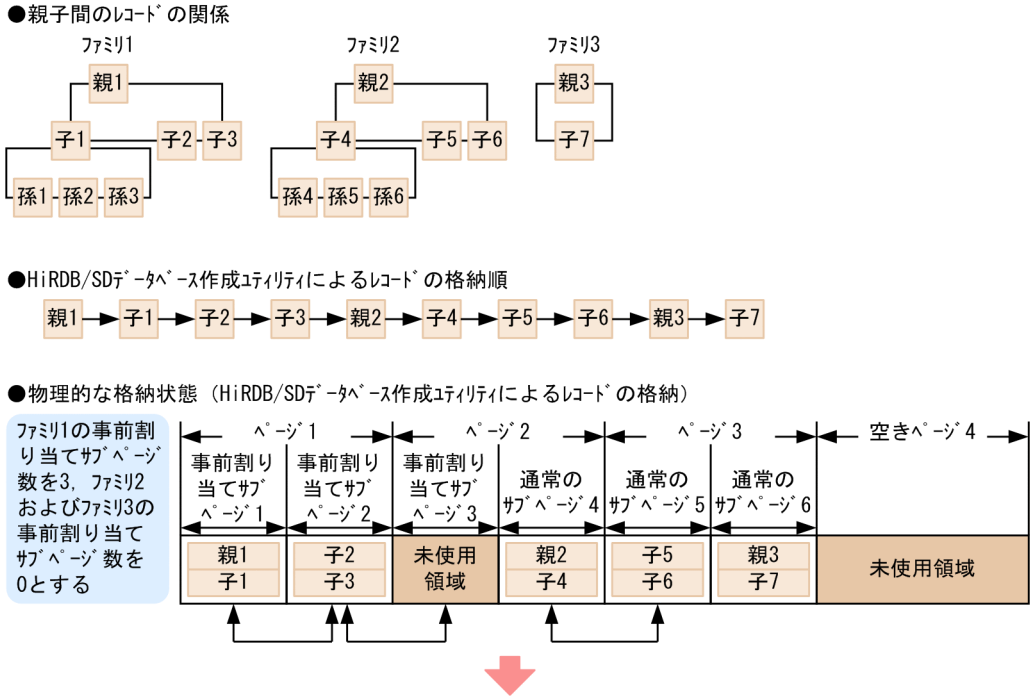
- 「図 2-88　更新可能なオンライン再編成の追い付き反映処理での事前割り当てサブページ数の引き継ぎ例」に示したように，SDB データベースを操作する API によるレコードの格納に対する追い付き反映処理によって，事前割り当てサブページが追加で確保された場合，オンライン（レプリカ RD エリア）側のトランザクションがロールバックしても，その追加された事前割り当てサブページは解放されませ

ん。なお，SDB データベースを操作する API によるルートレコードの格納時に確保された事前割り当てサブページの場合は解放されます。

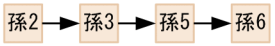
(e)　事前ページ割り当て機能を適用したファミリの格納例

データロード時に事前ページ割り当て機能を適用したファミリと適用していないファミリの格納例を次の図に示します。

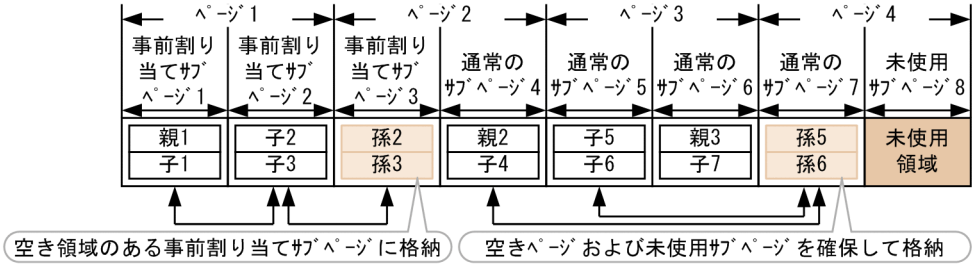
図 2-89　事前ページ割り当て機能の適用有無の違いによるファミリの格納例（サブページ分割をする場合）



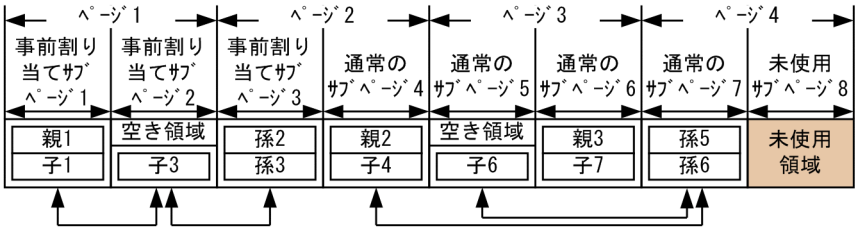
●SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの格納順



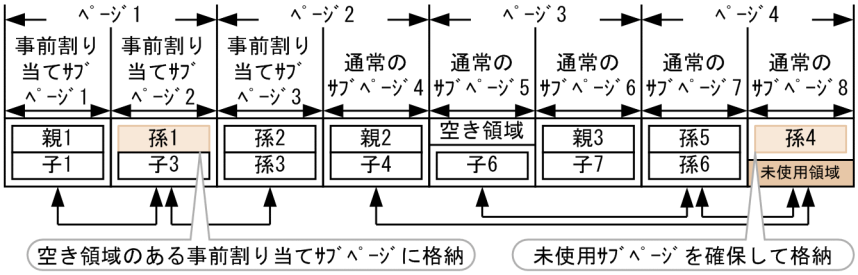
●物理的な格納状態（SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの格納）



●物理的な格納状態（SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの削除（子2，子5））



●物理的な格納状態（SDBﾃﾞｰﾀﾍﾞｰｽを操作するAPIによるﾚｺｰﾄﾞの格納（孫1，孫4））



（凡例）

：ｻﾌﾞﾍﾟｰｼﾞ間の関連づけ

このように，事前ページ割り当て機能を適用しているファミリの場合は，事前割り当てサブページ内にレコードを格納できる空き領域があるときは，その空き領域を再利用してレコードが格納されます。事前ページ割り当て機能を適用していないファミリの場合は，確保済みのサブページ内にレコードを格納できる空き領域があっても，その空き領域を再利用しないで，未使用サブページを確保してレコードが格納されることがあります。

なお，事前割り当てサブページ不足が発生し，追加で事前割り当てサブページが確保される場合の格納例については，「図 2-87　事前割り当てサブページ不足発生時の非連続の事前割り当てサブページの確保例」を参照してください。

(f)　適用基準

次のような場合に，事前ページ割り当て機能の適用を検討してください。

- 2 階層以上のレコードの格納およびレコードの削除を繰り返し実行するファミリへの連続したアクセスがあり，このとき一定のアクセス性能を維持したい場合（サブページ切り替え指定や，PCTFREE オペランドまたは recfree オペランド指定で厳密な空き領域設計ができる場合を除く）
- ほかの平均的なサイズのファミリと比べて，サイズが極端に大きいファミリであり，個別に格納領域の設計を行う必要がある場合
- サブページ切り替えを行うための厳密な見積もりが困難である場合

(g) 警告情報の出力

HiRDB/SD データベース作成ユティリティ（pdsdblod）での警告情報の出力

データロードによるレコードの格納時，レコードを格納するために確保したサブページの数が，指定した事前割り当てサブページ数を超えて事前ページを確保した場合，ファミリ単位に KFPB63518-I メッセージが出力されます。KFPB63518-I メッセージについては，「12.7　実行結果ファイルの出力形式」を参照してください。

HiRDB/SD データベース再編成ユティリティ（pdsdbrog）での警告情報の出力

アンロード時，事前ページ割り当て機能が適用されているファミリが存在するのに，アンロードデータファイル内にプリフィクス部が存在しないため，事前割り当てサブページ数が取得されなかった場合，KFPB63516-W メッセージが出力されます（事前割り当てサブページ数の引き継ぎが行われなかったことが通知されます）。

SDB データベースを操作する API によるレコードの格納（STORE）での警告情報の出力

SDB データベースを操作する API によるレコードの格納時に，事前割り当てサブページが不足して事前割り当てサブページが追加で確保された場合，レコード格納用 RD エリア単位に KFPB32000-W～KFPB32003-W メッセージが出力されます（事前割り当てサブページ数の再見積もりの要否確認が必要であることが通知されます）。

なお，このメッセージは 1 回出力されると，次回以降の事前割り当てサブページの不足時に，事前割り当てサブページを追加で確保してもメッセージが出力されません。この場合，次のどれかの操作が行われるまでメッセージが出力されません。メッセージの出力抑止状態の解除条件，および出力対象となる RD エリアを次に示します。

出力抑止状態の解除条件

- クローズ状態の該当するレコード格納用 RD エリアを pdopen コマンド，または pdrels -o コマンドでオープンしたとき
- 該当するレコード格納用 RD エリアに対して初期データロードを実行したとき
- HiRDB を再起動したとき

出力対象となる RD エリア

- インナレプリカ機能を使用していないレコード格納用 RD エリアの場合，出力対象となります。
- レコード格納用 RD エリアにインナレプリカ機能を使用している場合，オリジナル RD エリアが出力対象となります。

(5)　更新可能なオンライン再編成の追い付き反映処理でレコードをサブページに格納する場合【4V FMB】

更新可能なオンライン再編成の追い付き反映処理時に有効となる格納配置制御のオプションを次の表に示します。

表 2-32　追い付き反映処理時に有効となる格納配置制御のオプション

追い付き反映処理の種類	格納配置制御のオプション		
	ページ切り替えオプション	PCTFREE 有効化オプション	事前割り当てページ数
レコードの格納（STORE）の更新追い付き反映処理	○	○	○
レコードの削除（ERASE）の回復追い付き反映処理	×	×	−

（凡例）

○：オンライン（レプリカ RD エリア）側で，レコードの格納（STORE）時に指定した格納配置制御のオプションの指定を適用します。SDB データベースの再編成時に，事前ページ割り当て機能の適用有無を変更した場合の動作については，「(4)(d)　事前ページ割り当て機能に関する留意事項」を参照してください。

×：格納配置制御のオプションの指定を適用しません。

- サブページを切り替えません。
- SDB データベース格納定義の PCTFREE 句で指定したサブページ内の未使用領域の比率を適用しないで，レコード実現値を格納します。

−：該当しません。

(a)　レコードの格納（STORE）の更新追い付き反映処理時の格納配置制御

レコードの格納（STORE）の更新追い付き反映処理時の格納配置制御では，レコードを事前割り当てサブページに格納する場合を除き，オンライン（レプリカ RD エリア）側で指定した格納配置制御のオプションの指定が適用されます。

ページ切り替えオプションの指定の違いによる格納配置制御については，「図 2-80　ページ切り替えオプションによるレコードの格納例（4V FMB の SDB データベースの場合）」を参照してください。

PCTFREE 有効化オプションを適用した場合の格納配置制御については，「図 3-17　サブページに格納するレコード件数を制限する例」を参照してください。

事前ページ割り当て機能の適用有無の違いによる格納配置制御については，「図 2-89　事前ページ割り当て機能の適用有無の違いによるファミリの格納例（サブページ分割をする場合）」を参照してください。なお，この図では，HiRDB/SD データベース作成ユティリティ（pdsdblod）を実行する場合に，事前割り当てサブページ数を指定したときのファミリの格納例を示しています。SDB データベースを操作する API

でレコードの格納を実行する場合に，事前割り当てサブページ数を指定したときも同様の格納配置となります。

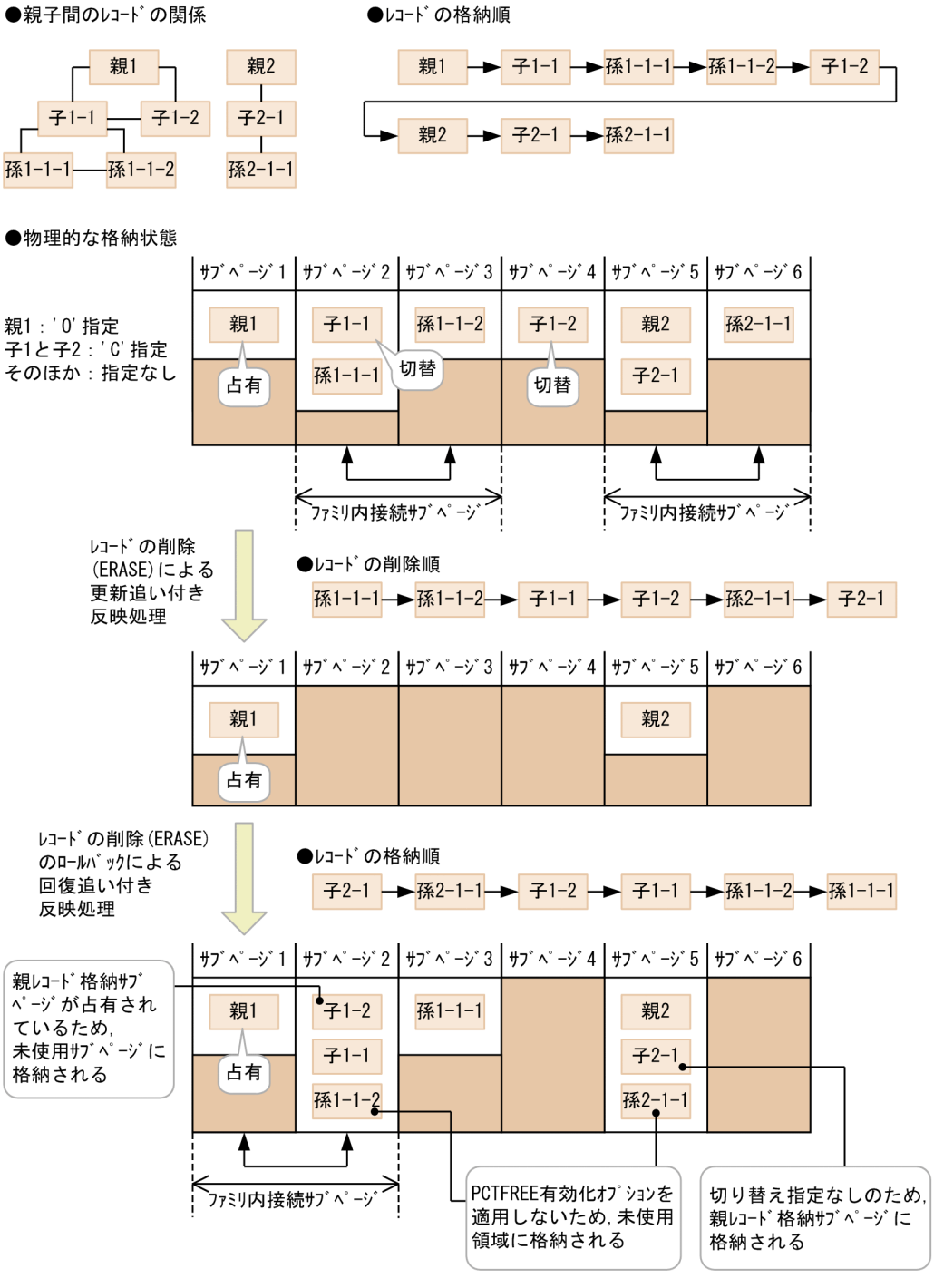
ルートレコード下のレコードの格納時に，事前割り当てサブページが不足した場合の格納配置制御については，「図 2-87　事前割り当てサブページ不足発生時の非連続の事前割り当てサブページの確保例」を参照してください。

(b)　レコードの削除（ERASE）の回復追い付き反映処理時の格納配置制御

レコードの削除（ERASE）の回復追い付き反映処理時の格納配置制御では，オンライン（レプリカ RD エリア）側で指定した格納配置制御のオプションの指定が適用されません。

ページ切り替えオプションを指定し，PCTFREE 有効化オプションを適用したときの回復追い付き反映処理時の格納配置制御を次の図に示します。

図 2-90　レコードの削除（ERASE）の回復追い付き反映処理時の格納配置制御



2.8　グローバルバッファ

グローバルバッファとは，ディスク上の RD エリアに格納されているデータを入出力するためのバッファのことで，共用メモリ上に確保されます。HiRDB/SD はデータベースの入出力処理にグローバルバッファを使用しています。

データを更新するためにバッファ上で更新され，データベースには未反映のバッファを更新バッファといいます。また，データを参照するためのバッファ，およびデータベースに反映済みのバッファを参照バッファといいます。

次に示す RD エリアには，必ずグローバルバッファを割り当ててください。

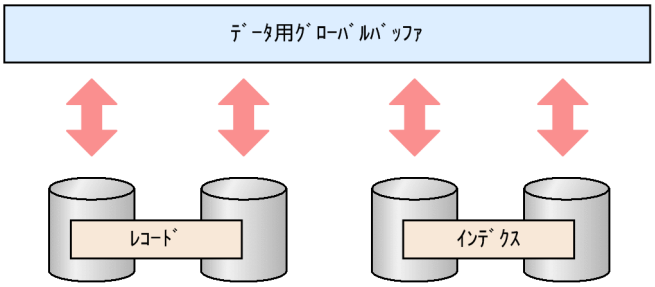
- レコード型，またはレコード型のインデクスを格納するユーザ用 RD エリア
- システム用 RD エリア
- 追い付き反映キー対応表を格納するユーザ用 RD エリア
- 追い付き反映キー対応表のインデクスを格納するユーザ用 RD エリア
- 追い付き状態管理表を格納するユーザ用 RD エリア
- 監査証跡表を格納するユーザ用 RD エリア

データ用グローバルバッファ

データの入出力に使用されるグローバルバッファです。データ用グローバルバッファは RD エリア単位に割り当てます。

グローバルバッファの概念を次の図に示します。

図 2-91　グローバルバッファの概念



なお，HiRDB の再起動を必要としない SDB データベースの定義追加や変更で RD エリアを追加した場合，データベース構成変更ユティリティ（pdmod）で割り当てたグローバルバッファは一時的なものです。HiRDB を正常停止，または計画停止したときに，システム共通定義の pdbuffer オペランドでグローバルバッファの割り当てを行う必要があります。

2.9　排他制御

HiRDB/SD の排他制御について説明します。

HiRDB/SD では，複数のユーザ（トランザクション）が SDB データベースに同時にアクセスしても，SDB データベースの整合性を保つように排他制御を行っています。

2.9.1　排他制御の単位

SDB データベースの排他制御の単位について説明します。

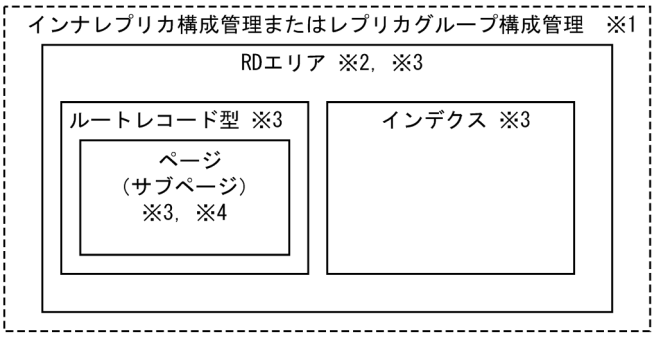
排他制御では，排他資源という単位で排他を掛けます。SDB データベースでは，RD エリアやルートレコード型などが排他資源に該当します。

排他制御では排他資源の上位から下位へと順番に排他が掛けられます。また，排他資源には包含関係があります。

例えば，RD エリアとルートレコード型の関係では，RD エリアが上位，ルートレコード型が下位の排他資源となります。あるトランザクションが，上位の排他資源である RD エリアに対して，排他資源を占有するモードで排他を掛けると，その RD エリアに含まれる下位の排他資源（ルートレコード型，インデクス，ページ，サブページ）は，ほかのトランザクションで同時に使用できなくなります。排他制御のモードについては，「2.9.2　排他制御のモード」を参照してください。

SDB データベースの排他資源とその包含関係を次の図に示します。

図 2-92　SDB データベースの排他資源とその包含関係



注

上記の図に示した排他資源以外に，セグメントに対して排他制御を行う場合があります。例えば，SDB データベースの削除を行う場合は，セグメント単位に排他制御を行います。

また，インデクスの一意性を保証するため，行に対して排他制御を行います。HiRDB/SD での排他資源の行とは，レコード実現値のことです。

注※1

インナレプリカ機能使用時の最上位排他資源は，インナレプリカ構成管理情報，またはレプリカグループ構成管理情報となります。

インナレプリカ構成管理情報の排他を取得できない場合に，レプリカグループ構成管理情報の排他を取得します。レプリカ RD エリアを定義していない RD エリアにアクセスする場合でも，排他が掛かります。これによって，業務実行中に通常 RD エリアにレプリカ RD エリアが定義されたり，インナレプリカグループ内の構成が変わったりすることを抑止します。

注※2

RD エリアは，ルートレコード型，インデクス，ページ，サブページの上位資源となります。

注※3

インナレプリカ機能を使用する場合，RD エリア，ルートレコード型，インデクス，ページ，サブページは，世代ごとの排他になります。

注※4

サブページ分割をする場合（SDB データベース格納定義の STORAGE RECORD 句下に SUBPAGE NUMBER 句を指定している場合），排他制御の単位はページではなく，サブページになります。

なお，サブページ分割をする場合は，「2.9.2　排他制御のモード」～「2.9.10　コミットしていない削除データの排他制御」を読まれる際，レコードを格納する RD エリアのページ（データページ）をサブページに読み替えてください。

2.9.2　排他制御のモード

排他制御では，ルートレコード型，ページなどのそれぞれの排他資源に対して，次に示す 5 種類の排他制御モードがあります。

1. **共用モード（PR モード：**Protected Retrieve）

   排他資源の参照だけが許されるモードです。ほかのトランザクションによる参照を許します。

2. **排他モード（EX モード：**Exclusive）

   1 つのトランザクションだけが排他資源を占有するモードです。ほかのトランザクションによる排他資源の参照，追加，更新および削除を許しません。

3. **意図共用モード（SR モード：**Shared Retrieve）

   排他資源の参照だけが許されるモードです。ほかのトランザクションによる排他資源の参照，追加，更新および削除を許します（ただし，排他資源がページの場合は参照だけを許します）。

4. **意図排他モード（SU モード：**Shared Update）

   排他資源の参照，追加，更新および削除が許されるモードです。ほかのトランザクションによる排他資源の参照，追加，更新および削除を許します。

5. **共用意図排他モード（PU モード：**Protected Update）

排他資源の参照，および更新が許されるモードです。ほかのトランザクションによる排他資源の参照だけを許します。

同一資源に対して 2 つのトランザクションが同時に排他を掛ける場合，排他制御モードによって，トランザクションを同時に実行できたりできなかったりします。排他制御モードとトランザクションの同時実行性を次の表に示します。

表 2-33　排他制御モードとトランザクションの同時実行性

排他制御モード	SR	PR	SU	PU	EX
SR	○	○	○	○	×
PR	○	○	×	×	×
SU	○	×	○	×	×
PU	○	×	×	×	×
EX	×	×	×	×	×

（凡例）

○：同時に実行できる。

×：同時に実行できない。

2 つのトランザクションが同時に実行できない場合，一般的に，一方のトランザクションがもう一方のトランザクションの終了待ちとなります。

(1)　排他制御モードの遷移

同一トランザクションが同一資源に対して，異なる種類の排他制御モードで二重，三重に排他を掛けていくと，排他制御モードは強い方へと遷移していきます。例えば，一度，EX モードで排他を掛けると，後から PR モードで排他を掛けても，その資源は EX モードのままであり，PR モードにはなりません。

現在の排他制御モードに対して，さらに排他制御を掛けた場合の排他制御モードの遷移規則を次の表に示します。

表 2-34　排他制御モードの遷移規則

後から掛ける排他制御モード	現在の排他制御モード				
	SR	PR	SU	EX	PU
SR	－	－	－	－	－
PR	PR	－	PU※	－	－
SU	SU	PU※	－	－	－
EX	EX	EX	EX	－	EX
PU	PU	PU	PU	－	－

（凡例）

－：排他制御モードは遷移しません。

－以外：遷移後の排他制御モードを表します。

注※

排他制御モード遷移の結果として，PU モードになることがあります。このモードは SR モードとだけ同時実行できます。

詳細については，マニュアル「HiRDB UAP 開発ガイド」の「モードの遷移」も参照してください。

(2) 排他制御モードの組み合わせ

排他制御は，実行する API または DML の種類，および実行条件などによって，モードの組み合わせが異なります。

実行する API または DML の種類，および実行環境の違いによる排他制御のモードの組み合わせの例を説明します。

参考

- SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。
- DML については，「17. DML リファレンス【SD FMB】」を参照してください。

表 2-35　排他制御のモードの組み合わせの例（4V FMB の SDB データベースの場合）

実行するAPIの種類	操作対象のﾚｺｰﾄﾞ	実行条件	資源												
			上位 ←－－－→ 下位												
			ｲﾝﾅﾚﾌﾟﾘｶ構成管理※1	ﾚﾌﾟﾘｶｸﾞﾙｰﾌﾟ構成管理※1	RDｴﾘｱ			ﾙｰﾄﾚｺｰﾄﾞ型※2	ページ					行	
					ﾚｺｰﾄﾞ格納用	ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽ格納用	二次ｲﾝﾃﾞｸｽ格納用		ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	親ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	PRIORﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	対象ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ※28※29	NEXTﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	一意性制約保証用※3	重複ｷｰ追加時の排他待ち用※35
ﾚｺｰﾄﾞの検索(FETCH)	ﾙｰﾄﾚｺｰﾄﾞ	排他ﾓｰﾄﾞ1が'S'※4	SR	SR	SR	SR	－	SR	SR	－	－	－	－	－	－
		排他ﾓｰﾄﾞ1が'E'※4 ｱｸｾｽﾓｰﾄﾞ1が'R'※5	SR	SR	SR	SR	－	SR	PU	－	－	－	－	－	－
		排他ﾓｰﾄﾞ1が'E'※4 ｱｸｾｽﾓｰﾄﾞ1が'U'※5	SR	SR	SU	SR	－	SU	PU	－	－	－	－	－	－
		排他ﾓｰﾄﾞ1が'O'※4 ｱｸｾｽﾓｰﾄﾞ1が'R'※5	SR	SR	SR	SR	－	SR	EX	－	－	－	－	－	－
		排他ﾓｰﾄﾞ1が'O'※4 ｱｸｾｽﾓｰﾄﾞ1が'U'※5	SR	SR	SU	SR	－	SU	EX	－	－	－	－	－	－
		排他ﾓｰﾄﾞ1が'N'※4	SR	SR	SR	SR	－	SR	－	－	－	－	－	－	－
	子ﾚｺｰﾄﾞ	排他ﾓｰﾄﾞ1が'S'※4	SR	SR	SR	－	SR※13	SR	－	－	－	SR	－	－	－
		排他ﾓｰﾄﾞ1が'E'※4 ｱｸｾｽﾓｰﾄﾞ1が'R'※5	SR	SR	SR	－	SR※13	SR	－	－	－	PU	－	－	－
		排他ﾓｰﾄﾞ1が'E'※4 ｱｸｾｽﾓｰﾄﾞ1が'U'※5	SR	SR	SU	－	SR※13	SU	－	－	－	PU	－	－	－
		排他ﾓｰﾄﾞ1が'O'※4 ｱｸｾｽﾓｰﾄﾞ1が'R'※5	SR	SR	SR	－	SR※13	SR	－	－	－	EX	－	－	－
		排他ﾓｰﾄﾞ1が'O'※4 ｱｸｾｽﾓｰﾄﾞ1が'U'※5	SR	SR	SU	－	SR※13	SU	－	－	－	EX	－	－	－
		排他ﾓｰﾄﾞ1が'N'※4	SR	SR	SR	－	SR※13	SR	－	－	－	－	－	－	－
複数ﾚｺｰﾄﾞの検索(FETCHDB ALL)	ﾙｰﾄﾚｺｰﾄﾞ※21	排他ﾓｰﾄﾞ1が'S'※4	SR	SR	SR	SR	－	SR	SR	－	－	－	－	－	－
		排他ﾓｰﾄﾞ1が'E'※4	SR	SR	SR	SR	－	SR	PU	－	－	－	－	－	－
		排他ﾓｰﾄﾞ1が'O'※4	SR	SR	SR	SR	－	SR	EX	－	－	－	－	－	－
	子ﾚｺｰﾄﾞ※22	なし	SR	SR	SR	－	－	SR	－	－	－	－	－	－	－
ﾚｺｰﾄﾞの更新(MODIFY)	ﾙｰﾄﾚｺｰﾄﾞ	なし	SR	SR	SU	－	－	SU	EX	－	－	－	－	－	－
	子ﾚｺｰﾄﾞ	ﾕｰｻﾞﾃﾞｰﾀ変更だけ	SR	SR	SU	－	－	SU	EX	－	－	EX	－	－	－
		USERﾎﾟｲﾝﾀ更新だけ(ﾎﾟｲﾝﾀｵﾌﾟｼｮﾝが'U'※6)	SR	SR	SU	－	－	SU	EX	PU※7	－	PU※7	－	－	－
		USERﾎﾟｲﾝﾀ更新だけ(ﾎﾟｲﾝﾀｵﾌﾟｼｮﾝが'C'※6)	SR	SR	SU	－	－	SU	EX	EX	－	－	－	－	－
		ﾕｰｻﾞﾃﾞｰﾀ変更およびUSERﾎﾟｲﾝﾀ更新(ﾎﾟｲﾝﾀｵﾌﾟｼｮﾝが'U'※6)	SR	SR	SU	－	－	SU	EX	PU※7※8	－	EX	－	－	－
		ﾕｰｻﾞﾃﾞｰﾀ変更およびUSERﾎﾟｲﾝﾀ更新(ﾎﾟｲﾝﾀｵﾌﾟｼｮﾝが'C'※6)	SR	SR	SU	－	－	SU	EX	EX	－	EX	－	－	－

（続く）

（続き）

実行するAPIの種類	操作対象のﾚｺｰﾄﾞ	実行条件	資源												
			上位 ←――→ 下位												
			ｲﾝﾅﾚﾌﾟﾘｶ構成管理※1	ﾚﾌﾟﾘｶｸﾞﾙｰﾌﾟ構成管理※1	RDｴﾘｱ			ﾙｰﾄﾚｺｰﾄﾞ型※2	ページ					行	
					ﾚｺｰﾄﾞ格納用	ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽ格納用	二次ｲﾝﾃﾞｸｽ格納用		ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	親ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	PRIORﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	対象ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ※28※29	NEXTﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	一意性制約保証用※3	重複ｷｰ追加時の排他待ち用※35
ﾚｺｰﾄﾞの格納(STORE)	ﾙｰﾄﾚｺｰﾄﾞ	なし	SR	SR	SU	SU	－	SU	EX	－	－	EX※29	－	EX※9	EX
	子ﾚｺｰﾄﾞ	なし	SR	SR	SU	－	SU※13	SU	EX	PU※7※8	PU※10	EX※29	PU※11	EX※13	－
ﾚｺｰﾄﾞの削除(ERASE)	ﾙｰﾄﾚｺｰﾄﾞ	一括ｵﾌﾟｼｮﾝが'S'※12	SR	SR	SU	SU	SU※13	SU	EX	－	－	EX※28	－	EX※27	－
	子ﾚｺｰﾄﾞ	一括ｵﾌﾟｼｮﾝが'S'※12	SR	SR	SU	－	SU※13	SU	EX	EX	EX	EX	EX	EX※13	－
		一括ｵﾌﾟｼｮﾝが'O'※12	SR	SR	SU	－	SU※13	SU	EX	EX	－	EX	－	EX※13	－

注

（凡例）と表中の※の説明は，「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」の後ろで説明しています。

表 2-36　排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合）

実行するAPIの種類	操作対象のﾚｺｰﾄﾞ	実行条件		資源												
				上位 ←－－－→ 下位												
				インナレプリカ構成管理※1	レプリカグループ構成管理※1	RDエリア			仮想ルートﾚｺｰﾄﾞ型※2	ページ					行	
						ﾚｺｰﾄﾞ格納用	シーケンシャルインデックス格納用	二次インデックス格納用		仮想ルートﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	親ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	PRIORﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	対象ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	NEXTﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	一意性制約保証用※3	重複ｷｰ追加時の排他待ち用※35
ﾚｺｰﾄﾞの検索（FETCH）	子ﾚｺｰﾄﾞ	排他ﾓｰﾄﾞ1が'S'※4		SR	SR	SR	SR	SR※13	SR	SR※26	－	－	SR※26	－	－	－
		排他ﾓｰﾄﾞ1が'E'※4	ｱｸｾｽﾓｰﾄﾞ1が'R'※5	SR	SR	SR	SR	SR※13	SR	SR※14	－	－	PU	－	－	－
			ｱｸｾｽﾓｰﾄﾞ1が'U'※5	SR	SR	SU	SR	SR※13	SU	SR※14	－	－	PU	－	－	－
		排他ﾓｰﾄﾞ1が'O'※4	ｱｸｾｽﾓｰﾄﾞ1が'R'※5	SR	SR	SR	SR	SR※13	SR	SR※14	－	－	EX	－	－	－
			ｱｸｾｽﾓｰﾄﾞ1が'U'※5	SR	SR	SU	SR	SR※13	SU	SR※14	－	－	EX	－	－	－
		排他ﾓｰﾄﾞ1が'N'※4		SR	SR	SR	SR	SR※13	SR	－	－	－	－	－	－	－
ﾚｺｰﾄﾞの更新（MODIFY）		なし		SR	SR	SU	－	－	SU	SR※20	－	－	EX	－	－	－
ﾚｺｰﾄﾞの格納（STORE）		1件目（仮想ﾙｰﾄﾚｺｰﾄﾞの追加を伴う子ﾚｺｰﾄﾞの追加）※15		SR	SR	SU	SU	SU※13	SU	EX	－	－	EX	－	EX※16	－
		2件目以降（仮想ﾙｰﾄﾚｺｰﾄﾞの追加を伴わない子ﾚｺｰﾄﾞの追加）		SR	SR	SU	SR※23	SU※13	SU	PU※20	－	PU※17	EX	PU※18	EX※13※24	－
		一括削除後のﾚｺｰﾄﾞ格納※19		SR	SR	SU	SR※23	SR※13	SU	PU※20	－	－	EX	－	－※24	－
ﾚｺｰﾄﾞの削除（ERASE）		一括ｵﾌﾟｼｮﾝが'S'※12		SR	SR	SU	－※23	SU※13	SU	EX	－	EX	EX	EX	EX※13※24	－
ﾚｺｰﾄﾞの一括削除	仮想ﾙｰﾄﾚｺｰﾄﾞ	なし		SR	SR	SU	SU※25	－	SU	EX	－	－	－	－	EX※24※25	－
ﾚｺｰﾄﾞの構成情報取得		なし		SR	SR	SR	SR	－	SR	－	－	－	－	－	－	－

注

（凡例）と表中の※の説明は，「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」の後ろで説明しています。

表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）

実行するDMLの種類	操作対象のﾚｺｰﾄﾞ	検索種別	実行条件					資源											
								上位←－－－－→下位											
			SD排他ﾓｰﾄﾞ ※30	ｱｸｾｽ目的 ※31	SDBﾃﾞｰﾀﾍﾞｰｽの階層数	ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞの共用排他指定 ※34	FOR UPDATE ｵﾌﾟｼｮﾝﾄﾞの指定	RDｴﾘｱ			ﾙｰﾄﾚｺｰﾄﾞ型 ※2	ﾙｰﾄﾚｺｰﾄﾞ型（NOWAIT検索時） ※36	ページ					行	
								ﾚｺｰﾄﾞ格納用	ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽ格納用	二次ｲﾝﾃﾞｸｽ格納用			ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	親ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	PRIORﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	対象ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	NEXTﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	一意性制約保証用 ※3	重複ｷｰ追加時の排他待ち用 ※35
FETCH, FIND	ﾙｰﾄﾚｺｰﾄﾞ	ﾚｺｰﾄﾞ型内の検索	shared	retrieve	－	－	なし	SR	SR ※38	SR ※38	SR	－	SR	－	－	－	－	－	－
				update	1	－	なし	SU	SR ※38	SR ※38	SU	－	SR	－	－	－	－	－	－
						－	あり	SU	SR ※38	SR ※38	SU	－	PU	－	－	－	－	－	－
					2以上	あり	なし	SU	SR ※38	SR ※38	SU	－	SR	－	－	－	－	－	－
							あり	SU	SR ※38	SR ※38	SU	－	PU	－	－	－	－	－	－
						なし	－	SU	SR ※38	SR ※38	SU	－	EX	－	－	－	－	－	－
			protected	retrieve	－	－	なし	PR	SR ※38	SR ※38	SR	－	－	－	－	－	－	－	－
				update	1	－	なし	PU	SR ※38	SR ※38	SU	－	SR	－	－	－	－	－	－
						－	あり	PU	SR ※38	SR ※38	SU	－	PU	－	－	－	－	－	－
					2以上	あり	なし	PU	SR ※38	SR ※38	SU	－	SR	－	－	－	－	－	－
							あり	PU	SR ※38	SR ※38	SU	－	PU	－	－	－	－	－	－
						なし	－	PU	SR ※38	SR ※38	SU	－	EX	－	－	－	－	－	－
			exclusive	retrieve	－	－	なし	EX	SR ※38	SR ※38	SR	－	－	－	－	－	－	－	－
				update	－	－	－	EX	SR ※38	SR ※38	SU	－	－	－	－	－	－	－	－
			nonprotected	retrieve	－	－	なし	SR	SR ※38	SR ※38	－	SR ※33	－	－	－	－	－	－	－
		位置指示子指定の検索	shared	retrieve	－	－	なし	SR	－	－	SR	－	SR	－	－	－	－	－	－
				update	－	あり	なし	SU	－	－	SU	－	SR	－	－	－	－	－	－
							あり	SU	－	－	SU	－	PU	－	－	－	－	－	－
						なし	－	SU	－	－	SU	－	EX	－	－	－	－	－	－
			protected	retrieve	－	－	なし	PR	－	－	SR	－	－	－	－	－	－	－	－
				update	－	あり	なし	PU	－	－	SU	－	SR	－	－	－	－	－	－
							あり	PU	－	－	SU	－	PU	－	－	－	－	－	－
						なし	－	PU	－	－	SU	－	EX	－	－	－	－	－	－
			exclusive	retrieve	－	－	なし	EX	－	－	SR	－	－	－	－	－	－	－	－
				update	－	－	－	EX	－	－	SU	－	－	－	－	－	－	－	－
			nonprotected	retrieve	－	－	なし	SR	－	－	－	SR ※33	－	－	－	－	－	－	－

（続く）

（続き）

実行するDMLの種類	操作対象のﾚｺｰﾄﾞ	検索種別	実行条件					資源											
								上位←－－－→下位											
			SD排他ﾓｰﾄﾞ ※30	ｱｸｾｽ目的 ※31	SDBﾃﾞｰﾀﾍﾞｰｽの階層数	ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞの共用排他指定 ※34	FOR UPDATEｵﾍﾟﾗﾝﾄﾞの指定	RDｴﾘｱ			ﾙｰﾄﾚｺｰﾄﾞ型 ※2	ﾙｰﾄﾚｺｰﾄﾞ型（NOWAIT検索時） ※36	ページ					行	
								ﾚｺｰﾄﾞ格納用	ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽ格納用	二次ｲﾝﾃﾞｸｽ格納用			ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	親ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	PRIORﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	対象ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	NEXTﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	一意性制約保証用 ※3	重複ｷｰ追加時の排他待ち用 ※35
FETCH, FIND	子ﾚｺｰﾄﾞ	親子集合型内の検索	shared	retrieve	－	－	なし	SR	－	－	SR	－	SR ※42	－	－	SR	－	－	－
				update	－	－	なし	SU	－	－	SU	－	SR ※42	－	－	SR	－	－	－
					－	－	あり	SU	－	－	SU	－	SR ※42	－	－	PU	－	－	－
			protected	retrieve	－	－	なし	PR	－	－	SR	－	－	－	－	－	－	－	－
				update	－	－	なし	PU	－	－	SU	－	SR ※42	－	－	SR	－	－	－
					－	－	あり	PU	－	－	SU	－	SR ※42	－	－	PU	－	－	－
			exclusive	retrieve	－	－	なし	EX	－	－	SR	－	－	－	－	－	－	－	－
				update	－	－	－	EX	－	－	SU	－	－	－	－	－	－	－	－
			nonprotected	retrieve	－	－	なし	SR	－	－	－	SR ※33	－	－	－	－	－	－	－
		ﾚｺｰﾄﾞ型内の検索	shared	retrieve	－	－	－	SR	－	SR	SR	－	SR	－	－	SR	－	－	－
				update	－	あり	なし	SU	－	SR	SU	－	SR	－	－	SR	－	－	－
					－	あり	あり	SU	－	SR	SU	－	PU	－	－	PU	－	－	－
					－	なし	なし	SU	－	SR	SU	－	EX	－	－	SR	－	－	－
					－	なし	あり	SU	－	SR	SU	－	EX	－	－	PU	－	－	－
			protected	retrieve	－	－	－	PR	－	SR	SR	－	－	－	－	－	－	－	－
				update	－	あり	なし	PU	－	SR	SU	－	SR	－	－	SR	－	－	－
					－	あり	あり	PU	－	SR	SU	－	PU	－	－	PU	－	－	－
					－	なし	なし	PU	－	SR	SU	－	EX	－	－	SR	－	－	－
					－	なし	あり	PU	－	SR	SU	－	EX	－	－	PU	－	－	－
			exclusive	retrieve	－	－	－	EX	－	SR	SR	－	－	－	－	－	－	－	－
				update	－	－	－	EX	－	SR	SU	－	－	－	－	－	－	－	－
			nonprotected	retrieve	－	－	－	SR	－	SR	－	SR ※33	－	－	－	－	－	－	－
		位置指示子指定の検索	shared	retrieve	－	－	なし	SR	－	－	SR	－	SR ※42	－	－	SR	－	－	－
				update	－	－	なし	SU	－	－	SU	－	SR ※42	－	－	SR	－	－	－
					－	－	あり	SU	－	－	SU	－	SR ※42	－	－	PU	－	－	－
			protected	retrieve	－	－	なし	PR	－	－	SR	－	－	－	－	－	－	－	－
				update	－	－	なし	PU	－	－	SU	－	SR ※42	－	－	SR	－	－	－
					－	－	あり	PU	－	－	SU	－	SR ※42	－	－	PU	－	－	－
			exclusive	retrieve	－	－	なし	EX	－	－	SR	－	－	－	－	－	－	－	－
				update	－	－	－	EX	－	－	SU	－	－	－	－	－	－	－	－
			nonprotected	retrieve	－	－	なし	SR	－	－	－	SR ※33	－	－	－	－	－	－	－

（続く）

（続き）

実行するDMLの種類	操作対象のﾚｺｰﾄﾞ	検索種別	実行条件					資源											
								上位←－－－－→下位											
			SD排他ﾓｰﾄﾞ ※30	ｱｸｾｽ目的 ※31	SDBﾃﾞｰﾀﾍﾞｰｽの階層数	ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞの共用排他指定 ※34	FOR UPDATEｵﾌﾟｼｮﾝの指定	RDｴﾘｱ			ﾙｰﾄﾚｺｰﾄﾞ型 ※2	ﾙｰﾄﾚｺｰﾄﾞ型（NOWAIT検索時） ※36	ページ					行	
								ﾚｺｰﾄﾞ格納用	ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽ格納用	二次ｲﾝﾃﾞｸｽ格納用			ﾙｰﾄﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	親ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	PRIORﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	対象ﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	NEXTﾚｺｰﾄﾞ格納ﾍﾟｰｼﾞ	一意性制約保証用 ※3	重複ｷｰ追加時の排他待ち用 ※35
MODIFY	ﾙｰﾄﾚｺｰﾄﾞ	－	shared	update	－	－	－	SU	－	SU ※13	SU	－	EX	－	－	－	－	EX ※37	－
			protected	update	－	－	－	PU	－	SU ※13	SU	－	EX	－	－	－	－	EX ※37	－
			exclusive	update	－	－	－	EX	－	SU ※13	SU	－	－	－	－	－	－	EX ※37	－
	子ﾚｺｰﾄﾞ	－	shared	update	－	－	－	SU	－	SU ※13	SU	－	EX ※32	EX ※40	EX ※40 ※41	EX	EX ※40 ※41	EX ※37	－
			protected	update	－	－	－	PU	－	SU ※13	SU	－	EX ※32	EX ※40	EX ※40 ※41	EX	EX ※40 ※41	EX ※37	－
			exclusive	update	－	－	－	EX	－	SU ※13	SU	－	－	－	－	－	－	EX ※37	－
STORE	ﾙｰﾄﾚｺｰﾄﾞ	－	shared	update	－	－	－	SU	SU	SU ※13	SU	－	EX	－	－	－	－	EX ※39	EX
			protected	update	－	－	－	PU	SU	SU ※13	SU	－	EX	－	－	－	－	EX ※39	EX
			exclusive	update	－	－	－	EX	SU	SU ※13	SU	－	－	－	－	－	－	EX ※39	－
	子ﾚｺｰﾄﾞ	－	shared	update	－	－	－	SU	－	SU ※13	SU	－	EX ※32	PU ※7 ※8	PU ※10	EX	PU ※11	EX ※37	EX
			protected	update	－	－	－	PU	－	SU ※13	SU	－	EX ※32	PU ※7 ※8	PU ※10	EX	PU ※11	EX ※37	EX
			exclusive	update	－	－	－	EX	－	SU ※13	SU	－	－	－	－	－	－	EX ※37	－
ERASE	ﾙｰﾄﾚｺｰﾄﾞ	－	shared	update	－	－	－	SU	SU	SU ※13	SU	－	EX	－	－	EX	－	EX ※27	－
			protected	update	－	－	－	PU	SU	SU ※13	SU	－	EX	－	－	EX	－	EX ※27	－
			exclusive	update	－	－	－	EX	SU	SU ※13	SU	－	－	－	－	－	－	EX ※27	－
	子ﾚｺｰﾄﾞ	－	shared	update	－	－	－	SU	－	SU ※13	SU	－	EX ※32	EX	EX	EX	EX	EX ※37	－
			protected	update	－	－	－	PU	－	SU ※13	SU	－	EX ※32	EX	EX	EX	EX	EX ※37	－
			exclusive	update	－	－	－	EX	－	SU ※13	SU	－	－	－	－	－	－	EX ※37	－

（凡例）

PU：共用意図排他モード

EX：排他モード

SR：意図共用モード

SU：意図排他モード

－：排他を掛けないことを示します。または，該当しません。

注※1

インナレプリカ機能使用時の最上位排他資源は，インナレプリカ構成管理情報，またはレプリカグループ構成管理情報となります。これらの排他資源を管理することで，インナレプリカ機能に関連する運用コマンドやユティリティとUAPの同時実行を抑止します。

インナレプリカ機能を使用しない場合，UAPを実行しても，インナレプリカ構成管理とレプリカグループ構成管理には排他を掛けません。

インナレプリカ機能を使用する場合，次に示す条件のうち，どちらかを満たすと，UAP 実行時に排他を掛ける範囲が小さくなります。

UAP 実行時の排他の単位が小さくなる条件

- システム共通定義の pd_inner_replica_lock_shift オペランドに Y を指定する。
- システム共通定義の pd_inner_replica_lock_shift オペランドに N を指定（インナレプリカ機能の同時実行性を向上させない指定）し，かつ運用コマンドやユティリティが動作中である。

排他を掛ける範囲が小さくなった場合の排他範囲

- インナレプリカ構成管理単位ではなく，レプリカグループ構成管理単位になる。
- サーバ単位ではなく，インナレプリカグループ単位になる。

詳細については，マニュアル「HiRDB システム定義」の pd_inner_replica_lock_shift オペランドの説明を参照してください。

注※2

更新ログ取得方式がログレスモード指定の場合，SU モードは EX モードの排他になります。

注※3

インデクスの一意性を保証するために HiRDB/SD が内部的に行に排他を掛けます。

注※4

SDB データベースを操作する API で指定する排他モード 1

'S'：共用モード

'E'：排他モード

'O'：占有モード

'N'：無排他モード

注※5

SDB データベースを操作する API で指定するアクセスモード 1

'R'：参照モード

'U'：更新モード

注※6

SDB データベースを操作する API で指定するポインタオプション

'U'：親レコードの USER ポインタに対象レコードの格納位置を設定します。

'C'：親レコードの USER ポインタの設定を解除します。

注※7

自身がルートレコードの場合，またはルートレコードの格納ページと同一の場合は，EX モードになります。

注※8

親レコードの格納ページが，対象レコードの格納ページと同一の場合は，EX モードになります。

注※9

シーケンシャルインデクスの一意性を保証するために，排他を掛けます。

注※10

PRIOR レコード（格納するレコードの PRIOR ポインタに接続されるレコード）がルートレコードの場合，または PRIOR レコードの格納ページがルートレコードか対象レコードの格納ページと同一の場合は，EX モードになります。

注※11

NEXT レコード（格納するレコードの NEXT ポインタに接続されるレコード）がルートレコードの場合，または NEXT レコードの格納ページがルートレコードか対象レコードの格納ページと同一の場合は，EX モードになります。

注※12

SDB データベースを操作する API で指定する一括オプション

'O'：最下位レベルのレコードについて，すべてのレコードを削除します。

'S'：削除対象のレコードが，最下位レベルのときは位置づけしているレコードだけを削除します。削除対象のレコードが最下位レベルでないときは，位置づけしているレコードとその下位レベルのレコードを削除します。

注※13

対象となるレコード型に二次インデクスが定義されている場合だけ排他を掛けます。ただし，レコードの検索で，検索に二次インデクスを使用しない場合は排他を掛けません。また，排他資源が行の場合，追加または削除対象のレコードに関する排他を掛けます。

注※14

操作対象レコードのレコード型の，SDB データベース格納定義の OCCURRENCE NUMBER 句の指定値が 0 の場合は，SDB データベースを操作する API で指定する排他モード 1 の指定によって，次のように排他制御モードが変更になります。

- 排他モード 1 の指定が'E'の場合：PU モード
- 排他モード 1 の指定が'O'の場合：EX モード

注※15

同一データベースキー値のレコードが 1 件も格納されていない場合，仮想ルートレコードを追加したあと，子レコードを追加します。ただし，レコードの削除によって当該データベースキー値を持つ子レコードが 0 件となる場合は，仮想ルートレコードは削除されないため，そのあとに実行する 1 件目のレコードの格納は該当しません。

注※16

シーケンシャルインデクス，および二次インデクスの一意性を保証するために，排他を掛けます。ただし，二次インデクスについては，対象となるレコード型に二次インデクスが定義されている場合だけ，排他を掛けます。

注※17

PRIOR レコードの格納ページが，対象レコードの格納ページと同一の場合は EX モードになります。

注※18

NEXT レコードの格納ページが，対象レコードの格納ページと同一の場合は EX モードになります。なお，4V AFM の SDB データベースは最後方への追加だけのため，NEXT レコードは必ずルートレコードとなります。

注※19

一括削除後のレコードの格納は，削除前の一連番号の最大値に達するまでは，内部的にレコードの更新（MODIFY）と同等の変更処理に置き換わります。

注※20

対象レコードの格納ページと同一の場合は，EX モードになります。

注※21

検索対象となるレコードに，ルートレコードが含まれる場合が対象となります。

注※22

検索対象となるレコードに，ルートレコードが含まれない場合が対象となります。

注※23

次に示すどちらかの場合は，HiRDB/SD が内部的にシーケンシャルインデクスを更新するため，SU モードで排他を掛けます。

- レコードの削除，またはレコードの一括削除によって，当該データベースキー値を持つ子レコードが 0 件となったあとに実行する，1 件目のレコードの格納の場合
- レコードの削除によって，当該データベースキー値を持つ子レコードが 0 件となる場合

注※24

次に示すどちらかの場合は，HiRDB/SD が内部的にシーケンシャルインデクスを更新するため，仮想ルートレコードに EX モードで排他を掛けます。

- レコードの削除，またはレコードの一括削除によって，当該データベースキー値を持つ子レコードが 0 件となったあとに実行する，1 件目のレコードの格納の場合
- レコードの削除，またはレコードの一括削除によって，当該データベースキー値を持つ子レコードが 1 件以上から 0 件となる場合

注※25

レコードの一括削除の実行前に当該データベースキー値を持つ子レコードが 0 件の場合，各排他資源の排他制御モードは次のようになります。

シーケンシャルインデクス格納用 RD エリア：SR

行：－

注※26

TAM のデータベースの無排他検索機能で動作する場合は，排他を掛けません。

注※27

対象となるレコード型にインデクスが定義されている場合に限り，削除対象のレコードに排他を掛けます。

注※28

削除対象のファミリを事前割り当てページに格納している場合，SDB データベースを操作する API（ERASE）実行時に子レコードが格納されていない事前割り当てページに対しても，ページを解放するために排他を掛けます。

注※29

SDB データベースを操作する API（STORE）実行時に確保する事前割り当てページに対しても，EX モードで排他を掛けます。

注※30

SD 排他モードは，SDB 用 UAP 環境定義の subschema オペランドの-e オプションで指定します。-e オプションの指定値の意味は次のとおりです。

- shared：ほかの UAP からの参照および更新を許します。
- protected：ほかの UAP からの更新を許しません。
- exclusive：ほかの UAP からの参照および更新を許しません。
- nonprotected：ほかの UAP からの参照および更新を許します（ページ（サブページ分割をする場合はサブページ）に対する排他制御を行いません）。

注※31

アクセス目的は，SDB 用 UAP 環境定義の subschema オペランドの-a オプションで指定します。-a オプションの指定値の意味は次のとおりです。

- retrieve：レコードの参照だけを行います。
- update：レコードの参照および更新を行います。

注※32

次の 2 つの条件が重なる場合だけ，排他を掛けます。

- レコードの検索（FETCH）または位置指示子の位置づけ（FIND）でルートレコードに位置づけている。
- 排他自動解除機能を使用している。またはルートレコード格納ページに対して排他制御モード SR または PU で排他制御をしている（subschema オペランドの-p オプションに shareroot を指定している）。

注※33

ルートレコード型に排他が掛かっている場合は，デッドロック防止のため，排他を掛けません。

注※34

ルートレコード格納ページの共用排他指定の意味は次のとおりです。

- ルートレコード格納ページの共用排他指定あり：ルートレコード格納ページに，排他制御モード SR または PU で排他制御を行います。
- ルートレコード格納ページの共用排他指定なし：ルートレコード格納ページに，排他制御モード EX で排他制御を行います。

ルートレコード格納ページの共用排他指定ありとなしは，SDB 用 UAP 環境定義の subschema オペランドの-p オプションで指定します。

ルートレコード格納ページの共用排他指定と，-p オプションの対応関係は次のとおりです。

- ルートレコード格納ページの共用排他指定あり：-p shareroot
- ルートレコード格納ページの共用排他指定なし：-p occupyroot

注※35

システム共通定義 pd_lock_uncommited_delete_data オペランドに WAIT を指定している場合だけ，排他を掛けます。同じデータベースキーを追加する後続のトランザクションを，即時ユニークエラーとしないで排他待ちにさせるためです。

子レコードの場合は，pd_lock_uncommited_delete_data オペランドに WAIT を指定していて，かつ二次インデクスの定義があるときに排他を掛けます。

注※36

NOWAIT 検索とは，無排他検索のことです。無排他検索機能，または無排他検索機能との同時実行を許容していない機能を実行すると，排他が掛かります。

注※37

対象となるレコード型に二次インデクスが定義されている場合に，二次インデクスの一意性を保証するために，排他を掛けます。

注※38

検索に使用するインデクスを格納している RD エリアだけに排他が掛かります。

注※39

シーケンシャルインデクスと二次インデクスの一意性を保証するために，排他を掛けます。

注※40

キー項目を更新する場合に排他を掛けます。

注※41

キー項目を更新する前の PRIOR レコード格納ページ（NEXT レコード格納ページ）と，更新したあとの PRIOR レコード格納ページ（NEXT レコード格納ページ）の，両方のページに排他を掛けます。

注※42

次の 3 つの条件が重なる場合だけ，排他を掛けます。

- 排他自動解除機能を使用している。
- SDB データベースのどれかの子レコード型に二次インデクスが定義されている。
- 2 階層目のレコード型を検索する。

なお，次の表には，SDB データベースを操作する API または DML がリターンした時点で確保している排他だけを記載しています。

- 「表 2-35　排他制御のモードの組み合わせの例（4V FMB の SDB データベースの場合）」
- 「表 2-36　排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合）」

- 「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」

上記の表には，処理途中で一時的に確保する排他については記載していません。処理途中で一時的に確保する排他※には次のものがあります。

- 1 つまたは複数のレコードの中から条件に合致するレコードを検索する場合，条件に合致しなかったレコードの格納ページ（排他制御モード：SR，PU，または EX（条件に合致するレコードの格納ページと同じになります））
- 1 つまたは複数のレコードの中からレコードの挿入位置を検索する場合，挿入位置とならなかったページ（排他制御モード：PU）
- 4V FMB または 4V AFM の SDB データベースで，終端検知（レコードの検索時に最終のレコードかどうかを判定する）のために検索する，LAST ポインタが指すレコードの格納ページ，または終端検知のために検索するルートレコード，仮想ルートレコードの格納ページ（排他制御モード：SR）
- 4V AFM の SDB データベースで，構成情報取得のために検索する仮想ルートレコードの格納ページ（排他制御モード：SR）

注※

ただし，次の場合には一時的な排他も確保しません。

- 4V FMB または 4V AFM の SDB データベースで，SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）を指定した場合
- TAM のデータベースの無排他検索機能で動作する場合

(3)　同一レコード格納用 RD エリアに対する，SDB データベースへのアクセスの同時実行可否

■4V FMB または 4V AFM の SDB データベースの場合

「表 2-35　排他制御のモードの組み合わせの例（4V FMB の SDB データベースの場合）」と「表 2-36　排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合）」に示すように，4V FMB または 4V AFM の SDB データベースを操作する API の実行時，レコード格納用 RD エリアには，SR モードまたは SU モードでしか排他が掛かりません。そのため，そのレコード格納用 RD エリアに対して排他を掛けるほかのトランザクションからの SDB データベースを操作する API の実行は，レコード格納用 RD エリアの排他によって，待ち状態になることはありません（ただし，そのほかのページなどの排他によっては待ち状態になることはあります）。

■SD FMB の SDB データベースの場合

「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」に示すように，SD FMB の SDB データベースに対する DML の実行時，SD 排他モードとアクセス目的の指定の組み合わせによっては，レコード格納用 RD エリアに PR モード，PU モード，または EX モードで排他が掛かります。そのため，そのレコード格納用 RD エリアに対して排他を掛けるほかのトランザクションからの DML の実行は，レコード格納用 RD エリアに掛かっている排他によっては，トランザク

ション（レコード格納用 RD エリアに PR モード，PU モード，または EX モードで排他を掛けているトランザクション）が終了するまで待ち状態になることがあります。

同一レコード格納用 RD エリアに対する，SDB データベースへのアクセスの同時実行可否を次の表に示します。

表 2-38　同一レコード格納用 RD エリアに対する，SDB データベースへのアクセスの同時実行可否（SD FMB の SDB データベースの場合）

SD 排他モードとアクセス目的の指定の組み合わせ	SD 排他モードとアクセス目的の指定の組み合わせ						
	nonprotected と retrieve	shared と retrieve	shared と update	protected と retrieve	protected と update	exclusive と retrieve	exclusive と update
nonprotected と retrieve	○	○	○	○	○	×	×
shared と retrieve	○	○	○	○	○	×	×
shared と update	○	○	○	×	×	×	×
protected と retrieve	○	○	×	○	×	×	×
protected と update	○	○	×	×	×	×	×
exclusive と retrieve	×	×	×	×	×	×	×
exclusive と update	×	×	×	×	×	×	×

（凡例）

○：同時実行できます。

×：同時実行できません。

(4)　同一ルートレコード下のレコード群に対する，SDB データベースを操作する API または DML の同時実行可否

同一ルートレコード下のレコード群に対する，SDB データベースの操作の同時実行可否を，SDB データベース種別ごとに説明します。

(a)　4V FMB の SDB データベースの場合

「表 2-35　排他制御のモードの組み合わせの例（4V FMB の SDB データベースの場合）」に示すように，4V FMB の SDB データベースに対する次の操作では，ルートレコードの格納ページに PU モードまたは EX モードで排他が掛かります。

- 4V FMB の SDB データベースの更新系の操作（レコードの更新（MODIFY），レコードの格納（STORE），レコードの削除（ERASE））
- SDB データベースを操作する API の排他モード 1 に，'E'（排他モード）または'O'（占有モード）を指定したレコードの検索（レコードの検索（FETCH），複数レコードの検索（FETCHDB ALL））

そのため，ルートレコードに排他が掛かっている間は，そのルートレコードおよびそのルートレコード下のレコード群に対して，排他を掛けるほかのトランザクションからのアクセスは待ち状態になることがあります。

同一ルートレコード下のレコード群に対する，SDB データベースを操作する API の同時実行可否を次の表に示します。なお，ルートレコードが異なるレコードについては，同時実行できます。

表 2-39　同一ルートレコード下のレコード群に対する，SDB データベースを操作する API の同時実行可否（4V FMB の SDB データベースの場合）

SDB データベースを操作する API の種類			レコードの検索（FETCH）または複数レコードの検索（FETCHDB ALL）				レコードの更新（MODIFY）	レコードの格納（STORE）	レコードの削除（ERASE）
			API の排他モード 1 の指定						
			'S'	'E'	'O'	'N'※			
レコードの検索（FETCH）または複数レコードの検索（FETCHDB ALL）	API の排他モード 1 の指定	'S'	○	○	×	○	×	×	×
		'E'	○	×	×	○	×	×	×
		'O'	×	×	×	○	×	×	×
		'N'※	○	○	○	○	○	○	○
レコードの更新（MODIFY）			×	×	×	○	×	×	×
レコードの格納（STORE）			×	×	×	○	×	×	×
レコードの削除（ERASE）			×	×	×	○	×	×	×

（凡例）

○：同時実行できます。

×：同時実行できません。

注※

レコードの検索（FETCH）の場合に限り，SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）を指定できます。

(b)　4V AFM の SDB データベースの場合

「表 2-36　排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合）」に示すように，4V AFM の SDB データベースでは同一仮想ルートレコード下のレコードの同時アクセスを可能とするため，レコードの検索（FETCH）およびレコードの更新（MODIFY）では仮想ルートレコードの格納ページに SR モードで排他を掛けます。※

注※

SDB データベースを操作する API の排他制御モード 1 の指定によって，SR モード以外の排他となることがあります。詳細については，「表 2-36　排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合）」を参照してください。

同一仮想ルートレコード下のレコード群に対する，SDB データベースを操作する API の同時実行可否を次の表に示します。なお，仮想ルートレコードが異なるレコードについては，同時実行できます。

表 2-40　同一仮想ルートレコード下のレコード群に対する，SDB データベースを操作する API の同時実行可否（4V AFM の SDB データベースの場合）

SDB データベースを操作する API の種類			レコードの検索（FETCH）				レコードの更新（MODIFY）	レコードの格納（STORE）	レコードの削除（ERASE）	レコードの一括削除※4	構成情報取得※4，※7	
			API の排他モード 1 の指定								API の排他モード 1 の指定	
			'S'	'E'	'O'	'N'					'S'/'E'/'O'	'N'
レコードの検索（FETCH）	API の排他モード 1 の指定	'S'	○	○	○※6	○	○※6	○※1	×	×	○	○
		'E'	○	○※1	○※1	○	○※1	○※1，※2	×	×	○	○
		'O'	○※6	○※1	○※1	○	○※1	○※1，※2	×	×	○※5	○
		'N'	○	○	○	○	○	○	○	○	○	○
レコードの更新（MODIFY）			○※6	○※1	○※1	○	○※1	○※1，※2	×	×	○※5	○
レコードの格納（STORE）			○※1	○※1，※2	○※1，※2	○	○※1，※2	×	×	×	○※3，※5	○
レコードの削除（ERASE）			×	×	×	○	×	×	×	−	×	○
レコードの一括削除※4			×	×	×	○	×	×	−	×	×	○
構成情報取得※4，※7	API の排他モード 1 の指定	'S'/'E'/'O'	○	○	○※5	○	○※5	○※5	×	×	○	○
		'N'	○	○	○	○	○	○	○	○	○	○

（凡例）

○：同時実行できます。

×：同時実行できません。

−：該当しません。

注※1

同一レコード（ページ）に対するアクセスは同時に実行できません。

注※2

レコードの格納時（STORE）には，対象となるレコードの NEXT レコードおよび PRIOR レコードの格納ページに PU モードで排他を掛けます。そのため，レコードの格納と次の操作との間で排他待ちが発生します。

- SDB データベースを操作する API の排他モード 1 に'E'（排他モード）または'O'（占有モード）が指定されたレコードの検索（FETCH）
- レコードの更新（MODIFY）

なお，レコードの一括削除後にレコードの格納をする場合，削除前の一連番号の最大値に達するまで，レコードの格納処理は，内部的にレコードの更新と同等の変更処理に置き換わります。この場合のレコードの格納は，上記に該当しません。

注※3

仮想ルートレコードの追加を伴う 1 件目のレコードの格納（STORE）とは同時に実行できません。

注※4

内部的に仮想ルートレコードにアクセスします。

注※5

次に示す条件をすべて満たす場合は，同時に実行できません。

- SDB データベース定義の OCCURRENCE NUMBER 句の指定値が 0 である。
- 次の操作のうちのどれかを，構成情報取得よりも先行して実行する。

  ・SDB データベースを操作する API の排他モード 1 に'O'（占有モード）が指定されたレコードの検索（FETCH）

  ・レコードの更新（MODIFY）

  ・レコードの格納（STORE）

注※6

同一レコード（ページ）に対するアクセスは同時に実行できません。ただし，TAM のデータベースの無排他検索機能が動作する場合は，同一レコード（ページ）に対するアクセスが同時に実行できます。TAM のデータベースの無排他検索機能については，「2.9.6　TAM のデータベースの無排他検索機能【4V TAM】」を参照してください。

注※7

TAM のデータベースの無排他検索機能を適用する場合，SDB データベースを操作する API の排他モード 1 に’N’（無排他モード）を指定したときと同様の同時実行性となります。

(c)　SD FMB の SDB データベースの場合

「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」に示すように，SD FMB の SDB データベースに対して次の操作を実行した場合，ルートレコードの格納ページに PU モードまたは EX モードで排他が掛かります。

- SD FMB の SDB データベースの更新系の操作（レコードの更新（MODIFY），レコードの格納（STORE），レコードの削除（ERASE））
- FOR UPDATE オペランドを指定するか，または 2 階層以上の SD FMB の SDB データベースに対する，レコードの参照系の操作（レコードの検索（FETCH），位置指示子の位置づけ（FIND））

そのため，ルートレコードに排他が掛かっている間は，そのルートレコードおよびそのルートレコード下のレコード群に対して，排他を掛けるほかのトランザクションからのアクセスは，待ち状態になることがあります。

同一ルートレコード下のレコード群に対する，DML の同時実行可否を次の表に示します。

なお，ルートレコードが異なるレコードについては，同時実行できます。

表 2-41　同一ルートレコード下のレコード群に対する，DML の同時実行可否（SD FMB の SDB データベースの場合）

DML の種類									レコードの検索(FETCH)または位置指示子の位置づけ(FIND)					レコードの更新(MODIFY)	レコードの格納(STORE)	レコードの削除(ERASE)
									SD 排他モード							
									nonprotected	shared または protected						
										アクセス目的						
										retrieve	update					
											ルートレコード格納ページの共用排他指定 ※1					
											あり		なし			
											FOR UPDATE 指定					
											なし	あり				
レコードの検索(FETCH)または位置指示子の位置づけ(FIND)	SD排他モード	nonprotected							○	○	○	○	○	○	○	○
		shared または protected	アクセス目的	retreive					○	○	○	○	×	×	×	×
				update	ルートレコード格納ページの共用排他指定 ※1	あり	FOR UPDATE 指定	なし	○	○	○ ※2	○ ※2	×	×	×	×
								あり	○	○	○ ※2	×	×	×	×	×
						なし			○	×	×	×	×	×	×	×
レコードの更新(MODIFY)									○	×	×	×	×	×	×	×
レコードの格納(STORE)									○	×	×	×	×	×	×	×
レコードの削除(ERASE)									○	×	×	×	×	×	×	×

（凡例）

○：同時実行できます。

×：同時実行できません。

注※1

SDB 用 UAP 環境定義の subschema オペランドの-p オプションで指定します。

あり：ルートレコード格納ページに，排他制御モード SR または PU で排他制御を行います（shareroot 指定）。

なし：ルートレコード格納ページに，排他制御モード EX で排他制御を行います（occupyroot 指定）。

注※2

SD 排他モード（SDB 用 UAP 環境定義の subschema オペランドの-e オプションの指定）が protected の場合，レコード格納用 RD エリアの排他が競合するため，同時実行できません。

2.9.3　排他の期間

トランザクションがある資源に対して掛けた排他は，一般にコミット，またはロールバックされるまで解除されません。例えば，あるページに対して排他が競合した場合，トランザクションがコミット，またはロールバックされるまで，ほかのトランザクション処理が待ち状態になることがあります。

ただし，排他自動解除機能を使用している場合は，トランザクションの終了を待たないで，ページの排他を解除できます。排他自動解除機能の対象となるデータページ（ページまたはサブページ）については，「付録 L　SDB データベースを操作する API または DML と，データページの排他解除の有無の関係」を参照してください。

2.9.4　排他自動解除機能

排他自動解除機能とは，SDB データベースを操作する API または DML で，SDB データベースを操作した結果，レコード位置指示子が指さなくなったレコード実現値を格納するページに対する排他を，API または DML の操作完了時に解除する機能です。

なお，複数レコードの検索時に排他自動解除機能を適用すると，複数レコードの検索の結果，検索対象のファミリが移動した場合に，移動元のファミリのルートレコード実現値を格納するページに対する排他を解除する機能となります。

排他自動解除機能を使用しない場合は，トランザクション内で一度確保された排他は，トランザクションが終了するまで保持されます。

排他自動解除機能の選択基準を次の表に示します。

表 2-42　排他自動解除機能の選択基準

排他自動解除機能	選択基準
使用する	• トランザクション内でデータの一貫性を保証する必要はなく，検索したレコード実現値を更新するほかのトランザクションとの同時実行性能を重視する場合 • 更新を前提とする検索の結果，検索したレコード実現値に対する更新が不要となることがあり，そのレコード実現値を操作するほかのトランザクションとの同時実行性能を重視する場合
使用しない	■4V FMB または 4V AFM の SDB データベースの場合 • トランザクション内でデータの一貫性を保証する必要があり，一度検索したレコード実現値がほかのトランザクションによって更新されるのをトランザクション終了まで抑止したい場合 • SDB データベースを操作する API の排他モード 1 に排他モードを指定した 2 つの検索で，一方のトランザクションが検索したレコード実現値を，もう一方のトランザクションが参照するのを，トランザクション終了まで抑止したい場合 • SDB データベースを操作する API の排他モード 1 に'O'（占有モード）を指定した検索で，自トランザクションが検索したレコード実現値をほかのトランザクションが参照するのを，トランザクション終了まで抑止したい場合 ■SD FMB の SDB データベースの場合

排他自動解除機能	選択基準
	• トランザクション内でデータの一貫性を保証する必要があり，一度検索したレコード実現値がほかのトランザクションによって更新されるのをトランザクション終了まで抑止したい場合 • FOR UPDATE オペランドを指定した 2 つの検索で，一方のトランザクションが検索したレコード実現値を，もう一方のトランザクションが参照するのを，トランザクション終了まで抑止したい場合

(1) 適用方法

4V FMB または 4V AFM の SDB データベースに対して排他自動解除機能を適用する場合は，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「排他制御」を参照してください。排他自動解除機能は，TP1/FSP の個別開始時に指定する排他パラメタの排他モード 2 で指定します。

SD FMB の SDB データベースに対して排他自動解除機能を適用する場合は，SDB 用 UAP 環境定義に lockrange オペランドを指定します。SDB 用 UAP 環境定義については，「9.3 SDB 用 UAP 環境定義【SD FMB】」を参照してください。

(2) SDB データベースを操作する API または DML と，データページの排他解除の有無の関係

SDB データベースを操作する API または DML と，データページ（ページまたはサブページ）の排他解除の有無（排他自動解除機能の対象となるデータページ）の関係については，「付録 L SDB データベースを操作する API または DML と，データページの排他解除の有無の関係」を参照してください。

(3) 注意事項

■4V FMB または 4V AFM の SDB データベースの場合

- 個別開始から個別終了までの間に，SDB データベースを操作する API で，排他自動解除機能の適用の有無を一致させる必要があります。
- 排他自動解除機能の対象となるかどうかは，SDB データベースを操作する API の排他モード 1 の指定（'S'（共用モード），'E'（排他モード），および'O'（占有モード））に依存しません。
- 次の場合，排他自動解除機能は適用されません。
  - TAM のデータベースの無排他検索機能が適用される検索（FETCH），および構成情報取得
  - SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）を指定している検索（FETCH），および構成情報取得

■SD FMB の SDB データベースの場合

- 排他自動解除機能の対象となるかどうかは，レコードの検索（FETCH）または位置指示子の位置づけ（FIND）の FOR UPDATE オペランドの指定に依存しません。

- SDB 用 UAP 環境定義の subschema オペランドの-e オプションに nonprotected を指定している場合，排他自動解除機能は適用されません。

2.9.5　排他自動解除機能を使用したレコードの検索例

排他自動解除機能を使用したレコードの検索例を次に示します。

- 排他自動解除機能を使用して，4V FMB または SD FMB の SDB データベースを階層順に検索する場合の例は，「2.9.5(1)例 1　排他自動解除機能を使用する場合の SDB データベースの階層順検索【4V FMB，SD FMB】」を参照してください。

  また，排他自動解除機能を使用しない場合の例については，「2.9.5(2)例 2　排他自動解除機能を使用しない場合の SDB データベースの階層順検索【4V FMB，SD FMB】」を参照してください。

  参考

  例 1 は，4V FMB の SDB データベースの場合は，SDB データベースを操作する API の排他モード 1 に'S'（共用モード）を指定したときの検索例です。SD FMB の SDB データベースの場合は，FOR UPDATE オペランドの指定を省略したときの検索例です。

- 排他自動解除機能を使用して各 API または DML を実行したあと，別のルートレコードのレコード実現値を検索してレコード位置指示子が移動する場合の例については，次の項目を参照してください。
  - 「2.9.5(3)例 3　排他自動解除機能を使用してレコードの格納（STORE）を実行する場合の動作【4V FMB，SD FMB】」
  - 「2.9.5(4)例 4　排他自動解除機能を使用してレコードの更新（MODIFY）を実行する場合の動作【4V FMB，SD FMB】」
  - 「2.9.5(5)例 5　排他自動解除機能を使用してレコードの削除（ERASE）を実行する場合の動作【4V FMB，SD FMB】」
  - 「2.9.5(6)例 6　排他自動解除機能を使用して一括削除を実行する場合の動作【4V DAM，4V SAM】」
  - 「2.9.5(7)例 7　構成情報取得を実行する場合の動作【4V AFM】」
- 排他自動解除機能を使用して複数レコードの検索（FETCHDB ALL）を実行した場合の例については，「2.9.5(8)例 8　排他自動解除機能を使用して複数レコードの検索（FETCHDB ALL）を実行する場合の動作【4V FMB】」を参照してください。

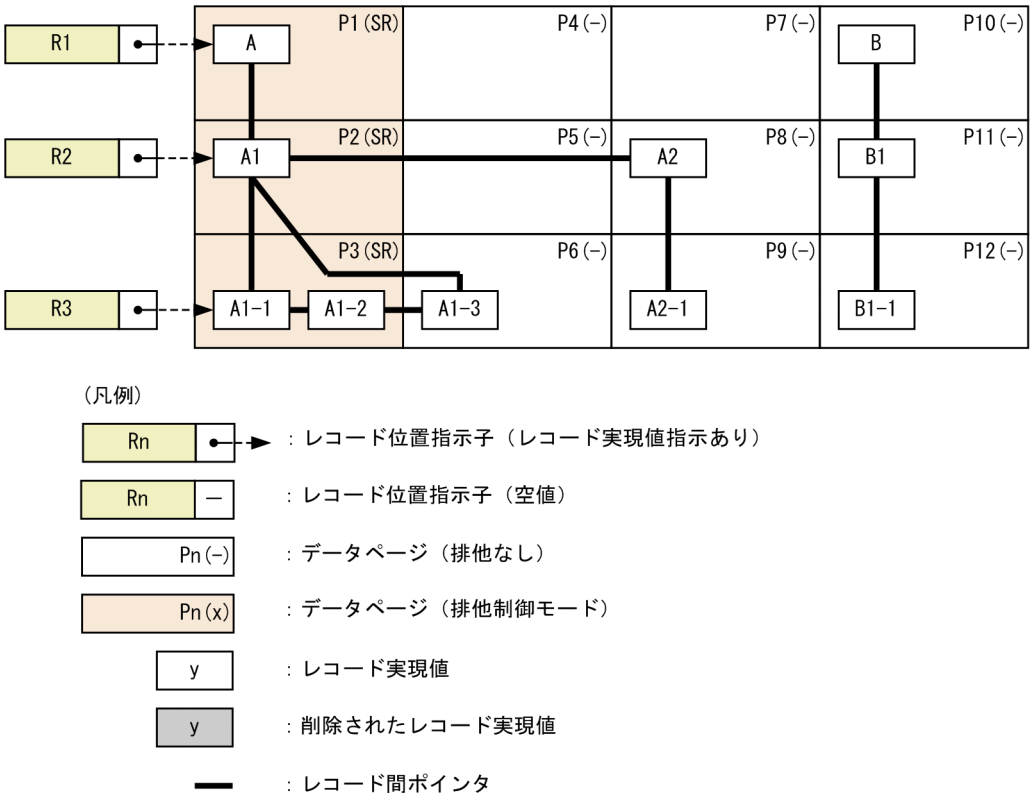
4V AFM の SDB データベースでも仮想ルートレコードのレコード実現値を操作して検索するため，基本的には 4V FMB の SDB データベースの例と同様の動作となります。

なお，上記の例の中で使用している記号の凡例については，「2.9.5(1)例 1　排他自動解除機能を使用する場合の SDB データベースの階層順検索【4V FMB，SD FMB】」の 1.にある図の凡例を参照してください。

(1) 例 1　排他自動解除機能を使用する場合の SDB データベースの階層順検索【4V FMB，SD FMB】

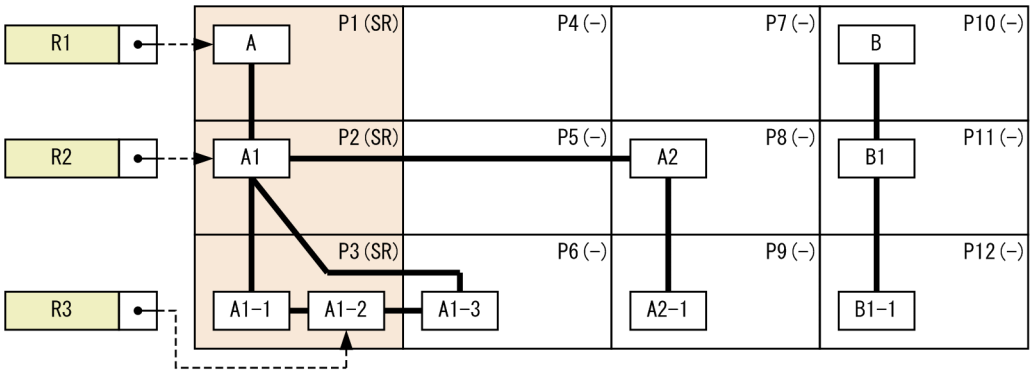
1. ルートレコードのレコード実現値 A，2 階層目のレコード実現値 A1，および 3 階層目のレコード実現値 A1-1 の順にレコードを検索（FETCH）します。

   これによって，検索したレコード実現値が格納されているデータページ P1，P2，および P3 に排他（SR モード）が掛かります。



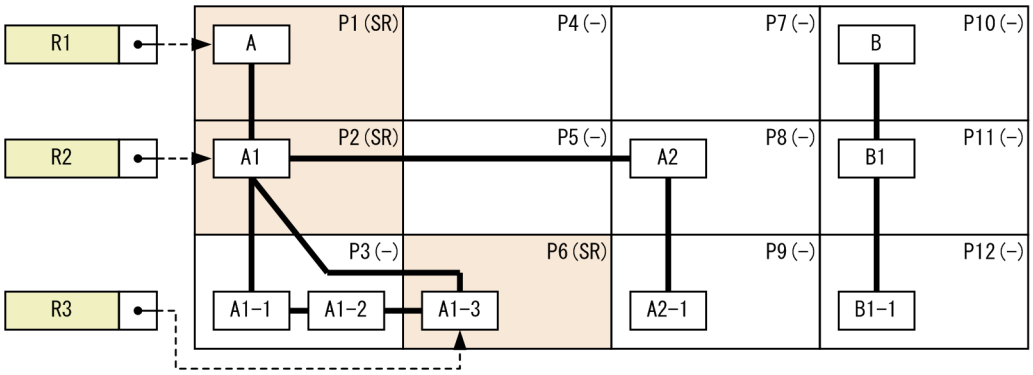
2. 3 階層目のレコード実現値 A1-2 を検索（FETCH）します。

   この場合，レコード位置指示子 R3 が指すレコード実現値は，A1-1 から A1-2 となりますが，これらのレコード実現値は同一データページに格納されているため，データページ P3 は排他（SR モード）が掛かった状態のままです。



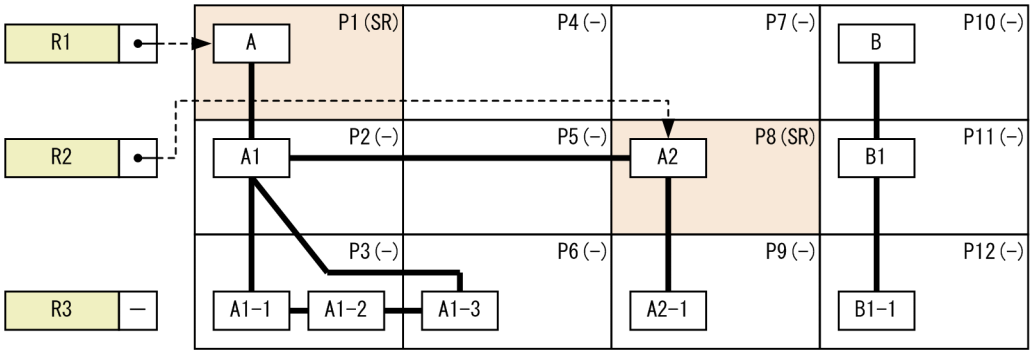
3. 3 階層目のレコード実現値 A1-3 を検索（FETCH）します。

この場合，レコード位置指示子 R3 が指すレコード実現値は，A1-2 から A1-3 となります。これらのレコード実現値は異なるデータページに格納されているため，データページ P6 に排他（SR モード）が掛かり，データページ P3 の排他（SR モード）は解除されます。



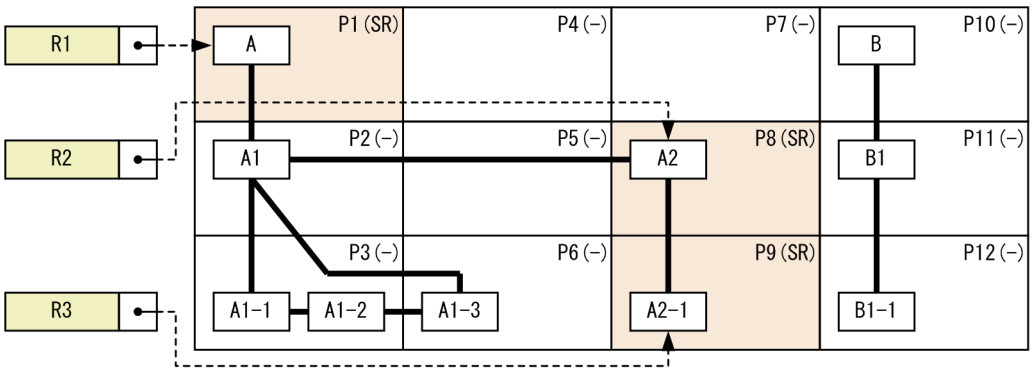
4. 2 階層目のレコード A2 を検索（FETCH）します。

この場合，レコード位置指示子 R2 が指すレコード実現値は A1 から A2 となります。これらのレコード実現値は異なるデータページに格納されているため，データページ P8 に排他（SR モード）が掛かり，データページ P2 の排他（SR モード）は解除されます。同時に，レコード位置指示子 R3 が空値となるため，3 階層目のレコード実現値 A1-3 が格納されているデータページ P6 の排他（SR モード）も解除されます。



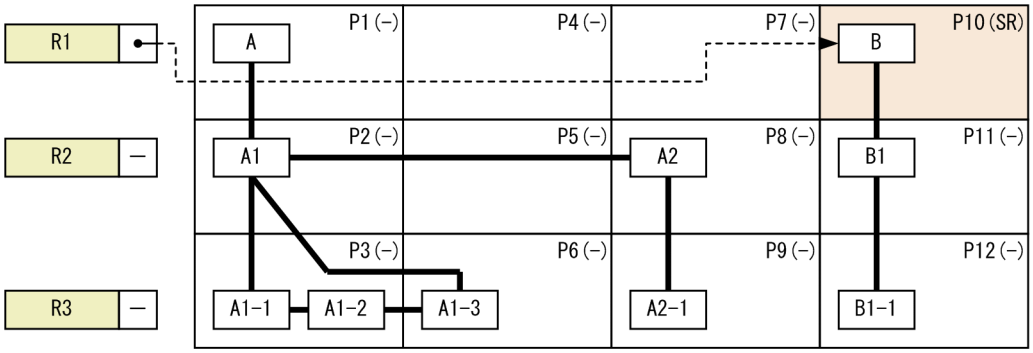
5. 3 階層目のレコード実現値 A2-1 を検索（FETCH）します。

この場合，レコード位置指示子 R3 がレコード実現値 A2-1 を指すので，レコード実現値 A2-1 が格納されているデータページ P9 に排他（SR モード）が掛かります。



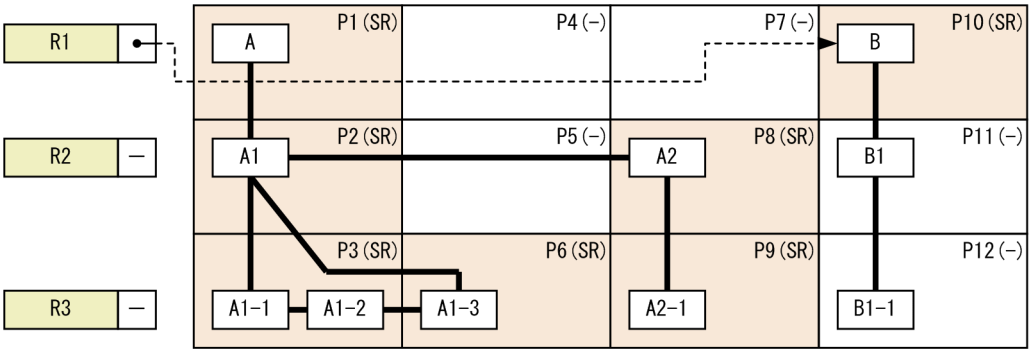
6. ルートレコードのレコード実現値 B を検索（FETCH）します。

この場合，レコード位置指示子 R1 が指すレコード実現値は A から B となるので，データページ P10 に排他（SR モード）が掛かり，データページ P1 の排他（SR モード）は解除されます。同時に，レコード位置指示子 R2 および R3 が空値となるので，それまでレコード位置指示子が指していたレコード実現値を格納しているデータページ P8 および P9 の排他（SR モード）も解除されます。



(2) 例 2 排他自動解除機能を使用しない場合の SDB データベースの階層順検索【4V FMB，SD FMB】

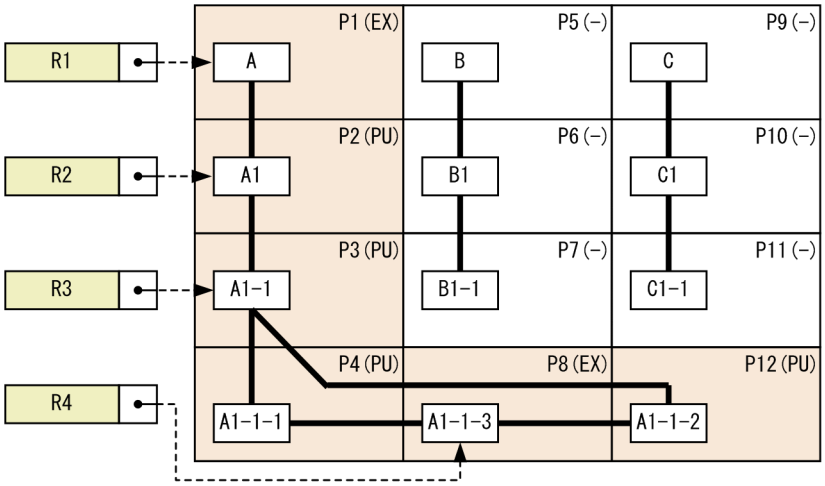
排他自動解除機能を使用しないで「2.9.5(1)例 1 排他自動解除機能を使用する場合の SDB データベースの階層順検索【4V FMB，SD FMB】」の 1.～6.で示した検索（FETCH）を実行すると，一度検索したレコード実現値を格納しているデータページ P1，P2，P3，P6，P8，および P9 の排他（SR モード）は解除されないで，トランザクションが終了するまで保持されます。



(3) 例 3 排他自動解除機能を使用してレコードの格納（STORE）を実行する場合の動作【4V FMB，SD FMB】

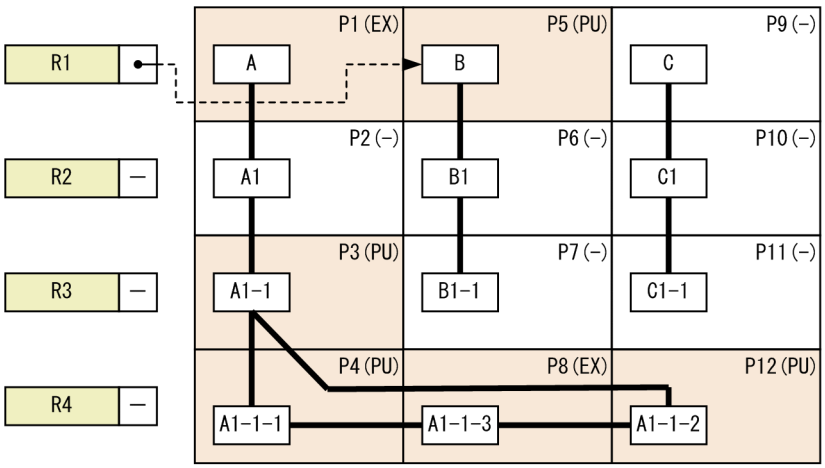
1. 次の順序で SDB データベースを操作する API または DML を実行します。
   a. ルートレコードのレコード実現値 A を検索（FETCH）します。
   b. 2 階層目のレコード実現値 A1 を検索（FETCH）します。
   c. 3 階層目のレコード実現値 A1-1 を検索（FETCH）します。
   d. 4 階層目のレコード実現値 A1-1-3 を格納（STORE）します。

   API または DML を実行すると，次の図のように排他が掛かります。



2. ルートレコードのレコード実現値 B を検索（FETCH）します。

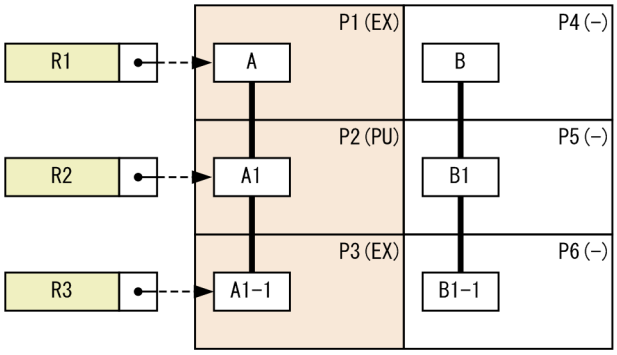
この場合，ルートレコードのレコード実現値 B が格納されているデータページ P5 に排他（PU モード）が掛かり，ルートレコードのレコード実現値 A が格納されているデータページ P1 を除き，更新されていないデータページ P2 の排他（PU モード）だけが解除されます。なお，データページ P3，P4，P12 については，レコード間ポインタなどの制御情報が更新されるため，排他が掛かった状態のままとなります。



(4) 例 4　排他自動解除機能を使用してレコードの更新（MODIFY）を実行する場合の動作【4V FMB，SD FMB】

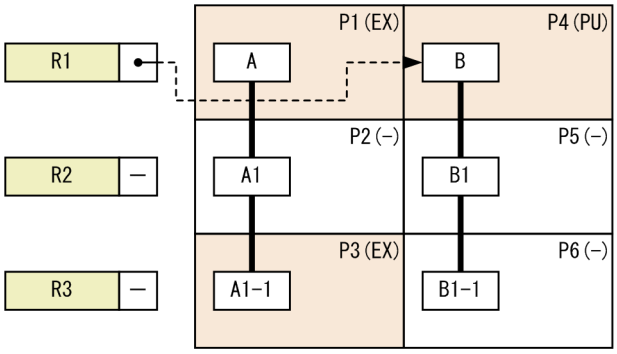
1. 次の順序で SDB データベースを操作する API または DML を実行します。
   a. ルートレコードのレコード実現値 A を検索（FETCH）します。
   b. 2 階層目のレコード実現値 A1 を検索（FETCH）します。
   c. 3 階層目のレコード実現値 A1-1 を検索（FETCH）します。
   d. 3 階層目のレコード実現値 A1-1 を更新（MODIFY）します。

API または DML を実行すると，次の図のように排他が掛かります。



2. ルートレコードのレコード実現値 B を検索（FETCH）します。

この場合，ルートレコードのレコード実現値 B が格納されているデータページ P4 に排他（PU モード）が掛かり，ルートレコードのレコード実現値 A が格納されているデータページ P1 を除き，更新されていないデータページ P2 の排他（PU モード）だけが解除されます。



(5) 例 5　排他自動解除機能を使用してレコードの削除（ERASE）を実行する場合の動作【4V FMB，SD FMB】

1. 次の順序で SDB データベースを操作する API または DML を実行します。

a. ルートレコードのレコード実現値 A を検索（FETCH）します。

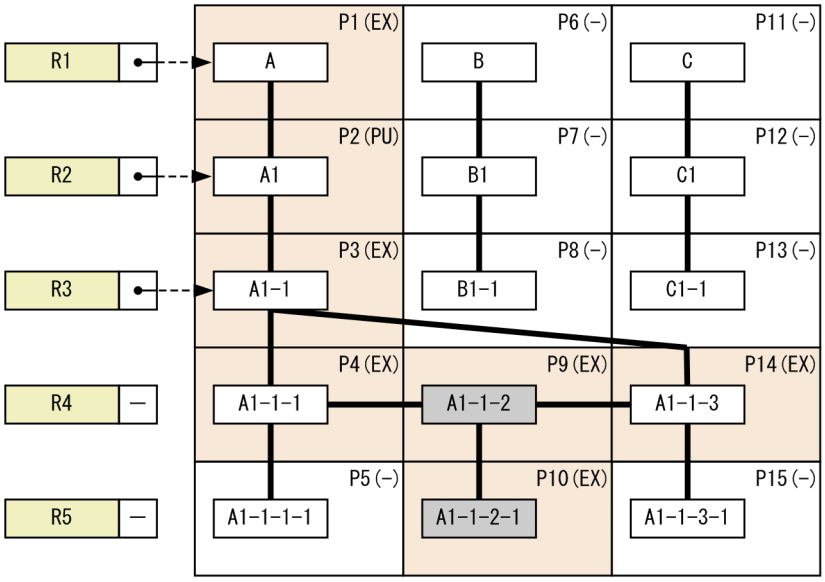
b. 2 階層目のレコード実現値 A1 を検索（FETCH）します。

c. 3 階層目のレコード実現値 A1-1 を検索（FETCH）します。

d. 4 階層目のレコード実現値 A1-1-2 を検索（FETCH）します。

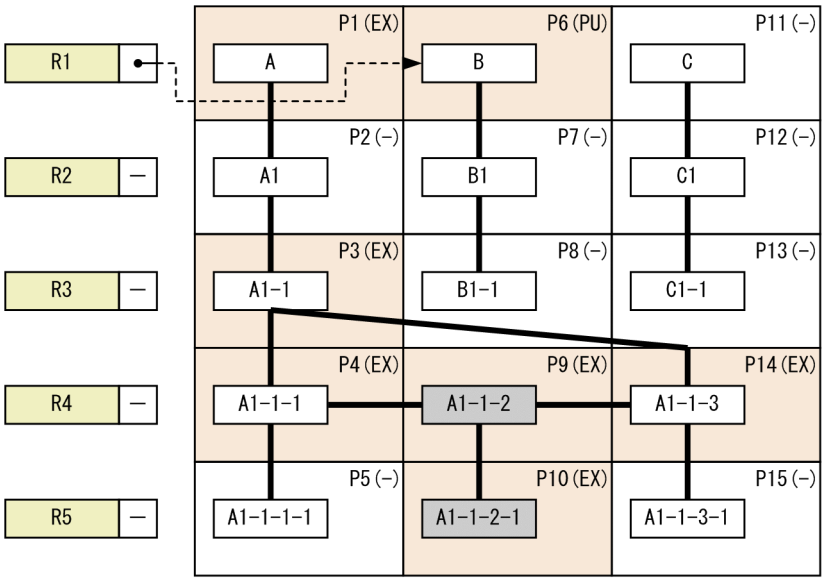
e. 4 階層目のレコード実現値 A1-1-2 を削除（ERASE）します（位置づけしたレコード実現値（その配下の子レコード実現値も含む）を削除します）。

API または DML を実行すると，次の図のように排他が掛かります。



2. ルートレコードのレコード実現値 B を検索（FETCH）します。

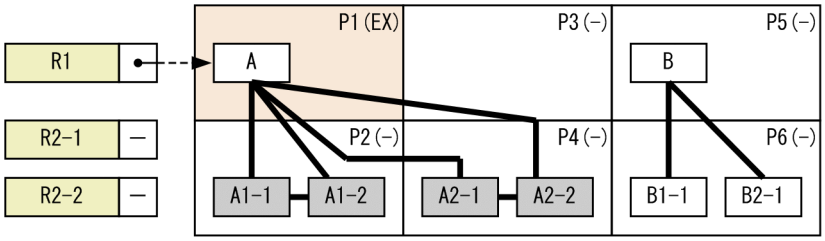
   この場合，ルートレコードのレコード実現値 B が格納されているデータページ P6 に排他（PU）が掛かり，ルートレコードのレコード実現値 A が格納されているデータページ P1 と削除対象のレコード実現値 A1-1-2 の親レコード実現値 A1-1 が格納されているデータページ P3 を除いて，更新されていないデータページ P2 の排他（PU モード）だけが解除されます。なお，データページ P4，P14 については，レコード間ポインタの制御情報が変更されるため，排他が掛かった状態のままとなります。



(6) 例 6　排他自動解除機能を使用して一括削除を実行する場合の動作【4V DAM，4V SAM】

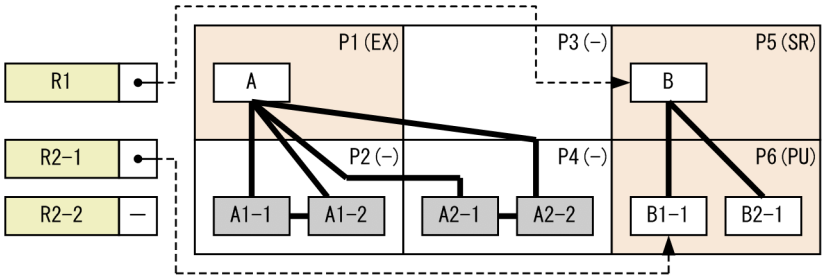
1. 仮想ルートレコードのレコード実現値 A 下のすべてのレコード実現値に対して一括削除を実行します。

   一括削除の結果，仮想ルートレコードのレコード実現値 A が格納されているデータページ P1 に排他（EX モード）が掛かります。



2. 2 階層目のレコード実現値 B1-1 を検索（FETCH）します。

   この場合，仮想ルートレコードのレコード実現値 B が格納されているデータページ P5，2 階層目のレコード実現値 B1-1 が格納されているデータページ P6 にそれぞれ排他（SR モード，PU モード）が掛かりますが，排他確保済みページの中で更新されていないページは存在しないため，排他は解除されません。

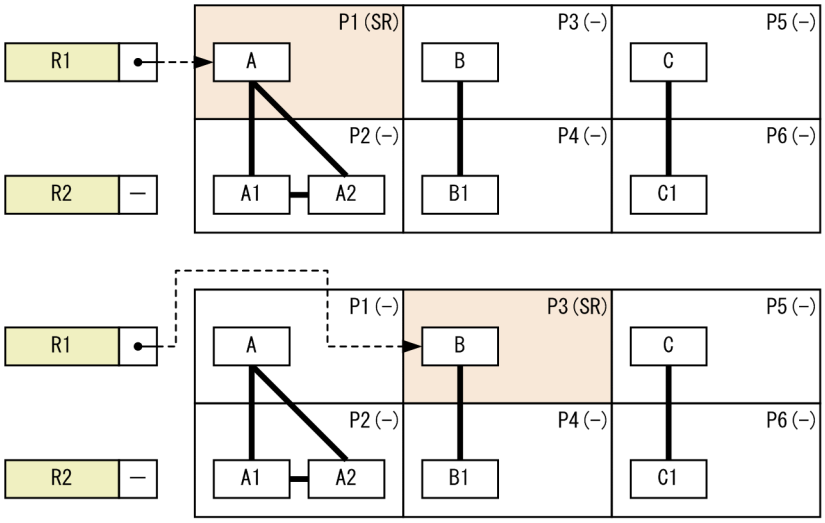


(7) 例 7　構成情報取得を実行する場合の動作【4V AFM】

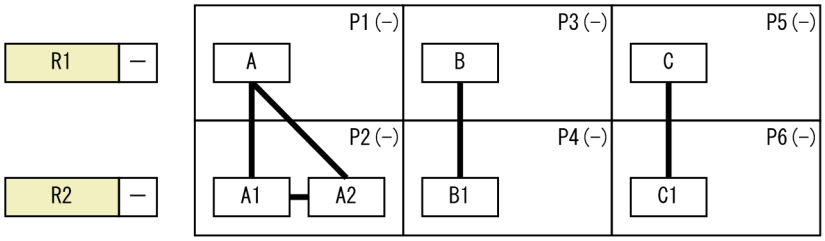
構成情報取得については，排他自動解除機能を適用しない場合の動作がこれまでの API と異なります。

1. 仮想ルートレコードのレコード実現値 A および B 下のレコード実現値の構成情報取得を実行します。

   この場合，仮想ルートレコードのレコード実現値 A から B の順に操作しますが，排他自動解除機能の適用有無に関係なく，排他自動解除機能が適用された場合の動作となります。



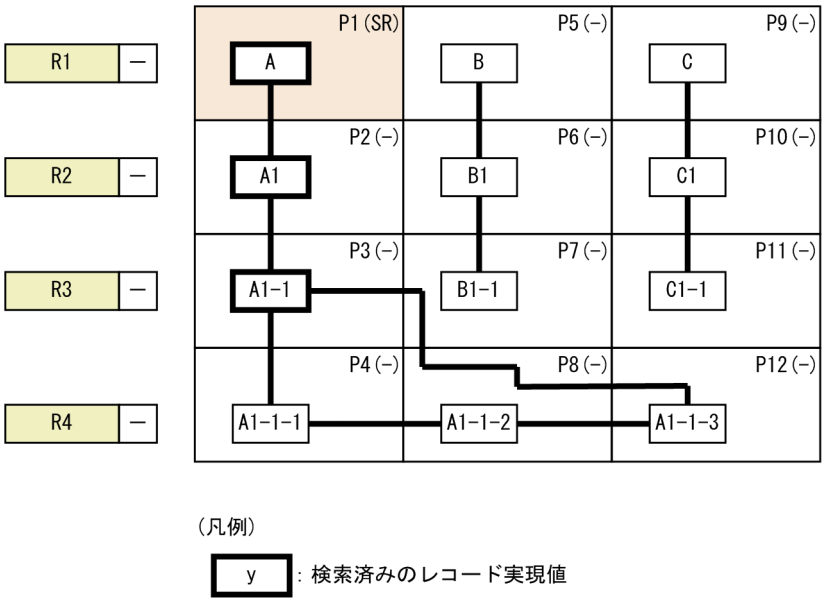
2. API が終了したときには，最後に操作した仮想ルートレコードのレコード実現値 B が格納されているデータページ P3 の排他（SR モード）も解除されます。



(8) 例 8　排他自動解除機能を使用して複数レコードの検索（FETCHDB ALL）を実行する場合の動作【4V FMB】

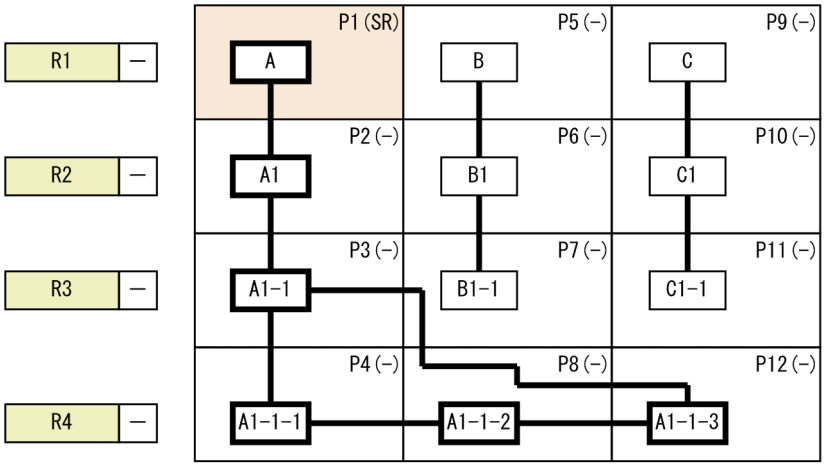
1. 指示コードに'F'を指定して，複数レコードの検索（FETCHDB ALL）を実行することで，ルートレコードのレコード実現値 A，2 階層目のレコード実現値 A1，および 3 階層目のレコード実現値 A1-1 を検索します。

   複数レコードの検索では，ルートレコード実現値が格納されているデータページ以外のページに排他を掛けないため，データページ P1 にだけ排他（SR モード）が掛かります。



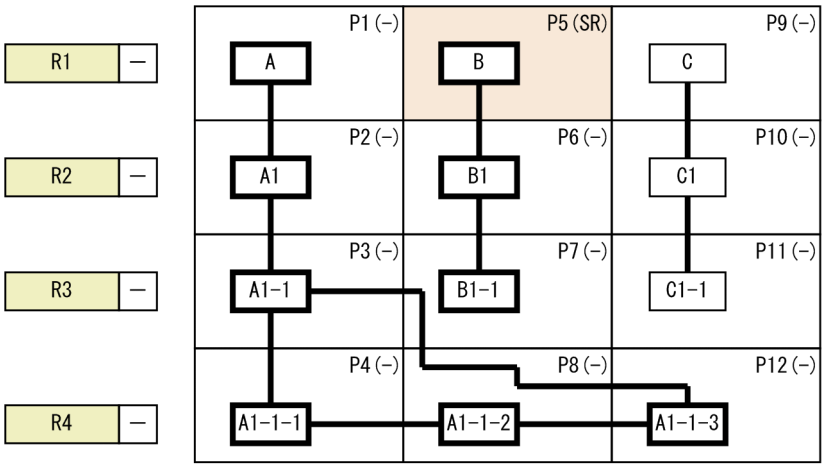
2. 指示コードに'N'を指定して，複数レコードの検索（FETCHDB ALL）を実行することで，4 階層目のレコード実現値 A1-1-1，A1-1-2，A1-1-3 を検索します。

   この場合，ファミリの移動はないため，データページ P1 にだけ排他（SR モード）が掛かった状態のままとなります。



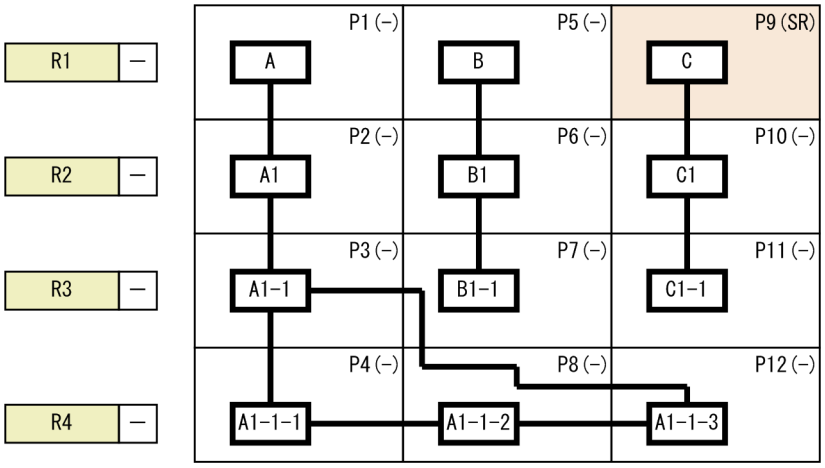
3. 指示コードに'N'を指定して，複数レコードの検索（FETCHDB ALL）でルートレコードのレコード実現値 B，2 階層目のレコード実現値 B1，および 3 階層目のレコード実現値 B1-1 を検索します。

   この場合，ファミリが移動したため，ルートレコード実現値 B が格納されているデータページ P5 にだけ排他（SR モード）が掛かり，データページ P1 の排他は解除されます。



4. 指示コードに'N'を指定して，複数レコードの検索（FETCHDB ALL）でルートレコードのレコード実現値 C，2 階層目のレコード実現値 C1，および 3 階層目のレコード実現値 C1-1 を検索します。

   ファミリが移動したため，ルートレコード実現値 C が格納されているデータページ P9 にだけ排他（SR モード）が掛かり，データページ P5 の排他は解除されます。



なお，排他自動解除機能を使用しないで 1.から 4.に示した検索を実行すると，一度検索したルートレコード実現値を格納しているデータページ P1，P5 および P9 の排他（SR モード）は解除されないで，トランザクション終了まで保持されます。

(9) 排他自動解除機能の使用時に排他が解除されないケース

■4V FMB または 4V AFM の SDB データベースの場合

次に示す場合は，排他自動解除機能を使用したときでも，データの完全性を保証するために排他解除の対象外となります。

- 更新したレコード実現値が格納されている場合（SDB データベースを操作する API の排他モード 1 に'E'（排他モード）または'O'（占有モード）を指定して，検索しただけのレコード実現値が格納されているページは排他解除の対象となります）
- レコード実現値が 1 つでも更新されているファミリのルートレコードのレコード実現値が格納されている場合

なお，4V FMB の SDB データベースで，USER ポインタが設定されていない中間レコード（FIRST ポインタ，LAST ポインタのどちらからも指されていないレコード実現値）を削除する場合，削除対象となるレコード実現値の親レコード実現値は更新されませんが，例外的に排他は解除されません。

■SD FMB の SDB データベースの場合

次に示す場合は，排他自動解除機能を使用したときでも，データの完全性を保証するために排他解除の対象外となります。

- 更新したレコード実現値が格納されている場合（FOR UPDATE オペランドを指定して，検索しただけのレコード実現値が格納されているページは排他解除の対象となります）
- レコード実現値が 1 つでも更新されているファミリのルートレコードのレコード実現値が格納されている場合

なお，中間レコード（FIRST ポインタ，LAST ポインタのどちらからも指されていないレコード実現値）を削除する場合，削除対象となるレコード実現値の親レコード実現値は更新されませんが，例外的に排他は解除されません。

2.9.6 TAM のデータベースの無排他検索機能【4V TAM】

SDB データベースを操作する API の指定と SDB データベース定義の TAMMODE 句の指定によって，ページまたはサブページに排他を掛けないで 4V TAM の SDB データベースのレコードの検索，および構成情報取得を実行できます。この機能を TAM のデータベースの無排他検索機能といいます。

TAM のデータベースの無排他検索機能は，次の指定でレコードの検索，または構成情報取得を実行した場合に適用されます。

- 4V TAM の SDB データベースで，SDB データベース定義の TAMMODE 句を省略するか，または NONPROTECTED RETRIEVE を指定する。

- 個別開始時に指定するSDBデータベースを操作するAPIで，排他モード1に'S'（共用モード）を指定する。

TAMのデータベースの無排他検索機能が適用されると，データページ（ページまたはサブページ）に対して排他を掛けないでレコードの検索，または構成情報取得が実行されます。データページ（ページまたはサブページ）に対して排他を掛けないので，排他自動解除機能は適用されません。

なお，個別開始時に指定するSDBデータベースを操作するAPIで，排他モード1に'N'（無排他モード）を指定する場合は，SDBデータベース定義のTAMMODE句の指定に関係なく，排他モード指定による無排他検索機能が適用されます。

排他モード指定による無排他検索機能については，「2.9.7　排他モード指定による無排他検索機能【4V FMB，4V AFM】」を参照してください。

(1)　TAMのデータベースの無排他検索機能の適用条件

TAMのデータベースの無排他検索機能は，レコードを参照した時点での，データの整合性を重視しない検索で使用するようにしてください。

- TAMのデータベースの無排他検索機能を使用すると，ほかのユーザが更新中（または排他モード1で'E'（排他モード）もしくは'O'（占有モード）を指定中）のデータでも，更新の完了またはトランザクションの終了を待たないで参照できます。一方で，同一トランザクション内で同じレコードを2度検索しても，結果が同じにならないことがあります。また，レコードを更新したユーザがそのトランザクションを取り消すと，データの整合性が取れていない状態のレコードを参照したことになります。
- 終端検知の結果が，ほかのトランザクションとの同時実行タイミングによっては，TAMのデータベースの無排他検索機能を適用しない場合の終端検知の結果とは異なることがあります。

(2)　TAMのデータベースの無排他検索機能で排他を掛けない資源

TAMのデータベースの無排他検索機能で排他を掛けない資源は，レコードを格納するデータページ（ページまたはサブページ）だけです。それ以外の上位の排他資源については，「表2-36　排他制御のモードの組み合わせの例（4V AFMのSDBデータベースの場合）」に従って排他が掛けられます。そのため，更新ログ取得方式がログレスモード指定で，4V TAMのSDBデータベースを更新するトランザクションが実行中の場合は，TAMのデータベースの無排他検索機能が適用される条件を満たしていても，排他待ちになります。

2.9.7　排他モード指定による無排他検索機能【4V FMB，4V AFM】

SDBデータベースを操作するAPIの排他モードの指定によって，ページまたはサブページに排他を掛けないでレコードの検索および構成情報取得を実行できます。この機能を排他モード指定による無排他検索機能といいます。

排他モード指定による無排他検索機能は，次の指定でレコードの検索，または構成情報取得を実行した場合に適用されます。

- 個別開始時に指定する SDB データベースを操作する API で，排他モード 1 に'N'（無排他モード）を指定した場合

排他モード指定による無排他検索機能が適用されると，データページ（ページまたはサブページ）に対して排他を掛けないため，排他自動解除機能は適用されません。

排他モード指定による無排他検索機能は，4V FMB または 4V AFM の SDB データベースの場合に適用されます。

(1) 排他モード指定による無排他検索機能の適用条件

排他モード指定による無排他検索機能は，レコードを参照した時点での，データの整合性を重視しない検索で使用するようにしてください。

(2) 排他モード指定による無排他検索機能で排他を掛けない資源

排他モード指定による無排他検索機能で排他を掛けない資源は，レコードを格納するデータページ（ページまたはサブページ）だけです。それ以外の上位の排他資源については，「表 2-35 排他制御のモードの組み合わせの例（4V FMB の SDB データベースの場合)」，または「表 2-36 排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合)」に従って排他が掛けられます。そのため，更新ログ取得方式がログレスモードで，操作対象データベースを更新するトランザクションが実行中の場合は，排他モード指定による無排他検索機能が適用される条件を満たしていても，排他待ちになります。

(3) 排他モード指定による無排他検索機能を適用する際の留意事項

排他モード指定による無排他検索機能を適用する際の留意事項を次に示します。

- 排他モード指定による無排他検索機能を使用した場合，ほかのユーザが占有しているデータでも，排他解除を待たないで参照できます。一方で，同一トランザクション内で同じレコードを 2 度検索しても，結果が同じにならないことがあります。また，レコードを更新したユーザがそのトランザクションを取り消すと，データの整合性が取れていない状態のレコードを参照したことになります。
- ほかのトランザクションによるデータの更新との同時実行のタイミングによっては，検索が続行できなくなり，ロールバックされることがあります。この場合は，このトランザクションを再実行するか，またはこのトランザクション中の個別開始時に指定する SDB データベースを操作する API で，排他モード 1 に'N'（無排他モード）以外を指定して，トランザクションを実行してください。
- システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定した場合でも，NOWAIT（省略値）を指定した場合と同様の動作となります。
- 子レコードを検索した際に返却される USER ポインタの情報は，親レコード検索時点のものです。
  - 親レコードの検索以降にほかのトランザクションが USER ポインタを更新した場合，実態とは異なる USER ポインタの情報が返却されることがあります。

- この不整合は，USER ポインタを使用した検索を実施した際，検知されます。その場合はロールバックされます。

- ほかのトランザクションとの同時実行タイミングによっては，終端検知の結果が，SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）以外を指定した場合の終端検知の結果とは異なることがあります。

2.9.8　SDB 用 UAP 環境定義の指定による無排他検索機能【SD FMB】

SDB 用 UAP 環境定義の指定によって，ページまたはサブページに排他を掛けないでレコードの検索および位置指示子の位置づけを実行できます。この機能を SDB 用 UAP 環境定義の指定による無排他検索機能といいます。

SDB 用 UAP 環境定義の指定による無排他検索機能は，次の指定をしてレコードの検索，または位置指示子の位置づけを実行した場合に適用されます。

- SDB 用 UAP 環境定義の subschema オペランドの-e オプションに nonprotected を指定し，かつ-a オプションに retrieve を指定した場合

SDB 用 UAP 環境定義の指定による無排他検索機能が適用されると，ページまたはサブページに対して排他を掛けないため，排他自動解除機能は適用されません。

SDB 用 UAP 環境定義の指定による無排他検索機能は，SD FMB の SDB データベースの場合に適用されます。

(1)　SDB 用 UAP 環境定義の指定による無排他検索機能の適用条件

SDB 用 UAP 環境定義の指定による無排他検索機能は，レコードを参照した時点での，データの整合性を重視しない検索で使用するようにしてください。

(2)　SDB 用 UAP 環境定義の指定による無排他検索機能で排他を掛けない資源

SDB 用 UAP 環境定義の指定による無排他検索機能で排他を掛けない資源は，レコードを格納するデータページ（ページまたはサブページ）だけです。それ以外の上位の排他資源については，「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」に従って排他が掛けられます。

(3)　SDB 用 UAP 環境定義の指定による無排他検索機能を適用する際の留意事項

SDB 用 UAP 環境定義の指定による無排他検索機能を適用する際の留意事項を次に示します。

- SDB 用 UAP 環境定義の指定による無排他検索機能を使用した場合，ほかのユーザが占有しているデータでも，排他解除を待たないで参照できます。一方で，同一トランザクション内で同じレコードを 2 度

検索しても，結果が同じにならないことがあります。また，レコードを更新したユーザがそのトランザクションを取り消すと，データの整合性が取れていない状態のレコードを参照したことになります。

- システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定した場合でも，NOWAIT（省略値）を指定した場合と同様の動作となります。
- ほかのトランザクションによるデータの更新との同時実行のタイミングによっては，検索が続行できなくなり，ロールバックされることがあります（システムから SQLCODE に-63005 が返されます）。この場合は，トランザクションを再実行してください。

2.9.9 デッドロック

2 つのトランザクションが 2 つ以上の資源の確保をめぐって互いに相手を待つ状態となり，そこから先へ処理が進まなくなることをデッドロックといいます。

詳細については，マニュアル「HiRDB UAP 開発ガイド」の「デッドロックと回避策」を参照してください。

デッドロックの例を次に示します。

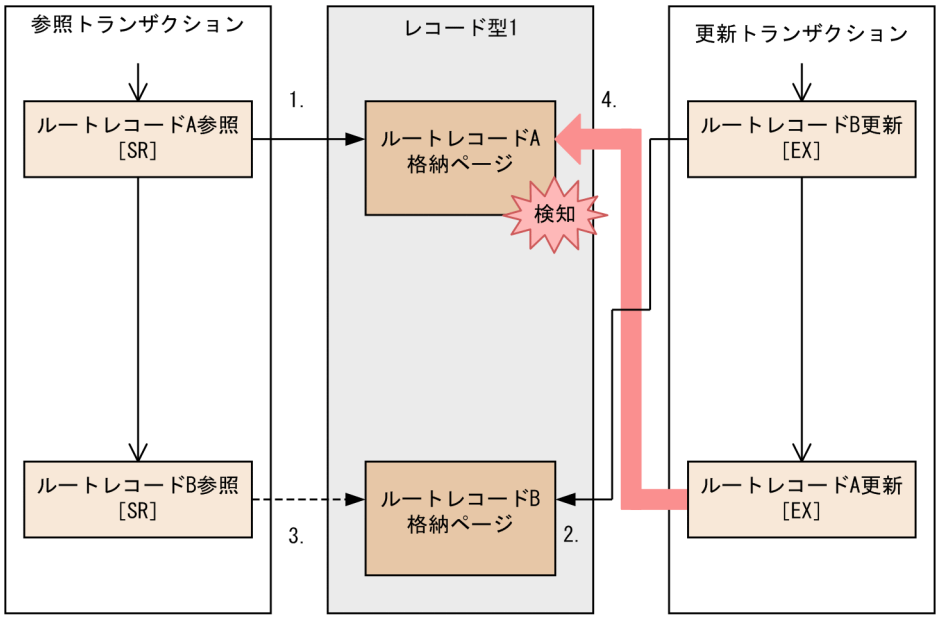
(1) ルートレコードを格納しているページ間のデッドロックの例

4V FMB または SD FMB の SDB データベースでは，2 つのトランザクションが複数のルートレコードに対して同時に逆順にアクセスした場合，デッドロックが発生するおそれがあります。次に示す場合にも同様にデッドロックが発生するおそれがあります。

- 4V AFM の SDB データベースで，異なる仮想ルートレコード下の子レコードを逆順にアクセスした場合
- SD FMB の SDB データベースに対して，二次インデクスを使用し，ファミリが異なる複数の子レコードを逆順にアクセスした場合

複数のルートレコードを逆順にアクセスした場合のデッドロックの例を，次の図に示します。

図 2-93　ルートレコードを格納しているページ間のデッドロックの例



（凡例）

：制御の流れ
：資源の排他
：資源の排他解除待ち
：デッドロックの検知

1.～4.　：資源の排他取得順序
[]　：排他制御モード

[説明]

1. 参照トランザクションがルートレコード A の格納ページに SR モードで排他を掛けて，レコードを参照します。
2. 更新トランザクションがルートレコード B の格納ページに EX モードで排他を掛けて，レコードを更新します。
3. 参照トランザクションがルートレコード B の格納ページに SR モードで排他を掛けようとしますが，更新トランザクションが EX モードで排他を掛けているため，ページの排他解除待ちとなります。
4. 更新トランザクションがルートレコード A の格納ページに EX モードで排他を掛けようとしますが，参照トランザクションが SR モードで排他を掛けているため，ページの排他解除待ちとなり，デッドロックが検知されます。

ここでは，同一のレコード型内の複数のルートレコードに対する例を挙げましたが，レコード型が異なる複数のルートレコードを逆順にアクセスした場合にもデッドロックが発生するおそれがあります。

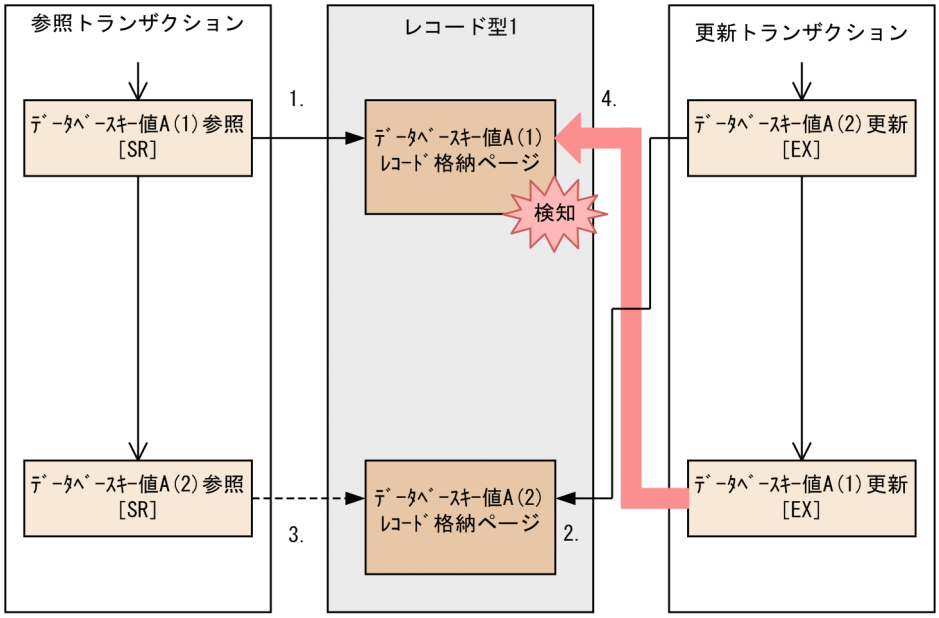
(2) 子レコードを格納しているページ間のデッドロックの例【4V AFM，SD FMB】

4V AFM の SDB データベースでは，2 つのトランザクションが複数のデータベースキー値のレコードを同時に逆順にアクセスした場合，デッドロックが発生するおそれがあります。

SD FMB の SDB データベースでは，二次インデクスを使用して子レコードを検索する場合，子レコードを逆順にアクセスすると，子レコード格納ページ間で，デッドロックが発生するおそれがあります。

デッドロックの例を次の図に示します。

図 2-94　子レコードを格納しているページ間のデッドロックの例



（凡例）

：制御の流れ
：資源の排他
：資源の排他解除待ち
：デッドロックの検知

1.～4.　：資源の排他取得順序
(1)，(2)：一連番号
[]　　：排他制御モード

[説明]

上記の図は，複数のデータベースキー値のレコードを逆順にアクセスした場合のデッドロックの例です。

1. 参照トランザクションがデータベースキー値 A(1)のレコードの格納ページに SR モードで排他を掛けて，レコードを参照します。
2. 更新トランザクションがデータベースキー値 A(2)のレコードの格納ページに EX モードで排他を掛けて，レコードを更新します。

3. 参照トランザクションがデータベースキー値 A(2)のレコードの格納ページに SR モードで排他を掛けようとしますが，更新トランザクションが EX モードで排他を掛けているため，ページの排他解除待ちとなります。
4. 更新トランザクションがデータベースキー値 A(1)のレコードの格納ページに EX モードで排他を掛けようとしますが，参照トランザクションが SR モードで排他を掛けているため，ページの排他解除待ちとなり，デッドロックが検知されます。

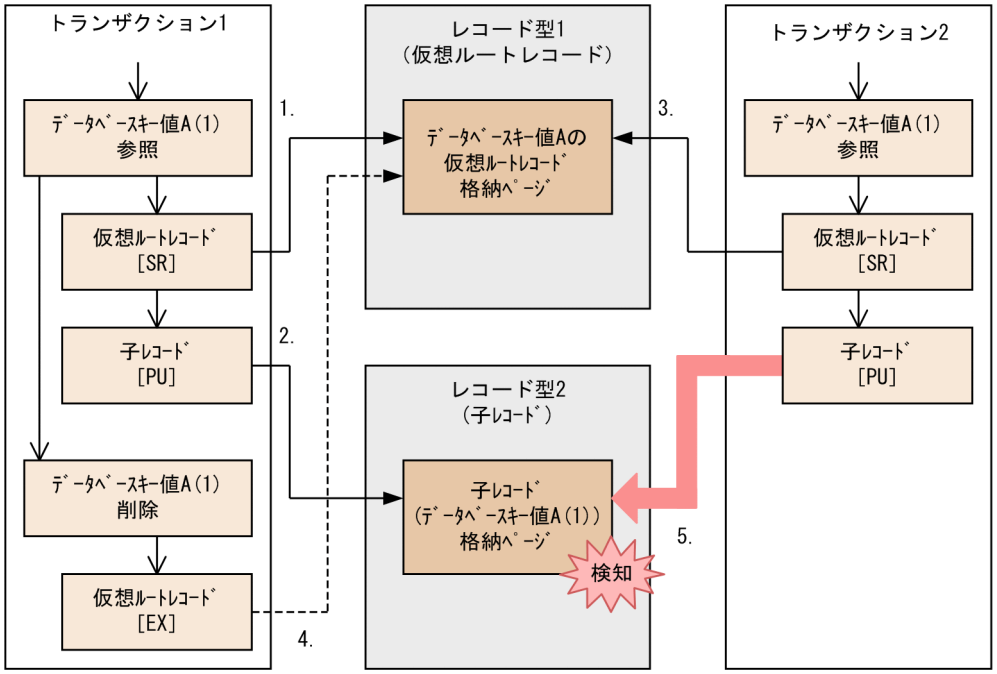
ここでは，同一のレコード型内の複数のデータベースキー値に対する例を挙げましたが，レコード型が異なる複数のデータベースキー値のレコードを逆順にアクセスした場合もデッドロックが発生するおそれがあります。

なお，4V AFM の SDB データベースのレコードの格納時には，格納位置を決めるために発生する検索の方向を指定する検索コードに従って子レコードの挿入位置を検索します。HiRDB/SD は指定された検索コードによってレコード間を NEXT 方向または PRIOR 方向に，子レコードが格納されているページに順次アクセスする場合があります。このように直接子レコードを操作しなくても，上記のデッドロックが発生するおそれがあるため，検索方向にも注意してください。

(3) 仮想ルートレコードを格納しているページと子レコードを格納しているページ間のデッドロックの例【4V AFM】

4V AFM の SDB データベースに対するアクセスでは，HiRDB/SD が内部的に仮想ルートレコードにアクセスすることがあるため，仮想ルートレコードを格納しているページと子レコードを格納しているページとの間でデッドロックが発生するおそれがあります。デッドロックの例を次の図に示します。

図 2-95　仮想ルートレコードを格納しているページと子レコードを格納しているページ間のデッドロックの例



（凡例）

：制御の流れ
：資源の排他
：資源の排他解除待ち
：デッドロックの検知

1.～5.　：資源の排他取得順序
(1)　：一連番号
[]　：排他制御モード

[説明]

上記の図は，仮想ルートレコードを格納しているページと子レコードを格納しているページ間のデッドロックの例です。

1. トランザクション 1 が，データベースキー値 A(1)を参照するため，仮想ルートレコードの格納ページに SR モードで排他を掛けます。
2. トランザクション 1 が，データベースキー値 A(1)を参照するため，子レコードの格納ページに PU モードで排他を掛けて，子レコードを参照します。
3. トランザクション 2 が，データベースキー値 A(1)を参照するため，仮想ルートレコードの格納ページに SR モードで排他を掛けます。
4. トランザクション 1 が，データベースキー値 A(1)を削除するため，仮想ルートレコードの格納ページに EX モードで排他を掛けようとしますが，トランザクション 2 が SR モードで排他を掛けているため，ページの排他解除待ちとなります。

5. トランザクション 2 が，データベースキー値 A(1)を参照するため，子レコードの格納ページに PU モードで排他を掛けようとしますが，トランザクション 1 が PU モードで排他を掛けているため，ページの排他解除待ちとなり，デッドロックが検知されます。

ここでは，レコードの削除とレコードの検索が競合した場合の例を挙げましたが，このほかにも次の SDB データベースを操作する API の競合時にも同じようにデッドロックが発生するおそれがあります。

- レコードの削除とレコードの格納
- レコードの削除とレコードの削除

また，異なるレコード型の間（例えば，4V FMB の SDB データベースのルートレコードを格納しているページと，4V AFM の SDB データベースの子レコードを格納しているページの間）でも，同じようにデッドロックが発生するおそれがあります。

(4) ルートレコードを格納しているページと子レコードを格納しているページ間のデッドロックの例【4V FMB，SD FMB】

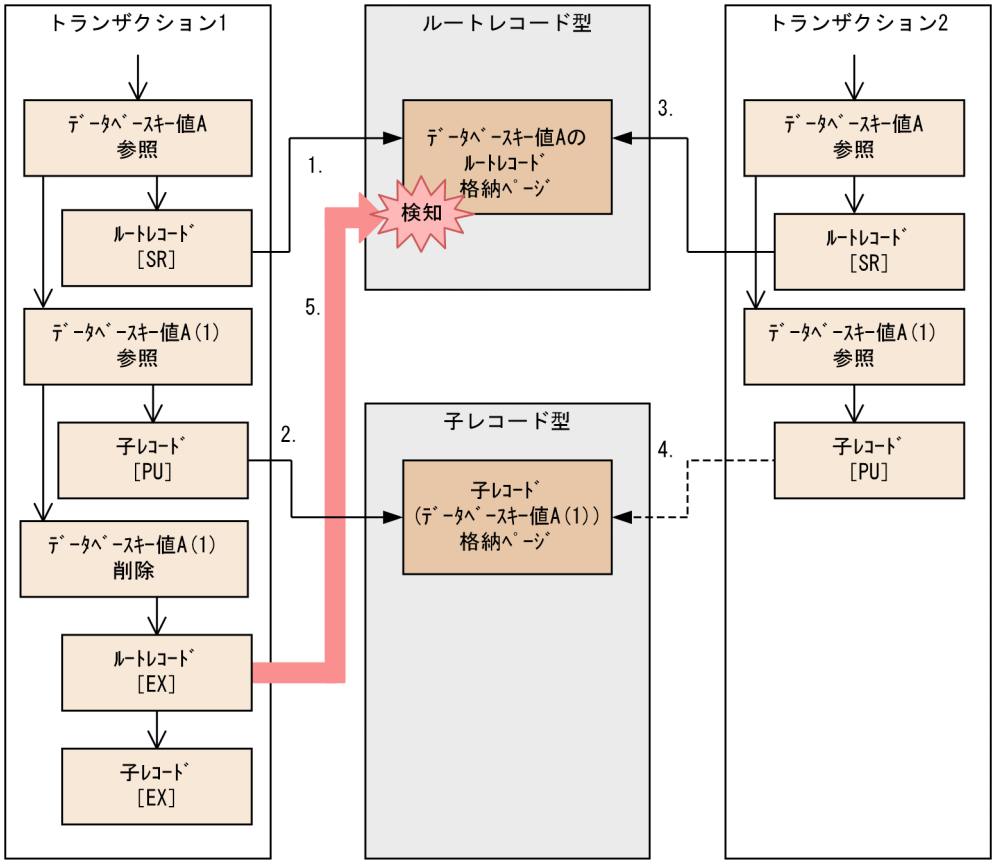
次のどちらかの条件で，ルートレコードの検索またはルートレコードの位置指示子の位置づけを実行すると，その後の子レコードの更新，格納，または削除の際，ルートレコード格納ページと子レコード格納ページとの間でデッドロックが発生するおそれがあります。

- SDB データベース種別が 4V FMB の SDB データベースで，かつ排他オプションが占有モード以外である。
- SDB データベース種別が SD FMB の SDB データベースで，かつ SDB 用 UAP 環境定義の subschema オペランドの-p オプションに shareroot を指定している。

また，SD FMB の SDB データベースでは，二次インデクスを使用して子レコードを検索する場合，子レコードを逆順にアクセスするときも，ルートレコード格納ページと子レコード格納ページ間で，デッドロックが発生するおそれがあります。

デッドロックの例を次の図に示します。

図 2-96　ルートレコードを格納しているページと子レコードを格納しているページ間のデッドロックの例



（凡例）

：資源の排他

：資源の排他解除待ち

：デッドロックの検知

1. ～5.　：資源の排他取得順序

(1)　：一連番号

[]　：排他制御モード

[説明]

上記の図は，ルートレコードを格納しているページと子レコードを格納しているページ間のデッドロックの例です。

1. トランザクション 1 が，データベースキー値 A を参照するため，ルートレコードの格納ページに SR モードで排他を掛けて，ルートレコードを参照します。
2. トランザクション 1 が，データベースキー値 A(1)を参照するため，子レコードの格納ページに PU モードで排他を掛けて，子レコードを参照します。
3. トランザクション 2 が，データベースキー値 A を参照するため，ルートレコードの格納ページに SR モードで排他を掛けて，ルートレコードを参照します。

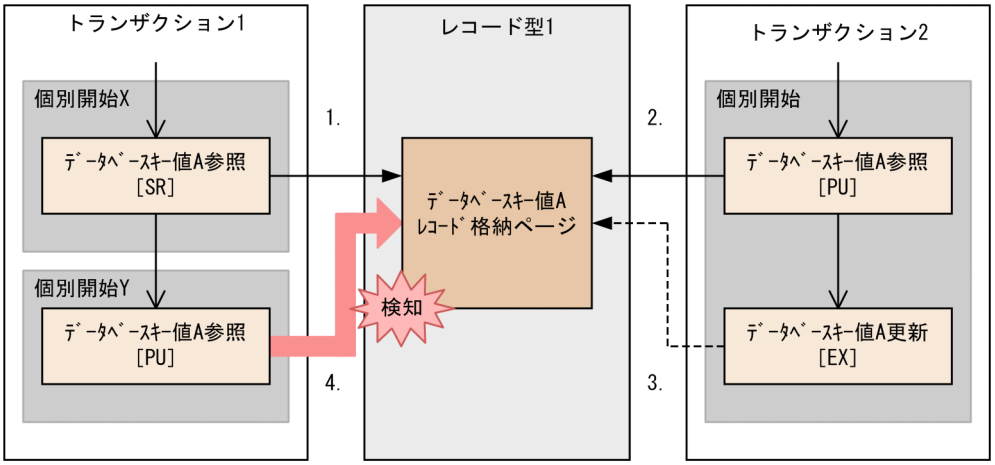
4. トランザクション 2 が，データベースキー値 A(1)を参照するため，子レコードの格納ページに PU モードで排他を掛けようとしますが，トランザクション 1 が PU モードで排他を掛けているため，ページの排他解除待ちとなります。

5. トランザクション 1 が，データベースキー値 A(1)を削除するため，ルートレコードの格納ページに EX モードで排他を掛けようとしますが，トランザクション 2 が SR モードで排他を掛けているため，ページの排他解除待ちとなり，デッドロックが検知されます。

(5) 排他モード 1 の指定が異なる検索が混在するトランザクションと，ほかのトランザクションが同じレコードに対してアクセスした場合のデッドロックの例【4V FMB，4V AFM】

4V FMB または 4V AFM の SDB データベースでは，SDB データベースを操作する API の，排他モード 1 の指定が異なる検索が混在するトランザクションと，ほかのトランザクションが同じレコードに対してアクセスした場合，デッドロックが発生するおそれがあります。デッドロックの例を次の図に示します。

図 2-97 排他モード 1 の指定が異なる検索が混在するトランザクションと，ほかのトランザクションが同じレコード（同じページ）に対してアクセスした場合の，デッドロックの例



（凡例）

：制御の流れ
：資源の排他
：資源の排他解除待ち
：デッドロックの検知

1. ～4. ：資源の排他取得順序
[] ：排他制御モード

[説明]

上記の図は，排他モード 1 の指定が異なる検索が混在するトランザクションと，ほかのトランザクションが同じレコード（同じページ）に対してアクセスした場合の，デッドロックの例です。

1. トランザクション 1（個別開始 X）が，データベースキー値 A を参照するため，レコードの格納ページに SR モードで排他を掛けます。

2. トランザクション 1（個別開始 X）が，個別終了を行います（個別終了処理では排他制御は行いません）。
3. トランザクション 2 が，データベースキー値 A を参照するため，レコードの格納ページに PU モードで排他を掛けます。
4. トランザクション 2 が，データベースキー値 A を更新するため，レコードの格納ページに EX モードで排他を掛けようとしますが，トランザクション 1（個別開始 X）が SR モードで排他を掛けているため，ページの排他解除待ちとなります。
5. トランザクション 1（個別開始 Y）が，データベースキー値 A を参照するため，レコードの格納ページに PU モードで排他を掛けようとしますが，トランザクション 2 が PU モードで排他を掛けているため，ページの排他解除待ちとなり，デッドロックが検知されます。

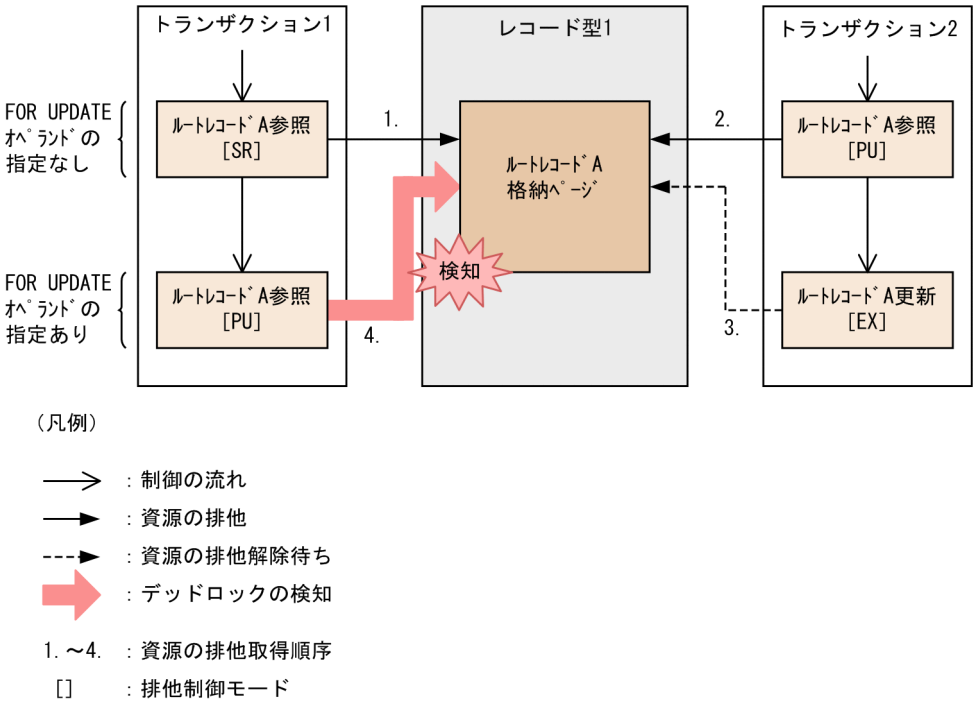
ここでは，同じページに格納された同じレコードに対するアクセスが競合した場合の例を挙げましたが，このほかに次の場合にも，同じようにデッドロックが発生するおそれがあります。

- 同じページに格納された異なるレコードに対するアクセス
- 異なるページ間でのアクセス

(6) FOR UPDATE オペランドの指定の有無が異なる検索が混在するトランザクションと，ほかのトランザクションが同じレコードに対してアクセスした場合のデッドロックの例【SD FMB】

SD FMB の SDB データベースでは，FOR UPDATE オペランドの指定の有無が異なる検索が混在するトランザクションと，ほかのトランザクションが，同じレコードに対してアクセスをすると，デッドロックが発生するおそれがあります。デッドロックの例を次の図に示します。

図 2-98　FOR UPDATE オペランドの指定の有無が異なる検索が混在するトランザクションと，ほかのトランザクションが同じレコード（同じページ）に対してアクセスした場合の，デッドロックの例



（凡例）

：制御の流れ
：資源の排他
：資源の排他解除待ち
：デッドロックの検知
1. ～4.　：資源の排他取得順序
[]　：排他制御モード

[説明]

上記の図は，FOR UPDATE オペランドの指定の有無が異なる検索が混在するトランザクションと，ほかのトランザクションが，同じレコード（同じページ）に対してアクセスした場合の，デッドロックの例です。

1. トランザクション 1 が，参照目的（FOR UPDATE オペランドの指定なし）でルートレコード A を参照するため，レコードの格納ページに SR モードで排他を掛けます。
2. トランザクション 2 が，更新目的（FOR UPDATE オペランドの指定あり）でルートレコード A を参照するため，レコードの格納ページに PU モードで排他を掛けます。
3. トランザクション 1 が，更新目的（FOR UPDATE オペランドの指定あり）でルートレコード A を参照するため，レコードの格納ページに PU モードで排他を掛けようとしますが，トランザクション 2 が PU モードで排他を掛けているため，ページの排他解除待ちとなります。
4. トランザクション 2 が，ルートレコード A を更新するため，レコードの格納ページに EX モードで排他を掛けようとしますが，トランザクション 1 が SR モードで排他を掛けているため，ページの排他解除待ちとなり，デッドロックが検知されます。

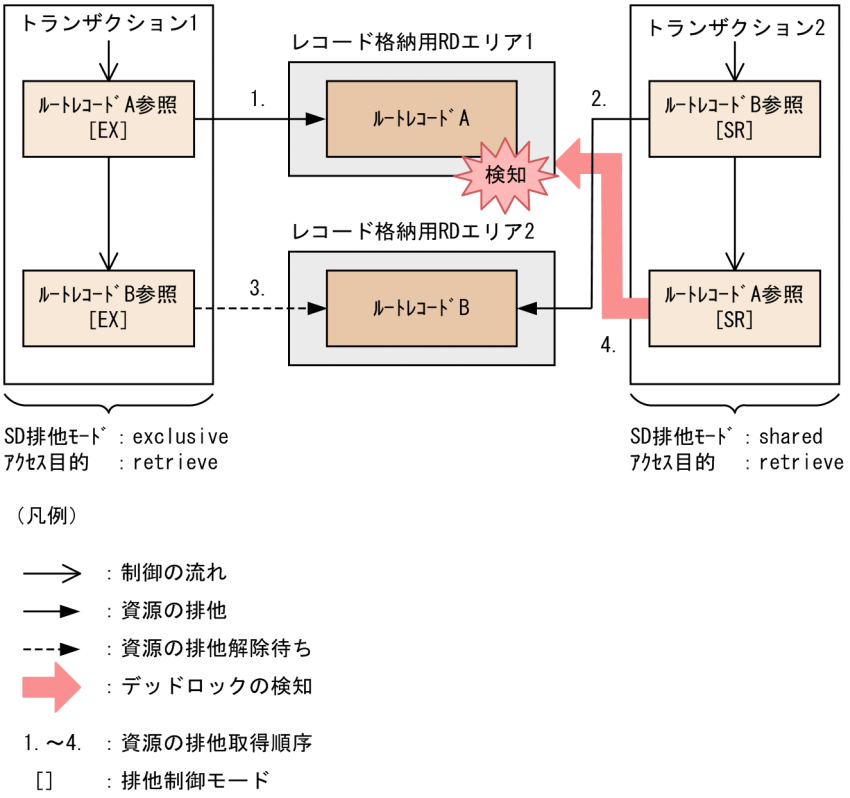
ここでは，同じページに格納された同じレコードに対するアクセスが競合した場合の例を挙げましたが，このほかに次の場合にも，同じようにデッドロックが発生するおそれがあります。

- 同じページに格納された異なるレコードに対するアクセス
- 異なるページ間でのアクセス

(7)　レコード格納用 RD エリア間のデッドロックの例【SD FMB】

SD FMB の SDB データベースでは，「表 2-38　同一レコード格納用 RD エリアに対する，SDB データベースへのアクセスの同時実行可否（SD FMB の SDB データベースの場合）」で説明しているように，レコード格納用 RD エリアの排他によって待ち状態になることがあります。そのため，レコード格納用 RD エリアに PR モード，PU モード，または EX モードで排他を掛けるトランザクションと，ほかのトランザクションが，複数のレコード格納用 RD エリアに対して同時に逆順にアクセスした場合，デッドロックが発生するおそれがあります。デッドロックの例を次の図に示します。

図 2-99　レコード格納用 RD エリア間のデッドロックの例



（凡例）

：制御の流れ
：資源の排他
：資源の排他解除待ち
：デッドロックの検知

1.～4.　：資源の排他取得順序
[]　：排他制御モード

[説明]

上記の図は，複数のレコード格納用 RD エリアに逆順にアクセスした場合の，レコード格納用 RD エリア間のデッドロックの例です。

1. トランザクション 1（SD 排他モード：exclusive，アクセス目的：retrieve）が，ルートレコード A を参照するため，レコード格納用 RD エリア 1 に EX モードで排他を掛けます。
2. トランザクション 2（SD 排他モード：shared，アクセス目的：retrieve）が，ルートレコード B を参照するため，レコード格納用 RD エリア 2 に SR モードで排他を掛けます。
3. トランザクション 1（SD 排他モード：exclusive，アクセス目的：retrieve）が，ルートレコード B を参照するため，レコード格納用 RD エリア 2 に EX モードで排他を掛けようとしますが，トランザクション 2 が SR モードで排他を掛けているため，レコード格納用 RD エリアの排他解除待ちとなります。

4. トランザクション 2（SD 排他モード：shared，アクセス目的：retrieve）が，ルートレコード A を参照するため，レコード格納用 RD エリア 1 に SR モードで排他を掛けようとしますが，トランザクション 1 が EX モードで排他を掛けているため，レコード格納用 RD エリアの排他解除待ちとなり，デッドロックが検知されます。

(8) デッドロックプライオリティ値による排他制御

デッドロックが発生した場合，どのトランザクションをエラーとするかをデッドロックプライオリティ値で制御できます。

この機能を使用する場合，システム共通定義の pd_deadlock_priority_use オペランドに Y を指定してください。クライアント環境定義などからプログラム単位に HiRDB/SD がデッドロック優先順位を決定します。

デッドロックプライオリティ値が小さい方が処理の優先度が高く，値が大きくなるほどエラーになってロールバックされる可能性が高くなります。値が同じときは，後発のトランザクションの方がエラーとなって暗黙的にロールバックされます。

なお，この機能を使用しない場合は，後発のトランザクションがエラーとなり，暗黙的にロールバックされます。

(9) デッドロックの回避策

デッドロックは排他の範囲を広くすることで発生頻度を低減できますが，同時実行性は低下します。排他の範囲を狭くすると同時実行性は向上しますが，デッドロックの発生率が増加します。

デッドロックを回避するために，例えば次のような対策があります。

- 2 つ以上のレコード型のレコードにアクセスする場合，または SD FMB の SDB データベースで 2 つ以上のレコード格納用 RD エリアにアクセスする場合，アクセス順序を統一します。また，A，B の順でアクセスした場合，再度 A をアクセスしないようにします（A の値は保存しておきます）。
- SD FMB の SDB データベースの場合で，2 つ以上のレコード格納用 RD エリアにアクセスするときは，SDB 用 UAP 環境定義の subschema オペランドの-e オプション（SD 排他モード）に shared を指定します。
- 2 つ以上のルートレコード（4V AFM の SDB データベースの場合，データベースキー値のレコード）にアクセスする場合，アクセス順序を統一します。また，A→B の順でアクセスしたら，再度 A はアクセスしないようにします（A の値は保存しておきます）。

  検索の方向※を指定する検索コードに，NEXT 方向の検索と PRIOR 方向の検索が混在している場合，アクセス順序が統一されていないため，混在しないように指定します。

  注※

  レコードの検索（FETCH）では，検索する方向のことです。レコードの格納（STORE）では，格納位置を決定するための検索の方向のことです。

- 検索トランザクションの場合，排他自動解除機能を使用すると，排他保持期間を短くできるため，排他待ちの発生を低減できます。

  排他自動解除機能については，「2.9.4　排他自動解除機能」を参照してください。
- SDB データベース格納定義で SUBPAGE NUMBER 句を指定し，サブページ分割機能を適用します。サブページ分割をすると，複数レコードを異なる排他資源（サブページ）に格納できるため，排他待ちの発生を低減できます。
- 検索時の SDB データベースを操作する API の，排他モード 1 の指定を'O'（占有モード）に変更します。同時実行性は低下しますが，SDB データベース定義での SUBPAGE NUMBER 句の指定の変更や SDB データベースを操作する API の実行順序の変更をすることなく，デッドロックの発生率を低下させることができます。
- SD FMB の SDB データベースの場合，検索時に FOR UPDATE オペランドを指定します。同時実行性は低下しますが，SDB データベース定義での SUBPAGE NUMBER 句の指定の変更や，DML の実行順序の変更をしなくても，デッドロックの発生率を低下させることができます。
- 業務要件に適合する場合，次のように無排他検索を適用してください。これによって，SDB データベースを操作する API または DML の実行順序を変更することなく，デッドロックの発生率を低下させることができます。
  - SDB データベースを操作する API の排他モード 1 の指定値を'N'（無排他モード）に変更する。
  - 操作対象が 4V TAM の SDB データベースの場合，TAM のデータベースの無排他検索機能を適用する。
  - 操作対象が SD FMB の SDB データベースの場合，SDB 用 UAP 環境定義の subschema オペランドの-e オプション（SD 排他モード）に nonprotected を指定する。
- 子レコードを更新，格納，または削除する場合，ルートレコードの検索では次のように指定します。
  - 操作対象が 4V FMB の SDB データベースの場合，SDB データベースを操作する API の排他モード 1 の指定値を'O'（占有モード）に変更する。
  - 操作対象が SD FMB の SDB データベースの場合，SDB 用 UAP 環境定義の subschema オペランドの-p オプションを指定しない。

デッドロックが発生する原因と対策を次の表に示します。

表 2-43　デッドロックが発生する原因と対策

排他資源	発生原因	対策
レコード型とレコード型	レコード型のアクセス順序の不定	**回避方法** UAP のアクセス順序を統一する。 **対処** デッドロック発生後に UAP を再実行する。
レコード格納用 RD エリア（SD FMB の SDB データベースの場合）	レコード格納用 RD エリアのアクセス順序の不定	**回避方法** • UAP のアクセス順序を統一する。

排他資源		発生原因	対策
			• SD 排他モードの指定を shared に変更する。 **対処** デッドロック発生後に UAP を再実行する。
ページ	複数ページ	• 4V FMB または SD FMB の SDB データベースのルートレコードアクセス順序の不定 • 4V AFM の SDB データベースの同一データベースキー値へのアクセス順序の不定	**回避方法（4V FMB または 4V AFM の SDB データベースの場合）** • UAP のアクセス順序を統一する。 • 排他自動解除機能を適用する。 • 無排他検索機能を適用する。 **回避方法（SD FMB の SDB データベースの場合）** • UAP のアクセス順序を統一する。 • 排他自動解除機能を適用する。 • 無排他検索機能を適用する。 **対処** デッドロック発生後に UAP を再実行する。
		子レコードの更新，格納，削除	**回避方法（4V FMB の SDB データベースの場合）** ルートレコードを検索する場合，SDB データベースを操作する API の，排他モード 1 の指定を'O'（占有モード）にする。 **回避方法（SD FMB の SDB データベースの場合）** SDB 用 UAP 環境定義の subschema オペランドの-p オプションを指定しない。 **対処** デッドロック発生後に UAP を再実行する。
		SD FMB の SDB データベースの，二次インデクスを使用した子レコードの検索	**回避方法** • UAP のアクセス順序を統一する。 • 排他自動解除機能を適用する。 • 無排他検索機能を適用する。 **対処** デッドロック発生後に UAP を再実行する。
	単一ページ※1	**■4V FMB または 4V AFM の SDB データベースの場合** 同一トランザクション内で，同一ページに対する検索を，次の順序で実行※2 する。 1. SDB データベースを操作する API の排他モード 1 に'S'（共用モード）を指定する。 2. SDB データベースを操作する API の排他モード 1 に'E'（排他モード）を指定する。	**回避方法** • 排他自動解除機能を適用する。 • 無排他検索機能を適用する。 • SDB データベースを操作する API の排他モード 1 の指定を'O'（占有モード）に変更する。 **対処** デッドロック発生後に UAP を再実行する。

排他資源		発生原因	対策
		■SD FMB の SDB データベースの場合 同一トランザクション内で，同一ページに対する検索を，次の順序で実行する。 1. FOR UPDATE オペランドを指定しないで検索する。 2. FOR UPDATE オペランドを指定して検索する。	回避方法 • 排他自動解除機能を適用する。 • FOR UPDATE オペランドを指定する。 • 無排他検索機能を適用する。 対処 デッドロック発生後に UAP を再実行する。

注※1

複数ページでも，同様のデッドロックが発生することがあります。

注※2

1 トランザクション中で複数の個別開始をすることで，実行できます。

デッドロックが発生すると，自動的に HiRDB/SD がデッドロックとなったトランザクションのどれかをエラーにしてロールバックさせます。デッドロックによって暗黙的にロールバックされた UAP は，ROLLBACK 文，または DISCONNECT 文でトランザクションを終了させないと，SDB データベースを操作する API を実行してもエラーとなります。また，OLTP 環境で X/Open に従ったアプリケーションプログラムをクライアントとした場合に，実行したアプリケーションプログラムがデッドロックとなったときもトランザクションの終了が必要です。

なお，デッドロック発生時にデッドロック情報を出力したい場合，システム共通定義またはユニット制御情報定義の pd_lck_deadlock_info オペランドに Y を指定する必要があります。pd_lck_deadlock_info オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

2.9.10　コミットしていない削除データの排他制御

次のどちらかの操作をする場合，SDB データベースを操作する API または DML（ERASE または MODIFY）でレコードを削除または更新するほかのトランザクションと競合することがあります。このとき，インデクス上で削除レコードのインデクスキー，または更新レコードの更新前インデクスキーを読み飛ばします。そのため，通常は，削除レコードまたは更新レコードを検索できません。

- SDB データベースを操作する API または DML（FETCH，FIND，または FETCHDB ALL）で，4V FMB または SD FMB の SDB データベースのルートレコードを検索する
- SDB データベースを操作する DML（FETCH，または FIND）で，二次インデクスを使用し，SD FMB の SDB データベースの子レコードを検索する

しかし，レコードを削除または更新したトランザクションがロールバックした場合，そのレコードの内容が回復されるため，検索トランザクションでは，結果的にそのレコードを読み飛ばしたことになります。

この問題は，検索トランザクションで削除中のレコードのインデクスキー，または更新中のレコードの更新前インデクスキーを参照したときに，レコード格納ページ（コミットしていない削除データ）に排他を掛けることで回避できます。

(1) 適用基準

次のどちらかの場合に，コミットしていない削除データに排他を掛けることを推奨します。それ以外の場合は，コミットしていない削除データに排他を掛けない運用を推奨します。

- 4V FMB または SD FMB の SDB データベースに対して，次の SDB データベースを操作する API または DML を同時実行する可能性があり，検索トランザクションで削除中または更新中のルートレコードを読み飛ばすことができない業務がある場合
  - ルートレコードを削除対象とするレコードの削除（ERASE），または二次インデクスキーを更新対象とするルートレコードの更新（MODIFY）
  - レコードの検索（FETCH），位置指示子の位置づけ（FIND），または複数レコードの検索（FETCHDB ALL）
- SD FMB の SDB データベースに対して，次の DML を同時実行する可能性があり，検索トランザクションで削除中または更新中の子レコードを読み飛ばすことができない業務がある場合
  - 子レコードを削除対象とするレコードの削除（ERASE），または二次インデクスキーを更新対象とする子レコードの更新（MODIFY）
  - 二次インデクスを使用して子レコードを検索対象とするレコードの検索（FETCH），または二次インデクスを使用して子レコードを走査対象とする位置指示子の位置づけ（FIND）

(2) 指定方法

システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定すると，コミットしていない削除データに排他を掛けることができます。詳細については，マニュアル「HiRDB システム定義」の pd_lock_uncommited_delete_data オペランドの説明を参照してください。
pd_lock_uncommited_delete_data オペランドの説明を参照する際の注意点を次に示します。

- 「DELETE 文」を「SDB データベースを操作する API（ERASE）によるレコードの削除」，または「DML（ERASE）によるレコードの削除」に読み替えてください。
- 「UPDATE 文」を「SDB データベースを操作する API（MODIFY）によるレコードの更新」，または「DML（MODIFY）によるレコードの更新」に読み替えてください。
- このオペランドの指定に関係なく，「一意性制約」に関しては「2.9.11　重複するインデクスキー値を格納する際の実行中トランザクション終了待ち合わせ【4V FMB，SD FMB】」を参照してください。

なお，HiRDB/SD が内部的に発行する，4V AFM の SDB データベースの仮想ルートレコードの検索では，このオペランドの指定値に関係なく，WAIT が指定されたものとして動作します。このオペランドに NOWAIT（省略値）を指定して 4V AFM の SDB データベースを操作する場合，マニュアル「HiRDB システム定義」の pd_lock_uncommited_delete_data オペランドの説明に記載されている HiRDB ファイルの使用可能サイズの制限はありません。

(3) コミットしていない削除データに排他を掛けた場合の効果

コミットしていない削除データに排他を掛けた場合の効果を次に示します。

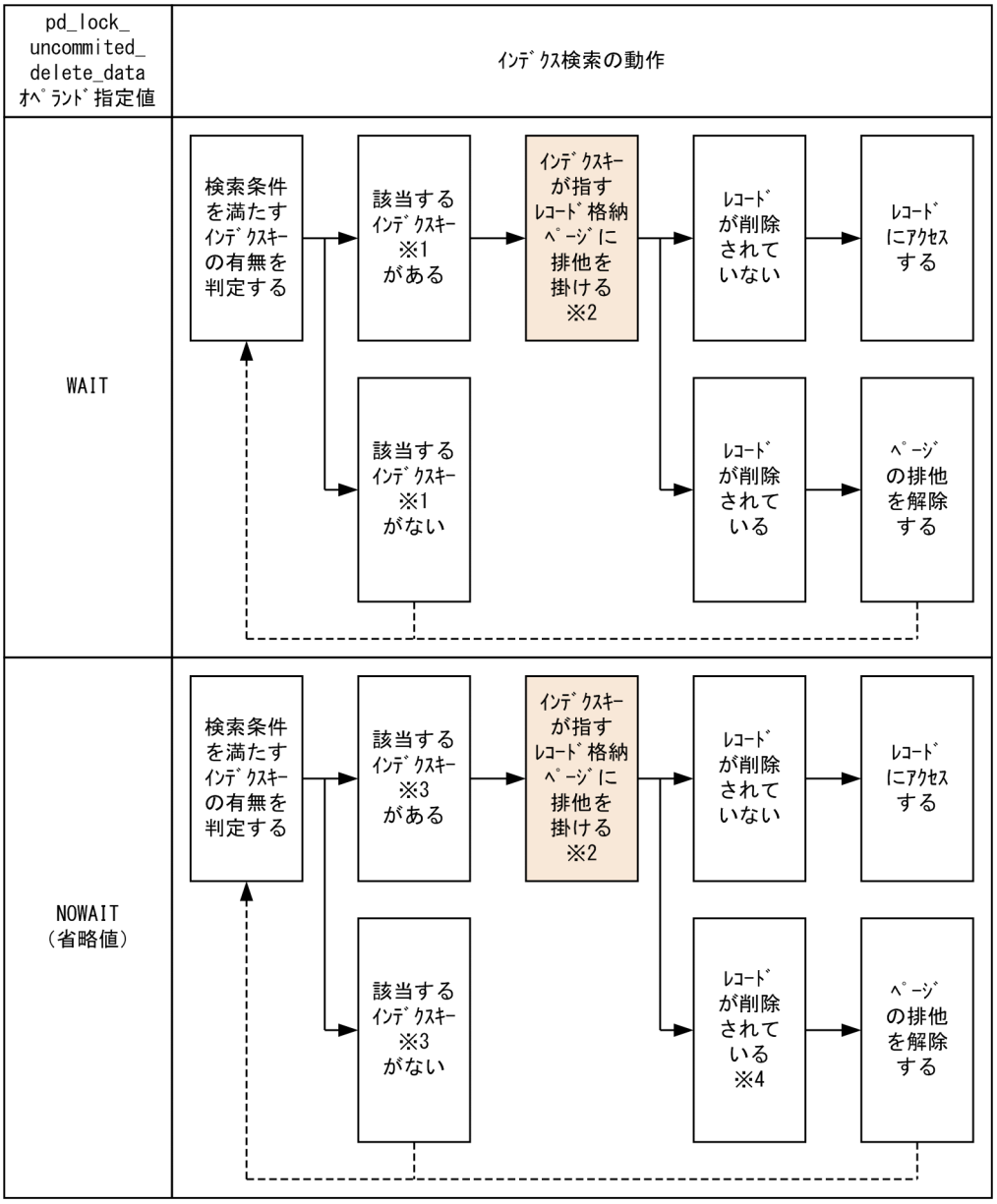
- 4V FMB または SD FMB の SDB データベースのルートレコードを検索中に，コミットする前の削除データまたは更新前のインデクスキーを検知した場合，検索トランザクションは，削除または更新トランザクションのコミットまたはロールバックの決着を待ってから検索処理を行います。これによって，削除または更新トランザクションでロールバックが発生した場合でも，検索トランザクションからの検索読み飛ばしを防止します。
- 二次インデクスを使用して，SD FMB の SDB データベースの子レコードを検索中に，コミットする前の削除データまたは更新前のインデクスキーを検知した場合，検索トランザクションは，削除または更新トランザクションのコミットまたはロールバックの決着を待ってから検索処理を行います。これによって，削除または更新トランザクションでロールバックが発生した場合でも，検索トランザクションからの検索読み飛ばしを防止します。

(4) インデクス検索時の排他制御

システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定した場合，インデクス検索時に削除中または削除済みのレコードを格納していたページにも排他を掛けます。

pd_lock_uncommited_delete_data オペランドに WAIT を指定した場合と，NOWAIT（省略値）を指定した場合のインデクス検索の動作の違いを次の図に示します。

図 2-100　pd_lock_uncommited_delete_data オペランド指定によるインデクス検索の動作の違い



注

上記の図は，削除中のレコードのインデクスキーを検索する例です。更新中のレコードの更新前のインデクスキーを検索する場合も，インデクス検索の動作は上記の図と同じ流れになります。

注※1

削除中または削除済みのレコードのインデクスキーも含みます。

注※2

排他制御モードについては，次の個所を参照してください。

- 「表 2-35　排他制御のモードの組み合わせの例（4V FMB の SDB データベースの場合）」
- 「表 2-36　排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合）」

- 「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」

注※3

削除中または削除済みのレコードのインデクスキーは含みません。

注※4

排他待ちが発生した場合に削除されていることがあります。

(5)　注意事項

(a)　インデクスの残存エントリ

SDB データベースを操作する API または DML で次のどちらかの操作を行った場合，インデクスキーエントリは削除されません。残存エントリとしてインデクス内に残ります。

- レコードの削除（ERASE）によって，インデクスが定義されたレコード型のレコードを削除した場合（この場合，削除レコードのインデクスキーエントリが残存エントリとして残ります）
- レコードの削除（ERASE）またはレコードの一括削除によって，仮想ルートレコード配下の子レコードをすべて削除した場合（この場合，仮想ルートレコードのシステム用構成要素が X'0000'のインデクスキーエントリが残存エントリとして残ります）
- レコードの更新（MODIFY）によって，インデクスが定義された構成要素を更新した場合（この場合，更新レコードの更新前のインデクスキーエントリが残存エントリとして残ります）

インデクスの残存エントリの注意事項については，マニュアル「HiRDB UAP 開発ガイド」の「コミットしていない削除データの排他制御」のインデクスの残存エントリの注意事項を参照してください。インデクスの残存エントリの注意事項を参照する際の注意点を次に示します。

- 残存エントリの発生条件については，前述したとおりです。
- 「pdrorg」を「pdsdbrog および pdsdblod」に読み替えてください。
- 回避策 2 は SDB データベースには適用できないため，無視してください。
- 「表の残存エントリ」の説明は無視してください。

2.9.11　重複するインデクスキー値を格納する際の実行中トランザクション終了待ち合わせ【4V FMB，SD FMB】

HiRDB/SD は，ルートレコードの格納時に，格納しようとしているレコードのインデクスキー値が，SDB データベース中にすでに存在しないかどうかをチェックします。また，ルートレコードの更新時も，更新後のインデクスキー値が，SDB データベース中にすでに存在しないかどうかをチェックします。

同じインデクスキー値が存在する場合は，ユニークエラーとすることで，追加されるインデクスキー値または更新後のインデクスキー値が一意であることを保証します。

4V FMB および SD FMB の SDB データベースの場合で，2 つのトランザクションから同じインデクスキー値を指定してルートレコードを格納または更新する際に，後発のトランザクションを即時ユニークエラーとしないで，先発のトランザクションが終了してからインデクスキー値が一意であるかどうかをチェックしたいときは，システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定してください。

表 2-44　重複するインデクスキー値を持つレコードを格納または更新する際の動作（4V FMB の SDB データベースの場合）

レコード型の種類	システム共通定義 pd_lock_uncommited_delete_data オペランドの値	同じインデクスキー値を指定してレコードを格納または更新する，後発のトランザクション
ルートレコード	NOWAIT	即時，ユニークエラーになります。※1
	WAIT	先発のトランザクションが終了したあと，ユニークエラーになります。※1，※2
ルートレコード以外	－	

（凡例）

－：該当しません。

注※1

SDB データベースを操作する API の実行を無視します。ただし，HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の場合，トランザクションを無効にします。

注※2

先発のトランザクションがロールバックした場合，レコードを格納または更新します。

表 2-45　重複するインデクスキー値を持つレコードを格納または更新する際の動作（SD FMB の SDB データベースの場合）

レコード型の種類	複数のトランザクションによるファミリの競合	システム共通定義 pd_lock_uncommited_delete_data オペランドの値	同じインデクスキー値を指定してレコードを格納または更新する，後発のトランザクション
ルートレコード	－	NOWAIT	即時，ユニークエラーになります。※1
		WAIT	先発のトランザクションが終了したあと，ユニークエラーになります。※1，※2
ルートレコード以外	同じファミリを操作する	－	
	異なるファミリを操作する	NOWAIT	即時，ユニークエラーになります。※1
		WAIT	先発のトランザクションが終了したあと，ユニークエラーになります。※1，※2

（凡例）

－：該当しません。

注※1

DML の実行を無視します。ただし，HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の場合，トランザクションを無効にします。

注※2

先発のトランザクションがロールバックした場合，レコードを格納または更新します。

なお，4V AFM の SDB データベースの場合，システム共通定義の pd_lock_uncommited_delete_data オペランドの指定値に関係なく，先発のトランザクションの終了を待ち合わせします。

2.10　データベースの更新ログの取得方式

HiRDB/SD は，UAP またはユティリティ※1 によって更新されたデータベースの履歴情報（システムログ中のデータベースの更新ログ）をシステムログファイルに取得しています。また，データベースの更新ログを取得しないこともできます。※2 データベースの更新ログを取得しないと，その分の処理時間が短縮されます。したがって，UAP（またはユティリティ）の実行時間を短縮できます。

注※1

次のユティリティが該当します。

- HiRDB/SD 定義ユティリティ（pdsdbdef）
- HiRDB/SD データベース作成ユティリティ（pdsdblod）
- HiRDB/SD データベースアクセスユティリティ（pdsdbexe）

注※2

HiRDB/SD 定義ユティリティ（pdsdbdef）によって更新されたデータベースのログは，常時取得しています。システムログ中のデータベースの更新ログを取得しない運用はできません。

このため，分割格納条件の変更機能を使用して格納 RD エリアの変更を行う場合は，SDB データベース定義の更新に必要なシステムログだけでなく，格納条件が変更されたことによって，変更対象の RD エリアに格納できなくなったデータの削除処理に必要なデータベースの更新ログを格納済みのデータ量に応じて出力します。

2.10.1　データベースの更新ログ取得方式の種類

UAP（または HiRDB/SD データベース作成ユティリティ（pdsdblod））を実行するときのデータベースの更新ログ取得方式には，次の表に示す 2 種類のモードがあります。

表 2-46　データベースの更新ログ取得方式

データベースの更新ログ取得方式	説明
ログ取得モード	ロールバックおよびロールフォワードに必要なデータベース更新ログを取得します。通常はログ取得モードで更新ログを取得します。
ログレスモード	データベース更新ログを取得しません。

2.10.2　データベースの更新ログ取得方式の指定方法

データベースの更新ログ取得方式の指定方法を次の表に示します。

表 2-47　データベースの更新ログ取得方式の指定方法

データベースの更新ログ取得方式の指定方法	説明
UAP の場合	クライアント環境定義の PDDBLOG オペランドで，データベースの更新ログ取得方式を指定します。PDDBLOG オペランドでは，ログ取得モードまたはログレスモードを指定できます。
HiRDB/SD データベース作成ユティリティ（pdsdblod）の場合	HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の logmode オペランドで，データベースの更新ログ取得方式を指定します。 HiRDB/SD データベース作成ユティリティについては「12.　HiRDB/SD データベース作成ユティリティ（pdsdblod）」を参照してください。

2.10.3　データベースの更新ログ取得方式による運用方法の違い

データベースの更新ログ取得方式が異なると，次に示す運用方法が異なります。

- UAP が異常終了したときの HiRDB/SD の処理とユーザの処置
- データベースを回復できる時点

(1)　UAP が異常終了したときの HiRDB/SD の処理とユーザの処置

UAP が異常終了したときの HiRDB/SD の処理とユーザの処置を次の表に示します。

表 2-48　UAP が異常終了したときの HiRDB/SD の処理とユーザの処置

データベースの更新ログ取得方式	HiRDB/SD の処理	ユーザの処置
ログ取得モード	更新した RD エリアの状態を UAP 実行前の状態または異常終了直前に取得した同期点までロールバックします。	UAP 実行前の状態にロールバックされた場合は，UAP を再実行してください。異常終了直前に取得した同期点までロールバックされた場合は，同期点以降の処理を実行してください。
ログレスモード	ロールバックしません。更新した RD エリアを**障害閉塞（ログレス閉塞）**します。RD エリアの内容は破壊されます。	UAP 実行前に取得したバックアップを入力情報として，データベース回復ユティリティ（pdrstr）で RD エリアを回復してください。その後，UAP を再実行してください。

(2)　データベースを回復できる時点

データベース回復ユティリティ（pdrstr）を使用してデータベースを回復できる時点を次の表に示します。

表 2-49　データベースを回復できる時点

データベースの更新ログ取得方式	データベースを回復できる時点
ログ取得モード	バックアップ取得時点またはバックアップ取得時点以降の任意の同期点

データベースの更新ログ取得方式	データベースを回復できる時点
ログレスモード	バックアップ取得時点

2.10.4 バックアップについての注意（重要）

1. ログレスモードまたは更新前ログ取得モードの UAP（またはユティリティ）の実行中に，更新可能モード（-M s 指定）でバックアップを取得しないでください。
2. ログレスモードまたは更新前ログ取得モードの UAP（またはユティリティ）の実行後は，次に示すモードでバックアップを取得してください。
   - 参照・更新不可能モード（-M x 指定）
   - 参照可能モード（-M r 指定）

2.11　システムファイル

障害時にシステムの状況を回復するための情報などを格納するファイルを**システムファイル**といいます。システムファイルは次に示すファイルの総称です。

- システムログファイル
- シンクポイントダンプファイル
- ステータスファイル

各システムファイルについては，マニュアル「HiRDB 解説」の「システムファイル」を参照してください。

2.11.1　作業表用ファイル

ユティリティを実行するとき，および HiRDB のディクショナリ表に格納されている HiRDB の定義情報を検索するときに必要となる一時的な情報を格納するためのファイルを**作業表用ファイル**といいます。作業表用ファイルは HiRDB/SD が自動的に作成します。HiRDB 管理者は，作業表用ファイル用の HiRDB ファイルシステム領域を作成してください。

作業表用ファイルの構成

次に示すサーバに作業表用ファイルが必要になります。

- ディクショナリサーバ
- バックエンドサーバ

作業表用ファイルを使用するユティリティ

次に示すユティリティで作業表用ファイルを使用します。

- HiRDB/SD 定義ユティリティ（pdsdbdef）
- HiRDB/SD データベース作成ユティリティ（pdsdblod）

2.12　HiRDB システム定義

HiRDB の稼働環境を HiRDB システム定義で設定します。HiRDB システム定義を格納するファイルを**HiRDB システム定義ファイル**といいます。

HiRDB システム定義ファイルについては，マニュアル「HiRDB 解説」の「HiRDB/パラレルサーバの HiRDB システム定義の体系」を参照してください。

■SDB 用 UAP 環境定義

HiRDB システム定義には，マニュアル「HiRDB 解説」の「HiRDB/パラレルサーバの HiRDB システム定義の体系」で説明されている HiRDB システム定義以外に，SDB 用 UAP 環境定義があります。

SDB 用 UAP 環境定義には，SD FMB の SDB データベースにアクセスする UAP の実行環境を定義します。

SDB 用 UAP 環境定義を指定したファイル（SDB 用 UAP 環境定義ファイル）は，フロントエンドサーバがあるサーバマシンに配置します。マルチフロントエンドサーバの場合は，SDB 用 UAP 環境定義を適用したいフロントエンドサーバのサーバマシンに配置します。

SDB 用 UAP 環境定義ファイルを含めた HiRDB システム定義の構成例については，「4.9.2　SDB 用 UAP 環境定義ファイルを配置するサーバマシンの検討」を参照してください。

2.13　HiRDB の開始・終了

HiRDB の開始・終了の詳細については，マニュアル「HiRDB 解説」の「HiRDB の開始・終了」を参照してください。

注意事項

HiRDB/SD データベース作成ユティリティ（pdsdblod）をログレスモードで実行中，または HiRDB/SD データベース再編成ユティリティ（pdsdbrog）を実行中に HiRDB が強制終了，または異常終了すると次回再開始時に当該トランザクションはロールバック対象となりません。

2.14　系切り替え機能

HiRDB と HA モニタが連携することによって，システムの信頼性向上，稼働率向上を目的とした系切り替え機能が使用できるようになります。HiRDB/SD で使用できる系切り替え機能は次のとおりです。

表 2-50　HiRDB/SD で使用できる系切り替え機能

項目	HiRDB/SD で使用できる機能
系切り替え機能の種類	スタンバイ型系切り替え機能 • 通常の系切り替え • 高速系切り替え
系切り替え機能の形態	• 自動系切り替え • 計画系切り替え

ここでは，これらの系切り替え機能の概要について説明します。

なお，HiRDB/SD で使用できない系切り替え機能については，「付録 A　未サポート機能の一覧」を参照してください。HA モニタについては，マニュアル「高信頼化システム監視機能 HA モニタ」を参照してください。

2.14.1　高速系切り替え機能とは

HiRDB の高速系切り替え機能については，マニュアル「HiRDB 解説」の「系切り替え機能とは」，および「系の切り替え時間を短縮する機能」を参照してください。

2.14.2　系切り替え機能の形態

系切り替え機能の形態については，マニュアル「HiRDB 解説」の「系切り替え機能の形態」を参照してください。

2.14.3　モニタモードとサーバモード

モニタモードとサーバモードについては，マニュアル「HiRDB 解説」の「系切り替え機能の運用方法」を参照してください。

モニタモード，サーバモードのどちらも使用できるクラスタソフトウェアは HA モニタだけです。

2.15　機密保護機能

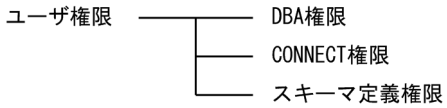
データベースを外部の人にアクセスされないようにする機能として，**機密保護機能**があります。機密保護機能では**ユーザ権限**という概念を使用していて，必要な権限を持っていないとデータベースにアクセスできないようになっています。

ここでは，機密保護機能の概要について説明します。機密保護機能の運用方法については，「5.16　セキュリティに関する運用」を参照してください。

2.15.1　ユーザ権限の種類

ここでは，ユーザ権限の種類について説明します。ユーザ権限の種類を次の図に示します。

図 2-101　ユーザ権限の種類



詳細については，マニュアル「HiRDB 解説」の「ユーザ権限の種類」を参照してください。

2.15.2　機密保護機能の運用方法

HiRDB 管理者，または DBA 権限保持者が，HiRDB のユーザに必要なユーザ権限を与えて，機密保護機能を運用します。権限の与え方については，「5.16.1　構造型 DB 機能に関するユーザ権限の設定」を参照してください。

3 HiRDB/SD のデータベース設計

この章では，HiRDB/SD のデータベース設計について説明します。

3.1 HiRDB/SD の設計

ここでは，HiRDB/SD の設計方針について説明します。

3.1.1 システム設計

HiRDB のシステム設計については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB/パラレルサーバのシステム設計」を参照してください。

3.1.2 HiRDB ファイルシステム領域の設計

HiRDB のシステム構築時に，HiRDB ファイルを作成する HiRDB ファイルシステム領域を作成します。ここでは，HiRDB ファイルシステム領域を作成するときの設計方針について説明します。

(1) HiRDB ファイルシステム領域に使用するファイル

HiRDB ファイルシステム領域に使用するファイルについては，マニュアル「HiRDB 解説」の「HiRDB ファイルシステム領域」を参照してください。

(2) HiRDB ファイルシステム領域の種類

HiRDB ファイルシステム領域の種類については，マニュアル「HiRDB 解説」の「HiRDB ファイルシステム領域」を参照してください。

(3) RD エリア用の HiRDB ファイルシステム領域の設計

RD エリア用の HiRDB ファイルシステム領域の設計方針については，マニュアル「HiRDB システム導入・設計ガイド」の「RD エリア用の HiRDB ファイルシステム領域の設計」を参照してください。

(4) システムファイル用の HiRDB ファイルシステム領域の設計

システムファイル用の HiRDB ファイルシステム領域の設計方針については，マニュアル「HiRDB システム導入・設計ガイド」の「システムファイル用の HiRDB ファイルシステム領域の設計」を参照してください。

(5) 作業表用ファイル用の HiRDB ファイルシステム領域の設計

作業表用ファイルとは，HiRDB/SD 定義ユティリティ（pdsdbdef）または HiRDB/SD データベース作成ユティリティ（pdsdblod）の実行時に作成される一時的な情報を格納するファイルです。

作業表用ファイル用の HiRDB ファイルシステム領域の設計方針については，マニュアル「HiRDB システム導入・設計ガイド」の「作業表用ファイル用の HiRDB ファイルシステム領域の設計」を参照してください。

(6) ユティリティ用の HiRDB ファイルシステム領域の設計

ユティリティ用の HiRDB ファイルシステム領域の設計方針については，マニュアル「HiRDB システム導入・設計ガイド」の「ユティリティ用の HiRDB ファイルシステム領域の設計」を参照してください。

(7) HiRDB ファイルシステム領域の最大長

HiRDB ファイルシステム領域の最大長は，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB ファイルシステム領域の最大長」を参照してください。

3.1.3 システムファイルの設計

ここでは，システムファイルの設計について説明します。

(1) システムログファイルの設計

システムログが出力されているファイルの容量いっぱいにシステムログが出力されると，システムログの出力先がほかのシステムログファイルに変更されます。これをシステムログファイルのスワップといいます。

システムログファイルのスワップ時に，スワップ先にできる状態のシステムログファイルがないと HiRDB（ユニット）は異常終了します。このため，システムログファイルの容量を適切に見積もっておく必要があります。

システムログファイルの設計については，マニュアル「HiRDB システム導入・設計ガイド」の「システムファイルの設計」の「システムログファイルの設計」を参照してください。なお，システムログ量を見積もる際には，「3.5.1 システムログ量の見積もり」で算出した値を加算してください。

HiRDB にはシステムログファイルの空き容量を監視する機能（システムログファイルの空き容量監視機能）があります。詳細については，マニュアル「HiRDB システム運用ガイド」の「システムログファイルの空き容量監視機能の運用方法」を参照してください。

(2) シンクポイントダンプファイルの設計

シンクポイントダンプファイルはシステムマネジャを除いた各サーバに必要です。シンクポイントダンプファイルの容量が不足すると，HiRDB を開始できません。

シンクポイントダンプファイルの設計については，マニュアル「HiRDB システム導入・設計ガイド」の「システムファイルの設計」の「シンクポイントダンプファイルの設計」を参照してください。

(3) ステータスファイルの設計

システムマネジャを除いた各サーバにサーバ用ステータスファイルが必要です。また，各サーバマシンにユニット用ステータスファイルが必要です。ステータスファイルが容量不足になると HiRDB が異常終了します。

ステータスファイルの設計については，マニュアル「HiRDB システム導入・設計ガイド」の「システムファイルの設計」の「ステータスファイルの設計」を参照してください。

3.1.4 RD エリアの配置

ここでは，次に示す RD エリアを配置するときの考慮点について説明します。

- システム用 RD エリア
- ユーザ用 RD エリア

HiRDB では，RD エリアを作成するサーバを意識する必要があります。RD エリアを作成するサーバマシンを次の表に示します。

表 3-1 RD エリアを作成するサーバマシン

RD エリアの種類	RD エリアを作成するサーバマシン
マスタディレクトリ用 RD エリア	ディクショナリサーバがあるサーバマシンに作成します。
データディクショナリ用 RD エリア	
データディレクトリ用 RD エリア	
ユーザ用 RD エリア	バックエンドサーバがあるサーバマシンに作成します。

注意事項

pd_max_rdarea_no オペランドの値に注意してください。このオペランドには RD エリアの最大数を指定します。RD エリアの数がこのオペランドの値を超えている場合は HiRDB を正常開始できません。

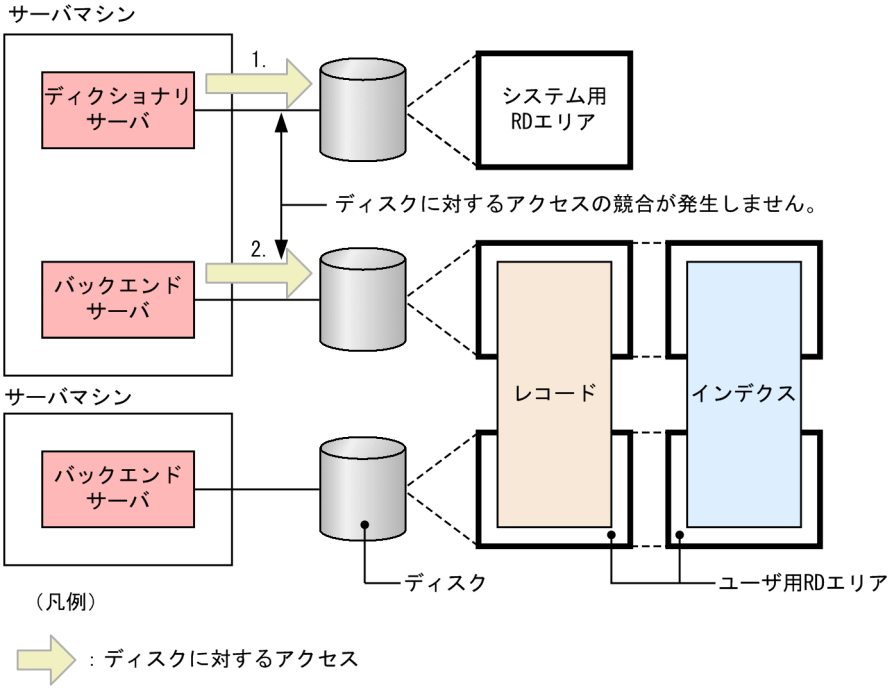
(1) システム用 RD エリアの配置

システム用 RD エリアは，ユーザ用 RD エリアの配置を考慮して配置します。システム用 RD エリアを配置するときの考慮点を次に示します。

- システム用 RD エリアはディクショナリサーバに配置します。
- ディクショナリサーバとバックエンドサーバが同一のサーバマシンにある場合は，ユーザ用 RD エリアを配置するディスクとは異なるディスクにシステム用 RD エリアを配置するようにします。

システム用 RD エリアの配置例を次の図に示します。

図 3-1　システム用 RD エリアの配置例



1. SDBディクショナリ情報やSDBディレクトリ情報の参照などのためのアクセス
2. レコードに対するアクセス

(2)　ユーザ用 RD エリアの配置

(a)　システムログファイルとの関連

システムログファイルを配置したディスクとは異なるディスクに，ユーザ用 RD エリアを配置するようにします。このようにすることで，シンクポイント時のシステムログファイルとユーザ用 RD エリアを構成する HiRDB ファイルへの入出力処理を複数のディスクに分散できるため，シンクポイントでの処理時間を削減できます。

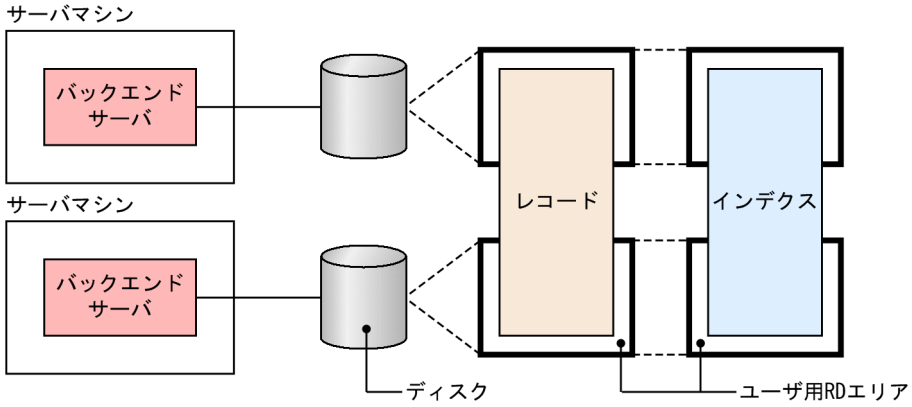
(b)　システム用 RD エリアとの関連

システム用 RD エリアを配置したディスクとは異なるディスクにユーザ用 RD エリアを配置するようにします。

(c)　SDB データベースを横分割した場合

SDB データベースを横分割した場合，横分割した SDB データベースを格納する RD エリアを異なるバックエンドサーバに配置します。また，横分割した SDB データベースを格納する RD エリアを異なるディスクに配置します。ユーザ用 RD エリアの配置例を次の図に示します。

図 3-2　ユーザ用 RD エリアの配置例



3.1.5　ユニット数またはサーバ数が多いシステムを構築する場合の考慮点

ユニット数またはサーバ数が多いシステムを構築および運用する場合，HiRDB で使用するポートを固定して通信負荷を軽減するシステム定義の設定，ユニット名を付けて実行するコマンドの種類などについての考慮が必要です。

目安として，10 ユニット以上，または 10 サーバ（フロントエンドサーバ，ディクショナリサーバ，およびバックエンドサーバ）以上のシステムを構築して運用する場合，マニュアル「HiRDB システム導入・設計ガイド」の「ユニット数またはサーバ数が多いシステムを構築する場合の考慮点」を参照してください。

3.2　RD エリアの設計

ここでは，RD エリアの設計するときの考慮点について説明します。

RD エリアを設計するときに考慮する点については，マニュアル「HiRDB システム導入・設計ガイド」の「RD エリアを設計するときの検討項目」も参照してください。また，RD エリアの配置については，「3.1.4 RD エリアの配置」を参照してください。

3.2.1　ユーザ用 RD エリアを設計する際の考慮点

ユーザ用 RD エリアを設計する際の考慮点を次に示します。

- SDB データベースを格納するユーザ用 RD エリアを定義する際，create rdarea 文の data model オペランドに structured を必ず指定してください。
- 次の表またはインデクスを格納するユーザ用 RD エリアを定義する際，create rdarea 文の data model オペランドに relational を指定するか，または data model オペランドの指定を省略してください。
  - 追い付き反映キー対応表
  - 追い付き反映キー対応表のインデクス
  - 追い付き状態管理表
  - 監査証跡表
- ユーザ用 RD エリアは，公用 RD エリアとして定義してください。ただし，監査証跡表を格納するユーザ用 RD エリアについては，マニュアル「HiRDB システム運用ガイド」の「監査人の登録，監査証跡表を格納する RD エリアの作成，および監査証跡表の作成」の「監査証跡表を格納する RD エリアの作成」の「RD エリアの作成」を参照して，格納できる RD エリアの種類を確認してください。
- レコード型とインデクスを同じユーザ用 RD エリアに格納できません。
- 複数の SDB データベースを同じユーザ用 RD エリアに格納できません。
- SDB データベースを横分割する場合，レコード型とインデクスの横分割数を同じにする必要があります。また，横分割したレコード型とインデクスを格納するユーザ用 RD エリアを同じバックエンドサーバに配置する必要があります。詳細については，「2.3.7　SDB データベースの横分割」を参照してください。

上記以外の RD エリアを設計するときに考慮する点については，マニュアル「HiRDB システム導入・設計ガイド」の「RD エリアを設計するときの検討項目」を参照してください。

3.2.2　セグメントの設計

ここでは，セグメントの設計をするために必要な情報について説明します。

(1) セグメントの状態

セグメントには次の表に示す状態があります。

表 3-2 セグメントの状態

セグメントの状態	説明
使用中セグメント	レコードまたはインデクスのデータを格納しているセグメントです。特に，データが満杯でセグメント内にデータを追加できないセグメントを**満杯セグメント**といい，データの削除でセグメント内の全ページが空きページ（未使用ページ）のセグメントを**使用中空きセグメント**といいます。
未使用セグメント	使用されたことがないセグメントです。このセグメントは RD エリア内のすべてのレコード（またはインデクス）が使用できます。

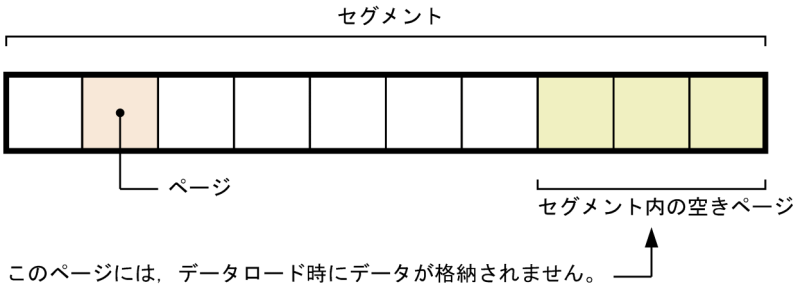
なお，データが格納されていないセグメントを**空きセグメント**といいます。使用中空きセグメントと未使用セグメントは，空きセグメントに分類されます。

(2) セグメント内の空きページの比率

HiRDB/SD データベース作成ユティリティ（pdsdblod）によるレコード格納時に，使用中セグメント内に空きページを確保できます。どれだけの空きページを確保するかは，セグメント内の空きページの比率で指定します。

セグメント内の空きページの比率を次の図に示します。

図 3-3 セグメント内の空きページの比率



[説明]

- この例ではセグメント内の空きページの比率を 30%としています。したがって，10 ページ中 3 ページが空きページになります。
- セグメント内の空きページの比率は，SDB データベース格納定義の PCTFREE 句で指定します。セグメント内の空きページの比率は 0～50%の範囲で指定でき，省略値は 10%となります。また，HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の recfree オペランドで，セグメント内の空きページの比率を指定することもできます。両方を指定した場合，recfree オペランドの指定値が優先されます。

なお，セグメント内の空きページの比率の設定は性能に影響を及ぼします。詳細については，「3.2.5　レコード格納時の空き領域の作成（サブページ分割をしない場合）」または「3.2.6　レコード格納時の空き領域の作成（サブページ分割をする場合）」を参照してください。

(3)　セグメントの確保と解放

RD エリアにレコードを格納する際，必要に応じてセグメントが確保されます。データの追加と削除を繰り返した場合，データ量が増えていないのに RD エリアが容量不足になることがあります。これを防ぐには HiRDB/SD データベース再編成ユティリティ（pdsdbrog）による SDB データベースの再編成を定期的に行ってセグメントを解放してください。

SDB データベースの再編成については，次のどれかを参照してください。

- 「5.11　SDB データベースの再編成（インナレプリカ機能を使用しない場合）」
- 「5.12　SDB データベースの再編成（インナレプリカ機能を使用したオンライン再編成）【4V FMB, 4V AFM】」
- 「5.13　SDB データベースの再編成（インナレプリカ機能を使用した更新可能なオンライン再編成）【4V FMB】」

なお，次に示す操作をした場合にもセグメントが解放されます。

- RD エリアの再初期化
- SDB データベースの削除
- HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の purge オペランドに yes を指定したデータロードまたはフォーマットライト

(4)　セグメントサイズの設計方針

セグメントサイズはデータの入出力時間およびディスク所要量に影響を与えるため，綿密に設計する必要があります。HiRDB/SD では，レコードの近傍配置やデータのアクセス性能（同時実行性能）を考慮して，あらかじめレコードを格納しない領域（セグメント内の空きページの比率）を設計しておく必要があります。

セグメントサイズについては，「3.5.5(4)　平均的なファミリが使用するページ数またはサブページ数の算出」で算出したサイズにすることを推奨します。

3.2.3　ページの設計

ここでは，ページの設計をするために必要な情報について説明します。

なお，1 ファミリのトータルのレコード長とページ長によっては，レコードを格納する際の配置制御との関係からレコードの格納効率が低下することがあります。「3.2.4　サブページの設計」で説明しているサブページの適用基準を参照して，該当する場合はサブページ分割の適用を検討してください。

ここでの説明は，サブページ分割をしないことを前提としています。

(1) ページの状態

ページには次の表に示す状態があります。

表 3-3 ページの状態

ページの状態	説明
未使用ページ	まだ割り当てられていないページ，またはレコードの削除によって解放されたページです。
使用中ページ	次のどちらかの条件を満たすページです。 • レコードが格納されていて，レコードを追加できる空き領域がある，通常のページまたは事前割り当てページ • レコードが格納されていない事前割り当てページ
使用中満杯ページ	次のどちらかの条件を満たすページです。 • レコードが格納されていて，レコードを追加できる空き領域がないページ • ページ切り替えオプションに'O'が指定されたレコードが格納されているページ

(2) ページ内の未使用領域の比率

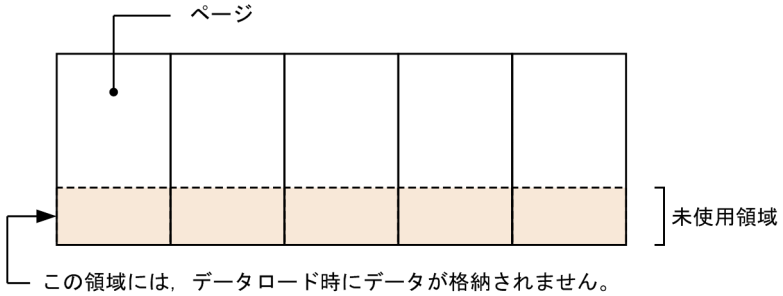
HiRDB/SD データベース作成ユティリティ（pdsdblod）によるレコード格納時に，使用中ページ内に未使用領域を確保できます※。どれだけの未使用領域を確保するかは，ページ内の未使用領域の比率で指定します。

注※

4V FMB または 4V AFM の SDB データベースを操作する API で，PCTFREE 有効化オプションを指定したレコードの格納時にも，使用中ページ内に未使用領域を確保できます。SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

ページ内の未使用領域の比率を次の図に示します。

図 3-4 ページ内の未使用領域の比率



［説明］

ページ内の未使用領域の比率は，SDB データベース格納定義の PCTFREE 句で指定します。ページ内の未使用領域の比率は 0～99%の範囲で指定でき，省略値は 30%となります。また，HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の recfree オペランドで，ページ内の未使用領域の比率を指定することもできます。両方を指定した場合，recfree オペランドの指定値が優先されます。

なお，ページ内の未使用領域の比率の設定は性能に影響を及ぼします。詳細については，「3.2.5　レコード格納時の空き領域の作成（サブページ分割をしない場合）」を参照してください。

(3)　ページの確保と解放

- **ページの確保**

  RD エリアにレコードを格納する際，必要に応じてページが確保されます。一度確保されたページ（一度使用したページ）はそのページが解放されないかぎり，再利用できません。

- **ページの解放**

  次のどれかの条件を満たす場合にページは解放されます。

  - セグメントが解放された場合
  - 事前割り当てページであり，ルートレコード実現値が削除された場合

    この場合，該当するファミリを格納するために確保している事前割り当てページがすべて解放されます。
  - 通常のページであり，ページ内のレコード実現値がすべて削除された場合

    なお，レコードの一括削除の場合は，レコードの管理情報を更新して，レコードが削除された状態にします。実際のレコード自体は削除されません。そのため，ページ内のレコード実現値がすべて削除された状態となっても，ページは解放されません。

(4)　ページ長の設計方針

ページ長は，データの入出力時間およびディスク所要量に影響を与えるため，次の項目を考慮して設計する必要があります。

- ページ長は，対象となる SDB データベースの最大のレコード長よりも大きくなければなりません。次に示す計算式を満たすように設計してください。

  ページ長≧mrl + 88（単位：バイト）

  （凡例）

  mrl：対象となる SDB データベースの最大のレコード長
- ページは入出力の単位であるため，ページ長を大きくすると入出力回数を軽減できます。逆にページ長が小さいと入出力回数が増加します。

- ページは排他制御の単位であるため，ページ長を小さくして格納するレコード数を制限することで同時実行性を向上させることができます。逆にページ長を大きくして同一ページに多くのレコードを格納すると同時実行性が低下します。
- ルートレコードが異なるレコードは別ページに格納されるため，1 ファミリのトータルのレコード長が小さい場合，不当に大きなページ長を設定すると格納効率が悪くなります。

HiRDB/SD では，レコードの近傍配置やデータのアクセス性能（同時実行性能）を考慮する必要があります。そのため，あらかじめレコードを格納しない領域（ページ内の未使用領域の比率）も含めて設計します。

3.2.4　サブページの設計

ここでは，サブページの設計をするために必要な情報について説明します。

サブページ分割する場合には，同時実行性とデータベース容量を考慮して設計してください。サブページ分割を適用した場合の配置制御については，「2.7　レコードの配置制御（サブページ分割をする場合）」を参照してください。

(1)　サブページの使用目的と適用基準

HiRDB/SD では，レコードを格納する際の配置制御によって，ルートレコードが異なるレコード（異なるファミリのレコード）は同じページに格納されません。そのため，1 ファミリに対して最低 1 ページの領域が必ず確保されます。1 ファミリのトータルのレコード長がページ長より小さい場合，ページ内に未使用領域ができてしまうため，レコードの格納効率が低下し，データベース容量が増加します。

このような場合，ページをサブページ分割すると，ページ内の未使用領域の発生を抑えることができます。ページ内の未使用領域の発生を抑えることで，レコードの格納効率が向上し，データベース容量の削減を図ることができます。

次の表に示す適用基準を満たす場合は，サブページの使用を検討してください。

表 3-4　サブページの適用基準

項番	SDB データベース種別	適用基準
1	4V FMB または SD FMB の場合	階層構造がなく（親レコードだけ），レコード長が 1,982 バイト以下の場合
2		1 ファミリ内のレコードが少なく，1 ファミリのトータルのレコード長が（1,992－10×格納レコード件数）バイト以下の場合
3	4V AFM の場合	1 ファミリ内のレコードが少なく，1 ファミリのトータルのレコード長が（1,992－10×格納レコード件数）バイト以下の場合
4		排他が競合したときの影響を局所化するために，1 ページに格納するレコード件数を制限した結果，1 ページに格納されるレコードの合計長の最大値が（1,992－10×格納レコード件数）バイト以下になった場合
5		次に示す 2 つの条件を満たす場合

項番	SDB データベース種別	適用基準
		• レコード長の最大値が 1,982 バイト以下の場合 • アクセス頻度が高い SDB データベースの同時実行性を向上させるために，ページ切り替え指定で 1 ページに 1 件のレコードを格納しようとしている場合

注

レコード長の計算方法については，「3.5.5(1)　レコード長の算出」を参照してください。

(2)　サブページの状態

サブページには次の表に示す状態があります。

表 3-5　サブページの状態

サブページの状態	説明
未使用サブページ	まだ割り当てられていないサブページ，またはレコードの削除によって解放されたサブページです。
使用中サブページ	次のどちらかの条件を満たすサブページです。 • レコードが格納されていて，レコードを追加できる空き領域がある，通常のサブページまたは事前割り当てサブページ • レコードが格納されていない事前割り当てサブページ
使用中満杯サブページ	次のどちらかの条件を満たすサブページです。 • レコードが格納されていて，レコードを追加できる空き領域がないサブページ • ページ切り替えオプションに'O'が指定されたレコードが格納されているサブページ

(3)　サブページ分割しているときのページの状態

サブページ分割しているときのページの状態を次の表に示します。

表 3-6　サブページ分割しているときのページの状態

ページの状態	説明
未使用ページ	まだ割り当てられていないページです。
使用中空きページ	レコードの削除によって，ページ内の全サブページが未使用サブページとなったページです。
使用中ページ	ページ内に使用中サブページまたは使用中満杯サブページがあり，かつ未使用サブページがあるページです。
使用中満杯ページ	ページ内の全サブページが使用中サブページまたは使用中満杯サブページであり，未使用サブページがないページです。

(4)　サブページ内の未使用領域の比率

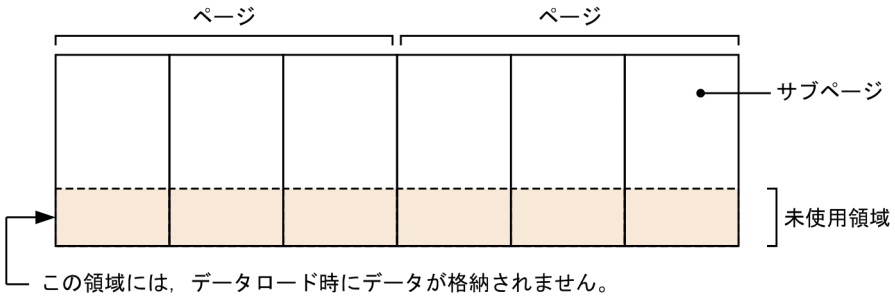
HiRDB/SD データベース作成ユティリティ（pdsdblod）によるレコード格納時に，使用中サブページ内に未使用領域を確保できます※。どれだけの未使用領域を確保するかは，サブページ内の未使用領域の比率で指定します。

注※

4V FMB または 4V AFM の SDB データベースを操作する API で，PCTFREE 有効化オプションを指定したレコードの格納時にも，使用中サブページ内に未使用領域を確保できます。SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

サブページ内の未使用領域の比率を次の図に示します。

図 3-5　サブページ内の未使用領域の比率



［説明］

サブページ内の未使用領域の比率は，SDB データベース格納定義の PCTFREE 句で指定します。サブページ内の未使用領域の比率は 0～99%の範囲で指定でき，省略値は 30%となります。また，HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の recfree オペランドで，サブページ内の未使用領域の比率を指定することもできます。両方を指定した場合，recfree オペランドの指定値が優先されます。

なお，サブページ内の未使用領域の比率の設定は性能に影響を及ぼします。詳細については，「3.2.6　レコード格納時の空き領域の作成（サブページ分割をする場合）」を参照してください。

(5)　サブページの確保と解放

- **サブページの確保**

  レコードを格納する際，必要に応じてサブページが確保されます。一度確保されたサブページはそのサブページが解放されないかぎり再利用できません。

- **サブページの解放**

  - セグメントが解放されたときに，セグメント内のページに属するサブページが解放されます。
  - 事前割り当てサブページの場合，ルートレコードが削除されたときに，該当するファミリを格納するために確保している事前割り当てサブページがすべて解放されます。

- 通常のサブページの場合，サブページ内のレコードがすべて削除されたときに，サブページが解放されます。なお，レコードの一括削除の場合は，レコードの管理情報を更新して，レコードが削除された状態にします。実際のレコード自体は削除されません。そのため，サブページ内のレコードがすべて削除された状態となっても，サブページは解放されません。

(6) サブページ分割しているときのページの確保と解放

- サブページ分割しているときのページの確保

  レコードを格納する際，必要に応じてページが確保されます。一度確保されたページはそのページが解放されないかぎり再利用できません。

- サブページ分割しているときのページの解放
  - セグメントが解放されたときに，セグメント内の全ページが解放されます。
  - ログレスモードの UAP によるレコードの削除で，ページ内の全サブページが未使用サブページになった場合に，そのページが解放されます。

(7) サブページ長（サブページ分割数）の設計方針

サブページ分割するページのページ長は 4,096 バイト固定になります。そのため，サブページ分割数（SDB データベース格納定義の STORAGE RECORD 句下の SUBPAGE NUMBER 句の指定値）によって，次の表に示すようにサブページ長が決まります。

表 3-7 サブページ分割数とサブページ長

サブページ分割数	サブページ長（単位：バイト）
2	2,028
3	1,352
4	1,012
5	808
6	676
7	576
8	504
9	448
10	404
11	368
12	336
13	312
14	288
15	268

サブページ分割数	サブページ長（単位：バイト）
16	252

■サブページ長（サブページ分割数）を決定する際の考慮点

性能（同時実行性）や，データベース容量を考慮してサブページ長（サブページ分割数）を決定します。また，次に示す点にも考慮してください。

- サブページ長は，対象となる SDB データベースの最大レコード長よりも大きくする必要があります。次に示す計算式を満たすように設計してください。

  サブページ長≧mrl + 48（単位：バイト）

  （凡例）

  mrl：対象となる SDB データベースの最大のレコード長

- ファミリが異なるレコードは別々のサブページに格納されます。そのため，1 ファミリのトータルのレコード長が小さい場合，サブページ分割数を小さくしてサブページ長を大きくすると，レコードの格納効率が悪くなります。
- サブページは排他制御の単位であるため，4V AFM の SDB データベースの場合は，サブページ分割数を大きくしてサブページ長を小さくすると，同一サブページに格納されるレコード数が少なくなり，同時実行性が向上します。逆に，サブページ分割数を小さくしてサブページ長を大きくすると，同一サブページに格納されるレコード数が多くなり，同時実行性が低下します。

■サブページ長（サブページ分割数）の設計

次に示す計算式からサブページ内の未使用領域長を計算し，サブページ内の未使用領域長ができるだけ小さい値になるようにサブページ分割数を決めてください。サブページ内の未使用領域長が小さいほど，レコードの格納効率が良くなります。

サブページ内の未使用領域長は，次に示す計算式から求められます。

サブページ内の未使用領域長＝ spl−36−N×（L + 10）（単位：バイト）

（凡例）

spl：サブページ長

N：1 サブページに格納するレコード件数

L：レコード長

なお，計算結果が負の値となる場合は，サブページ長が不足しているため，レコードが格納できません。この場合は，次に示す対処を行ってください。

- サブページ分割数を小さくしてサブページ長を大きくする。
- 1 サブページに格納するレコード件数を少なくする。
- サブページに格納するレコードのレコード長を小さくする。

レコード長の求め方については，「3.5.5(1)　レコード長の算出」を参照してください。

(8)　留意事項

- サブページ分割するページのページ長は 4,096 バイト固定になります。そのため，RD エリアの再初期化でページ長を 4,096 バイトから変更した場合，RD エリアの再初期化後のレコードの格納時や検索時などにエラーとなります。
- サブページ分割を適用した場合，排他制御の単位がページからサブページに変わるため，サブページ分割を適用しないときに比べて，排他制御によるメモリ使用量や処理時間が増加することがあります。例えば，1 ページに格納される複数のレコードが，サブページ分割を適用したことによって複数のサブページに分かれて格納されるような場合は，確保する排他資源数が増加するため，排他制御によるメモリ使用量や処理時間が増加します。
- サブページ分割を適用した RD エリアのバックアップファイルの容量は，サブページ分割を適用しない場合に比べて大きくなります。
- データベース状態解析ユティリティ（pddbst）の物理的解析では，RD エリア内の全セグメントおよび全ページの格納状態を解析します。この場合，解析結果の「満杯ページ」は，「(3)　サブページ分割しているときのページの状態」で説明している「使用中満杯ページ」のことを指します。

3.2.5　レコード格納時の空き領域の作成（サブページ分割をしない場合）

HiRDB/SD データベース作成ユティリティ（pdsdblod）でのレコード格納時に，ページ内にレコードを格納しない領域（未使用領域）を作成したり，セグメント内に使用しない空きページ（セグメント内の空きページ）を作成したりできます。

また，SDB データベースを操作する API によるレコード格納時に，PCTFREE 有効化オプションを指定すると，SDB データベース格納定義の PCTFREE 句で指定したページ内の未使用領域の比率を有効にしてレコードを格納できます。PCTFREE 有効化オプションは，4V FMB または 4V AFM の SDB データベースの場合に指定できます。

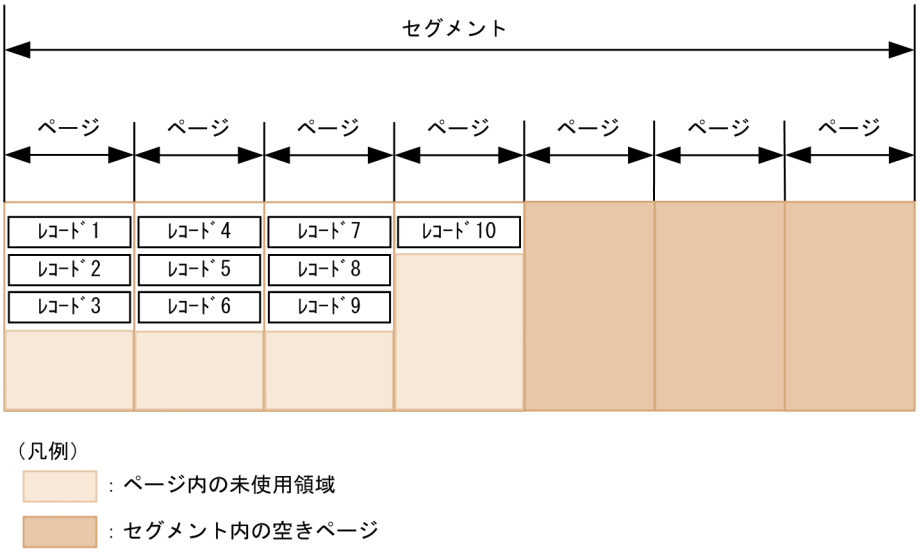
ページ内の未使用領域の比率やセグメント内の空きページの比率は，次の個所で指定できます。

- SDB データベース格納定義の PCTFREE 句
  PCTFREE 句については，次の個所を参照してください。
  - 4V FMB の SDB データベースの場合：「11.5.2(3)(j)　PCTFREE」
  - 4V AFM の SDB データベースの場合：「11.6.2(3)(j)　PCTFREE」
  - SD FMB の SDB データベースの場合：「11.7.2(3)(j)　PCTFREE」
- HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文 recfree オペランド
  environment 文の recfree オペランドについては，「12.3.1　environment 文」を参照してください。

PCTFREE 句と recfree オペランドの使い分けについては，「(3)　PCTFREE 句と recfree オペランドの使い分け」を参照してください。

空き領域の作成イメージを次の図に示します。

図 3-6　空き領域の作成イメージ



なお，この項の説明中の「データロード」については，HiRDB/SD データベース作成ユティリティ（pdsdblod）によるフォーマットライト機能を使用したレコードの格納も含みます。

(1)　空き領域を作成する目的

空き領域を作成する目的を次に示します。

(a)　データロード後のレコード格納に備えて空き領域を確保する

データロードをしたあとに，子レコードを格納する場合，データロード時に作成したページ内の未使用領域，セグメント内の空きページを使用することで，同一ルートレコード下のレコードの近傍に子レコードを格納できます。

ページ内の未使用領域は，次の場合に使用されます。

- SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定し，格納済みのレコードが持つユーザキーの中間のキーを持つレコードを格納する場合
- SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合や，ORDER 句で SORTED DUPLICATES を指定して，最大のユーザキーを持つレコードを格納する場合で，基準ページおよびそのファミリ内接続ページに未使用領域があるとき
- SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合で，親レコード格納ページおよびその前のファミリ内接続ページに未使用領域があるとき

ファミリ内接続ページについては，「2.6.1(2)　ページ間の関連づけとファミリ内接続ページ」を参照してください。

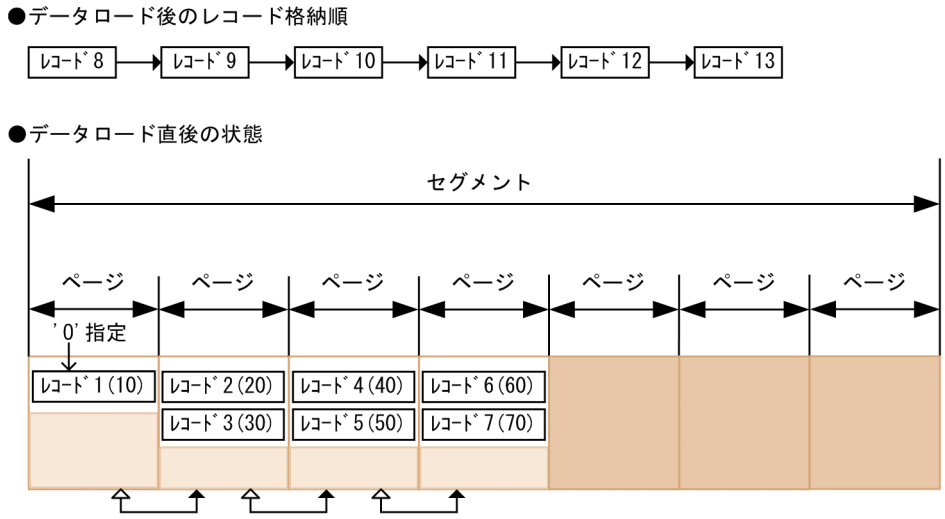
セグメント内の空きページは，次の場合に使用されます。

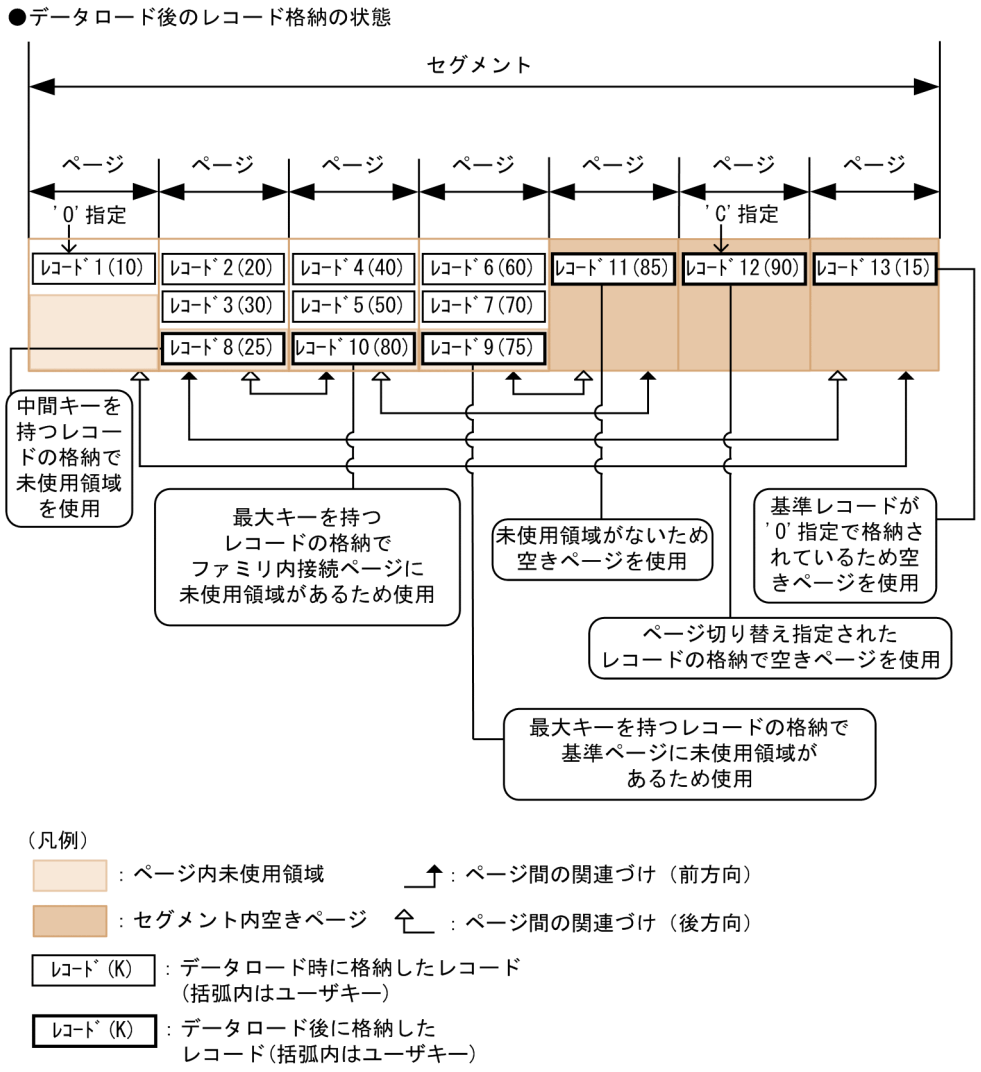
- 基準ページおよびその前後のファミリ内接続ページに未使用領域がない場合

- データロードをしたあとに，格納するレコードにページ切り替え（'O'または'C'）を指定した場合
- 基準レコードがページ切り替えの'O'指定をされたレコードのため，基準ページへの格納ができない場合
- 事前割り当てページ内にレコードを格納できる空き領域がないため，事前割り当てページへの格納ができない場合

データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定した場合）を次の図に示します。

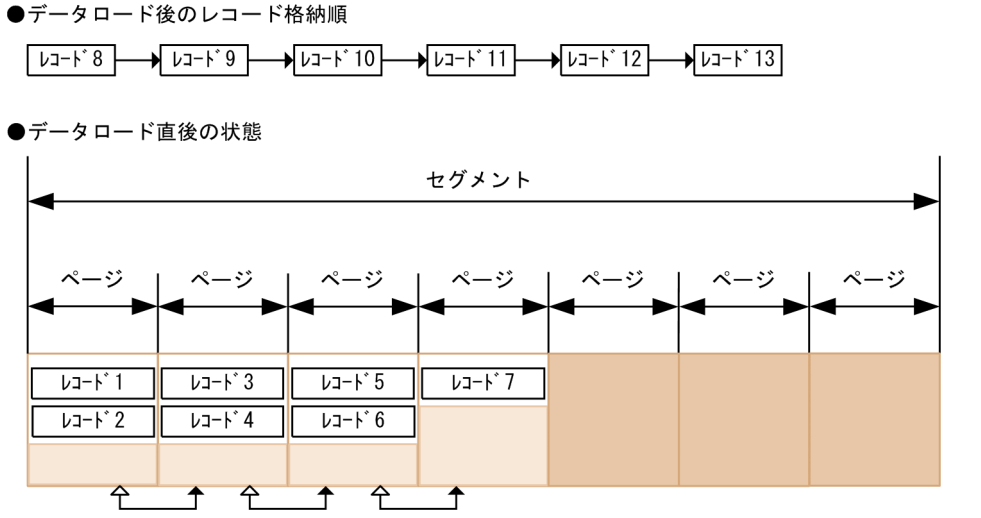
図 3-7　データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定した場合）

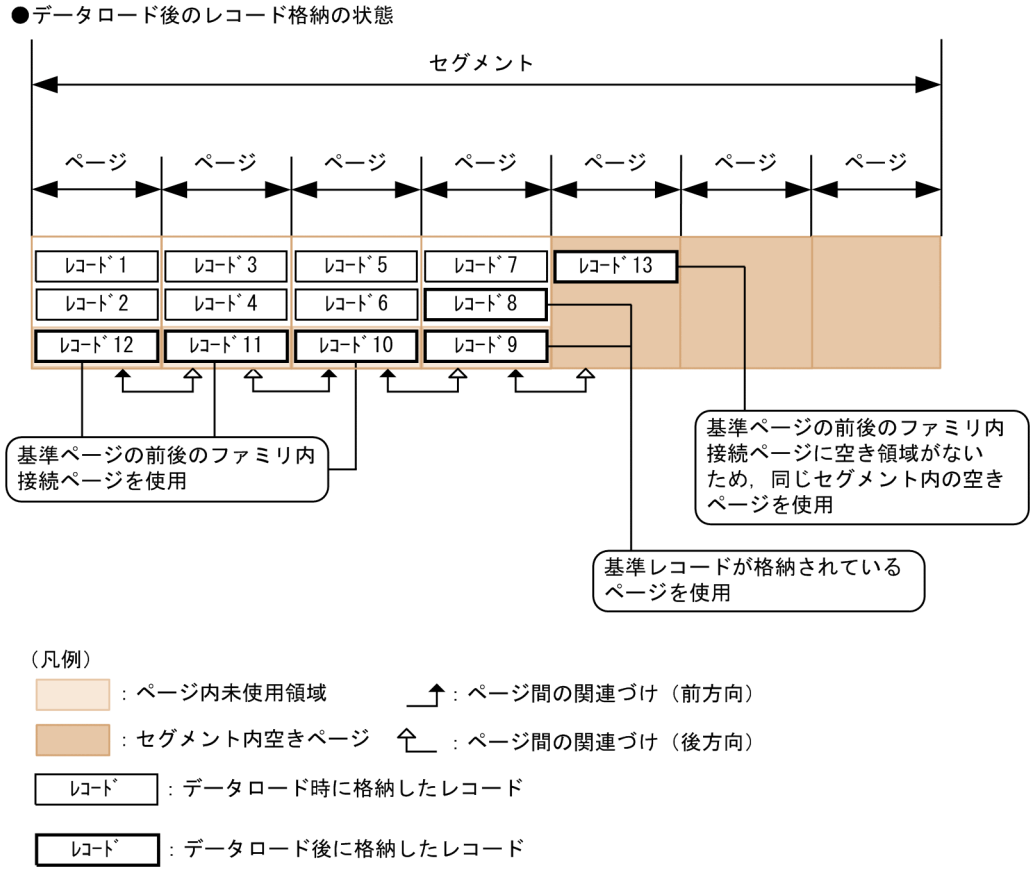




データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合）を次の図に示します。

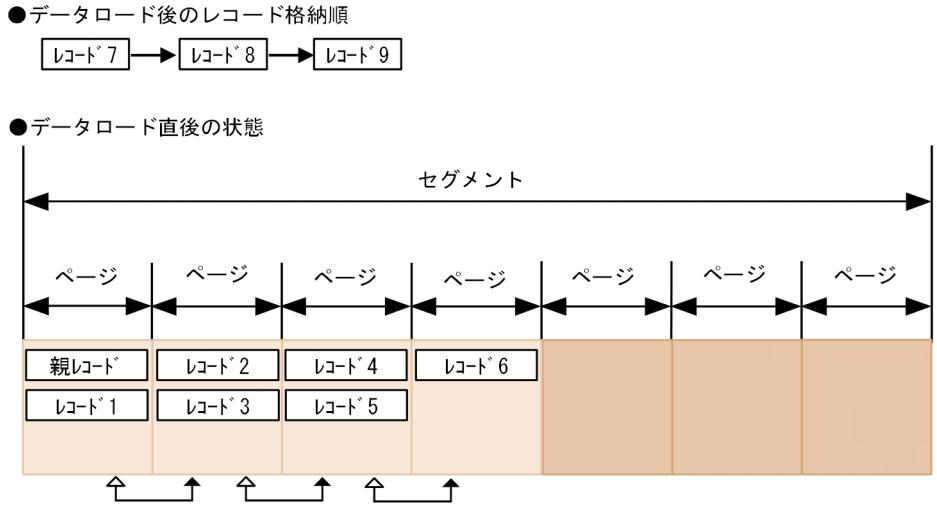
図 3-8　データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合）

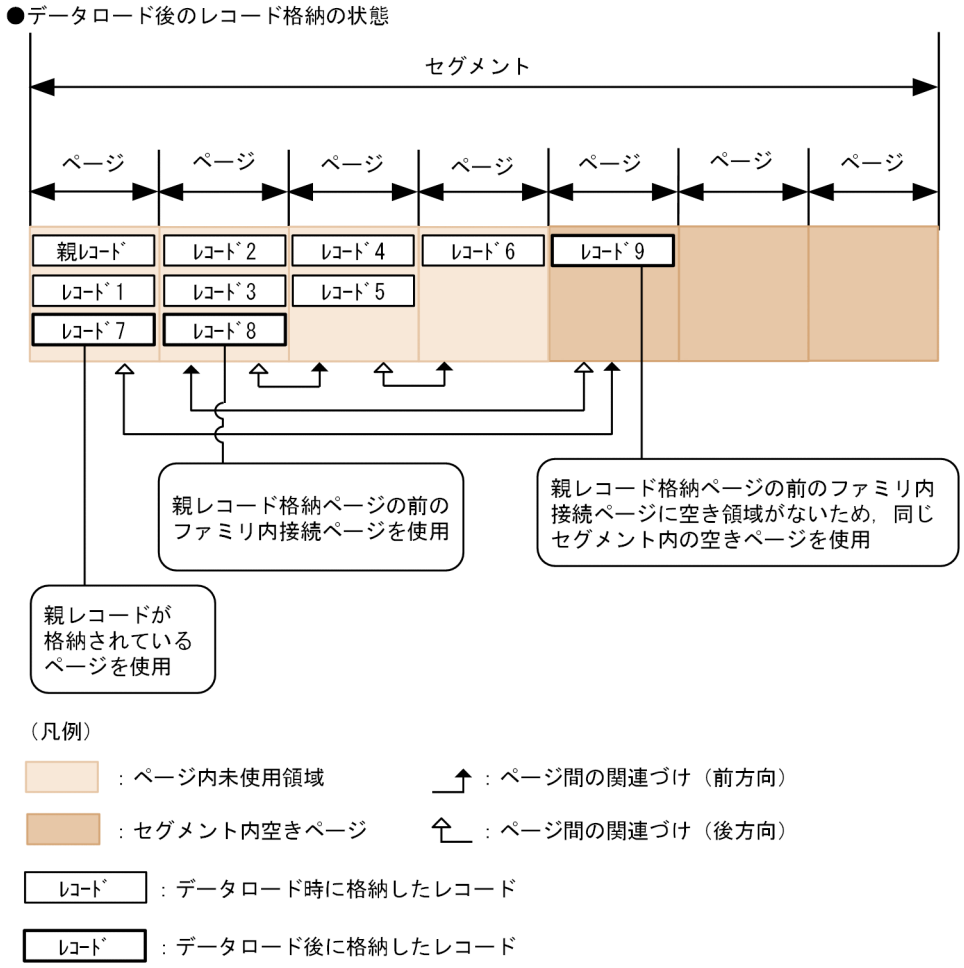




データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合）を次の図に示します。

図 3-9　データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合）





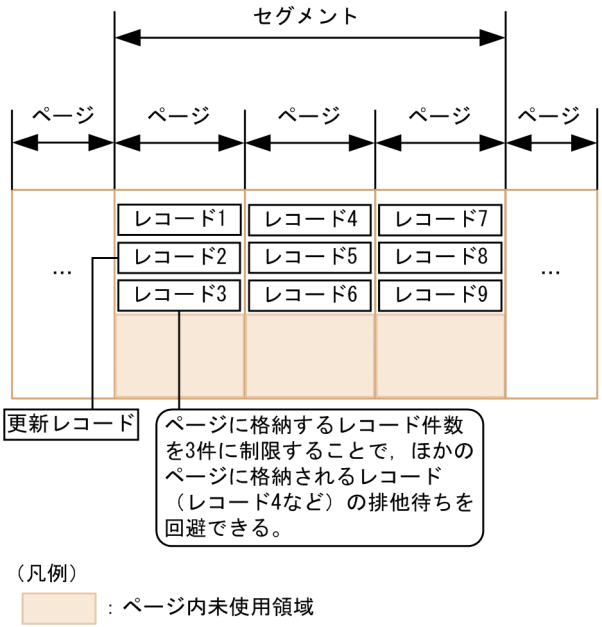
SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合，基準レコードは親レコードとなります。また，基準ページは親レコード格納ページとなります。上記の図では，基準ページを親レコード格納ページと表記しています。

(b) ページに格納するレコード件数を制限する【4V AFM】

HiRDB/SD では，複数のトランザクション間の整合性を保つために，ページ単位で排他制御をしています。そのため，1 ページ内に格納されるレコード件数が多いほど，排他の競合が発生しやすくなります。そこで，ページ内の未使用領域の比率を指定して 1 ページ内に格納するレコード件数を制限することで，排他の競合を抑えることができます。

ページ内の未使用領域の比率の指定によって，ページに格納するレコード件数を制限する例を次の図に示します。なお，ページに格納するレコード件数を制限する場合は，SDB データベースを操作する API によるレコード格納時に，PCTFREE 有効化オプションを指定する必要があります。SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

図 3-10　ページに格納するレコード件数を制限する例



(2)　ページ内の未使用領域の比率，セグメント内の空きページの比率の適用基準

ページ内の未使用領域の比率，セグメント内の空きページの比率の適用基準を次に示します。

(a)　ページ内の未使用領域の比率の適用基準

次のような場合は，ページ内の未使用領域の比率の設定を検討してください。

- SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定している場合で，データロード後に格納するレコードに，格納済みのレコードが持つユーザキーの中間のキーを持つレコードがあるとき
- ページ内に格納するレコード件数を制限して，ページによる排他の影響範囲を限定したいとき（4V AFM の SDB データベースの場合）

上記の条件に該当しないときは，ページ内の未使用領域の比率に 0 を指定してください。

(b)　セグメント内の空きページの比率の適用基準

次のような場合は，セグメント内の空きページの比率の設定を検討してください。

- SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合，または ORDER 句で SORTED DUPLICATES を指定しても，ユーザキーの昇順にレコードを格納することがわかっているとき
- データロード後に格納する子レコードが，ページ内の未使用領域だけでは格納しきれないとき
- データロード時にページ切り替えを指定していて，ページを占有したレコードがあるとき

- データロード後に，ページ切り替え指定（'C'指定または'O'指定）で格納するレコードがあるとき

上記の条件に該当しないときは，セグメント内の空きページの比率に 0 を指定してください。

(3) PCTFREE 句と recfree オペランドの使い分け

HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文で recfree オペランドを指定すると，SDB データベース格納定義の PCTFREE 句で指定したページ内の未使用領域の比率，およびセグメント内の空きページの比率は無視され，recfree オペランドの指定値に従って空き領域が作成されます。

そのため，ページ内の未使用領域の比率，およびセグメント内の空きページの比率の指定は次のように指定してください。

- データロード時に，データロード後のレコード格納に備えて空き領域を確保する場合
  recfree オペランドで指定してください。
- 4V AFM の SDB データベースで，ページに格納するレコード件数を制限する場合（PCTFREE 有効化オプションを指定して，SDB データベースを操作する API によるレコード格納時にレコード件数を制限する場合）
  SDB データベース格納定義の PCTFREE 句を指定してください。

4V AFM の SDB データベースのデータロード時に，データロード後のレコード格納に備えて空き領域を確保し，かつ SDB データベースを操作する API によるレコード格納時にレコード件数を制限したい場合は，それぞれの目的に合わせて recfree オペランド，PCTFREE 句の指定値を見積もってください。

(4) ページ内の未使用領域の比率，およびセグメント内の空きページの比率に関する留意事項

ページ内の未使用領域の比率，およびセグメント内の空きページの比率に関する留意事項を次に示します。

(a) 未使用領域の比率および空きページの比率の指定に関する留意事項

- セグメント内の空きページの比率には，50%を超える値を指定できません。レコード格納時にセグメント内の空きページを使い切った場合，HiRDB/SD は，ほかのセグメント内の空きページを割り当ててレコードを格納します。そのため，ファミリ単位の近傍配置ができないことがあります。
- 4V FMB または SD FMB の SDB データベースでは，ルートレコードの格納ページの排他によって，ルートレコード下のレコードへのアクセスをシリアライズするため，レコード件数の制限による同時実行性が向上する効果はありません。このため，レコード件数を制限する目的でのページ内の未使用領域の比率には 0 を指定してください。
- フォーマットライトを行う SDB データベースの場合，ページ内の未使用領域の比率，およびセグメント内の空きページの比率に 0 を指定してください。ただし，ページに格納するレコード件数を制限したい場合は，ページ内の未使用領域の比率を指定してください。

- 4V AFM の SDB データベースで OCCURRENCE NUMBER 句の指定が 0 の場合，仮想ルートレコード下には最大 1 件の子レコードしか存在しません。そのため，ページ内の未使用領域の比率には 0 を指定してください。

(b) 未使用領域および空きページの作成に関する留意事項

- ページ内の未使用領域の比率に関係なく，ページには最低 1 件のレコードが格納されます。このため，指定した比率の未使用領域が作成されないことがあります。
- HiRDB/SD データベース作成ユティリティ（pdsdblod）で事前割り当てページにレコードを格納する場合，ページ内の未使用領域の比率の指定は無視され，0%が指定されたものとして動作します。このため，指定した比率の未使用領域は作成されません。また，レコードは詰めて格納されます。
- HiRDB/SD データベース作成ユティリティ（pdsdblod）の実行時に，事前割り当てページを確保する際，セグメント内の空きページの比率の指定は無視され，0%が指定されたものとして動作します。このため，事前割り当てページを含むセグメント内の空きページの比率は，指定値より小さい値になることがあります。
- SDB データベースを操作する API で事前割り当てページにレコードを格納する際，PCTFREE 有効化オプションの指定によって，ページ内に未使用領域を作成することはできません。PCTFREE 有効化オプションを適用すると，SDB データベースを操作する API によるレコードの格納がエラーになります。

(c) 未使用領域へのレコード格納方法に関する留意事項

ページ内の未使用領域へのレコード格納方法は，レコード格納契機によって異なります。レコード格納契機ごとの差異を次の表に示します。

表 3-8 レコード格納契機によるレコード格納の差異

レコード格納契機	ページ内の未使用領域			
	レコード格納前		レコード格納後	
	指定比率以上の未使用領域あり	指定比率以上の未使用領域なし	指定比率以上の未使用領域あり	指定比率以上の未使用領域なし
HiRDB/SD データベース作成ユティリティ（pdsdblod）の実行時	○	×	−	−
SDB データベースを操作する API によるレコードの格納時	−	−	○	×
DML によるレコードの格納時	−	−	○※	○※

（凡例）

○：当該ページにレコードが格納されます。

×：当該ページにレコードは格納されません。

−：レコード格納の有無に関係しません。

注※

レコードを格納できるだけの未使用領域がある場合は，PCTFREE の指定に関係なくそのページにレコードが格納されます。

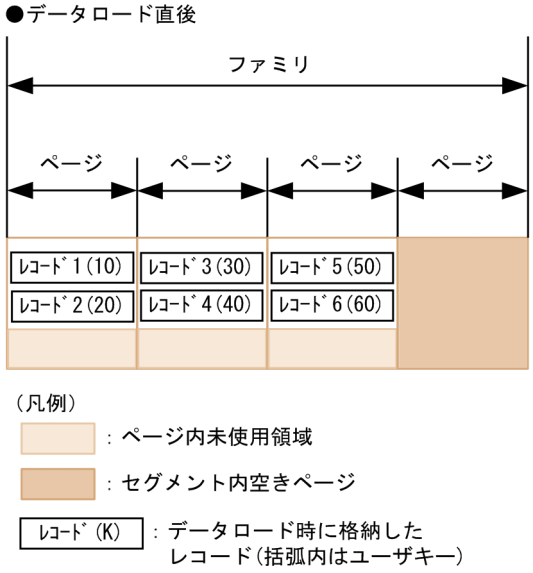
HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時には，レコード格納前のページに指定した比率の未使用領域が存在すれば，そのページにレコードを格納するため，指定した比率で未使用領域が作成されないことがあります。

(5)　データロード後のレコード格納に備えて空き領域を確保する場合の見積もり例

(a)　基本的な考え方

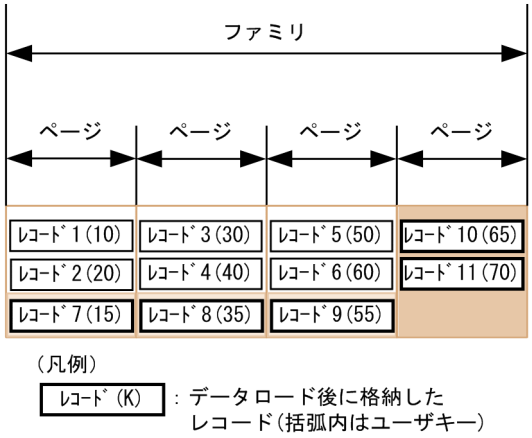
- 事前割り当てページに格納するファミリについては，除外して考えます。事前割り当てページに格納するファミリの容量の見積もりについては，「3.5.5(3)　事前割り当てページ数または事前割り当てサブページ数の算出【4V FMB】」を参照してください。
- 作成する SDB データベースの特性を考慮して，平均的なファミリを想定します。データロード時に作成した空き領域に，データロード後に格納するレコードがすべて格納できる状態とします。この状態を次の図に示します。

図 3-11　見積もりで想定するレコード格納状態



●データロード後にレコードを格納した結果

データロード後に格納したレコードが空き領域にすべて格納される。



(b)　見積もりの進め方

作成するSDBデータベースの特性から平均的なファミリを想定して，次の表に示す項目を見積もってください。レコード長の見積もりについては，「3.5.5(1)　レコード長の算出」を参照してください。

見積もり時の留意事項を次に示します。

- 4V AFMのSDBデータベースで，SDBデータベース定義のOCCURRENCE NUMBER句の指定値が1以上のレコードについては，ページ切り替え指定が'O'指定の仮想ルートレコードとして見積もってください。
- SD FMBのSDBデータベースの場合は，ページ切り替えの指定はなしとして見積もってください。

表3-9　見積もりに必要な項目

項番	項目	レコードの格納契機	ページ切り替え指定	空き領域の種別	見積もり式で使用する変数	備考
1	レコードを格納するRDエリアのページ長	−	−	−	b	−
2	レコード長	−	−	−	L	レコード型ごとに見積もります。
3	レコード件数	データロード時に格納するレコード件数	'O'指定	−	t1	平均的なファミリまたは事前割り当てページに格納するファミリについて，レコード型ごとに見積もります。
4			'C'指定	−	c1	
5			指定なし	−	n1	
6		データロード後に格納するレコード件数	'O'指定	−	t2	
7			'C'指定	−	c2	
8			指定なし	ページ内の未使用領域を使用するレコード件数	u2	

項番	項目	レコードの格納契機	ページ切り替え指定	空き領域の種別	見積もり式で使用する変数	備考
9				セグメント内の空きページを使用するレコード件数	s2	

（凡例）

－：該当しません。

「表 3-9　見積もりに必要な項目」の「空き領域の種別」で，データロード後に格納するレコードをページ内の未使用領域，およびセグメント内の空きページのどちらに見積もるかは，次の表に従って決定してください。

表 3-10　見積もり対象とする空き領域種別の決定方針

SDB データベース定義の SET 句下の指定	格納するレコードの特性	見積もり対象とする空き領域の種別
ORDER 句に SORTED DUPLICATES 指定	格納するレコードが持つユーザキーの中間のキーを持つレコード	ページ内の未使用領域
	最大のユーザキーを持つレコード	セグメント内の空きページ
ORDER 句に LAST 指定	－	セグメント内の空きページ
ORDER 句に FIRST 指定	－	セグメント内の空きページ

（凡例）

－：該当しません。

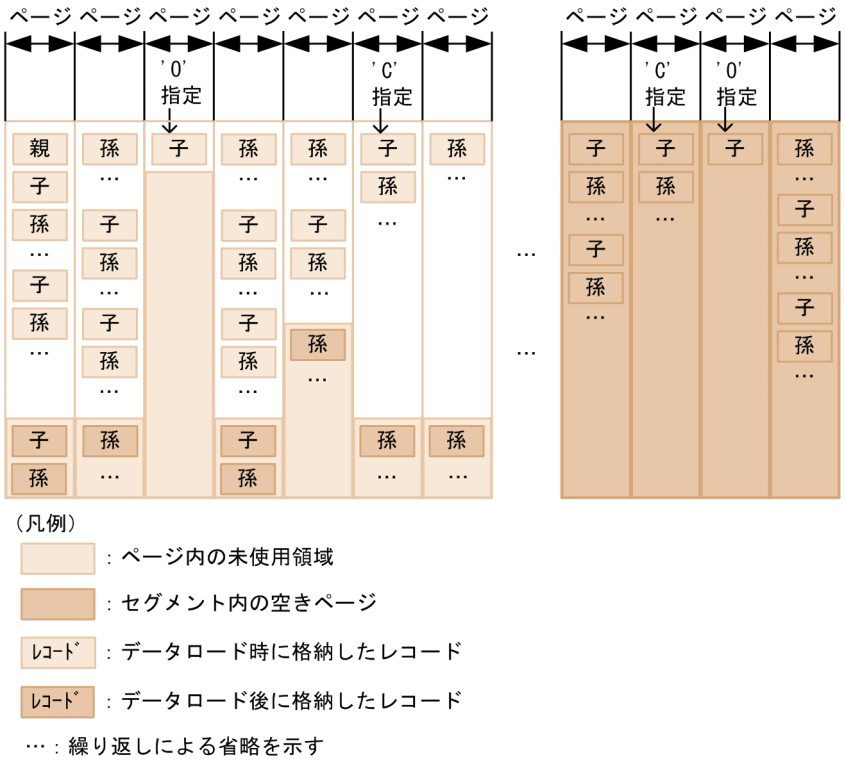
「表 3-9　見積もりに必要な項目」に従った見積もり項目のレコード長，およびレコード件数について，見積もり例を次の表に示します。

表 3-11　レコード長，およびレコード件数の見積もり例

レコード型	レコード長（バイト）	データロード時に格納するレコード			データロード後に格納するレコード			
		'O'指定（件数）	'C'指定（件数）	指定なし（件数）	'O'指定（件数）	'C'指定（件数）	指定なし	
							ページ内の未使用領域を使用するレコード（件数）	セグメント内の空きページを使用するレコード（件数）
REC1（親）	40	0	0	1	0	0	0	0
REC2（子）	100	5	5	40	1	1	8	8
REC3（孫）	200	0	0	150	0	0	40	20

「表 3-11　レコード長，およびレコード件数の見積もり例」に従った格納状態のイメージを次の図に示します。

図 3-12　見積もり例に従った格納状態のイメージ



(c)　注意事項

セグメント内の空きページの比率には，50%を超える値を指定できないため，想定したファミリの格納イメージで，セグメント内の空きページの比率が 50%を超える場合には，データロード後に格納するレコードの一部をページ内の未使用領域を使用するものとして見積もるようにしてください。ただし，ページ切り替え（'O'指定，'C'指定）をするレコードは格納時にページ内の未使用領域を使用しないため，ページ内の未使用領域を使用するものとして見積もることはできません。

(d)　見積もり式の前提

見積もり式では，「表 3-9　見積もりに必要な項目」の「見積もり式で使用する変数」で示した変数を使用します。また，「表 3-9　見積もりに必要な項目」の項番 2～9 については，レコード型ごとに見積もるため，ファミリ単位の件数の場合には，すべてのレコード型について総和が必要です。すべてのレコード型についての総和は，総和記号を使って次のように表記します。

データロード時に'0'指定で格納するレコードの総数 = Σt1

(e)　ページ内の未使用領域の比率の見積もり

ページ内の未使用領域の比率の見積もりで，すべてのレコードに対してページ切り替えを指定する場合には，見積もり値を 0 としてください。

ページ内の未使用領域の比率を見積もる場合，「表 3-9　見積もりに必要な項目」の項番 1，2，5，8 を使用します。項番 5，8 については，次の式でレコード型が使用する総領域長を計算してください。

$$Sumn1 = \sum \{ n1 \times (L+10) \}$$
$$Sumu2 = \sum \{ u2 \times (L+10) \}$$

また，項番 5，8 を使用して，次のように平均領域長を計算してください。

$$Aven1 = \uparrow \frac{Sumn1}{\sum n1} \uparrow$$
$$Aveu2 = \uparrow \frac{Sumu2}{\sum u2} \uparrow$$

上記で計算した結果を基に，次の R，A，B を計算してください。

R：ページ内の領域に対するデータロード時の未使用領域の比率

A：データロード時とデータロード後の総レコード長に対する，データロード時の総レコード長の比率

B：ページ内の領域に対するデータロード時の使用可能領域の比率

$$R = MIN\left(99,\ 100 - \downarrow \frac{A \times B}{100} \downarrow\right)$$
$$A = \downarrow \frac{Sumn1 \times 100}{Sumn1 + Sumu2} \downarrow$$
$$B = \downarrow \frac{MIN\ (b-76,\ 255 \times Aven1)\ \times 100}{b-76} \downarrow$$

ページ内の未使用領域の比率を次の式で見積もってください。

$$\text{ページ内の未使用領域の比率} = \uparrow \left\{ \frac{100 \times \left(Aveu2 \times \uparrow \frac{b \times R}{100 \times Aveu2} \uparrow \right)}{b} \right\} \uparrow$$

(f)　ページ内の未使用領域の比率の見積もり例

「表 3-11　レコード長，およびレコード件数の見積もり例」に示す値を基に，ページ内の未使用領域の比率を見積もった例を次に示します。この例では，レコードを格納する RD エリアのページ長を 4,096 バイトとして計算します。

$$Sumn1 = \{ 1 \times (40+10) \} + \{40 \times (100+10) \} + \{150 \times (200+10) \} = 35,950$$
$$Sumu2 = \{ 0 \times (40+10) \} + \{ 8 \times (100+10) \} + \{40 \times (200+10) \} = 9,280$$

$$Aven1 = \uparrow \frac{35,950}{1+40+150} \uparrow = 189$$

$$Aveu2 = \uparrow \frac{9,280}{0+8+40} \uparrow = 194$$

$$A = \downarrow \frac{35,950 \times 100}{35,950+9,280} \downarrow = 79$$

$$B = \downarrow \frac{MIN\ (4,096-76,\ 255 \times 189)\ \times 100}{4096-76} \downarrow = 100$$

$$R = \min\left(99,\ 100- \downarrow \frac{79 \times 100}{100} \downarrow\right) = 21$$

$$\text{ページ内未使用領域比率} = \uparrow \left\{ \frac{100 \times \left(194 \times \uparrow \frac{4,096 \times 21}{100 \times 194} \uparrow \right)}{4,096} \right\} \uparrow = 24\ (\%)$$

(g)　セグメント内の空きページの比率の見積もり

セグメント内の空きページの比率を見積もる場合，「表 3-9　見積もりに必要な項目」の項番 1～7 および 9 を使用して計算してください。ページ内の未使用領域の比率を C とすると，セグメント内の空きページの比率は，ファミリが使用するページ数と，データロード後に格納するレコードが使用するページ数の比率として見積もります。

項番 5，9 について，次に示す式を使用してレコード型が使用する総領域長を計算します。

$$Sumn1 = \sum \{ n1 \times (L+10) \}$$
$$Sums2 = \sum \{ s2 \times (L+10) \}$$

また，それぞれについて，次のように平均領域長を計算してください。

$$Aven1 = \uparrow \frac{Sumn1}{\sum n1} \uparrow$$
$$Aves2 = \uparrow \frac{Sums2}{\sum s2} \uparrow$$

次の計算式に従って，平均的なファミリが使用するページ数を計算してください。

$$p0 = \sum t1 + \sum c1 + \left(\uparrow \frac{\sum n1}{MIN \left\{ 255,\ \downarrow \frac{\downarrow \frac{b \times (100-C)}{100} \downarrow -76}{Aven1} \downarrow \right\}} \uparrow \right)$$

$$+ \sum t2 + \sum c2 + \left(\uparrow \frac{\sum s2}{MIN \left\{ 255,\ \downarrow \frac{b-76}{Aves2} \downarrow \right\}} \uparrow \right)$$

次の計算式に従って，データロード後に格納するレコードが使用する空きページ数を計算してください。

$$p1 = \Sigma t2 + \Sigma c2 + \left(\uparrow \frac{\Sigma s2}{\mathrm{MIN}\left\{ 255,\ \downarrow \frac{b-76}{Aves2} \downarrow \right\}} \uparrow \right)$$

以上を基に，セグメント内の空きページの比率を計算してください。

$$\text{空きページ比率} = \mathrm{MIN}\left\{ 50,\ \uparrow (1.1 \times \uparrow \frac{p1 \times 100}{p0} \uparrow) \uparrow \right\}$$

(h)　セグメント内の空きページの比率の見積もり例

「表 3-11　レコード長，およびレコード件数の見積もり例」の表に示す値を基に見積もり計算の例を次に示します。レコードを格納する RD エリアのページ長を 4,096 バイト，ページ内の未使用領域の比率を 24%として計算します。

$$Sumn1 = \{ 1 \times (40+10) \} + \{ 40 \times (100+10) \} + \{ 150 \times (200+10) \} = 35,950$$
$$Sums2 = \{ 0 \times (40+10) \} + \{ 8 \times (100+10) \} + \{ 20 \times (200+10) \} = 5,080$$

$$Aven1 = \uparrow \frac{35,950}{1+40+150} \uparrow = 189$$

$$Aves2 = \uparrow \frac{5,080}{0+8+20} \uparrow = 182$$

$$p0 = (0+5+0) + (0+5+0) + \left(\uparrow \frac{(1+40+150)}{\mathrm{MIN}\left\{ 255,\ \downarrow \frac{\downarrow \frac{4,096 \times (100-24)}{100} \downarrow -76}{189} \downarrow \right\}} \uparrow \right)$$

$$+ (0+1+0) + (0+1+0) + \left(\uparrow \frac{(0+8+20)}{\mathrm{MIN}\left\{ 255,\ \downarrow \frac{4,096-76}{182} \downarrow \right\}} \uparrow \right) = 26 \text{（ページ）}$$

$$p1 = (0+1+0) + (0+1+0) + \left(\uparrow \frac{(0+8+20)}{\mathrm{MIN}\left\{ 255,\ \downarrow \frac{4,096-76}{182} \downarrow \right\}} \uparrow \right) = 4 \text{（ページ）}$$

$$\text{空きページ比率} = \mathrm{MIN}\left\{ 50,\ \uparrow (1.1 \times \uparrow \frac{4 \times 100}{26} \uparrow) \uparrow \right\} = 18 \text{（%）}$$

(6) ページに格納するレコード件数を制限する場合のページ内の未使用領域の比率の見積もり【4V FMB，4V AFM】

(a) 見積もりの前提

- 事前割り当てページに格納するファミリについては，ページ内の未使用領域を確保しないで詰めて格納していくため，対象外とします。
- 4V AFM の SDB データベースで OCCURRENCE NUMBER 句の指定が 1 以上の場合を前提にします。4V FMB の SDB データベースの場合，または 4V AFM の SDB データベースで OCCURRENCE NUMBER 句の指定が 0 の場合は，レコード件数の制限による同時実行性の向上に効果がないため，対象外とします。
- SD FMB の SDB データベースの場合，ページに格納するレコード件数を制限できないため，対象外とします。

(b) 見積もりの進め方

作成するデータベースの中で，格納件数を制限する対象のレコード型を，次の表に従って決定してください。

表 3-12 格納件数を制限する対象のレコード型の決定

項番	親子集合の定義数	格納件数の制限対象とするレコード型	備考
1	1	子レコードのレコード型	－
2	2 以上	同時実行性が最も要求される子レコードのレコード型	SDB データベースを操作する API によるレコード格納時，対象外のレコード型に対してページ内の未使用領域を作成したくない場合は，制限対象のレコード型のレコード格納時だけ PCTFREE 句の指定を有効にしてください。 なお，HiRDB/SD データベース作成ユティリティ（pdsdblod）の recfree オペランドでは，レコード型ごとにページ内の未使用領域の比率を設定することはできません。

（凡例）

－：該当しません。

ページに格納するレコード件数を制限する場合のページ内の未使用領域の比率は次の計算式で見積もってください。レコード長の見積もりについては，「3.5.5(1) レコード長の算出」を参照してください。

なお，見積もり式では，N は 255 以下にする必要があります。また，計算結果が負となる場合は，ページに N 件のレコードを格納できないため，レコード件数を見直してください。

●API実行時の見積もり

$$未使用領域比率 = MIN\left\{99,\ 100-\downarrow\frac{100\times N}{min\left(255,\downarrow\frac{b-76}{L+10}\downarrow\right)}\downarrow\right\}$$

●HiRDB/SDデータベース作成ユティリティ（pdsdblod）実行時の見積もり

$$未使用領域比率 = MIN\left\{99,\ 100-\downarrow\frac{100\times N}{min\left(255,\downarrow\frac{b-76}{L+10}\downarrow+1\right)}\downarrow\right\}$$

（凡例）
b：レコードを格納するRDエリアのページサイズ
L：格納するレコード型のレコード長
N：ページに格納するレコード件数

(c) 留意事項

- レコードを格納する RD エリアのページ内には，HiRDB/SD がシステムで使用する制御情報が格納されています。レコードの格納，または削除による制御情報長の変化や，レコードの格納状況によっては，見積もったレコード件数よりも少ないレコードしか格納できない場合があります。
- レコード長の異なる複数のレコード型をページに格納する場合，レコードの格納状況によって，見積もったレコード件数より少ないレコードしか格納できなかったり，逆に多くのレコードが格納できたりすることがあります。
- SDB データベースを操作する API 実行時，または HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時では，見積もりの計算式が異なります。そのため，同じレコード件数に制限したい場合は，それぞれで見積もる必要があります。

  見積もり結果が異なる場合は，SDB データベース格納定義の PCTFREE 句，および HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の recfree オペランドにそれぞれの見積もり結果を指定してください。

(d) 見積もりの計算例

レコード長が 120 バイト，レコードを格納する RD エリアのページ長を 4,096 バイトとして，SDB データベースを操作する API 実行時にページに格納するレコード件数を 20 件としたい場合の見積もりの計算例を次に示します。

$$未使用領域比率 = MIN\left\{99,\ 100-\downarrow\frac{100\times 20}{MIN\left(255,\downarrow\frac{4,096-76}{120+10}\downarrow\right)}\downarrow\right\} = 34\ (\%)$$

3.2.6 レコード格納時の空き領域の作成（サブページ分割をする場合）

HiRDB/SD データベース作成ユティリティ（pdsdblod）でのレコード格納時に，サブページ内にレコードを格納しない領域（サブページ内の未使用領域）を作成したり，セグメント内に使用しない空きページ（セグメント内の空きページ）を作成したりできます。

また，SDB データベースを操作する API によるレコード格納時に，PCTFREE 有効化オプションを指定すると，SDB データベース格納定義の PCTFREE 句で指定したサブページ内の未使用領域の比率を有効にしてレコードを格納できます。PCTFREE 有効化オプションは，4V FMB または 4V AFM の SDB データベースの場合に指定できます。

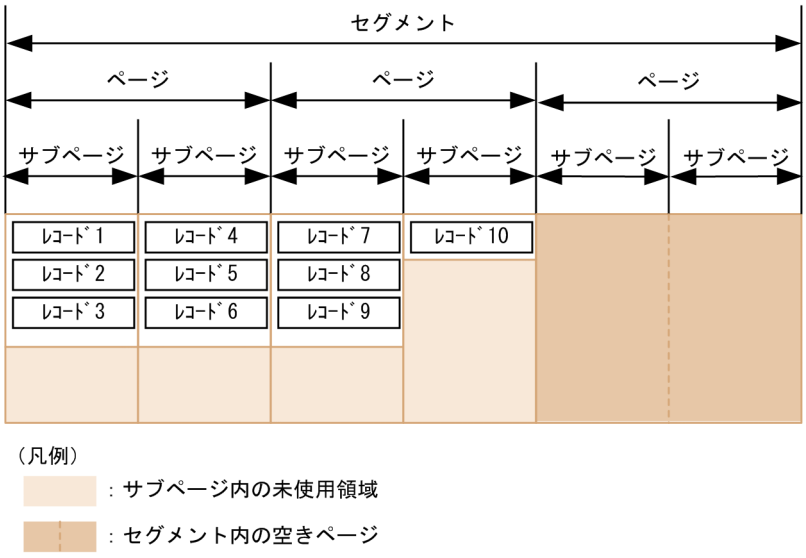
サブページ内の未使用領域の比率やセグメント内の空きページの比率は，次の個所で指定できます。

- SDB データベース格納定義の PCTFREE 句
  PCTFREE 句については，次の個所を参照してください。
  - 4V FMB の SDB データベースの場合：「11.5.2(3)(j)　PCTFREE」
  - 4V AFM の SDB データベースの場合：「11.6.2(3)(j)　PCTFREE」
  - SD FMB の SDB データベースの場合：「11.7.2(3)(j)　PCTFREE」
- HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文 recfree オペランド
  environment 文の recfree オペランドについては，「12.3.1　environment 文」を参照してください。

PCTFREE 句と recfree オペランドの使い分けについては，「(3)　PCTFREE 句と recfree オペランドの使い分け」を参照してください。

空き領域の作成イメージを次の図に示します。

図 3-13　空き領域の作成イメージ



なお，この項の説明中の「データロード」については，HiRDB/SD データベース作成ユティリティ（pdsdblod）によるフォーマットライト機能を使用したレコードの格納も含みます。

(1)　空き領域を作成する目的

空き領域を作成する目的を次に示します。

(a) データロード後のレコード格納に備えて空き領域を確保する

データロードをしたあとに，子レコードを格納する場合，データロード時に作成したサブページ内の未使用領域，セグメント内の空きページを使用することで，同一ルートレコード下のレコードの近傍に子レコードを格納できます。

サブページ内の未使用領域は，次の場合に使用されます。

- SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定し，格納済みのレコードが持つユーザキーの中間のキーを持つレコードを格納する場合
- SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合や，ORDER 句で SORTED DUPLICATES を指定して，最大のユーザキーを持つレコードを格納する場合で，基準サブページおよびそのファミリ内接続サブページに未使用領域があるとき
- SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合で，親レコード格納サブページおよびその前のファミリ内接続サブページに未使用領域があるとき

ファミリ内接続サブページについては，「2.7.1(2)　サブページ間の関連づけとファミリ内接続サブページ」を参照してください。

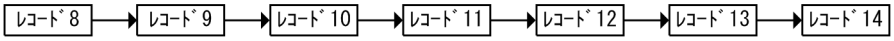
セグメント内の空きページは，次の場合に使用されます。

- 基準サブページおよびその基準サブページが属するページ内のファミリ内接続サブページに未使用領域がない場合
- データロードをしたあとに，格納するレコードにサブページ切り替え（'O'または'C'）を指定した場合
- 基準レコードがサブページ切り替えの'O'指定をされたレコードのため，基準サブページへの格納ができない場合
- 事前割り当てサブページ内にレコードを格納できる空き領域がないため，事前割り当てサブページへの格納ができない場合

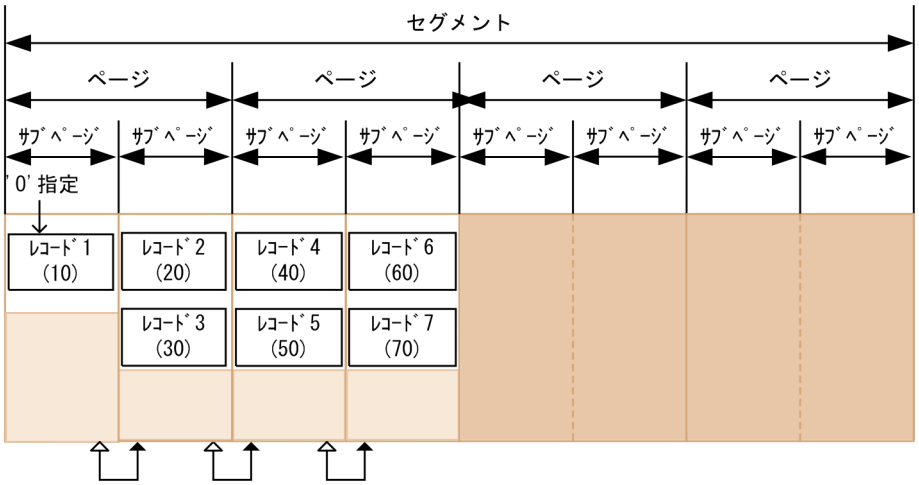
データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定した場合）を次の図に示します。

図 3-14　データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定した場合）

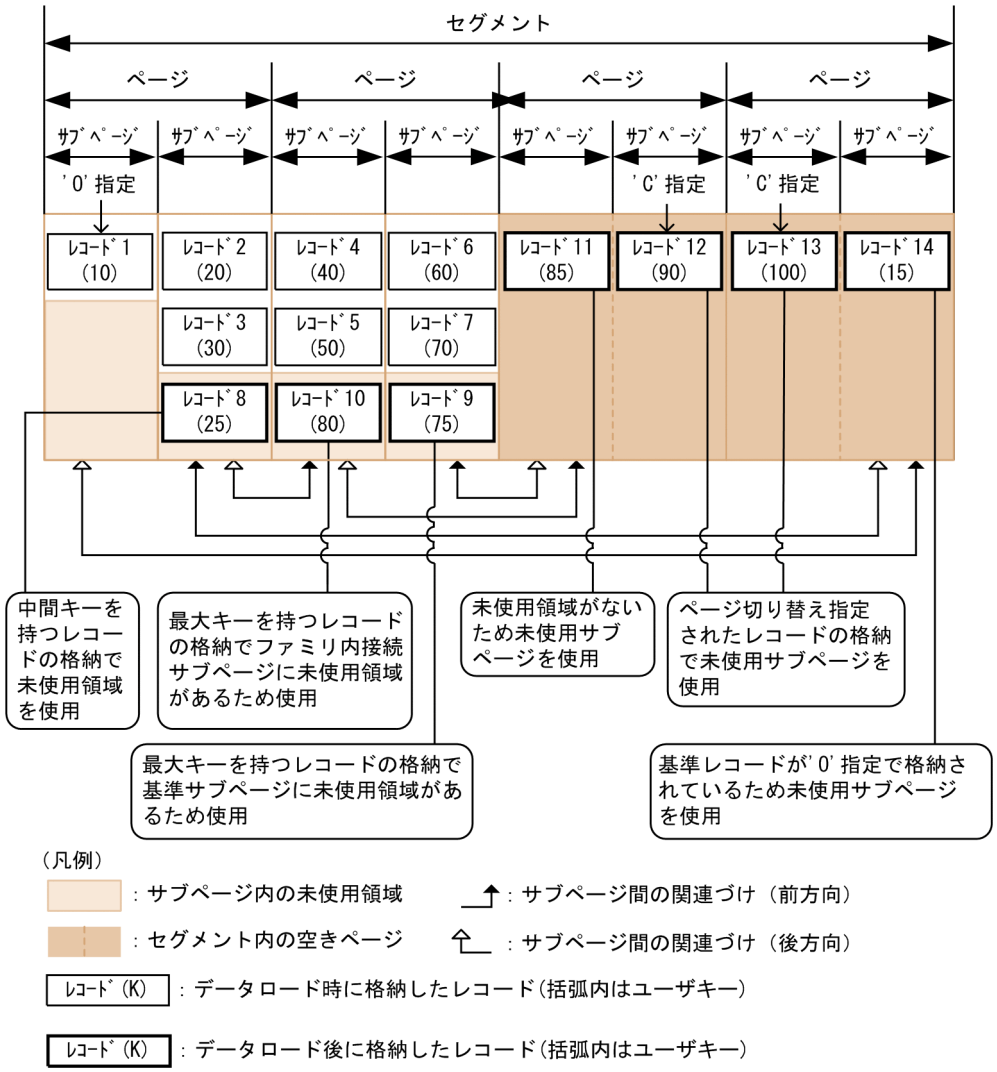
●データロード後のレコード格納順



●データロード直後の状態



●データロード後のレコード格納の状態

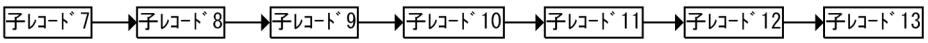


データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合）を次の図に示します。

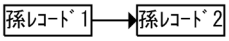
図 3-15　データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合）

●データロード後のレコード格納順

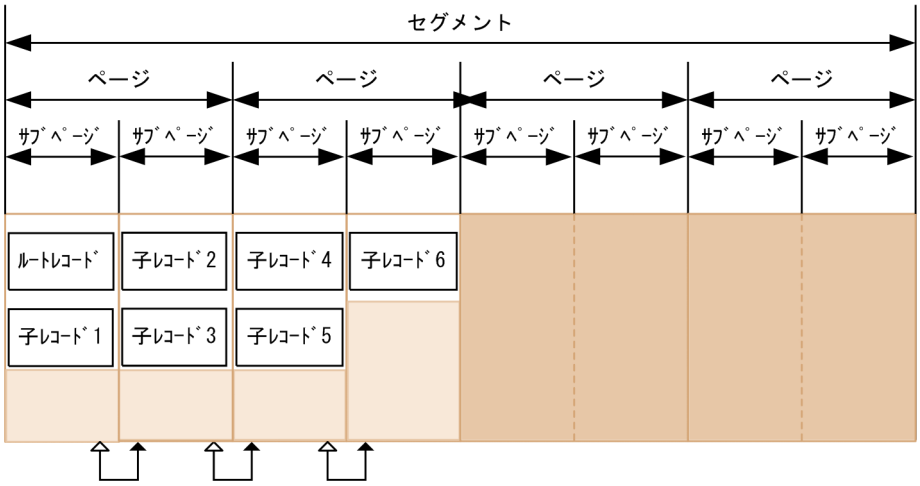
ルートレコードを親レコードとするレコード群の追加



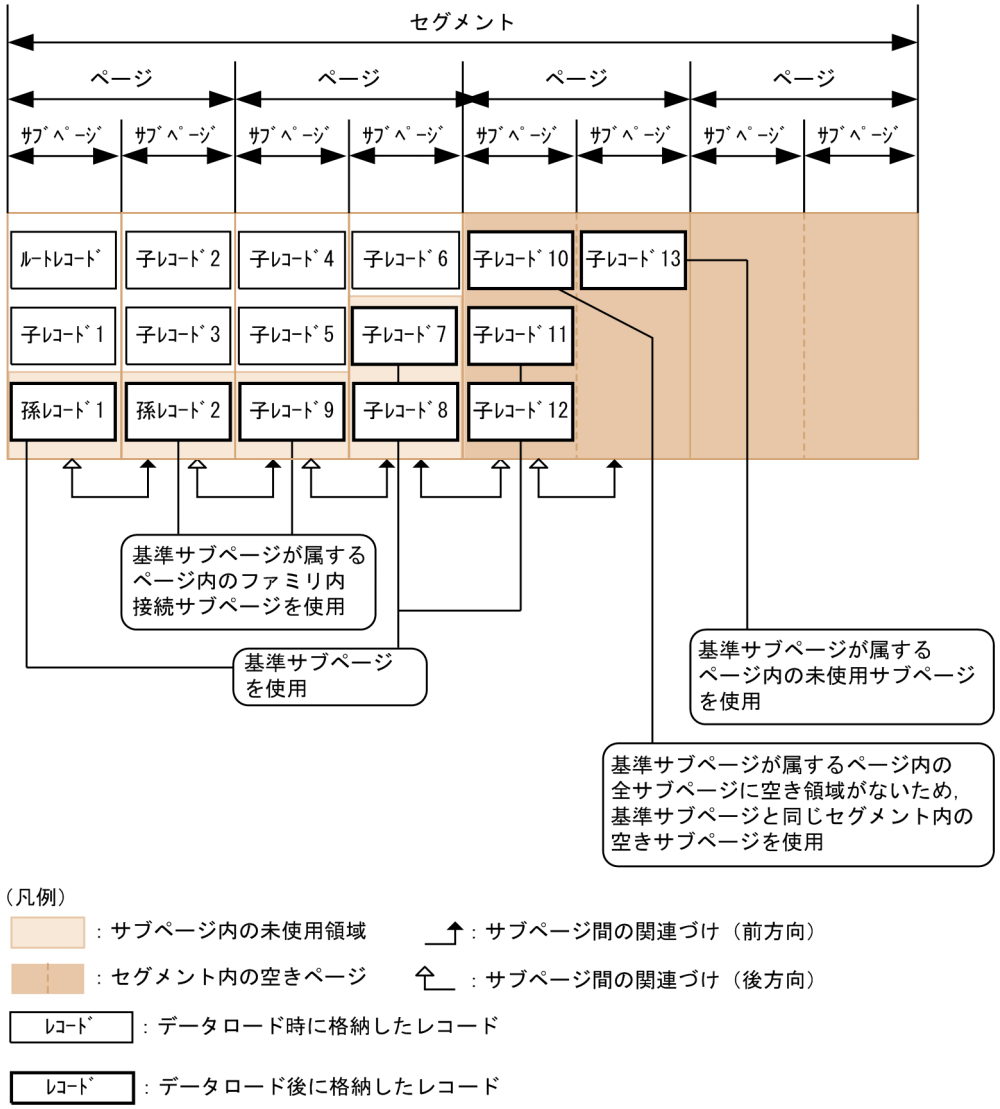
子レコード1を親レコードとするレコード群の追加



●データロード直後の状態

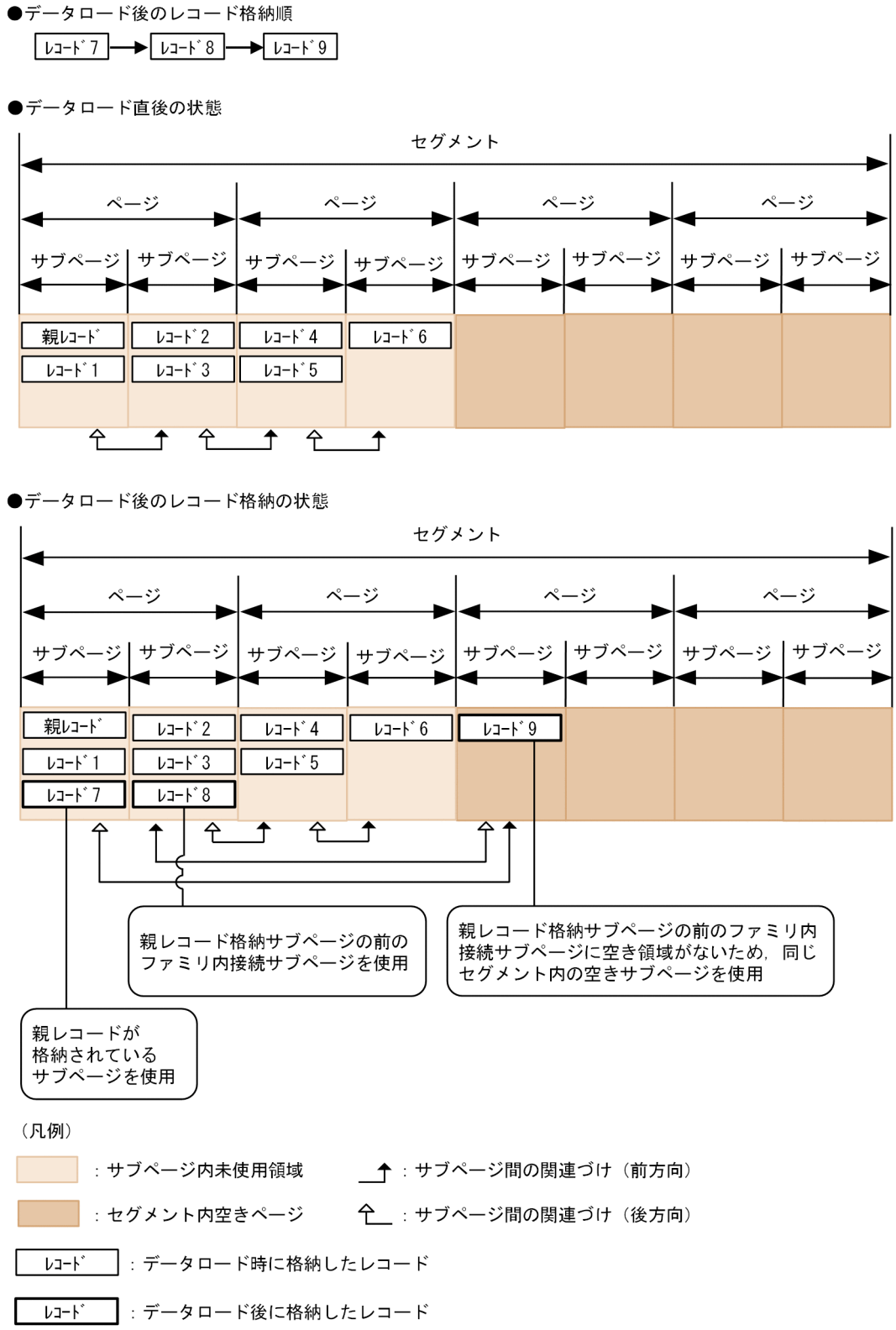


●データロード後のレコード格納の状態



データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合）を次の図に示します。

図 3-16　データロードをしたあとのレコード格納による空き領域の使用例（SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合）



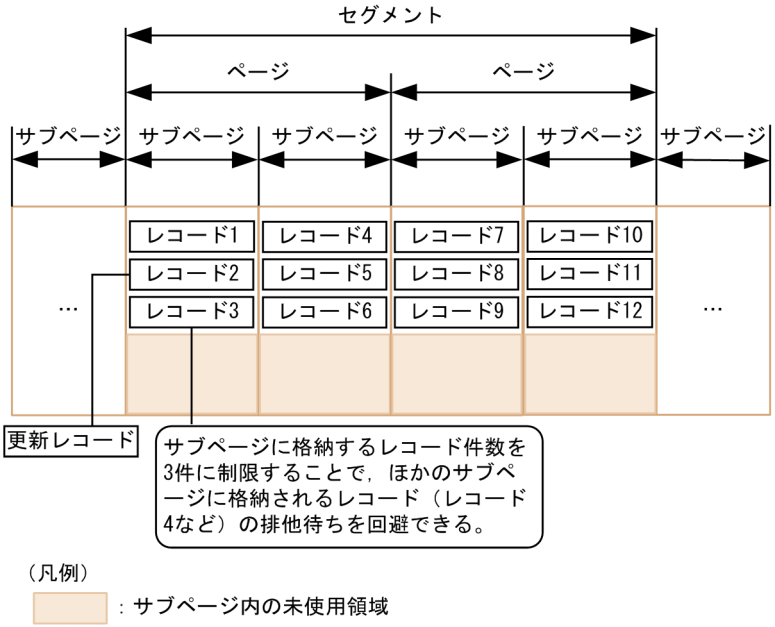
SDB データベース定義の SET 句下の ORDER 句で FIRST を指定した場合，基準レコードは親レコードとなります。また，基準サブページは親レコード格納サブページとなります。上記の図では，基準サブページを親レコード格納サブページと表記しています。

(b)　サブページに格納するレコード件数を制限する【4V AFM】

HiRDB/SD では，複数のトランザクション間の整合性を保つために，サブページ単位で排他制御をしています。そのため，1 サブページ内に格納されるレコード件数が多いほど，排他の競合が発生しやすくなります。そこで，サブページ内の未使用領域の比率を指定して 1 サブページ内に格納するレコード件数を制限することで，排他の競合を抑えることができます。

サブページ内の未使用領域の比率の指定によって，サブページに格納するレコード件数を制限する例を次の図に示します。なお，サブページに格納するレコード件数を制限する場合は，SDB データベースを操作する API によるレコード格納時に，PCTFREE 有効化オプションを指定する必要があります。SDB データベースを操作する API については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

図 3-17　サブページに格納するレコード件数を制限する例



(2)　サブページ内の未使用領域の比率，セグメント内の空きページの比率の適用基準

サブページ内の未使用領域の比率，セグメント内の空きページの比率の適用基準を次に示します。

(a)　サブページ内の未使用領域の比率の適用基準

次のような場合は，サブページ内の未使用領域の比率の設定を検討してください。

- SDB データベース定義の SET 句下の ORDER 句で SORTED DUPLICATES を指定している場合で，データロード後に格納するレコードに，格納済みのレコードが持つユーザキーの中間のキーを持つレコードがあるとき
- サブページ内に格納するレコード件数を制限して，サブページによる排他の影響範囲を限定したいとき（4V AFM の SDB データベースの場合）

上記の条件に該当しないときは，サブページ内の未使用領域の比率に 0 を指定してください。

(b) セグメント内の空きページの比率の適用基準

次のような場合は，セグメント内の空きページの比率の設定を検討してください。

- SDB データベース定義の SET 句下の ORDER 句で LAST を指定した場合，または ORDER 句で SORTED DUPLICATES を指定しても，ユーザキーの昇順にレコードを格納することがわかっているとき
- データロード後に格納する子レコードが，サブページ内の未使用領域だけでは格納しきれないとき
- データロード時にページ切り替えを指定していて，サブページを占有したレコードがあるとき
- データロード後に，ページ切り替え指定（'C'指定または'O'指定）で格納するレコードがあるとき

上記の条件に該当しないときは，セグメント内の空きページの比率に 0 を指定してください。

(3) PCTFREE 句と recfree オペランドの使い分け

HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文で recfree オペランドを指定すると，SDB データベース格納定義の PCTFREE 句で指定したサブページ内の未使用領域の比率，およびセグメント内の空きページの比率は無視され，recfree オペランドの指定値に従って空き領域が作成されます。

そのため，サブページ内の未使用領域の比率，およびセグメント内の空きページの比率の指定は次のように指定してください。

- データロード時に，データロード後のレコード格納に備えて空き領域を確保する場合
  recfree オペランドで指定してください。
- 4V AFM の SDB データベースで，サブページに格納するレコード件数を制限する場合（PCTFREE 有効化オプションを指定して，SDB データベースを操作する API によるレコード格納時にレコード件数を制限する場合）
  SDB データベース格納定義の PCTFREE 句を指定してください。

4V AFM の SDB データベースのデータロード時に，データロード後のレコード格納に備えて空き領域を確保し，かつ SDB データベースを操作する API によるレコード格納時にレコード件数を制限したい場合には，それぞれの目的に合わせて recfree オペランド，PCTFREE 句の指定値を見積もってください。

(4) サブページ内の未使用領域の比率，およびセグメント内の空きページの比率に関する留意事項

サブページ内の未使用領域の比率，およびセグメント内の空きページの比率に関する留意事項を次に示します。

(a) 未使用領域の比率および空きページの比率の指定に関する留意事項

- セグメント内の空きページの比率には，50%を超える値を指定できません。レコード格納時にセグメント内の空きページを使い切った場合，HiRDB/SD は，ほかのセグメント内の空きページを割り当ててレコードを格納します。そのため，ファミリ単位の近傍配置ができないことがあります。
- 4V FMB または SD FMB の SDB データベースでは，ルートレコードの格納サブページの排他によって，ルートレコード下のレコードへのアクセスをシリアライズするため，レコード件数の制限による同時実行性が向上する効果はありません。このため，レコード件数を制限する目的でのサブページ内の未使用領域の比率には 0 を指定してください。
- フォーマットライトを行う SDB データベースの場合，サブページ内の未使用領域の比率，およびセグメント内の空きページの比率に 0 を指定してください。ただし，サブページに格納するレコード件数を制限したい場合は，サブページ内の未使用領域の比率を指定してください。
- 4V AFM の SDB データベースで OCCURRENCE NUMBER 句の指定が 0 の場合，仮想ルートレコード下には最大 1 件の子レコードしか存在しません。そのため，サブページ内の未使用領域の比率には 0 を指定してください。

(b) 未使用領域および空きページの作成に関する留意事項

- サブページ内の未使用領域の比率に関係なく，サブページには最低 1 件のレコードが格納されます。このため，指定した比率の未使用領域が作成されないことがあります。
- HiRDB/SD データベース作成ユティリティ（pdsdblod）で事前割り当てサブページにレコードを格納する場合，サブページ内の未使用領域の比率の指定は無視され，0%が指定されたものとして動作します。このため，指定した比率の未使用領域は作成されません。また，レコードは詰めて格納されます。
- HiRDB/SD データベース作成ユティリティ（pdsdblod）の実行時に，事前割り当てサブページが属するページを確保する際，セグメント内の空きページの比率の指定は無視され，0%が指定されたものとして動作します。このため，事前割り当てサブページが属するページを含むセグメント内の空きページの比率は，指定値より小さい値になることがあります。
- SDB データベースを操作する API で事前割り当てサブページにレコードを格納する際，PCTFREE 有効化オプションの指定によって，サブページ内に未使用領域を作成することはできません。PCTFREE 有効化オプションを適用すると，SDB データベースを操作する API によるレコードの格納がエラーになります。

(c) 未使用領域へのレコード格納方法に関する留意事項

サブページ内の未使用領域へのレコード格納方法は，レコード格納契機によって異なります。次に示す契機ごとの差異を次の表に示します。

表 3-13　レコード格納契機によるレコード格納の差異

レコード格納契機	サブページ内の未使用領域			
	レコード格納前		レコード格納後	
	指定比率以上の未使用領域あり	指定比率以上の未使用領域なし	指定比率以上の未使用領域あり	指定比率以上の未使用領域なし
HiRDB/SD データベース作成ユティリティ（pdsdblod）の実行時	○	×	−	−
SDB データベースを操作する API によるレコードの格納時	−	−	○	×
DML によるレコードの格納時	−	−	○※	○※

（凡例）

○：当該サブページにレコードが格納されます。

×：当該サブページにレコードは格納されません。

−：レコード格納の有無に関係しません。

注※

レコードを格納できるだけの未使用領域がある場合は，PCTFREE の指定に関係なくそのサブページにレコードが格納されます。

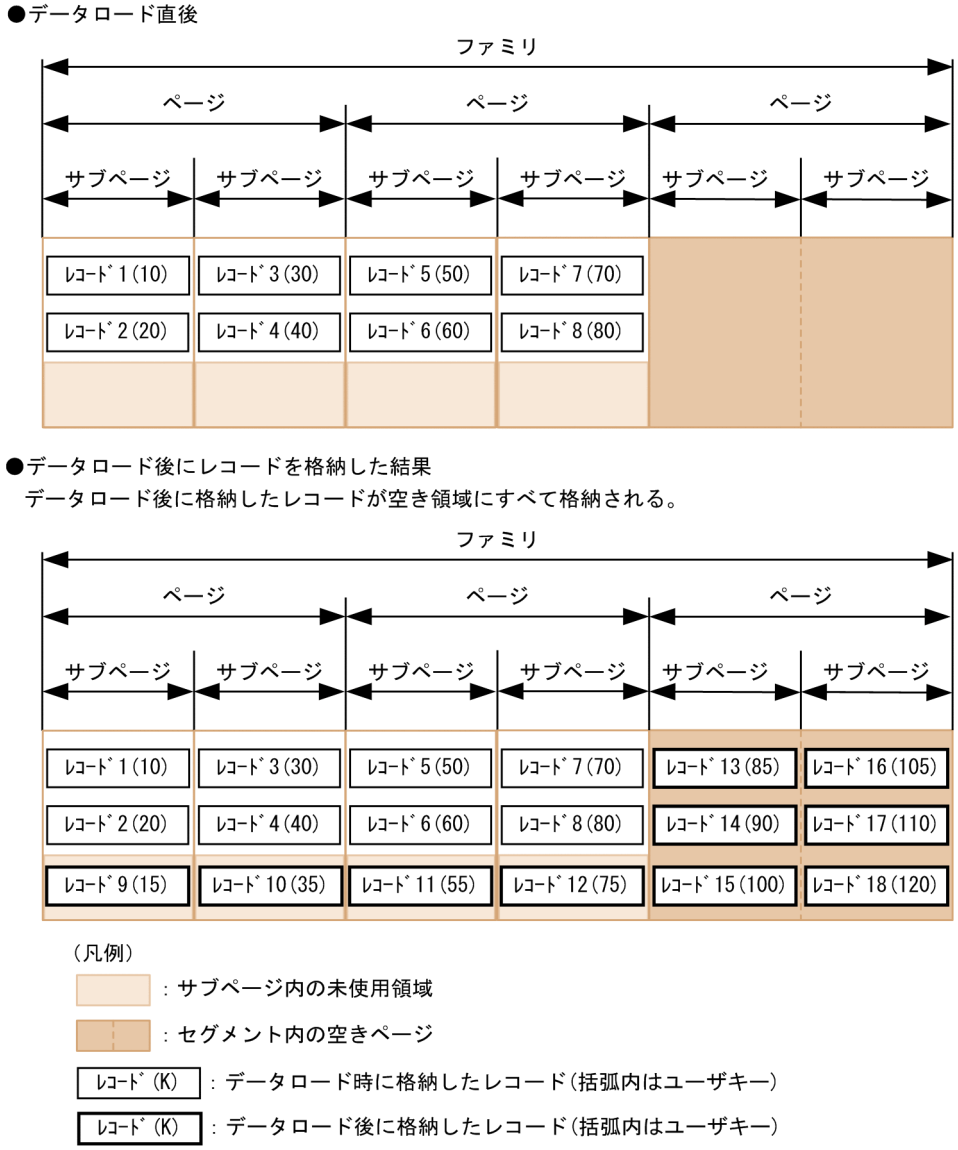
HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時には，レコード格納前のサブページに指定した比率の未使用領域が存在すれば，そのサブページにレコードを格納するため，指定した比率で未使用領域が作成されないことがあります。

(5)　データロード後のレコード格納に備えて空き領域を確保する場合の見積もり例

(a)　基本的な考え方

- 事前割り当てサブページに格納するファミリについては，除外して考えます。事前割り当てサブページに格納するファミリの容量の見積もりについては，「3.5.5(3)　事前割り当てページ数または事前割り当てサブページ数の算出【4V FMB】」を参照してください。
- 作成する SDB データベースの特性を考慮して，平均的なファミリを想定します。データロード時に作成した空き領域に，データロード後に格納するレコードがすべて格納できる状態とします。この状態を次の図に示します。

図 3-18　見積もりで想定するレコード格納状態



(b)　見積もりの進め方

作成する SDB データベースの特性から平均的なファミリを想定して，次の表に示す項目を見積もってください。レコード長の見積もりについては，「3.5.5(1)　レコード長の算出」を参照してください。

見積もり時の留意事項を次に示します。

- 4V AFM の SDB データベースで，SDB データベース定義の OCCURRENCE NUMBER 句の指定値が 1 以上のレコードについては，ページ切り替え指定が'O'指定の仮想ルートレコードとして見積もってください。
- SD FMB の SDB データベースの場合は，ページ切り替えの指定はなしとして見積もってください。

表 3-14　見積もりに必要な項目

項番	項目	レコードの格納契機	ページ切り替え指定	空き領域の種別	見積もり式で使用する変数	備考
1	レコードを格納するデータページのサブページ分割数	−	−	−	spn	−
2	レコードを格納するデータページのサブページ長	−	−	−	spl※	−
3	レコード長	−	−	−	L	レコード型ごとに見積もります。
4	レコード件数	データロード時に格納するレコード件数	'O'指定	−	t1	平均的なファミリまたは事前割り当てサブページに格納するファミリについて，レコード型ごとに見積もります。
5			'C'指定	−	c1	
6			指定なし	−	n1	
7		データロード後に格納するレコード件数	'O'指定	−	t2	
8			'C'指定	−	c2	
9			指定なし	サブページ内の未使用領域を使用するレコード件数	u2	
10				セグメント内の空きページを使用するレコード件数	s2	

（凡例）

−：該当しません。

注※

サブページ分割数によってサブページ長が決まります。サブページ長については，「3.2.4(7)　サブページ長（サブページ分割数）の設計方針」を参照してください。

「表 3-14　見積もりに必要な項目」の「空き領域の種別」で，データロード後に格納するレコードをサブページ内の未使用領域，およびセグメント内の空きページのどちらに見積もるかは，次の表に従って決定してください。

表 3-15　見積もり対象とする空き領域種別の決定方針

SDB データベース定義の SET 句下の指定	格納するレコードの特性	見積もり対象とする空き領域の種別
ORDER 句に SORTED DUPLICATES 指定	格納するレコードが持つユーザキーの中間のキーを持つレコード	サブページ内の未使用領域
	最大のユーザキーを持つレコード	セグメント内の空きページ
ORDER 句に LAST 指定	−	セグメント内の空きページ

SDB データベース定義の SET 句下の指定	格納するレコードの特性	見積もり対象とする空き領域の種別
ORDER 句に FIRST 指定	－	セグメント内の空きページ

（凡例）

－：該当しません。

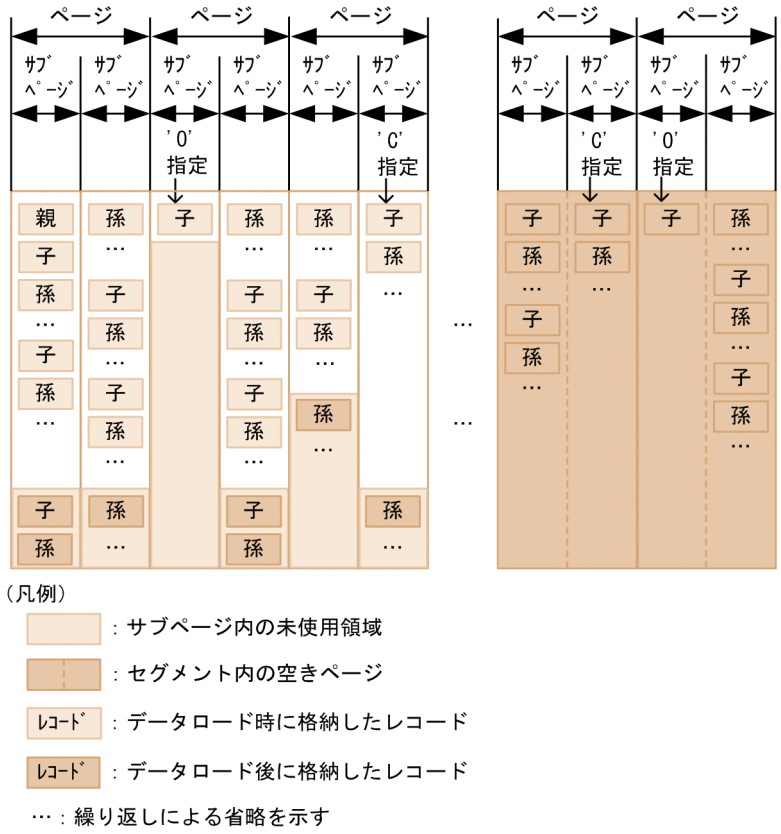
「表 3-14　見積もりに必要な項目」に従った見積もり項目のレコード長，およびレコード件数について，見積もり例を次の表に示します。

表 3-16　レコード長，およびレコード件数の見積もり例

レコード型	レコード長（バイト）	データロード時に格納するレコード			データロード後に格納するレコード			
		'O'指定（件数）	'C'指定（件数）	指定なし（件数）	'O'指定（件数）	'C'指定（件数）	指定なし	
							サブページ内の未使用領域を使用するレコード（件数）	セグメント内の空きページを使用するレコード（件数）
REC1（親）	40	0	0	1	0	0	0	0
REC2（子）	100	5	5	40	1	1	8	8
REC3（孫）	200	0	0	150	0	0	40	20

上記の表に従った格納状態のイメージを次の図に示します。

図 3-19　見積もり例に従った格納状態のイメージ



(c)　注意事項

セグメント内の空きページの比率には，50%を超える値を指定できないため，想定したファミリの格納イメージで，セグメント内の空きページの比率が 50%を超える場合には，データロード後に格納するレコードの一部をサブページ内の未使用領域を使用するものとして見積もるようにしてください。ただし，サブページ切り替え（'O'指定，'C'指定）をするレコードは格納時にサブページ内の未使用領域を使用しないため，サブページ内の未使用領域を使用するものとして見積もることはできません。

(d)　見積もり式の前提

見積もり式では，「表 3-14　見積もりに必要な項目」の「見積もり式で使用する変数」で示した変数を使用します。また，「表 3-14　見積もりに必要な項目」の項番 3～10 については，レコード型ごとに見積もるため，ファミリ単位の件数の場合には，すべてのレコード型について総和が必要です。すべてのレコード型についての総和は，総和記号を使って次のように表記します。

データロード時に'0'指定で格納するレコードの総数 ＝ ∑t1

(e)　サブページ内の未使用領域の比率の見積もり

サブページ内の未使用領域の比率の見積もりで，すべてのレコードに対してページ切り替えを指定する場合には，見積もり値を 0 としてください。

サブページ内の未使用領域の比率を見積もる場合，「表 3-14　見積もりに必要な項目」の項番 1，2，3，6，9 を使用します。項番 5，8 については，次の式でレコード型が使用する総領域長を計算してください。

$$Sumn1 = \sum \{ n1 \times (L+10) \}$$
$$Sumu2 = \sum \{ u2 \times (L+10) \}$$

また，項番 6，9 を使用して，次のように平均領域長を計算してください。

$$Aven1 = \uparrow \frac{Sumn1}{\sum n1} \uparrow$$
$$Aveu2 = \uparrow \frac{Sumu2}{\sum u2} \uparrow$$

上記で計算した結果を基に，次の R および A を計算してください。

R：サブページ内の領域に対するデータロード時の未使用領域の比率

A：データロード時とデータロード後の総レコード長に対する，データロード時の総レコード長の比率

$$R = MIN(99,\ 100-A)$$
$$A = \downarrow \frac{Sumn1 \times 100}{Sumn1+Sumu2} \downarrow$$

サブページ内の未使用領域の比率を次の式で見積もってください。

$$\text{サブページ内の未使用領域の比率} = \uparrow \left\{ \frac{100 \times \left(Aveu2 \times \uparrow \frac{spl \times R}{100 \times Aveu2} \uparrow \right)}{spl} \right\} \uparrow$$

(f)　サブページ内の未使用領域の比率の見積もり例

「表 3-16　レコード長，およびレコード件数の見積もり例」に示す値を基に，サブページ内の未使用領域の比率を見積もった例を次に示します。この例では，レコードを格納するデータページのサブページ分割数を 2 として計算します。

$$Sumn1 = \{\ 1\times(40+10)\ \}+\{40\times(100+10)\ \}+\{150\times(200+10)\ \}= 35,950$$
$$Sumu2 = \{\ 0\times(40+10)\ \}+\{\ 8\times(100+10)\ \}+\{40\times(200+10)\ \}= 9,280$$

$$Aven1 = \uparrow \frac{35,950}{1+40+150} \uparrow = 189$$

$$Aveu2 = \uparrow \frac{9,280}{0+8+40} \uparrow = 194$$

$$A = \downarrow \frac{35,950\times 100}{35,950+9,280} \downarrow = 79$$

$$R = \min\left(99,\ 100-79\right) = 21$$

$$\text{サブページ内の未使用領域の比率} = \uparrow \left\{ \frac{100\times\left(194\times\uparrow\frac{2,028\times 21}{100\times 194}\uparrow\right)}{2,028} \right\} \uparrow = 29\ (\%)$$

(g)　セグメント内の空きページの比率の見積もり

セグメント内の空きページの比率を見積もる場合，「表 3-14　見積もりに必要な項目」の項番 1～8 および 10 を使用して計算してください。サブページ内の未使用領域の比率を C とすると，セグメント内の空きページの比率は，ファミリが使用するページ数と，データロード後に格納するレコードが使用するページ数の比率として見積もります。

項番 6，9 について，次に示す式を使用してレコード型が使用する総領域長を計算します。

$$Sumn1 = \sum\ \{\ n1\times(L+10)\ \}$$
$$Sums2 = \sum\ \{\ s2\times(L+10)\ \}$$

また，それぞれについて，次のように平均領域長を計算してください。

$$Aven1 = \uparrow \frac{Sumn1}{\sum n1} \uparrow$$
$$Aves2 = \uparrow \frac{Sums2}{\sum s2} \uparrow$$

次の計算式に従って，平均的なファミリが使用するサブページ数を計算してください。

$$sp0 = \sum t1 + \sum c1 + \left(\uparrow \frac{\sum n1}{\downarrow \frac{spl\times(100-C)}{100} \downarrow - 36 \atop \downarrow \frac{}{Aven1} \downarrow} \uparrow \right)$$

$$+ \sum t2 + \sum c2 + \left(\uparrow \frac{\sum s2}{\downarrow \frac{spl-36}{Aves2} \downarrow} \uparrow \right)$$

次の計算式に従って，データロード時に格納するレコードが使用するサブページ数を計算してください。

$$sp1 = \Sigma t1 + \Sigma c1 + \left(\uparrow \frac{\Sigma n1}{\downarrow \frac{sp1 \times (100-C)}{100} \downarrow - 36 \downarrow} \uparrow \right)$$
Aven1

以上を基に，セグメント内の空きページの比率を計算してください。

$$空きページ比率 = MIN \left(50, \uparrow \frac{\left\{ \uparrow \frac{sp0}{spn} \uparrow \times 1.1 - \uparrow \frac{sp1}{spn} \uparrow \right\} \times 100}{\uparrow \frac{sp0}{spn} \uparrow} \uparrow \right)$$

(h)　セグメント内の空きページの比率の見積もり例

「表 3-16　レコード長，およびレコード件数の見積もり例」の表に示す値を基に見積もり計算の例を次に示します。レコードを格納するデータページのサブページ分割数を 2，サブページ内の未使用領域の比率を 29%として計算します。

Sumn1 ＝ ｛ 1×（40＋10）｝＋｛ 40×（100＋10）｝＋｛ 150×（200＋10）｝＝ 35,950
Sums2 ＝ ｛ 0×（40＋10）｝＋｛ 8×（100＋10）｝＋｛ 20×（200＋10）｝ ＝ 5,080

$$Aven1 = \uparrow \frac{35,950}{1+40+150} \uparrow = 189$$

$$Aves2 = \uparrow \frac{5,080}{0+8+20} \uparrow = 182$$

$$sp0 = (0+5+0) + (0+5+0) + \left(\uparrow \frac{(1+40+150)}{\downarrow \frac{\downarrow \frac{2,028 \times (100-29)}{100} \downarrow -36}{189} \downarrow} \uparrow \right)$$

$$+ (0+1+0) + (0+1+0) + \left(\uparrow \frac{(0+8+20)}{\downarrow \frac{2,028-36}{182} \downarrow} \uparrow \right) = 43（サブページ）$$

$$sp1 = (0+5+0) + (0+5+0) + \left(\uparrow \frac{(1+40+150)}{\downarrow \frac{\downarrow \frac{2,028 \times (100-29)}{100} \downarrow -36}{189} \downarrow} \uparrow \right) = 38（サブページ）$$

$$空きページ比率 = MIN \left(50, \uparrow \frac{\left(\uparrow \frac{43}{2} \uparrow \times 1.1 - \uparrow \frac{38}{2} \uparrow \right) \times 100}{\uparrow \frac{43}{2} \uparrow} \uparrow \right) = 24（%）$$

(6) サブページに格納するレコード件数を制限する場合のサブページ内の未使用領域の比率の見積もり【4V FMB，4V AFM】

(a) 見積もりの前提

- 事前割り当てサブページに格納するファミリについては，サブページ内の未使用領域を確保しないで詰めて格納していくため，対象外とします。
- 4V AFM の SDB データベースで OCCURRENCE NUMBER 句の指定が 1 以上の場合を前提にします。4V FMB の SDB データベースの場合，または 4V AFM の SDB データベースで OCCURRENCE NUMBER 句の指定が 0 の場合は，レコード件数の制限による同時実行性の向上に効果がないため，対象外とします。
- SD FMB の SDB データベースの場合，サブページに格納するレコード件数を制限できないため，対象外とします。

(b) 見積もりの進め方

作成するデータベースの中で，格納件数を制限する対象のレコード型を，次の表に従って決定してください。

表 3-17　格納件数を制限する対象のレコード型の決定

項番	親子集合の定義数	格納件数の制限対象とするレコード型	備考
1	1	子レコードのレコード型	－
2	2 以上	同時実行性が最も要求される子レコードのレコード型	SDB データベースを操作する API によるレコード格納時，対象外のレコード型に対してサブページ内の未使用領域を作成したくない場合は，制限対象のレコード型のレコード格納時だけ PCTFREE 句の指定を有効にしてください。 なお，HiRDB/SD データベース作成ユティリティ（pdsdblod）の recfree オペランドでは，レコード型ごとにサブページ内の未使用領域の比率を設定することはできません。

（凡例）

－：該当しません。

サブページに格納するレコード件数を制限する場合のサブページ内の未使用領域の比率は次の計算式で見積もってください。レコード長の見積もりについては，「3.5.5(1)　レコード長の算出」を参照してください。

なお，見積もり式で計算結果が負となる場合は，サブページに N 件のレコードを格納できないため，レコード件数を見直してください。

●API実行時の見積もり

$$未使用領域比率 = MIN\left\{99,\ 100-\downarrow\frac{100\times N}{\downarrow\frac{spl-36}{L+10}\downarrow}\downarrow\right\}$$

●HiRDB/SDデータベース作成ユティリティ（pdsdblod）実行時の見積もり

$$未使用領域比率 = MIN\left\{99,\ 100-\downarrow\frac{100\times N}{\downarrow\frac{spl-36}{L+10}\downarrow+1}\downarrow\right\}$$

（凡例）
spn：レコードを格納するデータページのサブページ分割数
spl：レコードを格納するデータページのサブページ長
L：格納するレコード型のレコード長
N：サブページに格納するレコード件数

(c)　留意事項

- レコードを格納する RD エリアのサブページ内には，HiRDB/SD がシステムで使用する制御情報が格納されています。レコードの格納，または削除による制御情報長の変化や，レコードの格納状況によっては，見積もったレコード件数よりも少ないレコードしか格納できない場合があります。
- レコード長の異なる複数のレコード型をサブページに格納する場合，レコードの格納状況によって，見積もったレコード件数より少ないレコードしか格納できなかったり，逆に多くのレコードが格納できたりすることがあります。

- SDB データベースを操作する API 実行時，または HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時では，見積もりの計算式が異なります。そのため，同じレコード件数に制限したい場合は，それぞれで見積もる必要があります。

  見積もり結果が異なる場合は，SDB データベース格納定義の PCTFREE 句，および HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の recfree オペランドにそれぞれの見積もり結果を指定してください。

(d) 見積もりの計算例

レコード長が 120 バイト，レコードを格納するデータページのサブページ分割数を 2 として，SDB データベースを操作する API 実行時にサブページに格納するレコード件数を 10 件としたい場合の見積もりの計算例を次に示します。

$$未使用領域比率 = MIN\left\{99,\ 100-\downarrow\frac{100\times10}{\downarrow\frac{2,028-36}{120+10}\downarrow}\downarrow\right\} = 34\ (\%)$$

3.3　グローバルバッファの設計

グローバルバッファは，RD エリアに格納されているディスク上のデータを入出力するためのメモリ領域（バッファ）のことで，共用メモリ上に確保されます。

システム共通定義の pdbuffer オペランドで，グローバルバッファをどの RD エリアに割り当てるかを定義します。グローバルバッファは，システムで 1,600 個まで指定できます。詳細については，マニュアル「HiRDB システム定義」の「グローバルバッファに関するオペランド」の「pdbuffer」を参照してください。

3.3.1　グローバルバッファを割り当てる単位

- すべての RD エリアにデータ用グローバルバッファを割り当てる必要があります。1 つのグローバルバッファには，1 つ以上の RD エリアを割り当ててください。
- システム用 RD エリアにもグローバルバッファを割り当ててください。
- グローバルバッファの設計方針（グローバルバッファをどのように RD エリアに割り当てるか）については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

3.3.2　グローバルバッファの割り当て方法

(1)　HiRDB の開始時にグローバルバッファを割り当てる方法

システム共通定義の pdbuffer オペランドでグローバルバッファを割り当てます。pdbuffer オペランドでのグローバルバッファの割り当て例を次に示します。

（例）

```
pdbuffer -a gbuf01 -r RDAREA01,RDAREA02 -n 1000
```

［説明］

RD エリア（RDAREA01，RDAREA02）にデータ用グローバルバッファを割り当てます。

pdbuffer オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

(2)　RD エリアの追加時にグローバルバッファを一時的に割り当てる方法

pdmod コマンドの globalbuffer オペランドで，既存のグローバルバッファを一時的に割り当てます。globalbuffer オペランドでグローバルバッファを一時的に割り当てる例を次に示します。

（例）

```
create rdarea RDAREA03
      globalbuffer gbuf01
```

```
    for user used by PUBLIC
    server name bes1
    data model structured
    page 8192 characters
    storage control segment 100 pages
    file name "/dbarea/area1/rsarea03"
        initial 1000 segments;
```

［説明］

追加する RD エリア（RDAREA03）に既存のデータ用グローバルバッファ（gbuf01）を割り当てます。

globalbuffer オペランドについては，マニュアル「HiRDB コマンドリファレンス」の「データベース構成変更ユティリティ（pdmod）」の「制御文（create rdarea 文）」を参照してください。

3.3.3 グローバルバッファを割り当てる際の考慮点

グローバルバッファを割り当てる際に次の点を考慮してください。

- ユーザ用 RD エリアにグローバルバッファを割り当てる場合，pdbuffer オペランドの-r オプションに RD エリア名を必ず指定してください。
- システム用 RD エリアにグローバルバッファを割り当てる場合，pdbuffer オペランドの-r オプションに RD エリア名を必ず指定してください。
- 次に示す RD エリアを同じグローバルバッファに割り当てることはできません。
  - SDB データベースのレコード型を格納する RD エリア
  - SDB データベースのインデクスを格納する RD エリア
  - リレーショナル DB を格納する RD エリア

  それぞれ別のグローバルバッファに割り当ててください。

  なお，リレーショナル DB を格納する RD エリアとは，次の RD エリアのことです。
  - システム用 RD エリア
  - 追い付き反映キー対応表を格納する RD エリア
  - 追い付き反映キー対応表のインデクスを格納する RD エリア
  - 追い付き状態管理表を格納する RD エリア
  - 監査証跡表を格納する RD エリア

  参考

  - create rdarea 文の data model オペランドに structured を指定した RD エリアは，SDB データベースを格納する RD エリアとなります。

- create rdarea 文の data model オペランドに relational を指定（または data model オペランドを省略）した RD エリアは，リレーショナル DB を格納する RD エリアとなります。

- SDB データベースのインデクスを格納する RD エリアに，インデクス用グローバルバッファを割り当てることはできません（pdbuffer オペランドの-i オプションは指定できません）。
- 異なるページ長の RD エリアが複数ある場合は，同じかまたは近いページ長の RD エリアをまとめて 1 つのグローバルバッファに割り当ててください。詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「異なるページ長の RD エリアが複数ある場合」を参照してください。
- 1 つのグローバルバッファに複数の RD エリアを割り当てる場合は，それぞれの RD エリアを構成する HiRDB ファイルが，同じ HiRDB ファイルシステム領域に存在する RD エリアをまとめるようにしてください。詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「1 つのグローバルバッファに複数の RD エリアを割り当てる場合」を参照してください。
- グローバルバッファのバッファ面数の設定をする場合は，共用メモリの上限，およびバッファヒット率を考慮してください。詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「グローバルバッファのバッファ面数の設定」を参照してください。
- SDB データベースを格納する RD エリアに，同時実行最大プリフェッチ数の指定はできません（pdbuffer オペランドの-m オプションは指定できません）。
- SDB データベースのレコード型を格納する RD エリアの場合，次の計算式を基にバッファ面数を計算してください。

  同時実行ユーザ数×1トランザクション内の平均同時アクセスレコード型数×4×n

  n は，どのくらいの余裕をグローバルバッファに持たせるか（グローバルバッファのヒット率をどこまで期待するか）を検討して指定してください。グローバルバッファの面数のチューニングについては，マニュアル「HiRDB システム運用ガイド」を参照してください。

  pdbuffer オペランドの各オプションについては，マニュアル「HiRDB システム定義」を参照してください。
- pdmod コマンドで割り当てたグローバルバッファは一時的な割り当てとなります。次回の HiRDB の正常開始時，または計画停止後の再開始時には無効となります。そのため，次回の HiRDB の正常開始時，または計画停止後の再開始時には，システム共通定義の pdbuffer オペランドでグローバルバッファの割り当てを変更する必要があります。

3.3.4 グローバルバッファの LRU 管理方式

HiRDB は，グローバルバッファを一括した LRU で管理します。グローバルバッファの不足時には，グローバルバッファ内のアクセスしたバッファで，最も古いバッファがメモリから追い出されます。

また，UAP ごとに LRU 管理を抑止することもできます。詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「UAP ごとの LRU 管理抑止設定」を参照してください。

3.3.5　グローバルバッファの先読み入力

グローバルバッファの先読み入力とは，指定したルートレコード型または仮想ルートレコード型のレコードおよびその子レコード群のデータや，インデクスのデータをあらかじめグローバルバッファに読み込んでおく機能です。

あらかじめ SDB データベースのデータ（レコードやインデクス）をグローバルバッファに読み込んでおくことで，ディスクからデータを読み込まないで（物理的な入出力を発生させないで），レコードにアクセスできます。入出力が多いと思われるレコードやインデクスがある場合は，グローバルバッファの先読み入力の適用を検討してください。

グローバルバッファの先読み入力には，グローバルバッファ常駐化ユティリティ（pdpgbfon）を使用します。詳細については，マニュアル「HiRDB コマンドリファレンス」の「グローバルバッファ常駐化ユティリティ（pdpgbfon）」を参照してください。

3.4　メモリ所要量の見積もり

HiRDB を使用するために必要なメモリ所要量の見積もりについて説明します。

HiRDB の各ユニットが使用するために必要なメモリ所要量は，この節に示すすべての項目を，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB/パラレルサーバのメモリ所要量の見積もり」の「メモリ所要量の計算式」に加算した値です。

3.4.1　HiRDB の各ユニットのメモリ配置

HiRDB の各ユニットのメモリ配置については，マニュアル「HiRDB システム導入・設計ガイド」の「メモリ配置」を参照してください。

3.4.2　共用メモリの見積もり

HiRDB/SD 定義ユティリティ（pdsdbdef）で生成された SDB ディレクトリ情報は，ユニットコントローラ用共用メモリ上に展開されます。このため，ユニットコントローラ用共用メモリの容量に，HiRDB が使用するメモリ所要量を加算する必要があります。

マニュアル「HiRDB システム導入・設計ガイド」の「メモリ所要量の計算式」を参照して，「ユニットコントローラ用共用メモリ中，各サーバが使用する領域」に，次の表に示す計算式から算出した値を加算してください。

表 3-18　ユニットコントローラ用共用メモリのメモリ所要量

項目		メモリ所要量（キロバイト）	変数の説明	
ユニットコントローラ用共用メモリ中，HiRDB が使用する領域	MGR，FES，または BES があるユニット	↑(unit ＋ shm×advr)÷4↑×4	unit	HiRDB が使用するユニットコントローラ用共用メモリ 詳細については，「3.4.2(1)　ユニットコントローラ用共用メモリの見積もり」を参照してください。
			shm	システム共通定義 pd_structured_shmpool_dicsize オペランドの指定値 詳細については，「3.4.2(2)　システム共通定義 pd_structured_shmpool_dicsize オペランドに指定する値」を参照してください。
			advr	システム共通定義の pd_structured_advance_resident オペランドに use を指定した場合は 2 を，nouse を指定するかまたはオペランドを省略した場合は 1 を代入してください。

項目		メモリ所要量（キロバイト）	変数の説明	
	上記以外のユニット	0	－	－

（凡例）

－：該当しません。

(1) ユニットコントローラ用共用メモリの見積もり

次の計算式から算出してください。

```
・ユニット内にFESまたはBESがない場合
  1.5
・ユニット内にFESまたはBESのどちらかがある場合
  1.5＋｛2.1＋｛0.6×（pd_max_usersオペランドの値＋3）｝｝×S
                                                （単位：キロバイト）
```

変数の説明

S：ユニット内の FES および BES の合計数

(2) システム共通定義 pd_structured_shmpool_dicsize オペランドに指定する値

pd_structured_shmpool_dicsize オペランドの指定値には，SDB ディレクトリ情報長＋ SDB データベース名管理情報長よりも大きな値を指定してください。

ただし，SDB データベース情報長の値と SDB データベース名管理情報長の値の合計値が 2,000,000 キロバイトを超える場合は，メモリに SDB ディレクトリ情報を常駐できないため，データベース構成を見直して 2,000,000 キロバイト以内に収まるようにしてください。

SDB ディレクトリ情報長，SDB データベース名管理情報，それぞれの計算式を次に示します。

(a) SDB ディレクトリ情報長の計算式

SDB ディレクトリ情報長は，次の計算式から算出してください。

$$\text{SDBディレクトリ情報長} = \sum_{i=1}^{db} \{ dbs(i) + sub(i) \} \quad （単位：キロバイト）$$

SDB ディレクトリ情報長は，HiRDB/SD 定義ユティリティ（pdsdbdef）の実行時に出力される KFPB61231-I のメッセージに表示される SDB ディレクトリ情報の合計長から求めることもできます。

変数の説明

- db：SDB データベース定義文数

- dbs：SDB データベース情報長（単位：キロバイト）

  次の計算式から算出してください。

$$
\begin{aligned}
&(1{,}008+rec\times568+mem\times80+set\times296+setkey\times24\\
&+(cmp+2)\times144+(cmptyp+2)\times32+seqidx\times200+secidx\times200\\
&+(seqkey+2)\times40+seckey\times40+allara\times248+recara\times112\\
&+idxara\times16+idxarai\times112+idxara8\times4+divrec\times24+divkey\times8\\
&+\sum_{x=1}^{divcmp}\uparrow\left(\sum_{y=1}^{divkey_x}divkel_{xy}\right)\div8\uparrow\times8+dividx\times24+dividx\times dividxkey\times8\\
&+\sum_{x=1}^{dividx}\uparrow(dividxkey\times dividxkel_x)\div8\uparrow\times8+para\times16+bes\times16+keydef\times72+keydat\times32\\
&+\sum_{x=1}^{keydef}\uparrow\left(\sum_{y=1}^{keydat_x}keydatl_{xy}\right)\div8\uparrow\times8+recpck\times16+cmppck\times8+occurs\times32)\div1{,}024
\end{aligned}
$$

- rec：SDB データベース定義 RECORD 句の指定数
- mem：SDB データベース定義 SET 句下の MEMBER 句の指定数
- set：SDB データベース定義 SET 句の指定数
- setkey：SDB データベース定義 SET 句下の KEY 句に指定した構成要素数
- cmp：SDB データベース定義 RECORD 句で指定した構成要素数

  集団項目を含みます。また，SDB データベース定義で OCCURS 句を指定する場合，システム内で生成される構成要素を含みます。
- cmptyp：SDB データベース定義 RECORD 句で TYPE 句を指定した構成要素数

  SDB データベース定義で OCCURS 句を指定する場合，システム内で生成される構成要素を含みます。
- seqidx：対応する SDB データベース格納定義 SEQUENTIAL 句の指定数
- secidx：対応する SDB データベース格納定義 STORAGE RECORD 句下の SECONDARY INDEX 句の指定数
- seqkey：対応する SDB データベース格納定義 SEQUENTIAL 句に指定した構成要素の数
- seckey：対応する SDB データベース格納定義 STORAGE RECORD 句下の SECONDARY INDEX 句に指定した構成要素の数
- allara：対応する SDB データベース格納定義に指定した RD エリア数

  格納レコード用 RD エリア数，シーケンシャルインデクス用 RD エリア数，および二次インデクス用 RD エリア数の和で，同一 RD エリア名はカウントしません。
- recara：対応する SDB データベース格納定義の STORAGE RECORD 句下の WITHIN 句に指定した格納レコード用 RD エリアの延べ数（格納レコード用 RD エリア数を格納レコード型ごとに重複排除してカウントした数の総数）
- idxara：対応する SDB データベース格納定義 STORAGE RECORD 句下に指定した次の RD エリア数の和（同一 RD エリア名はカウントしません）

  ・SEQUENTIAL 句下の WITHIN 句に指定したシーケンシャルインデクス用 RD エリア数

  ・SECONDARY INDEX 句下の WITHIN 句に指定した二次インデクス用 RD エリア数

- idxarai：対応する SDB データベース格納定義 STORAGE RECORD 句下に指定した次の RD エリア数の延べ数（インデクス格納用 RD エリア数をインデクスごとに重複排除してカウントした数の総数）

  ・SEQUENTIAL 句下の WITHIN 句に指定したシーケンシャルインデクス用 RD エリア数

  ・SECONDARY INDEX 句下の WITHIN 句に指定した二次インデクス用 RD エリア数
- idxara8：対応する SDB データベース格納定義の STORAGE RECORD 句下に指定した次の値の総数

  ・SEQUENTIAL 句下の WITHIN 句に指定したシーケンシャルインデクス用 RD エリア数を 1 インデクスごとに重複排除して 2 の倍数に切り上げた値

  ・SECONDARY INDEX 句下の WITHIN 句に指定した二次インデクス用 RD エリア数を 1 インデクスごとに重複排除して 2 の倍数に切り上げた値
- divrec：対応する SDB データベース格納定義 STORAGE RECORD 句下の WITHIN 句で格納条件または境界値を指定したレコード型数
- divkey：対応する SDB データベース格納定義 STORAGE RECORD 句下の WITHIN 句で指定したレコード型の RD エリア分割キー値数（ただし，境界値分割の場合は，レコード型の RD エリア分割キー値数＋ 1 とします）
- divcmp：対応する SDB データベース格納定義 STORAGE RECORD 句下の WITHIN 句の格納条件または DEPENDING 句に指定した構成要素数（同一構成要素名はカウントしません）
- divkel：対応する SDB データベース格納定義 STORAGE RECORD 句下の WITHIN 句で指定したレコード型の RD エリア分割キー値のキー長
- dividx：DEPENDING 句を指定したインデクス数
- dividxkey：DEPENDING 句を指定したインデクスのレコード型の RD エリア分割キー値数＋ 1
- dividxkel：DEPENDING 句を指定したインデクスのレコード型の RD エリア分割キー値のキー長
- para：1
- bes：対応する SDB データベース格納定義で指定した RD エリアが配置される BES 数（同一 RD エリア名はカウントしません）
- keydef：対応する SDB データベース格納定義 SDBOPTION 句下の KEYDEF 句の指定数
- keydat：対応する SDB データベース格納定義 SDBOPTION 句下の DATA 句の指定数
- keydatl：対応する SDB データベース格納定義 SDBOPTION 句下の DATA 句に指定したキー値長
- recpck：SDB データベース定義 RECORD 句で指定したレコード型のうち，PACKED 属性を指定した構成要素を含むレコード型の指定数
- cmppck：SDB データベース定義 RECORD 句で指定した構成要素のうち，PACKED 属性を指定した構成要素の指定数

  SDB データベース定義で OCCURS 句を指定する場合，システム内で生成される構成要素を含みます。
- occurs：SDB データベース定義の RECORD 句下の OCCURS 句の指定数

- sub：SDB データベースビュー情報長

  次の計算式から算出してください。

```
（376＋rec×312＋idxrec×16＋ownrec×16＋own×16＋mem×24
＋set×112＋setsrt×16＋setkey×8＋（cmp＋2）×56＋（anykey＋2）×8
＋seqidx×40＋secidx×24＋（seqkey＋2）×24＋seckey×16＋occurs×16）÷1,024
```

- idxrec：SDB データベース格納定義 SEQUENTIAL 句または SECONDARY INDEX 句を指定した STORAGE RECORD 句数
- ownrec：SDB データベース定義 SET 句下の OWNER 句に指定したレコード型数（同一レコード型名はカウントしません）
- own：SDB データベース定義 SET 句下の OWNER 句の指定数
- anykey：次のどれかに該当する構成要素の延べ数（1 つの構成要素が複数の条件に該当しても 1 とカウントします）

  ・SDB データベース定義 SET 句の KEY 句に指定した構成要素

  ・SDB データベース格納定義 SEQUENTIAL 句または SECONDARY INDEX 句に指定した構成要素
- setsrt：SDB データベース定義 SET 句で指定した親子集合のうち，ORDER 句に SORTED DUPLICATES PROHIBITED を指定した親子集合数

(b) SDB データベース名管理情報長の計算式

SDB データベース名管理情報長は，次の計算式から算出してください。

```
0.2×db＋（40＋（112×db））÷1,024　（単位：キロバイト）
```

変数の説明

db：SDB データベース定義数

(3) スケジューラプロセス，およびプロセスサーバが使用するメモリサイズに関する注意事項

HiRDB の開始から終了までの間にユニットコントローラが使用する共用メモリは，各 HiRDB のプロセスの項目すべてを加算します。

スケジューラプロセス，およびプロセスサーバが使用するメモリサイズを求める計算式は次のようになります。

詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「ユニットコントローラが使用する共用メモリの計算式」を参照してください。

スケジューラプロセスが使用するメモリサイズ（単位：バイト）

```
・pd_utl_exec_modeの値が0の場合
  {↑（432＋304×n）÷1,024↑＋494＋x} ×1,024
・pd_utl_exec_modeの値が1の場合
  {↑（432＋304×n）÷1,024↑＋↑（s×2,000＋136）÷1,024↑＋y} ×1,024
```

変数の説明

x：ユニット内に MGR がある場合：37

ユニット内に FES がある場合：57 ＋ 1×（s ＋ 3）＋ 14

ユニット内に DIC がある場合：102 ＋ 5×（t ＋ 3）＋ 14

ユニット内に BES がある場合：{192 ＋ 9×（u ＋ 3）＋ 14} ×BES 数

y：ユニット内に MGR がある場合：0

ユニット内に FES がある場合：1×（s ＋ 3）＋ 14

ユニット内に DIC がある場合：5×（t ＋ 3）＋ 14

ユニット内に BES がある場合：{9×（u ＋ 3）＋ 14} ×BES 数

n：ユニット内のサーバ数＋ユニット内ユティリティサーバ数＋ 1

ユニット内ユティリティサーバ数：26 ＋ α

α：ユニット内に MGR がある場合：3

ユニット内に FES がある場合：3

ユニット内に DIC がある場合：7

ユニット内に BES がある場合：BES 数×15

s：pd_max_users の値＋ pd_max_reflect_process_count の値

t：pd_max_dic_process の値＋ pd_max_reflect_process_count の値

u：pd_max_bes_process の値＋ pd_max_reflect_process_count の値

プロセスサーバが使用するメモリサイズ

```
208＋528×MAX（a, 256）＋80＋256
＋（pd_max_server_processの値＋50）×（256＋160）
＋16＋8×34＋16＋16＋64＋64×（pd_module_trace_maxの値＋1）
```

変数の説明

- a：↑（50 ＋ b ＋ c ＋ d ＋（e×ユニット内の BES 数））÷16↑×16
- b：ユニット内に MGR がある場合は 59，ない場合は f
- c：ユニット内に DS がある場合は 17，ない場合は 0
- d：ユニット内に FES がある場合は 11，ない場合は 0
- e：ユニット内に BES がある場合は 26，ない場合は 0
- f：pd_mlg_msg_log_unit オペランドに manager を指定しているときは 1，local を指定しているときは 2

3.4.3　プロセス固有メモリの見積もり

マニュアル「HiRDB システム導入・設計ガイド」の「メモリ所要量の計算式」を参照して，「HiRDB/パラレルサーバの各ユニットが使用するメモリ所要量」のうち，「各サーバプロセスが使用するプロセス固有領域」に，次の表に示す計算式から算出した値を加算してください。

表 3-19　各サーバプロセスが使用するプロセス固有領域のメモリ所要量

サーバプロセス	メモリ所要量 （単位：キロバイト）	変数の説明	
フロントエンドサーバ	psdc ＋ bpsts ＋ fdml ＋ con	psdc	1
		bpsts	0.1
		fdml	「3.4.3(1)　SDB データベースを操作する API または DML の実行時に，フロントエンドサーバで確保するメモリ」を参照して，トランザクションごとのメモリ所要量を算出してください。 算出した結果から，最大値を代入してください。
		con	「3.4.3(3)　HiRDB サーバへの接続（CONNECT）時に確保するメモリ【SD FMB】」を参照して，メモリ所要量を算出してください。 算出した結果から，最大値を代入してください。
バックエンドサーバ	psdc ＋ psda ＋ bpsts ＋ bdml ＋↑（4×MAX（mxra÷2,1,024）＋ 8×mxra）÷1,024↑	psda	102
		bdml	「3.4.3(2)　SDB データベースを操作する API または DML の実行時に，バックエンドサーバで確保するメモリ」を参照して，トランザクションごとのメモリ所要量を算出してください。 算出した結果から，最大値を代入してください。
		mxra	pd_max_rdarea_no の値を代入してください。
		上記以外の変数	フロントエンドサーバの変数の説明を参照してください。
ディクショナリサーバ	0	－	－

（凡例）

－：該当しません。

(1)　SDB データベースを操作する API または DML の実行時に，フロントエンドサーバで確保するメモリ

SDB データベースを操作する API または DML の実行時に，フロントエンドサーバで確保するメモリを次の計算式で求めます。

SDI＋308※＋0.3×個別開始要求回数　（単位：キロバイト）

注※

トラブルシュート機能（UAP 統計レポートにアクセスパス情報の出力指定）適用時に加算します。

(a) 変数 SDI の求め方

■4V FMB または 4V AFM の SDB データベースの場合

変数 SDI は次の計算式で求めます。

$$
\begin{aligned}
&262.6+(\uparrow lkey \div 1024 \uparrow +0.7)\times 2\\
&+\sum_{i=1}^{a}\left\{\begin{array}{l}\{1.2+\uparrow(maxrs(i)+sdidout(i)+maxukey(i))\div 1024\uparrow\}\times sdimlutr\\ +0.3\times fvrcpc+0.02\times bes+0.67\times divkey+4.25\end{array}\right\}\\
&+\{0.35+\uparrow(0.6+aududmax\times 8)\div 1024\uparrow\}\times audrmax\\
&+\uparrow audrsmax\div 1024\uparrow
\end{aligned}
$$

変数の説明

- lkey：操作対象のルートレコード型に定義されたシーケンシャルインデクスのキー長（バイト）
- a：個別開始要求回数
- maxrs(i)：個別開始対象 SDB データベースの最大レコード長（バイト）
  ただし，個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は 0
- sdidout(i)：個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は，複数レコードの検索時に指定したデータ格納エリア長（バイト）
  'D'または'S'を指定していない場合は 0
- maxukey(i)：個別開始対象 SDB データベースの SDB データベース定義 RECORD 句で TYPE U，K を指定した構成要素の最大構成要素長（バイト）
  TYPE U，K を指定した構成要素がない場合は 0
- sdimlutr：4V FMB の SDB データベースの場合は，1 回の API でまとめて操作要求するレコード型数（ただし，個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は 1）
  4V AFM の SDB データベースの場合は 1
- fvrcpc：4V FMB の SDB データベースの場合は 0，4V AFM の SDB データベースの場合は 1 回の容量情報取得要求で要求するキー数の最大値
- bes：個別開始対象の SDB データベースの SDB データベース格納定義で指定した RD エリアが配置される BES 数（重複を排除して計算する）
- divkey：個別開始対象 SDB データベースの SDB データベース格納定義 STORAGE RECORD 句下の WITHIN 句で指定したレコード型の RD エリア分割キー値数（ただし，境界値分割の場合は，レコード型の RD エリア分割キー値数＋ 1）
- aududmax：各 SDB データベースに定義されているレコード型に含まれるユーザデータの長さの最大値（バイト）
- audrmax：各 SDB データベースに定義されているレコード型数の最大値
- audrsmax：各 SDB データベースのレコード長の総和の最大値（バイト）

■SD FMB の SDB データベースの場合

変数 SDI は次の計算式で求めます。

$$263.3+(\uparrow lkey \div 1024 \uparrow +0.7)\times 2$$
$$+\sum_{i=1}^{a}\left\{\begin{array}{l}\{1.5+\uparrow maxrs(i)\div 1024\uparrow\}\\ +0.02\times bes+0.67\times divkey+4.25\end{array}\right\}$$
$$+\ \{0.35+\uparrow(0.6+aududmax\times 8)\div 1024\uparrow\}\ \times audrmax$$
$$+\uparrow audrsmax\div 1024\uparrow$$

変数の説明

- lkey：操作対象のルートレコード型に定義されたシーケンシャルインデクスのキー長（バイト）
- a：操作対象の SDB データベース数
- maxrs(i)：操作対象の SDB データベースの最大レコード長（バイト）
- bes：操作対象の SDB データベースの SDB データベース格納定義で指定した RD エリアが配置される BES 数（重複を排除して計算する）
- divkey：操作対象の SDB データベースの SDB データベース格納定義の STORAGE RECORD 句下の WITHIN 句で指定したレコード型の RD エリア分割キー値数（ただし，境界値分割の場合は，レコード型の RD エリア分割キー値数＋ 1）
- aududmax：各 SDB データベースに定義されているレコード型に含まれるユーザデータの長さの最大値（バイト）
- audrmax：各 SDB データベースに定義されているレコード型数の最大値
- audrsmax：各 SDB データベースのレコード長の総和の最大値（バイト）

(2) SDB データベースを操作する API または DML の実行時に，バックエンドサーバで確保するメモリ

SDB データベースを操作する API または DML の実行時に，バックエンドサーバで確保するメモリは次の計算式で求めます。

SDI＋SDP＋SDA＋0.3×個別開始要求回数　（単位：キロバイト）

(a) 変数 SDI の求め方

■4V FMB または 4V AFM の SDB データベースの場合

変数 SDI は次の計算式で求めます。

$$
\begin{aligned}
&14.9+\uparrow lkey \div 1024 \uparrow + appbdt + roglog \\
&+\sum_{i=1}^{a}\left\{\begin{array}{l}
1.3\times r(i)+0.1\times set(i)+0.01\times elmct+0.05\times divkey+nxtuksv+2.95 \\
+\uparrow \{maxrs(i)+rsize(i)\} \div 1024 \uparrow \\
+\uparrow maxrs(i) \div 1024 \uparrow \times 8 \\
+\{1.2+\uparrow \{maxrs(i)+sdidout(i)+maxukey(i)\} \div 1024 \uparrow\} \times sdimlutr
\end{array}\right\} \\
&+0.08\times audrmax \\
&+\uparrow audrsmax \div 1024 \uparrow
\end{aligned}
$$

変数の説明

- lkey：操作対象のルートレコードに定義されたシーケンシャルインデクスのキー長（バイト）
- appbdt：256

  トラブルシュート機能（UAP 統計レポートにアクセスパス情報の出力指定）適用時に加算します。
- roglog：更新可能なオンライン再編成中の SDB データベースを更新アクセスする場合は 31.6（アクセスしない場合は 0）
- a：個別開始要求回数
- r(i)：個別開始対象 SDB データベースのレコード型数

  ただし，個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は 1
- set(i)：個別開始対象 SDB データベースの親子集合数
- elmct：4V FMB の SDB データベースの場合は 0

  4V AFM の SDB データベースの場合は，レコード型ごとにデータ種別 1 が U の構成要素数をカウントしたときの最大数
- divkey：個別開始対象の SDB データベースの，SDB データベース格納定義の STORAGE RECORD 句の WITHIN 句で指定したレコード型の RD エリア分割キー値数

  ただし，境界値分割の場合は，レコード型の RD エリア分割キー値数＋ 1
- nxtuksv：4V MAM の SDB データベースの場合は 0.3

  4V MAM の SDB データベース以外の場合は 0
- maxrs(i)：個別開始対象 SDB データベースの最大レコード長（バイト）

  ただし，個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は 0
- rsize(i)：個別開始対象 SDB データベースのレコード型ごとのレコード長の総和（バイト）

  ただし，個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は，108 ＋個別開始の対象となる SDB データベースのルートレコードのレコード長（バイト）
- sdidout(i)：個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は，複数レコードの検索時に指定したデータ格納エリア長（バイト）

  'D'または'S'を指定していない場合は 0
- maxukey(i)：個別開始対象の SDB データベースの SDB データベース定義の RECORD 句で TYPE U，K を指定した構成要素の最大構成要素長（バイト）

  TYPE U，K を指定した構成要素がない場合は 0

- sdimlutr：4V FMB の SDB データベースの場合は，1 回の API でまとめて操作要求するレコード型数（ただし，個別開始の複数レコードの検索専用オプションで'D'または'S'を指定した場合は 1）
  4V AFM の SDB データベースの場合は 1
- audrmax：各 SDB データベースに定義されているレコード型数の最大値
- audrsmax：各 SDB データベースのレコード長の総和の最大値（バイト）

■SD FMB の SDB データベースの場合

変数 SDI は次の計算式で求めます。

$$16.1+\lceil lkey\div1024\rceil+appbdt+\sum_{i=1}^{a}\left\{\begin{array}{l}1.3\times r(i)+0.1\times set(i)+0.05\times divkey+2.95\\+\lceil\{maxrs(i)+rsize(i)\}\div1024\rceil\\+\lceil maxrs(i)\div1024\rceil\times8\\+\{1.5+\lceil maxrs(i)\div1024\rceil\}\end{array}\right\}+0.08\times audrmax+\lceil audrsmax\div1024\rceil$$

変数の説明

- lkey：操作対象のルートレコードに定義されたシーケンシャルインデクスのキー長（バイト）
- appbdt：256
  トラブルシュート機能（UAP 統計レポートにアクセスパス情報の出力指定）適用時に加算します。
- a：操作対象の SDB データベース数
- r(i)：操作対象の SDB データベースのレコード型数
- set(i)：操作対象の SDB データベースの親子集合数
- divkey：操作対象の SDB データベースの，SDB データベース格納定義の STORAGE RECORD 句の WITHIN 句で指定したレコード型の RD エリア分割キー値数
  ただし，境界値分割の場合は，レコード型の RD エリア分割キー値数＋ 1
- maxrs(i)：操作対象の SDB データベースの最大レコード長（バイト）
- rsize(i)：操作対象の SDB データベースのレコード型ごとのレコード長の総和（バイト）
- audrmax：各 SDB データベースに定義されているレコード型数の最大値
- audrsmax：各 SDB データベースのレコード長の総和の最大値（バイト）

(b) 変数 SDP の求め方

変数 SDP は次の計算式で求めます。

SDP個別開始時算出式＋SDP_API算出式＋API全般※　（単位：キロバイト）

- SDP 個別開始時算出式：個別開始時に必要な領域の領域長
- SDP_API 算出式：SDB データベースを操作する API または DML ごとに必要な領域の領域長
- API 全般：SDB データベースを操作する API または DML で，共通に必要な領域の領域長

注※

トラブルシュート機能（UAP 統計レポートにアクセスパス情報の出力指定）適用時に加算します。

SDP 個別開始時算出式：

$$\left\{ 0.1 + \sum_{i=1}^{a} \left\{ \begin{array}{l} 0.3 \times recvn(i) + 0.2 \times setvn(i) + 0.1 \times sdbid(i) \\ + 0.6 \times idxid(i) + 3 \times maxrl(i) + 18 \end{array} \right\} \right\} \times 1.1$$

変数の説明

- a：個別開始実行回数
- recvn(i)：SDB データベースに定義したレコード型数
- setvn(i)：SDB データベースに定義した親子集合数
- sdbid(i)：SDB データベースの基 SDB データベース番号
- idxid(i)：インデクス定義総数－1
- maxrl(i)：31

SDP_API 算出式：

$$\sum_{i=1}^{a} (\text{API共通算出式}(i) + \text{API個別算出式}(i))$$

変数の説明

- a：個別開始実行回数
- API 共通算出式(i)：複数の SDB データベースを操作する API または DML 間で共通に利用する領域の領域長

  個別開始した SDB データベースについて算出します。
- API 個別算出式(i)：SDB データベースを操作する API または DML の種別ごとに確保する領域長

  個別開始した SDB データベースについて算出します。

API 共通算出式(i)：

個別開始から個別終了の間で実行する要求を対象に求めます。

$$\{0.001 \times lkey + MAX1^{※1} + MAX2^{※1} + MAX3 + MAX4 + MAX5 + 0.55^{※2}\} \times 1.1$$

（単位：キロバイト）

注※1

個別開始後の初回のレコードの検索，位置指示子の位置づけ，複数レコードの検索，またはレコードの格納時に確保します。以後のレコードの検索，位置指示子の位置づけ，複数レコードの検索，またはレコードの格納時に領域長が不足した場合は，領域を解放して再取得します。個別終了で領域を解放します。

注※2

個別開始後の初回 API または DML の実行時に確保し，個別終了で領域を解放します。

変数の説明

- lkey：操作対象のルートレコードに定義されたシーケンシャルインデクスのキー長（バイト）
- MAX1：0.55 ＋ 0.1×ccnd ＋ 0.1×faltn ＋ 0.1×cndn

  個別開始から個別終了の間に発行する要求に対して上記の計算を行い，最も大きい値を MAX1 としてください。

  ccnd：共通条件（キーの条件）数

  faltn：択一条件数

  cndn：連結条件数
- MAX2：5.03 ＋（0.248 ＋ 96×nidx）×（dbkcnd ＋ 2）

  個別開始から個別終了の間に発行する要求に対して上記の計算を行い，最も大きい値を MAX2 としてください。

  nidx：インデクスの構成列数

  dbkcnd：シーケンシャルインデクスのキーに対する条件の数
- MAX3：（0.02 ＋ 0.002×nrec）×2

  個別開始から個別終了の間に発行する要求に対して上記の計算を行い，最も大きい値を MAX3 としてください。

  nrec：操作対象の SDB データベースを構成するレコード型数
- MAX4：0.15×nset

  個別開始から個別終了の間に発行する要求に対して上記の計算を行い，最も大きい値を MAX4 としてください。

  nset：操作対象のレコードが子レコードとして関与する親子集合型数
- MAX5：0.4 ＋（0.1×ncmp）

  個別開始から個別終了の間に発行する要求に対して上記の計算を行い，最も大きい値を MAX5 としてください。

  ncmp：SDB データベース定義の KEY 句に指定した構成要素型数

API 個別算出式(i)：

```
0.83
```

API 全般：

```
546　（単位：キロバイト）
```

(c)　変数 SDA の求め方

変数 SDA は次の計算式で求めます。

A＋B　（単位：キロバイト）

計算式A：↑（48,228＋xpgsz×4）÷1,024↑

計算式B：$\uparrow\{\sum_{i=1}^{a}(1,256+72\times r(i))\}\div 1,024\uparrow$

SD FMB の SDB データベースの場合，計算式 B の値は 0 になります。

変数の説明

- xpgsz：インデクスを格納している RD エリアの中で最大のページ長（バイト）
- a：複数レコードの検索専用オプションで'D'または'S'を指定した個別開始要求回数
- r(i)：個別開始対象データベースのレコード型数

> **参考**
>
> 計算式 B は，バックエンドサーバ用の簡易ダンプファイルの容量計算式（FETCHDB ALL 用ワークエリアサイズの計算式）として使用します。バックエンドサーバ用の簡易ダンプファイルの容量計算式については，「付録 F.1　簡易ダンプ」を参照してください。

(3)　HiRDB サーバへの接続（CONNECT）時に確保するメモリ【SD FMB】

HiRDB サーバへの接続（CONNECT）時に確保するメモリは次の計算式で求めます。

SDA×1.1　（単位：キロバイト）

SD FMB の SDB データベースに SDB 用 UAP 環境定義を使用してアクセスする場合に，上記の計算をしてください。SDB 用 UAP 環境定義を使用しない場合は，変数 SDA は 0 になります。

(a)　変数 SDA の求め方

変数 SDA は次の計算式で求めます。

oper＋↑（subn×（0.2＋0.04×rda）×1.34）÷36↑×36＋subn×（0.2＋0.04×rda）＋163　（単位：キロバイト）

変数の説明

- oper：SDB 用 UAP 環境定義で，1 オペランドの継続行数が 100 行を超えるオペランドがない場合は 8，100 行を超えるオペランドがある場合は 160
- subn：SDB 用 UAP 環境定義の subschema オペランドの指定数
- rda：SDB 用 UAP 環境定義の subschema オペランドに指定する，RD エリア名の指定数の最大値

3.5　ディスク容量の見積もり

HiRDB/SD を使用するために必要なディスク容量の見積もりについて説明します。

3.5.1　システムログ量の見積もり

システムログ量の見積もりについて説明します。

(1)　システムログ量の見積もり式

出力されるシステムログ量は次の計算式から求めます。

$$\text{システムログ量} = \uparrow\left\{\sum_{i=1}^{n} Li\right\} \times \{1+(r \div n)\} \times 1.1\uparrow \quad (\text{単位：バイト})$$

変数の説明

- Li：SDB データベースを操作する API または DML を 1 回実行したときのシステムログ量，またはユティリティを 1 回実行したときのシステムログ量
- n：SDB データベースを操作する API または DML，およびユティリティの実行回数
- r：ロールバック対象となる SDB データベースを操作する API または DML，およびロールバック対象となるユティリティの数

SDB データベースを操作する API または DML の実行時のシステムログ量については，「(2)　SDB データベースを操作する API または DML の実行時のシステムログ量」および「(3)　更新可能なオンライン再編成実行時のシステムログ量【4V FMB】」を参照してください。

ユティリティ実行時のシステムログ量については，「(4)　HiRDB/SD 定義ユティリティ（pdsdbdef）実行時のシステムログ量」および「(5)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のシステムログ量」を参照してください。

ここで算出した値をシステムログファイルの総容量に加算してください。

システムログファイルの総容量，および次に示す以外のシステムログファイルの容量については，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

(2)　SDB データベースを操作する API または DML の実行時のシステムログ量

SDB データベースを操作する API または DML の実行時のシステムログ量について，次の表に示します。

なお，表中にあるレコード長については，「3.5.5(1)　レコード長の算出」を参照して求めてください。

表 3-20　SDB データベースを操作する API または DML の実行時のシステムログ量

実行する API または DML	SDB データベース種別と格納するレコードの条件	更新前ログ（単位：バイト）	更新後ログ（単位：バイト）	説明
レコードの格納（STORE）	4V FMB または SD FMB で，ルートレコードを格納する場合※4	1,556 ＋（↑976÷sbn↑＋416）×ppn	レコード長＋1,400 ＋（↑976÷sbn↑＋416）×ppn	sbn：サブページ分割数。サブページ分割をしない場合は 1 を代入してください。 ppn：確保予定の事前割り当てページまたは事前割り当てサブページの数。事前ページ割り当て機能を適用しない場合は 0 を代入してください。
	4V FMB または SD FMB で，ルートレコード以外のレコードを格納する場合※5	2,908×G ＋（↑976÷sbn↑＋416）×ppn	（レコード長＋2,768）×G ＋（↑976÷sbn↑＋416）×ppn	G には，SDB データベース定義で定義したレコード型の数が 3 以上の場合，1.2 を指定してください。そのほかの場合は，1 を指定してください。ただし，各レコード型のレコード長にばらつきがある場合，ERASE によってできた空き領域を再利用するとき，計算結果以上にログ容量が増加することがあります。このように，容量の増加が大幅に見込まれる場合，1.2 より大きい値を指定してください。 sbn：サブページ分割数。サブページ分割をしない場合は 1 を代入してください。 ppn：追加で確保予定の事前割り当てページまたは事前割り当てサブページの数。事前ページ割り当て機能を適用しない場合は 0 を代入してください。
	4V MAM の SDB データベースのレコードを格納する場合※1※2	2,712	レコード長＋2,572	－
	4V DAM または 4V SAM の SDB データベースで，同一データベースキー内のレコード件数が過去に格納したレコード件数の最大値を超えるレコードを格納する場合	4V MAM の SDB データベースのレコードを格納する場合と同じです。		－
	4V DAM または 4V SAM の SDB データベースで，同一データベースキー内のレコード件数が過去に格納したレコード件数の最大値を超えないレコードを格納する場合※3	「レコードの更新（MODIFY）」を参照して，見積もってください。ただし，過去に格納したレコード件数の最大値を想定できない場合は，「4V MAM の SDB データベースのレコードを格納する場合」を参照して見積もってください。		－

実行する API または DML	SDB データベース種別と格納するレコードの条件	更新前ログ (単位：バイト)	更新後ログ (単位：バイト)	説明
レコードの更新（MODIFY）	－	レコード長＋ 172 ＋ u ＋ ov ＋ su		u：USER ポインタ更新時は 184，USER ポインタの更新が不要なときは 0 を代入してください。 ov：4V DAM または 4V SAM の SDB データベースの場合で，同一データベースキー内のレコード件数が，過去に格納したレコード件数の最大値を超えないレコード格納の場合は 184，それ以外は 0 を代入してください。 su：SD FMB の SDB データベースで，キー項目を更新する場合は 1,376，それ以外は 0 を代入してください。
レコードの削除（ERASE）	事前ページ割り当て機能を適用したファミリのルートレコードを削除する場合	レコード長＋ 1,732 ＋（↑148÷sbn↑＋ 236）×ppn	1,728 ＋（↑148÷sbn↑＋ 236）×ppn	sbn：サブページ分割数。サブページ分割をしない場合は 1 を代入してください。 ppn：確保済みの事前割り当てページまたは事前割り当てサブページの数。
	事前ページ割り当て機能を適用したファミリのルートレコード以外のレコードを削除する場合	レコード長＋ 1,740	1,736	－
レコードの一括削除	－	184	184	－

（凡例）

－：該当しません。

注※1

仮想ルートレコードを格納する場合は，次の値を加算してください。

更新前ログ：1,368 バイト

更新後ログ：仮想ルートレコード長＋ 1,212 バイト

注※2

仮想ルートレコードを格納しない場合は，次の値を加算してください。

更新前ログ：144 バイト

注※3

「レコードの更新（MODIFY）」を参照して見積もる場合は，次の値を加算してください。

更新前ログ：144 バイト

注※4

システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定している場合は，次の値を加算してください。

更新後ログ：156 バイト

注※5

次の条件をすべて満たす場合は，次の値を加算してください。

・システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定している
・SD FMB の SDB データベースである
・二次インデクスを定義している
更新後ログ：156 バイト

インデクスログ量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「インデクスログ量の見積もり」を参照してください。「インデクスログ量の見積もり」を参照して計算する際の注意点を次に示します。

- 「表」を「レコード型」に読み替えて計算してください。
- 「行」を「レコード」に読み替えて計算してください。
- 新しいキーを追加する場合のインデクスログ量の計算式は，「インデクスログ量の見積もり」の「1 本のインデクスログ量の計算式」で説明している「インデクスキー無排他を適用した UNIQUE 指定のインデクスのときの計算式」を使用してください。
- インデクスログ量の見積もりをする際，インデクスのキー長が必要になります。インデクスのキー長には，「3.5.5(8)　インデクスを格納するページ数の見積もり」で説明している「キーとなる構成要素の長さ」の計算結果を代入してください。
- 4V AFM の SDB データベースのレコードの格納（STORE）時に，仮想ルートレコードを格納する場合は，シーケンシャルインデクスのキーが 2 つ追加されるため，該当するシステムログ量を 2 倍して見積もってください。
- 4V AFM の SDB データベースのレコードの格納（STORE）時に，仮想ルートレコードを格納しない場合※で，かつ 1 件目のレコードを格納する場合は，シーケンシャルインデクスのキーが 1 つ追加されるため，新しいキーが追加されるときのシステムログ量を見積もり，その値を加算してください。

  注※

  4V DAM または 4V SAM の SDB データベースの場合，同一データベースキー内のレコード件数が，過去に格納したレコード件数の最大値を超えない，レコードの格納のケースを含みます。
- 4V AFM の SDB データベースで一括削除またはレコードの削除（ERASE）を実行した場合に，仮想ルートレコード配下のレコード件数が 0 件となるときは，シーケンシャルインデクスのキーが 1 つ削除されるため，キー値が削除されるときのシステムログ量を見積もり，その値を加算してください。

(3)　更新可能なオンライン再編成実行時のシステムログ量【4V FMB】

更新可能なオンライン再編成中の SDB データベースを更新する場合，「(2)　SDB データベースを操作する API または DML の実行時のシステムログ量」で計算したシステムログ量に，次に示すシステムログ量を加算してください。

■レコードの格納（STORE）時に出力されるシステムログ量

228 ＋ ↑（a ＋ b）÷4↑×4　　（単位：バイト）

■レコードの更新（MODIFY）時に出力されるシステムログ量

448 ＋ ↑（a ＋ b）÷4↑×8　　（単位：バイト）

■レコードの削除（ERASE）時に出力されるシステムログ量（一括オプションに'S'を指定する場合）

$$220 + \uparrow a \div 4 \uparrow \times 4 + \sum_{i=1}^{m} \{228 + \uparrow (a+b) \div 4 \uparrow \times 4\} + \sum_{i=1}^{n} \{220 + \uparrow a \div 4 \uparrow \times 4\}$$

（単位：バイト）

■レコードの削除（ERASE）時に出力されるシステムログ量（一括オプションに'O'を指定する場合）

$$220 + \uparrow a \div 4 \uparrow \times 4 + \sum_{i=1}^{p} \{228 + \uparrow (a+b) \div 4 \uparrow \times 4\}$$

（単位：バイト）

■計算式中の変数の説明

a：ルートレコードのキー長＋（更新対象レコードの階層－1）×4

b：SDB データベース定義の RECORD 句で指定したレコードサイズ

m：削除レコードとその下位階層のすべてのレコードの総数

n：削除レコードとその下位レコードの各階層で USER ポインタが設定されているレコードの総数

p：削除レコードの階層のすべてのレコードの総数

(4) HiRDB/SD 定義ユティリティ（pdsdbdef）実行時のシステムログ量

HiRDB/SD 定義ユティリティ（pdsdbdef）実行時のシステムログ量の見積もりについて説明します。

(a) *ENTRY DICTIONARY 文または*DELETE DICTIONARY 文の実行時に出力されるシステムログ量

- ディクショナリサーバで出力されるシステムログ量

$$2{,}925 + \uparrow a \div (e-59) \uparrow \times (e+152) + \uparrow b \div (e-59) \uparrow \times (e+152) + \uparrow c \div (e-59) \uparrow \times (e+152) + \{l \times f + m \times j\} \text{※}$$

（単位：バイト）

注※　*DELETE DICTIONARY 文の実行時に加算します。

(b) *ALTER DICTIONARY 文の実行時に出力されるシステムログ量【4V FMB，4V AFM】

- ディクショナリサーバで出力されるシステムログ量

$$3{,}606 + \{\uparrow a \div (e-59) \uparrow \times (e+152)\} \times 2 + \{\uparrow b \div (e-59) \uparrow \times (e+152)\} \times 2 + \{\uparrow c \div (e-59) \uparrow \times (e+152)\} \times 2$$

（単位：バイト）

(c) *CHECK DICTIONARY 文の実行時に出力されるシステムログ量

- ディクショナリサーバで出力されるシステムログ量

```
1,302＋｛↑c÷（e-59）↑×（e＋152）｝×2
＋l×f＋m×j
                                                （単位：バイト）
```

(d) *ENTRY DIRECTORY 文または*DELETE DIRECTORY 文の実行時に出力されるシステムログ量

- ディクショナリサーバで出力されるシステムログ量

```
5,722＋｛↑b÷（e-59）↑×（e＋152）｝×2
＋｛↑c÷（e-59）↑×（e＋152）｝×2
＋｛l×f＋m×j｝※
＋↑d1÷（e-59）↑×（e＋152）
＋↑d2÷（e-59）↑×（e＋152）
                                                （単位：バイト）
```

注※　*CHECK DICTIONARY 文を同時に指定していない場合に加算します。

- 各バックエンドサーバで出力されるシステムログ量

```
l×f＋m×j　（単位：バイト）
```

(e) *ALTER DIRECTORY 文の実行時に出力されるシステムログ量【4V FMB，4V AFM】

- ディクショナリサーバで出力されるシステムログ量

```
8,840＋｛↑b÷（e-59）↑×（e＋152）｝×2
＋｛↑c÷（e-59）↑×（e＋152）｝×2
＋｛l×f＋m×j｝※
＋｛↑d1÷（e-59）↑×（e＋152）｝×2
＋｛↑d2÷（e-59）↑×（e＋152）｝×2
                                                （単位：バイト）
```

注※　*CHECK DICTIONARY 文を同時に指定していない場合に加算します。

- 各バックエンドサーバで出力されるシステムログ量

```
n　（単位：バイト）
```

変数の説明

- a：SDB データベース定義情報長（バイト）
  「表 3-26　計算式 2（SDB データベース定義情報長）」を参照して算出してください。
- b：SDB データベースビュー定義情報長（バイト）
  「表 3-27　計算式 3（SDB データベースビュー定義情報長）」を参照して算出してください。
- c：SDB データベース格納定義情報長（バイト）
  「表 3-28　計算式 4（SDB データベース格納定義情報長）」を参照して算出してください。
- d1：SDB ディレクトリ情報長（SDB データベース情報）（バイト）
  「3.4.2　共用メモリの見積もり」の，変数 dbs の計算式で算出してください。

- d2：SDB ディレクトリ情報長（SDB データベースビュー情報）（バイト）
  「3.4.2　共用メモリの見積もり」の，変数 sub の計算式で算出してください。
- e：データディクショナリ用 RD エリアのページ長（バイト）
- f：1SDB データベース当たりのレコード型数（個）
- j：1SDB データベース格納定義当たりのインデクス数（個）
- l：レコード型定義時および削除時のシステムログ量※1（バイト）
- m：レコード型へのインデクス定義時および削除時のシステムログ量※2（バイト）
- n：格納条件が変更となる RD エリア内のデータを削除するときに必要なシステムログ量※3
  分割格納条件の変更機能で格納 RD エリア数を変更する場合に加算します。

注※1

レコード型定義時および削除時のシステムログ量については，マニュアル「HiRDB システム導入・設計ガイド」の「表定義時に出力されるシステムログ量」を参照してください。

なお，マニュアル参照時には，次の用語を読み替えてください。

- 「表」を「レコード型」に読み替えてください。
- 「列」を「構成要素」に読み替えてください。なお，データ属性を持たない構成要素は，構成要素数を数える際の対象にはしないでください。また，SDB データベース定義で OCCURS 句を指定する場合，システム内で生成される構成要素は，構成要素数を数える際の対象にしてください。

注※2

インデクス定義時のシステムログ量については，マニュアル「HiRDB システム導入・設計ガイド」の「インデクス定義時に出力されるシステムログ量」を参照してください。

なお，マニュアル参照時には，次の用語を読み替えてください。

- 「列」を「構成要素（データ属性を持たない構成要素は含みません）」に読み替えてください。
- 「インデクスを格納するために必要なインデクスの格納ページ数」は 0 に読み替えてください。

注※3

RD エリア内のデータを削除するときに必要なシステムログ容量については，マニュアル「HiRDB システム導入・設計ガイド」の「PURGE TABLE 文実行時に出力されるシステムログ量」を参照してください。格納条件が変更となる格納レコード用 RD エリアおよびインデクスを格納するユーザ用 RD エリアに対して，RD エリアごとに求めたシステムログ量の総容量を求めてください。

なお，マニュアル「HiRDB システム導入・設計ガイド」の「PURGE TABLE 文実行時に出力されるシステムログ量」を参照する際，「表」を「レコード型」に読み替えてください。

(5) HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のシステムログ量

HiRDB/SD データベース作成ユティリティ（pdsdblod）を実行する場合，次に示すシステムログが出力されます。

下記(a)～(e)で求めた値をデータロードの対象となる SDB データベースを管理するバックエンドサーバのログ量に加算してください。なお，SDB データベースを横分割している場合は，レコードおよびインデクスを格納する RD エリアごとにシステムログ量を計算します。

(a) 必ず取得するシステムログ量

項番	条件		出力されるシステムログ量（単位：バイト）
1	インデクス一括作成モード（idxmode=create）でデータロードする場合	purge=yes	2,000×（2 + IXRD）
		purge=no	2,000×（1 + IXRD）
2	インデクス更新モード（idxmode=sync）でデータロードする場合	purge=yes	4,000
		purge=no	2,000
3	インデクス再作成する場合		4,000

変数の説明

- IXRD：インデクス格納 RD エリア数

(b) logmode=no の場合に取得するシステムログ量

（DTRD + IXRD）×400 （単位：バイト）

変数の説明

- DTRD：データ格納 RD エリア数
- IXRD：インデクス格納 RD エリア数

(c) logmode=all の場合に取得するシステムログ量

項番	条件	出力されるシステムログ量（単位：バイト）
1	インデクス一括作成モード（idxmode=create）でデータロードする場合	280×m + 1,940×↑m÷s↑＋格納するレコードのログ総量※1 $+\sum_{i=1}^{n}\{(132+xi\times\uparrow(100-f)\div100\uparrow)\times Wi+280\times Wi+1,940\times\uparrow Wi\div Vi\uparrow\}$
2	インデクス更新モード（idxmode=sync）でデータロードする場合	280×m + 1,940×↑m÷s↑＋格納するレコードのログ総量※1 $+\sum_{i=1}^{n}\{280\times Wi+1,940\times\uparrow Wi\div Vi\uparrow+$（キー追加のログ量）※2$\times r+e\times d\}$

項番	条件	出力されるシステムログ量（単位：バイト）
3	インデクス再作成する場合	$\sum_{i=1}^{n}\{280Wi+1,940\times\uparrow Wi\div Vi\uparrow+\alpha+\beta\times\gamma\}$

注※1

レコードの格納（STORE）の容量見積もりから求めた，格納するレコードのログの総量です。「(2) SDB データベースを操作する API または DML の実行時のシステムログ量」を参照してください。ただし，更新前のログは，総量に含めないでください。

注※2

- マニュアル「HiRDB システム導入・設計ガイド」の「表データ更新時に出力されるシステムログ量」にある「インデクスログ量の見積もり」を参照して求めた値です。参照する際の注意事項は「(2)SDB データベースを操作する API または DML の実行時のシステムログ量」と同じです。
- 4V AFM の SDB データベースの場合，仮想ルートレコード型に定義したシーケンシャルインデクスでは，仮想ルートレコード 1 件につき 2 つのキーを追加するため，この「キー追加のログ量」を 2 倍して見積もってください。

変数の説明

- d：インデクススプリット時に出力されるシステムログ量
- e：インデクスのスプリット回数
- f：インデクス定義時に指定する PCTFREE 句（ページ内の未使用領域の比率）の値（%）
- m：レコードの格納ページ数
- n：インデクスの個数
- r：レコード型のレコード件数
- s：レコードを格納するユーザ用 RD エリアのセグメントサイズ（ページ）
- Wi：インデクスの格納ページ数

  4V AFM の SDB データベースの場合，仮想ルートレコード型に定義したシーケンシャルインデクスは，インデクスキーとレコード実現値の関係が通常 2 対 1 になります。仮想ルートレコード型に定義したシーケンシャルインデクスを格納するために必要なインデクスの格納ページ数（変数 Wi）は，「3.5.5(8)　インデクスを格納するページ数の見積もり」を参照してから見積もってください。
- Vi：インデクスを格納するユーザ用 RD エリアのセグメントサイズ（ページ）
- xi：インデクスを格納するユーザ用 RD エリアのページ長（バイト）
- α：インデクス定義時に出力されるシステムログ量

  マニュアル「HiRDB システム導入・設計ガイド」の「インデクス定義時に出力されるシステムログ量」にある「HiRDB/パラレルサーバの場合　バックエンドサーバで出力されるシステムログ量」を参照して求めてください。
- β：ページ／セグメントの確保ログ量の計算式

マニュアル「HiRDB システム導入・設計ガイド」の「表データ更新時に出力されるシステムログ量」にある「ページ／セグメントの確保ログ量の計算式」を参照して求めてください。

- γ：PURGE TABLE 文実行時に出力されるシステムログ量のインデクスのシステムログ量
  マニュアル「HiRDB システム導入・設計ガイド」の「PURGE TABLE 文実行時に出力されるシステムログ量」を参照して求めてください。

(d) purge=yes の場合に取得するシステムログ量

マニュアル「HiRDB システム導入・設計ガイド」の「PURGE TABLE 文実行時に出力されるシステムログ量」を参照して求めてください。

(e) oreload 文を指定した場合に取得するシステムログ量

1,102×格納するレコード件数 （単位：バイト）

3.5.2 作業表用ファイルの容量の見積もり

作業表用ファイルの容量の見積もりについて説明します。

作業表用ファイルの容量を見積もる際には，マニュアル「HiRDB システム導入・設計ガイド」の「作業表用ファイルの容量の見積もり」を参照し，次に示す作業用ファイルの見積もりを考慮してください。

(1) HiRDB/SD 定義ユティリティ（pdsdbdef）が使用する作業表用ファイルの容量の見積もり

HiRDB/SD 定義ユティリティ（pdsdbdef）で SDB ディレクトリ情報を削除する場合，次に示す計算式の分の作業表用ファイルが必要になります。

算出した値は，マニュアル「HiRDB システム導入・設計ガイド」の「SQL 文が使用する作業表用ファイルの容量」の「列情報作業表の容量」として計算してください。

```
↑a÷MIN｛↓（b-48）÷ 20↓，255｝↑×b×2 （単位：バイト）
```

変数の説明

- a：全 SDB データベース内の全レコード型内で構成要素数が最大の構成要素数
  ただし，データ属性を持たない構成要素を除きます。また，SDB データベース定義で OCCURS 句を指定する場合は，システム内で生成される構成要素数を含みます。
- b：作業表ページ長
  次の計算式から求めてください。

  ```
  MAX｛↑（20＋48）÷2,048↑×2,048，4,096｝
  ```

(2) HiRDB/SD データベース作成ユティリティ（pdsdblod）が使用する作業表用ファイルの容量の見積もり

インデクス一括作成モードで HiRDB/SD データベース作成ユティリティ（pdsdblod）を実行する場合，作業表用ファイルが必要になります。

詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「ユティリティが使用する作業表用ファイルの容量」を参照してください。

なお，マニュアルを参照するときには次の点に注意してください。

- 「pdload」を「pdsdblod」に読み替えてください。
- 「CREATE INDEX で指定するページ内の未使用領域の比率」は，「SDB データベース格納定義でインデクスに指定するページ内の未使用領域の比率（PCTFREE 句の値）」に読み替えてください。
- シーケンシャルインデクス，または二次インデクスのキーとなる構成要素の長さについては，「3.5.5(8) インデクスを格納するページ数の見積もり」を参照して計算してください。

なお，複数のユティリティを同時に実行する場合は，各ユティリティが使用する作業表用ファイルの容量を合計してください。

3.5.3 監査証跡ファイルの容量の見積もり

監査証跡ファイルの容量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「監査証跡ファイルの容量の見積もり」を参照してください。

(1) SDB ユティリティ操作イベントの監査証跡を取得する場合

見積もりをする際，次の表に示すコマンドを実行したときに出力される監査証跡のレコード数を加算してください。

表 3-21 出力される監査証跡のレコード数（SDB ユティリティ操作イベントの場合）

項番	コマンド名	イベント取得個所	1 イベントで出力されるレコード数	見積もりの目安
1	pdsdbdef	コマンド実行時の CONNECT 権限チェック	1	pdsdbdef コマンドの実行回数
2		SDB ディクショナリ情報の操作	1	次の SDB 定義文の合計数 • *ENTRY DICTIONARY 文 • *ALTER DICTIONARY 文 • *DELETE DICTIONARY 文
3		データベース定義のチェック	1	次の SDB 定義文の合計数

項番	コマンド名	イベント取得個所	1 イベントで出力されるレコード数	見積もりの目安
				• *CHECK DICTIONARY 文
4		SDB ディレクトリ情報の操作	1	次の SDB 定義文の合計数 • *ENTRY DIRECTORY 文 • *ALTER DIRECTORY 文 • *DELETE DIRECTORY 文
5		SDB ディレクトリ情報ファイルの作成	1	pdsdbdef コマンドの dirinf 文の指定がある SDB 制御文の個数
6		pdsdbdef コマンドの終了	1	pdsdbdef コマンドの実行回数
7	pdsdblod	コマンド実行時の CONNECT 権限チェック	1	pdsdblod コマンドの実行回数
8		pdsdblod コマンドの終了	1	pdsdblod コマンドの実行回数
9	pdsdbrog	コマンド実行時の CONNECT 権限チェック	1	pdsdbrog コマンドの実行回数
10		pdsdbrog コマンドの終了	1	pdsdbrog コマンドの実行回数

(2) SDB データベース操作イベントの監査証跡を取得する場合

見積もりをする際，次の表に示す SDB データベース操作を実行したときに出力される監査証跡のレコード数を加算してください。

表 3-22 出力される監査証跡のレコード数（SDB データベース操作イベントの場合）

項番	SDB データベース操作	イベント取得個所	1 イベントで出力されるレコード数	見積もりの目安
1	• レコードの検索（FETCH） • 位置指示子の位置づけ（FIND）	イベント終了時	1	レコードの検索（FETCH）および位置指示子の位置づけ（FIND）の実行回数 ただし，FIRST オプションの指定時は，個別開始単位にファミリ内のレコード型に対して 1 件となります
2	レコードの格納（SOTRE）		1	レコードの格納（STORE）の実行回数
3	レコードの更新（MODIFY）		1	レコードの更新（MODIFY）の実行回数
4	レコードの削除（ERASE）		1	レコードの削除（ERASE）の実行回数

項番	SDB データベース操作	イベント取得個所	1 イベントで出力されるレコード数	見積もりの目安
5	レコードの一括削除		1	レコードの一括削除の実行回数
6	複数レコードの検索（FETCHDB_ALL）		1	複数レコードの検索（FETCHDB ALL）の実行回数
7	レコードの取得（GET）		1	レコードの取得（GET）の実行回数

3.5.4　システム用 RD エリアの容量の見積もり

システム用 RD エリアの容量の見積もりについて説明します。

(1)　データディクショナリ用 RD エリアの容量の見積もり

データディクショナリ用 RD エリアの容量は，次に示す計算式で求めます。

```
a×b×1.3＋c×125＋1,600,000　（単位：バイト）
```

変数の説明

- a：データディクショナリ用 RD エリアのページ長

  データベース初期設定ユティリティ（pdinit）またはデータベース構成変更ユティリティ（pdmod）の create rdarea 文で指定するページ長です。

- b：データディクショナリ用 RD エリアの総ページ数

  「表の格納ページ数＋インデクスの格納ページ数」です。「3.5.4(1)(a)　表の格納ページ数の計算方法」および「3.5.4(1)(b)　インデクスの格納ページ数の計算方法」を参照してしてください。

  なお，レコード型およびインデクスの定義情報をデータディクショナリ用 RD エリアで管理するため，SDB データベースの追加に伴って行を追加するディクショナリ表についても，使用するページ数を見積もって加算してください。

- c：データディクショナリ用 RD エリアのセグメントサイズ

  データベース初期設定ユティリティ（pdinit）またはデータベース構成変更ユティリティ（pdmod）の create rdarea 文で指定するセグメントサイズです。

(a)　表の格納ページ数の計算方法

表の格納ページ数（単位：ページ）は，計算式 1～計算式 4 の総和に，HiRDB/SD が使用する HiRDB のディクショナリ表の格納ページ数を加算して求めます。

HiRDB のディクショナリ表の格納ページ数については，マニュアル「HiRDB システム導入・設計ガイド」の「通常のデータディクショナリ用 RD エリアの容量の見積もり」の「表の格納ページ数の計算方法」を参照してください。ただし，SQL_RDAREAS 表および SQL_AUDIT_REGULARIZE 表については，次の表に示す計算式を使用してください。

表 3-23　SQL_RDAREAS 表の格納ページ数

ディクショナリ表名	計算式
SQL_RDAREAS	$\left\uparrow \frac{b}{\left\downarrow \frac{a-48}{\uparrow (c\times 2+101) \div 2 \uparrow \times 2} \right\downarrow} \right\uparrow$

注

計算式中で使用している変数については，マニュアル「HiRDB システム導入・設計ガイド」の「通常のデータディクショナリ用 RD エリアの容量の見積もり」の「表の格納ページ数の計算方法」を参照してください。

表 3-24　SQL_AUDIT_REGULARIZE 表の格納ページ数

ディクショナリ表名	計算式
SQL_AUDIT_REGULARIZE	$\left\uparrow \frac{b}{\left\downarrow \frac{a-48}{\uparrow (c+d+e+f+406) \div 2 \uparrow \times 2} \right\downarrow} \right\uparrow$

注

計算式中で使用している変数については，マニュアル「HiRDB システム導入・設計ガイド」の「通常のデータディクショナリ用 RD エリアの容量の見積もり」の「表の格納ページ数の計算方法」を参照してください。

表 3-25　計算式 1

ディクショナリ表名	計算式
SQL_SDB_DB	$\left\uparrow \frac{b1}{\left\downarrow \frac{a-48}{(\uparrow (c+d1+12\times g+2\times 7+43+8) \div 2 \uparrow \times 2)} \right\downarrow} \right\uparrow + b1 \times \left\uparrow \frac{e1}{a-48} \right\uparrow$
SQL_SDB_DB_VIEW	$\left\uparrow \frac{b2}{\left\downarrow \frac{a-48}{(\uparrow (c\times 2+d1+d2+12\times g+2\times 11+63+8) \div 2 \uparrow \times 2)} \right\downarrow} \right\uparrow + b2 \times \left\uparrow \frac{e2}{a-48} \right\uparrow$
SQL_SDB_STORAGE_DB	$\left\uparrow \frac{b3}{\left\downarrow \frac{a-48}{(\uparrow (c\times 2+d1+d3+12\times g+2\times 11+63+8) \div 2 \uparrow \times 2)} \right\downarrow} \right\uparrow + b3 \times \left\uparrow \frac{e3}{a-48} \right\uparrow$
SQL_SDB_NAME_CODE	$\left\uparrow \frac{b1+b2+b3}{\left\downarrow \frac{a-48}{(\uparrow (d1+MEMN+2\times 9+28+8) \div 2 \uparrow \times 2)} \right\downarrow} \right\uparrow$

ディクショナリ表名	計算式
SQL_SDB_DIR	$\uparrow \frac{b1+b2}{a-48} \uparrow \downarrow \frac{}{(\uparrow (c\times 2+d1+d2+12\times g+2\times 11+49+8) \div 2 \uparrow \times 2)} \downarrow + b1 \times \uparrow \frac{f1}{a-48} \uparrow + b2 \times \uparrow \frac{f2}{a-48} \uparrow$

変数の説明

- a：データディクショナリ用 RD エリアのページ長（バイト）
- b1：SDB データベース定義の総数（個）
- b2：SDB データベースビュー定義の総数（個）
- b3：SDB データベース格納定義の総数（個）
- c：認可識別子の長さの平均値（バイト）
- d1：SDB データベース名の長さの平均値（バイト）
- d2：SDB データベースビュー名の長さの平均値（バイト）
- d3：SDB データベース格納定義名の長さの平均値（バイト）
- e1：SDB データベース定義情報長（バイト）
- e2：SDB データベースビュー定義情報長（バイト）
- e3：SDB データベース格納定義情報長（バイト）
- f1：SDB ディレクトリ情報長（SDB データベース情報）（バイト）
  「3.4.2　共用メモリの見積もり」の，システム共通定義の pd_structured_shmpool_dicsize オペランドの指定値を求める計算式の変数 dbs の値となります。
- f2：SDB ディレクトリ情報長（SDB データベースビュー情報）（バイト）
  「3.4.2　共用メモリの見積もり」の，システム共通定義の pd_structured_shmpool_dicsize オペランドの指定値を求める計算式の変数 sub の値となります。
- g：データ長が 256 バイト以上で，かつ分岐するバイナリデータの平均値（個）
  バイナリデータの格納ページ数の分岐条件については，マニュアル「HiRDB システム導入・設計ガイド」の「データ長一覧」を参照してください。
- MEMN：番号に変換される名称長の平均長（バイト）
  SQL_SDB_NAME_CODE 表の NAME_LEN 列の値です。次の計算式から求めてください。

```
↑（d1＋d2＋d3）÷3↑
```

表 3-26　計算式 2（SDB データベース定義情報長）

計算式	変数の説明
e1 ＝ {442 ＋（272 ＋ 248×c）×a ＋（480 ＋ 176×d）×b ＋ 496}	• a：1SDB データベース定義のレコード型指定数の平均値（個） • b：1SDB データベース定義の親子集合指定数の平均値（個） • c：1SDB データベース定義の 1 レコード型ごとの構成要素指定数の平均値（個）※ • d：1SDB データベース定義の 1 親子集合ごとのキーの構成要素指定数（個）

注※

SDB データベース定義で OCCURS 句を指定する場合は，システム内で生成される構成要素数を考慮する必要があります。

表 3-27　計算式 3（SDB データベースビュー定義情報長）

計算式	変数の説明
e2 ＝ （588 ＋ 240×a ＋ 208×b ＋ 216×c ＋ 416）	• a：1SDB データベース定義のレコード型指定数の平均値（個） • b：1SDB データベース定義の親子集合指定数の平均値（個） • c：1SDB データベース定義の 1 レコード型ごとの最小レベル番号の構成要素指定数の平均値（個）※

注※

SDB データベース定義で OCCURS 句を指定する場合は，システム内で生成される構成要素数を考慮する必要があります。

表 3-28　計算式 4（SDB データベース格納定義情報長）

計算式	変数の説明
e3 ＝ {432 ＋ 840×a ＋ 280×b ＋（168×d ＋ 368）×c ＋（120×f ＋ 96）×e ＋ 336}	• a：1SDB データベース格納定義の格納レコード指定数の平均値（個） • b：1SDB データベース格納定義の格納親子集合指定数の平均値（個） • c：1SDB データベース格納定義のインデクス指定数の平均値（個） • d：1SDB データベース格納定義のインデクスごとの構成要素指定数の平均値（個） • e：1SDB データベース格納定義の KEYDEF 句指定数の平均値（個） • f：1SDB データベース格納定義の KEYDEF 句ごとの DATA 句指定数の平均値（個）

(b)　インデクスの格納ページ数の計算方法

インデクスの格納ページ数は，次の表に示すディクショナリ表のインデクスを格納するページ数に，HiRDB/SD が使用する HiRDB のディクショナリ表のインデクスを格納するページ数を加算して求めます。

HiRDB のディクショナリ表のインデクスを格納するページ数，およびその計算方法については，マニュアル「HiRDB システム導入・設計ガイド」の「通常のデータディクショナリ用 RD エリアの容量の見積もり」の「インデクスの格納ページ数の計算方法」を参照してください。また，「インデクスの格納ページ数の計算方法」に記載されている計算式には，次の表に示す変数を代入してください。

なお，マニュアルを参照する場合，次の用語を読み替えてください。

- 「表」を「レコード型」に読み替えてください。
- 「列」を「構成要素（データ属性を持たない構成要素は含みません）」に読み替えてください。

表 3-29　インデクスの格納ページ数を求める計算式に代入する変数一覧

ディクショナリ表名	種別	キー長※2 (変数 g※1)	キーの種類の個数 (変数 c※1)	キーの重複数の平均値 (変数 d※1)
SQL_SDB_DB	129	SCN + DBN + 2	システム内の SDB データベース数	1
	130	2	システム内の SDB データベース数	1
	131	4	すべての SDB データベース定義情報内で SDB データベース定義情報長が異なる数	SDB データベース定義情報長の平均重複数
SQL_SDB_DB_VIEW	132	SCN×2 + DBN + DBVN + 2	システム内の SDB データベースビュー数	1
	133	2	1	システム内の SDB データベースビュー数
	134	2	システム内の SDB データベース数	1
	135	4	すべての SDB データベースビュー定義情報内で SDB データベースビュー定義情報長が異なる数	SDB データベースビュー定義情報長の平均重複数
SQL_SDB_STORAGE_DB	136	SCN + STDBN + 2	システム内の SDB データベース格納定義数	1
	137	SCN + DBN + 2	システム内の SDB データベース数	1
	138	2	システム内の SDB データベース格納定義数	1
	139	2	システム内の SDB データベース数	1

ディクショナリ表名	種別	キー長※2 (変数 g※1)	キーの種類の個数 (変数 c※1)	キーの重複数の平均値 (変数 d※1)
	140	4	すべての SDB データベース格納定義情報内で SDB データベース格納定義情報長が異なる数	SDB データベース格納定義情報長の平均重複数
SQL_SDB_NAME_CODE	141	DBN + MEMN + 4	今まで定義したことのある名称の，システム内の SDB ディクショナリ情報数	1
	142	6	今まで定義したことのある名称の，システム内の SDB ディクショナリ情報数	1
	143	DBN + 6	今まで定義したことのある名称の，システム内の SDB ディクショナリ情報数	1
	144	MEMN + 6	今まで定義したことのある名称の，システム内の SDB ディクショナリ情報数	1
SQL_SDB_DIR	145	SCN + DBN + 4	システム内の SDB ディレクトリ情報数	1
	146	SCN×2 + DBN + DBVN + 4	システム内の SDB ディレクトリ情報数	1
	147	4	すべての SDB ディレクトリ情報内で SDB ディレクトリ情報長が異なる数	SDB ディレクトリ情報長の平均重複数

変数の説明

- SCN：認可識別子の長さの平均値（バイト）
- DBN：SDB データベース名の長さの平均値（バイト）
- DBVN：SDB データベースビュー名の長さの平均値（バイト）
- STDBN：SDB データベース格納定義名の長さの平均値（バイト）
- MEMN：番号に変換される名称長の平均長（バイト）
  SQL_SDB_NAME_CODE 表の NAME_LEN 列の値です。次の計算式から求めてください。

```
↑（DBN＋DBVN＋STDBN）÷3↑
```

注※1

マニュアル「HiRDB システム導入・設計ガイド」の「インデクスの格納ページ数の計算方法」の「計算式中で使用する変数」に記載されている変数です。

注※2

キー長は４バイト単位で切り上げになります。次に示す計算式で求めてください。

↑キー長÷4↑×4

(2)　そのほかのシステム用 RD エリアの容量の見積もり

次のシステム用 RD エリアの容量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」のそれぞれの項目を参照してください。

- マスタディレクトリ用 RD エリアの容量の見積もり
- データディレクトリ用 RD エリアの容量の見積もり

なお，マニュアルを参照する際，次の用語を読み替えてください。

- 「表」を「レコード型」に読み替えてください。
- 「列」を「構成要素（データ属性を持たない構成要素は含みません）」に読み替えてください。

また，マスタディレクトリ用 RD エリアの容量を見積もる際，次の変数は 0 として計算してください。

- 変数 d：定義するビュー表の総数
- 変数 e：定義するデータ型およびインデクス型の合計

3.5.5　ユーザ用 RD エリアの容量の見積もり

ユーザ用 RD エリアの容量は，レコードを格納するデータページ，またはインデクスを格納するためのインデクスページ数を計算したあと，総ページ数を計算して求めることができます。

ユーザ用 RD エリアの容量は次の手順で見積もります。

手順

1. SDB データベースに格納するレコードのレコード長を算出します。
2. ページ長，およびサブページ長を設計します。
3. 事前割り当てページ数または事前割り当てサブページ数を算出します。
4. 平均的なファミリが使用するページ数またはサブページ数を算出します。
5. セグメントサイズを設計します。
6. レコードを格納するデータページ数を算出します。
7. ユーザ用 RD エリアに必要な容量を算出します。

それぞれの説明を次に示します。

(1)　レコード長の算出

システム要件から，SDB データベースのレコード長を算出します。レコード長の計算式を次に示します。

$$\uparrow \frac{\sum_{i=1}^{f} d_i + sc}{4} \uparrow \times 4 + \uparrow \frac{y}{4} \uparrow \times 4 + 4 + rpn + b \times 24 + c + ow + nu \times 12 + ny \times 4$$

変数の説明

- f：レコード型のすべての構成要素数

  SDB データベース定義で OCCURS 句を指定する場合，システム内で繰り返し回数分生成される構成要素数も含みます。システム内で生成される構成要素については，「11.7.1(4)(g)　OCCURS」を参照してください。

- d_i：各構成要素のデータ長

  データ型とデータ長の対応を次の表に示します。

表 3-30　データ型とデータ長の対応

データ型	データ長（単位：バイト）
CHARACTER	n（1～30,000）※1
XCHARACTER	n（1～30,000）※1
PACKED DECIMAL FIXED	↑（m + n + 1）÷2↑※2
INTEGER	4
SMALLINT	2

注※1

SDB データベース定義で指定した値です。n は指定したデータ長です。

注※2

SDB データベース定義で指定した値です。m は整数部の桁数です。n は小数部の桁数です。m + n は最大 38 となります。

- sc：システム用構成要素の長さ

  4V AFM の SDB データベースの仮想ルートレコード型の場合：4 バイト

  上記以外の場合：0 バイト

- y：ポインタ管理領域の予備領域の長さ（SDB データベース格納定義の POINTER AREA SIZE 句の指定値）

- rpn：4V FMB または SD FMB の SDB データベースのルートレコード型の場合は 4，それ以外の場合は 0 を代入してください。

- b：当該レコード型を親レコード型とする親子集合数
- c：ルートレコード型以外のレコード型の場合は 20，ルートレコード型の場合は 0 を代入してください。
- ow：OWNER ポインタを持つ子レコードの場合は 8，それ以外の場合は 0 を代入してください。
- nu：当該レコード型を親レコード型とする親子集合のうち，USER ポインタを持つ子レコード型の数
- ny：当該レコード型を親レコード型とする親子集合のうち，一括削除機能を適用する子レコード型の数

(2)　ページ長およびサブページ長の設計

ユーザ用 RD エリアのページ長，およびサブページ長の設計指針を次の表に示します。

表 3-31　ページ長およびサブページ長の設計指針

設計項目		説明
ページ	ページ長の決定	ページ長については，「3.2.3　ページの設計」を参照して決定してください。 また，マニュアル「HiRDB システム導入・設計ガイド」の「ページ長の決定」も参照してください。 なお，サブページ分割をする場合は，ページ長は 4,096 バイト固定になります。
	ページ内の未使用領域の比率の設定	「3.2.5　レコード格納時の空き領域の作成（サブページ分割をしない場合）」を参照してください。
サブページ	サブページ長の決定	サブページ長については，「3.2.4　サブページの設計」を参照してください。
	サブページ内の未使用領域の比率の設定	「3.2.6　レコード格納時の空き領域の作成（サブページ分割をする場合）」を参照してください。

(3)　事前割り当てページ数または事前割り当てサブページ数の算出【4V FMB】

事前割り当てページまたは事前割り当てサブページに格納するファミリについては，そのファミリが使用するページ数またはサブページ数を個別に見積もります。

次に示す計算式から，レコードが使用する総領域長を計算してください。サブページ分割をしない場合は，「表 3-9　見積もりに必要な項目」の項番 5，8 について※，レコードが使用する総領域長を計算してください。サブページ分割をする場合は，「表 3-14　見積もりに必要な項目」の項番 6，9 について※，レコードが使用する総領域長を計算してください。

注※

事前割り当てページまたは事前割り当てサブページにレコードを格納する場合，ページ切り替え指定によって，ページまたはサブページを切り替えることはできません。

$$Sum = \sum_{i=1}^{e} \{(n1_i + u2_i) \times (L_i + 10)\}$$

変数の説明

- e：RD エリアに格納するレコード型の総数
- n1i：各レコード型のデータロード時に格納するレコード件数（ページ切り替え指定なし）
- u2i：各レコード型のデータロード後に格納するレコード件数（ページ切り替え指定なし）
- Li：各レコード型のレコード長

また，次の計算式を使って，平均領域長を計算してください。

$$Ave = \uparrow \frac{Sum}{\sum_{i=1}^{e} (n1_i + u2_i)} \uparrow$$

上記の平均領域長を使って，次の計算式からファミリが使用するページ数またはサブページ数を計算してください。

- ファミリが使用するページ数を求める場合

●ファミリが使用するページ数

$$= \uparrow \frac{\sum_{i=1}^{e} (n1_i + u2_i)}{MIN(255, \downarrow \frac{b-76}{Ave} \downarrow)} \uparrow \times 1.3^{※}$$

- ファミリが使用するサブページ数を求める場合

●ファミリが使用するサブページ数

$$= \uparrow \frac{\sum_{i=1}^{e} (n1_i + u2_i)}{\downarrow \frac{sp1-36}{Ave} \downarrow} \uparrow \times 1.3^{※}$$

変数の説明

- e：RD エリアに格納するレコード型の総数
- n1i：各レコード型のデータロード時に格納するレコード件数（ページ切り替え指定なし）
- u2i：各レコード型のデータロード後に格納するレコード件数（ページ切り替え指定なし）
- b：レコードを格納する RD エリアのページ長
- spl：サブページ長
  サブページ長の求め方については，「3.2.4(7)　サブページ長（サブページ分割数）の設計方針」を参照してください。

注※

1.3 は次の内容を考慮した値です。ファミリ内のレコードが非連続のデータページへ分散することを防止したい場合，1.3 より大きい値を設定してください。

- データロード後に格納するレコード件数が想定よりも多くなり，事前割り当てページまたは事前割り当てサブページが不足すると，追加で事前割り当てページまたは事前割り当てサブページが確保されます。

(4)　平均的なファミリが使用するページ数またはサブページ数の算出

平均的なファミリが使用するページ数またはサブページ数は，事前割り当てページまたは事前割り当てサブページに格納するファミリを除外して見積もります。

次に示す計算式から，レコードが使用する総領域長を計算してください。サブページ分割をしない場合は，「表 3-9　見積もりに必要な項目」の項番 5，9 について，レコードが使用する総領域長を計算してください。サブページ分割をする場合は，「表 3-14　見積もりに必要な項目」の項番 6，10 について，レコードが使用する総領域長を計算してください。

$$Sumn1 = \sum_{i=1}^{e} \{ n1_i \times (L_i + 10) \}$$

$$Sums2 = \sum_{i=1}^{e} \{ s2_i \times (L_i + 10) \}$$

変数の説明

- e：RD エリアに格納するレコード型の総数
- Li：各レコード型のレコード長
- n1i：各レコード型のデータロード時に格納するレコード件数（ページ切り替え指定なし）
- s2i：各レコード型のデータロード後に格納するレコード件数（ページ切り替え指定なし，かつセグメント内の空きページを使用するレコード）

また，次の計算式を使って，平均領域長を計算してください。

$$Aven1 = \uparrow \frac{Sumn1}{\sum_{i=1}^{e} n1_i} \uparrow$$

$$Aves2 = \uparrow \frac{Sums2}{\sum_{i=1}^{e} s2_i} \uparrow$$

変数の説明

- e：RD エリアに格納するレコード型の総数
- n1i：各レコード型のデータロード時に格納するレコード件数（ページ切り替え指定なし）
- s2i：各レコード型のデータロード後に格納するレコード件数（ページ切り替え指定なし，かつセグメント内の空きページを使用するレコード）

上記の平均領域長を使って，次の計算式から平均的なファミリが使用するページ数またはサブページ数を計算してください。

- 平均的なファミリが使用するページ数を求める場合

●平均的なファミリが使用するページ数

$$= \sum_{i=1}^{e} t1_i + \sum_{i=1}^{e} c1_i + \left(\uparrow \frac{\sum_{i=1}^{e} n1_i}{\min\left(255, \ \downarrow \frac{\downarrow \frac{b \times (100-C)}{100} \downarrow - 76}{Aven1} \downarrow \right)} \uparrow \right)$$

$$+ \sum_{i=1}^{e} t2_i + \sum_{i=1}^{e} c2_i + \left(\uparrow \frac{\sum_{i=1}^{e} s2_i}{\min\left(255, \ \downarrow \frac{b-76}{Aves2} \downarrow \right)} \uparrow \right)$$

- 平均的なファミリが使用するサブページ数を求める場合

●平均的なファミリが使用するサブページ数

$$= \sum_{i=1}^{e} t1_i + \sum_{i=1}^{e} c1_i + \left(\uparrow \frac{\sum_{i=1}^{e} n1_i}{\downarrow \frac{\downarrow \frac{spl \times (100-D)}{100} \downarrow - 36}{Aven1} \downarrow} \uparrow \right)$$

$$+ \sum_{i=1}^{e} t2_i + \sum_{i=1}^{e} c2_i + \left(\uparrow \frac{\sum_{i=1}^{e} s2_i}{\downarrow \frac{spl-36}{Aves2} \downarrow} \uparrow \right)$$

変数の説明

- e：RD エリアに格納するレコード型の総数
- b：レコードを格納する RD エリアのページ長
- C：ページ内の未使用領域の比率
- spl：サブページ長（サブページ長の求め方については，「3.2.4(7)　サブページ長（サブページ分割数）の設計方針」を参照してください）
- D：サブページ内の未使用領域の比率
- t1i：各レコード型のデータロード時に格納するレコード件数（ページ切り替え'O'指定）
- c1i：各レコード型のデータロード時に格納するレコード件数（ページ切り替え'C'指定）
- n1i：各レコード型のデータロード時に格納するレコード件数（ページ切り替え指定なし）
- t2i：各レコード型のデータロード後に格納するレコード件数（ページ切り替え'O'指定）
- c2i：各レコード型のデータロード後に格納するレコード件数（ページ切り替え'C'指定）
- s2i：各レコード型のデータロード後に格納するレコード件数（ページ切り替え指定なし，かつセグメント内の空きページを使用するレコード）

(5)　セグメントサイズの設計

ユーザ用 RD エリアのセグメントサイズの設計指針を次の表に示します。

表 3-32　セグメントサイズの設計指針

設計項目	説明
セグメントサイズの決定	セグメントサイズ（1 セグメント当たりのページ数）は，平均的なファミリが使用するページ数にすることを推奨します。 詳細については，「3.2.2　セグメントの設計」，および「3.5.5(4)　平均的なファミリが使用するページ数またはサブページ数の算出」を参照してください。 なお，サブページ分割をする場合は，次に示す計算式で平均的なファミリが使用するページ数を求めてください。 • ↑平均的なファミリが使用するサブページ数÷サブページ分割数↑ また，マニュアル「HiRDB システム導入・設計ガイド」の「セグメントサイズの決定」も参照してください。
セグメント内の空きページの比率の設定	• サブページ分割をしない場合 「3.2.5　レコード格納時の空き領域の作成（サブページ分割をしない場合）」を参照してください。 • サブページ分割をする場合 「3.2.6　レコード格納時の空き領域の作成（サブページ分割をする場合）」を参照してください。

(6)　レコードを格納するデータページ数の算出

レコードを格納するデータページ数は，次に示す計算式で求めることができます。

- サブページ分割をしない場合

  計算式中の平均的なファミリが使用するページ数は，「表 3-9　見積もりに必要な項目」の項番 1～7 および 9 を使用して計算してください。また，計算式中の事前割り当てページ数の合計値は，「表 3-9 見積もりに必要な項目」の項番 5 および 8 を使用して計算してください。

$$\uparrow \frac{\text{（平均的なファミリが使用するページ数}\times\text{ファミリ数）}^{※2} + \text{ppn}}{S} \uparrow \times S \times 1.3^{※1}$$

- サブページ分割をする場合

  計算式中の平均的なファミリが使用するサブページ数は，「表 3-14　見積もりに必要な項目」の項番 1～8 および 10 を使用して計算してください。また，計算式中の事前割り当てサブページ数の合計値は，「表 3-14　見積もりに必要な項目」の項番 6 および 9 を使用して計算してください。

$$\uparrow \frac{\uparrow \frac{\text{（平均的なファミリが使用するサブページ数}\times\text{ファミリ数）}^{※2} + \text{psn}}{\text{spn}} \uparrow}{S} \uparrow \times S \times 1.3^{※1}$$

変数の説明

- S：データベース初期設定ユティリティ（pdinit），またはデータベース構成変更ユティリティ（pdmod）で指定した 1 セグメントのページ数（セグメントサイズ）
- ppn：事前割り当てページの合計数
- spn：レコードを格納するデータページのサブページ分割数
- psn：事前割り当てサブページの合計数

注※1

1.3 は次の内容を考慮した値です。内容を確認して，容量が大きくなることが予想される場合，1.3 より大きい値を設定してください。

- SDB データベースを操作する API または DML によるレコード追加時に，ほかのトランザクションによって排他を取得できない場合，想定した格納位置とは異なる場所にレコードが格納されることがあります。その結果，格納効率が悪くなり容量が増加することがあります。
- 各レコード型の平均領域長から計算するため，実際のレコード格納時のレコードサイズ，およびレコードの格納順序によっては，格納効率が悪くなり容量が増加することがあります。
- 平均的なファミリが使用するページ数またはサブページ数から見積もるため，ファミリ間にばらつきがある場合，計算結果以上に容量が大きくなることがあります。

注※2

事前割り当てページまたは事前割り当てサブページを確保するファミリを除外して計算してください。

(7) ユーザ用 RD エリアに必要な容量の算出

全レコードの格納に必要なページ数から，ユーザ用 RD エリアに必要な容量を算出します。計算式を次に示します。

```
a×（b＋c）　　（単位：バイト）
```

変数の説明

- a：ユーザ用 RD エリアのページ長
- b：全レコードの格納に必要なページ数
- c：ディレクトリページ部分の総ページ数

ディレクトリページの総ページ数については，マニュアル「HiRDB システム導入・設計ガイド」の「ユーザ用 RD エリアの容量の見積もり」の「ディレクトリページ部分の総ページ数の計算式」を参照してください。

(8)　インデクスを格納するページ数の見積もり

インデクスを格納するページ数の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「ユーザ用 RD エリアの容量の見積もり」の「インデクスの格納ページ数の計算方法」を参照してください。

なお，マニュアルを参照するときには次の点に注意してください。

- 「CREATE INDEX」を「SDB データベース定義」に読み替えてください。
- クラスタキーの記載は無視してください。
- 仮想ルートレコード型に定義したシーケンシャルインデクスについては，インデクスキーとレコード実現値の関係が通常 2 対 1※になります。そのため，マニュアル「HiRDB システム導入・設計ガイド」の「ユーザ用 RD エリアの容量の見積もり」の「インデクスの格納ページ数の計算方法」で求めたインデクスページ数を 2 倍して，そのインデクスのインデクスページ数としてください。

  注※

  詳細については，「2.3.9　シーケンシャルインデクス」を参照してください。
- インデクスの格納ページ数の計算で使用する変数を，次のとおり読み替えてください。

  b：SDB データベース格納定義のシーケンシャルインデクス定義，または二次インデクス定義で指定した PCTFREE 句のページ内の未使用領域の比率※1（%）

  c，h：キーの種類の個数

  d：1

  e，f：0

  g：シーケンシャルインデクス，または二次インデクスのキーとなる構成要素の長さ※2（バイト）

  注※1

  ページ内の未使用領域の比率を指定しない場合は，30%とします。

  注※2

  キーとなる構成要素の長さは，次に示す式で求めます。

  ↑キー長÷4↑×4

  式中のキー長は，次の表を参照して求めてください。インデクスを構成する構成要素が複数の場合は，各構成要素に対してキー長を求め，キー長の合計を上記の式中のキー長としてください。

  4V AFM の SDB データベースの仮想ルートレコード型に定義するシーケンシャルインデクスのキー長は，次の手順で算出してください。

  1.　次の表の「インデクスを構成する構成要素が複数の場合」のキー長から，構成要素のキー長を求めます。インデクスを構成する構成要素が複数の場合は，各構成要素に対してキー長を求め，キー長の合計を算出します。

  2.　1.で求めたキー長に 6 バイトを加算します。この値を上記の式中のキー長としてください。

  4V AFM の SDB データベースの仮想ルートレコード型には，HiRDB/SD が内部的に使用するシステム用構成要素が定義されます。そのため，SDB データベース格納定義の ORDER KEY 句に指定した構成要素が 1 つだけであっても，上記の手順でキー長を算出してください。

表 3-33　データ型ごとのキー長

データ型	キー長	
	インデクスを構成する構成要素が 1 つの場合	インデクスを構成する構成要素が複数の場合
INTEGER	4	5
SMALLINT	2	3
PACKED [DECIMAL FIXED] 整数部〔，小数部〕	↓(整数部の桁数＋小数部の桁数)÷2↓＋1	↓(整数部の桁数＋小数部の桁数)÷2↓＋2
CHARACTER，XCHARACTER	定義した構成要素の長さ	定義した構成要素の長さ＋1

3.5.6　ユーザ用 RD エリア（追い付き反映キー対応表を格納する）の容量の見積もり【4V FMB】

更新可能なオンライン再編成の実行時に使用する追い付き反映キー対応表，およびそのインデクスを格納するユーザ用 RD エリアの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「ユーザ用 RD エリアの容量の計算方法」を参照してください。

(1)　表の格納ページ数を計算する際の留意事項

表の格納ページ数を計算する際は，マニュアル「HiRDB システム導入・設計ガイド」の「ユーザ用 RD エリアの容量の計算方法」の「表の格納ページ数の計算方法」の説明に従ってください。ただし，次の点に留意してください。

- 「表の格納ページ数の計算方法」では，「FIX 指定がない場合」の計算式を使用してください。
- 「計算式中で使用する変数」は，次の変数を使用してください。

  a：表に格納する行の総数（件）

  更新可能なオンライン再編成を実行する SDB データベースに格納されているレコードの件数＋更新可能なオンライン再編成中に再編成対象の SDB データベースに対してレコードの格納（STORE）を実行するレコードの件数

  b：ユーザ用 RD エリアのページ長（バイト）

  c：未使用領域の比率（%）

  未使用領域の比率は，0%で計算してください。

  di：各列のデータ長（バイト）

  各列のデータ長の合計は，418 バイトで計算してください。

  ei：列のデータ長の平均値（バイト）

  列のデータ長の平均値は，0 バイトで計算してください。

f：表に定義する列の総数（個）

表に定義されている列の数は，10 個で計算してください。

g：表を格納するユーザ用 RD エリアのセグメントサイズ（ページ）

h：セグメント内の空きページ比率（%）

セグメント内の空きページ比率は，0%で計算してください。

(2) インデクスの格納ページ数を計算する際の留意事項

インデクスの格納ページ数を計算する際は，マニュアル「HiRDB システム導入・設計ガイド」の「ユーザ用 RD エリアの容量の計算方法」の「インデクスの格納ページ数の計算方法」の説明に従ってください。ただし，次の点に留意してください。

- 「計算式中で使用する変数」は，次の変数を使用してください。

  a：ユーザ用 RD エリアのページ長（バイト）

  b：未使用領域の比率（%）

  未使用領域の比率は，0%で計算してください。

  c：キー値の重複が 200 以下のキーの種類の個数（個）

  更新可能なオンライン再編成を実行する SDB データベースに格納されているレコードの件数＋更新可能なオンライン再編成中に再編成対象の SDB データベースに対してレコードの格納（STORE）を実行するレコードの件数

  d：キー値の重複が 200 以下のキーの重複数の平均値（個）

  0 を代入して計算してください。

  e：キー値の重複が 201 以上のキーの種類の個数（個）

  0 を代入して計算してください。

  f：キー値の重複が 201 以上のキーの重複数の平均値（個）

  0 を代入して計算してください。

  g：DB 格納キー長（バイト）

  73 ＋ルートレコードのデータベースキー長※＋ SDB データベース定義中の最大階層数×4

  注※

  ルートレコード型で指定したデータベースキー（SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素の合計長）です。

  h：次の計算結果の値

  更新可能なオンライン再編成を実行する SDB データベースに格納されているレコードの件数＋更新可能なオンライン再編成中に再編成対象の SDB データベースに対してレコードの格納（STORE）を実行するレコードの件数

3.6　ユティリティ実行時の容量の見積もり

ユティリティ実行時のファイル容量およびメモリ所要量の見積もり方法について説明します。

3.6.1　ユティリティ実行時のファイル容量の見積もり

(1)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量

HiRDB/SD データベース作成ユティリティ（pdsdblod）で使用するファイルの容量の計算式を次の表に示します。

表 3-34　ファイル容量の計算式（pdsdblod）

ファイルの種類	計算式（単位：バイト）
入力データファイル	$\sum_{i=1}^{d}\{(34+a_i+g)\times b_i\}$
実行結果ファイル	$(1,800+130\times c\times d)+e\times 256+pgwng\times 750$
論理エラー情報ファイル	$\sum_{i=1}^{d}\{(34+a_i+g+20)\times e_i\}$
インデクス情報ファイル	$(K_{ij}+10)\times(indxrec+stkrec)+1536$ 上記の計算式で，1 インデクス情報に必要な容量が求まります。作成対象のインデクスごとに，該当するインデクス格納用 RD エリア数分を計算して合計してください。
ワークファイル（infmsglvl=lvl2 指定時）	$154+q\times 2$ $+\sum_{i=1}^{m}[\{386+(r+s+t)\times 3+u\times 2\}\times P_i]$
中間ファイル	$\sum_{i=1}^{d}\{22+v+(W_i-1)\times 8\}\times b_i$

変数の説明

- a：レコード長
- b：入力行数
- c：レコード格納用 RD エリア数
- d：レコード型の数
- e：エラーデータ件数
- g：プリフィクス長
- Kij：インデクスのキー長

インデクスのキー長の計算式については，「3.5.5(8)　インデクスを格納するページ数の見積もり」を参照してください。

- m：インデクス数（1 レコード型当たり）
- q：レコード型名長
- r：インデクス名称長
- s：RD エリア名称長
- t：サーバ名称長
- u：MIN（150，インデクス情報ファイルの絶対パス名長）
- v：SDB データベースのキー長
- Wi：レコード型の階層レベル（ルートレコードを 1 とする階層番号）
- indxrec：

  ・データロードの場合

  インデクスを定義したレコード型の入力データファイル中のレコード件数（RD エリア単位の値）を代入してください。ただし，4V AFM の SDB データベースのシーケンシャルインデクスについて見積もる場合は，「入力データファイル中に存在するデータベースキーの数（RD エリア単位の値）×2」を代入してください。

  ・インデクスの再作成の場合

  0 を代入してください。
- stkrec：

  ・初期データロードの場合（environment 文の purge オペランドに yes を指定した場合）

  0 を代入してください。

  ・追加データロードの場合（environment 文の purge オペランドに no を指定した場合）

  インデクスを定義したレコード型の SDB データベース中に格納済みのレコード件数（RD エリア単位の値）を代入してください。

  ・インデクスの再作成の場合

  インデクスを定義したレコード型の SDB データベース中に格納済みのレコード件数（RD エリア単位の値）を代入してください。ただし，4V AFM の SDB データベースのシーケンシャルインデクスについて見積もる場合は，「SDB データベース中に格納済みのデータベースキーの数（RD エリア単位の値）×2」を代入してください。
- pgwng：事前ページ割り当て警告件数

(2)　HiRDB/SD データベース再編成ユティリティ（pdsdbrog）実行時のファイルの容量

HiRDB/SD データベース再編成ユティリティ（pdsdbrog）で使用するファイルの容量の計算式を次の表に示します。

表 3-35　ファイル容量の計算式（pdsdbrog）

ファイルの種類	計算式（単位：バイト）
アンロードデータファイル	$\sum_{i=1}^{d}(34+a_i+b) \times c_i$
アンロードの実行結果ファイル	500 + 140× d

変数の説明

- a：レコード長
- b：プリフィクス長
- c：レコード実現値
- d：レコード型数

(3)　データベース複写ユティリティ（pdcopy）実行時のファイルの容量

データベース複写ユティリティ実行時のファイル容量については，マニュアル「HiRDB システム導入・設計ガイド」の「データベース複写ユティリティ（pdcopy）実行時のファイルの容量」を参照してください。

参照時には，差分バックアップファイルの容量算出式について，次の計算式で算出してください。

$$\sum_{i=1}^{n}(P_i \times L_i)$$

変数の説明

- P：バックアップ対象 RD エリアの更新ページ数
- L：ページ長
- n：複写する RD エリアの数

変数 P のバックアップ対象 RD エリアの更新ページ数は，前回のフルバックアップまたは差分バックアップの取得時から，現在までの間に更新されたページ数を意味しています。

レコードを格納している RD エリアとインデクスを格納している RD エリアをそれぞれ計算してください。

・レコードを格納している RD エリアの場合

バックアップ対象 RD エリアの更新ページ数（P）は，次の計算式に従って計算します。

```
P = MIN (R, fi)
```

格納，更新，削除対象以外のレコードに対して，レコード中のポインタを更新することによって，P の数が見積もりより多くなることがあります。

変数の説明

- R：前回のフルバックアップまたは差分バックアップの取得時から，次の機能で操作したレコードの数（同一レコードの操作は加算しない）

  ・レコードの格納（HiRDB/SD データベース作成ユティリティ（pdsdblod）で追加するレコードを含む）

  ・レコードの更新

  ・レコードの削除

  ・一括削除（SDB データベースを操作する API1 回の実行について，操作したレコードの数を 1 とする）

- fi：RD エリアの使用ページ数

  データベース状態解析ユティリティ（pddbst）の RD エリア単位の状態解析（物理的解析）結果から得られる使用ページ数です。

・インデクスを格納している RD エリアの場合

インデクスを格納するユーザ用 RD エリアの更新ページ数（P）は，次の計算式に従って計算します。

```
P = MIN (X, fi)
```

変数の説明

- X：インデクスが定義されたレコード型に対して，前回のフルバックアップまたは差分バックアップの取得時から，次の機能で操作したレコードの数（同一レコードの操作は加算しない）

  ・レコードの格納（HiRDB/SD データベース作成ユティリティ（pdsdblod）で追加するレコードを含む）

  ・レコードの更新

  ・レコードの削除

- fi：RD エリアの使用ページ数

  データベース状態解析ユティリティ（pddbst）の RD エリア単位の状態解析（物理的解析）結果から得られる使用ページ数です。

(4) 統計解析ユティリティ（pdstedit）実行時のファイルの容量

統計解析ユティリティ（pdstedit）で使用するファイルの容量の計算式を次の表に示します。その他，統計解析ユティリティ（pdstedit）実行時に使用するファイルの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「統計解析ユティリティ（pdstedit）実行時のファイルの容量」を参照してください。

表 3-36 ファイル容量の計算式（pdstedit）

ファイルの種類		容量の計算式（単位：バイト）
ワーク用ファイル	UAP に関する統計情報	（872×実行 UAP 数または実行トランザクション数）×2

ファイルの種類		容量の計算式（単位：バイト）
DAT 形式ファイル		2,024×実行 UAP 数または実行トランザクション数

(5) データベース状態解析ユティリティ（pddbst）実行時のファイルの容量

データベース状態解析ユティリティ（pddbst）の「RD エリア単位の論理的解析」で使用するファイルの容量の計算式を次の表に示します。

表 3-37 ファイル容量の計算式（pddbst）

ファイルの種類		容量の計算式（単位：キロバイト）
ワーク用ファイル	RD エリア単位の論理的解析	$1+6.2\times\sum_{i=1}^{a}$（RDエリア内の表数i ＋インデクス数i ※ ＋RDエリア内のルートレコード型数i）

注※

上記計算式中のインデクスは，SDB データベースのインデクス，および RDB の B-tree インデクスの両方が該当します。

変数の説明

a：解析対象の RD エリア数

データベース状態解析ユティリティ（pddbst）実行時に使用するそのほかのファイルの容量見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「データベース状態解析ユティリティ（pddbst）実行時のファイルの容量」を参照してください。

なお，マニュアル「HiRDB システム導入・設計ガイド」に記載されている計算式を使って計算する際，計算式中の係数を 4.1 から 6.2 に変更して計算してください。

（例）

「RD エリア単位の物理的解析」のワーク用ファイルの容量を求める場合は，次のように計算式中の係数を 4.1 から 6.2 に変更して計算してください。

1 ＋ 4.1×a※ → 1 ＋ 6.2×a

注※

マニュアル「HiRDB システム導入・設計ガイド」に記載されている計算式です。

(6) DML プリプロセサ（pdsdbcbl）実行時のファイルの容量【SD FMB】

DML プリプロセサ（pdsdbcbl）実行時のファイル容量の計算式を次の表に示します。

表 3-38 ファイル容量の計算式（pdsdbcbl）

ファイルの種類	容量の計算式（単位：キロバイト）
ポストソースファイル	uapsrc ＋ 150×0.1×dmln ＋ 100×0.1

ファイルの種類	容量の計算式（単位：キロバイト）
簡易ダンプファイル	コマンドプロセス（pdsdbcbl）が使用するプロセス固有領域サイズ＋（0.1×16）

変数の説明

uapsrc：UAP ソースファイルの容量

dmln：UAP ソースファイル内の DML の数

3.6.2 ユティリティ実行時のメモリ所要量の見積もり

(1) HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のメモリ所要量

HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のメモリ容量の計算式を次の表に示します。

表 3-39 メモリ所要量（pdsdblod）

項目	メモリ所要量の計算式（キロバイト）
コマンドプロセス（pdsdblod）が使用するプロセス固有領域	49,000 ＋ 4.7 ＋ luntsvr ＋ lsdbcpc ＋ lsdbdir ＋ dbs
ユティリティサーバプロセス（pdsdblodm）が使用するプロセス固有領域	58,900 ＋ 0.6 ＋ luntsvr ＋ lsdbcpc ＋ lsdbdir
ユーザサーバプロセス（pdbes）が使用するプロセス固有領域	94,900 ＋ 0.3 ＋ luntsvr ＋ lsdbcpc ＋ lsdbdir ＋ dbs ＋ lload ＋ lloada ＋ lfwrt ＋ lindx ＋ lindxa ＋ lscana ＋ lpurge

計算式中で使用している変数の値および意味を次に示します。

変数	変数の値および意味
luntsvr	0.02 ＋（unit×0.04）＋（1 ＋ ds ＋ fes ＋ bes）×0.05 • unit：全ユニット数 • ds：全ディクショナリサーバ数 • fes：全フロントエンドサーバ数 • bes：全バックエンドサーバ数
lsdbcpc	9.6 ＋ 0.1×recara ＋ 0.2×rec • recara：格納対象の格納レコード用 RD エリア数 • rec：格納対象 SDB データベースの全レコード型数
lsdbdir	0.18 ＋ 0.05×area ＋ 0.05×set ＋ 0.06×rec ＋ 0.06×cmp ＋ 0.14×recara ＋ 0.07×（1 ＋ secidx） • area：格納対象 SDB データベースの全 RD エリア数 • set：格納対象 SDB データベースの全親子集合の数 • rec：格納対象 SDB データベースの全レコード型数

変数	変数の値および意味
	• cmp：格納対象 SDB データベースの全構成要素数 SDB データベース定義で OCCURS 句を指定する場合，システム内で生成される構成要素数を含みます。 • recara：lsdbcpc で記載した recara と同じです。 • secidx：格納対象 SDB データベースの全二次インデクス数
dbs	SDB データベース情報長（単位：キロバイト） 「3.4.2(2)(a)　SDB ディレクトリ情報長の計算式」の変数 dbs の説明を参照してください。
lload	インデクス再作成機能を実行しない場合に加算します。 40.3 ＋ lmaxrec ＋ 0.02×recara ＋ 0.01×divrec ＋ 2×lbuf ＋ lstor • lmaxrec：格納対象 SDB データベースの全レコード型のうち，最大のレコード長 • recara：lsdbcpc で記載した recara と同じです。 • divrec：ルートレコードの横分割格納 RD エリア数 • lbuf：システム共通定義の pd_utl_file_buff_size オペランドに指定した値 • lstor：「3.4.3(2)　SDB データベースを操作する API または DML の実行時に，バックエンドサーバで確保するメモリ」を参照して算出してください。使用する pdsdblod の値を次に示します。 ・個別開始要求回数：1 ・API 個別算出式：STORE 算出式
lloada	インデクス再作成機能を実行しない場合に加算します。 ↑（(128 ＋ 24×recarea）÷1,024）↑ • recarea：格納対象の格納レコード用 RD エリア数
lfwrt	フォーマットライトまたはインデクス再作成機能を実行しない場合に加算します。 2×lmaxrec ＋ 0.1×lkeydef • lmaxrec：lload で記載した lmaxrec と同じです。 • lkeydef：SDB データベース格納定義の KEYDEF DATA に指定した数
lindx	インデクス一括作成モードでインデクスを作成またはインデクス再作成機能を実行する場合に加算します。 1.5 ＋（linxf ＋ 0.1 ＋ 0.1×inxkey）×（1 ＋ secidx）×inxara ＋ lsort • linxf：インデクス情報ファイルの絶対パス名長 • inxkey：各インデクスキーの構成要素数 • secidx：lsdbdir で記載した secidx と同じです。 • inxara：格納対象のインデクス格納用 RD エリア数 • lsort：idxload 文 sortdir オペランドのソート用バッファサイズ指定値（省略時は 1）
lindxa	インデクス一括作成モードでインデクスを作成またはインデクス再作成機能を実行する場合に加算します。 ↑（(4,424 ＋ 16×（rec ＋ idx ＋ 2）＋ 8×idx×（recarea ＋ 2）＋ 8,256×MIN（idx×recarea, 7,000)）÷1,024）↑ • rec：格納対象 SDB データベースの全レコード型数 • idx：格納対象 SDB データベースの全インデクス数 • recarea：格納対象の格納レコード用 RD エリア数
lscana	追加データロードまたはインデクスの再作成を実行する場合に加算します。 ↑ {(56,290 ＋ 2×xpgsz ＋ 24×recarea2 ＋ 2×rec2）÷1024} ↑

変数	変数の値および意味
	• xpgsz：インデクスを格納している RD エリアの中で最大のページサイズ • recarea2：格納対象の格納レコード用 RD エリア数（インデクスの再作成の場合は 1） • rec2：格納対象の SDB データベースの全レコード型数（インデクスの再作成の場合は 1）
lpurge	environment 文の purge オペランドに yes を指定する場合に加算します。 2.3 + 0.01×recara +（0.3 + 0.1×inxkey）×（1 + secidx）×inxara • recara：lsdbcpc で記載した recara と同じです。 • inxkey：lindx で記載した inxkey と同じです。 • secidx：lsdbdir で記載した secidx と同じです。 • inxara：lindx で記載した inxara と同じです。

(2) HiRDB/SD データベース再編成ユティリティ（pdsdbrog）実行時のメモリ所要量

HiRDB/SD データベース再編成ユティリティ（pdsdbrog）実行時のメモリ容量の計算式を次の表に示します。

表 3-40 メモリ所要量（pdsdbrog）

項目	メモリ所要量の計算式（キロバイト）
コマンドプロセス（pdsdbrog）が使用するプロセス固有領域	49,100 + 4.5 + runtsvr + rsdbcpc + rsdbdir + rsdbrog + dbs + rgen
ユティリティサーバプロセス（pdsdbrogm）が使用するプロセス固有領域	58,900 + 0.6 + runtsvr + rsdbrog
ユーザサーバプロセス（pdbes）が使用するプロセス固有領域	93,600 + 0.3 + rsdbcpc + rsdbrog + dbs + runlod

計算式中で使用している変数の値および意味を次に示します。

変数	変数の値および意味
runtsvr	0.02 +（unit×0.04）+（1 + ds + fes + bes）×0.05 • unit：全ユニット数 • ds：全ディクショナリサーバ数 • fes：全フロントエンドサーバ数 • bes：全バックエンドサーバ数
rsdbcpc	$0.15+\{0.04\times(1+\mathrm{idxara})\}+\sum_{i=1}^{rec}\{0.22+(0.01\times1)+(0.03\times\mathrm{CMP}_i)\}$ • idxara：アンロード対象エリアに対応する SDB データベース格納定義の STORAGE RECORD 句下にある SEQUENTIAL 句下の WITHIN 句に指定したシーケンシャルインデクス用 RD エリア数と，SECONDARY INDEX 句下にある WITHIN 句に指定した二次インデクス用 RD エリア数の和 同一 RD エリア名はカウントしません（重複排除）。

変数	変数の値および意味
	• rec：SDB データベース定義の RECORD 句の指定数（レコード型の数） • CMPi：SDB データベース定義の RECORD 句で指定した構成要素数 集団項目を含みます。また，SDB データベース定義で OCCURS 句を指定する場合，システム内で生成される構成要素数を含みます。
rsdbdir	0.18 ＋ 0.05×rec ＋ 0.14×1 ＋（0.07×（seqidx ＋ secidx）×rec） • rec：rsdbcpc で記載した rec と同じです。 • seqidx：対応する SDB データベース格納定義の STORAGE RECORD 句下にある SEQUENTIAL 句の指定数 • secidx：対応する SDB データベース格納定義の STORAGE RECORD 句下にある SECONDARY INDEX 句の指定数
rsdbrog	2.32 ＋（1 ＋ idxara）×0.17 • idxara：rsdbcpc で記載した idxara と同じです。
dbs	SDB データベース情報長（単位：キロバイト） 「3.4.2(2)(a)　SDB ディレクトリ情報長の計算式」の変数 dbs の説明を参照してください。
rgen	0.04 ＋ {0.1×（allara ＋ 1 ＋ idxara)} • allara：対応する SDB データベース格納定義に指定した RD エリア数※。同一 RD エリア名はカウントしません（重複排除)。 • idxara：rsdbcpc で記載した idxara と同じです。
runlod	0.6 ＋ rsdi ＋ rrecun ＋ rwrite ＋ rdml • rsdi：0.8 ＋ maxrl • maxrl：SDB データベースで最大のレコード長 • rrecun：0.04 ＋ prefix×0.01 ＋ maxrl • prefix：制御文ファイルの prefix オペランド指定値 • maxrl：rsdi で記載した maxrl と同じです。 • rwrite：8.02 ＋ fbuff×2 • fbuff：システム共通定義の pd_utl_file_buff_size オペランド指定値 • rdml：「3.4.3　プロセス固有メモリの見積もり」から算出してください。使用する pdbes の値を次に示します。 ・個別開始要求回数：1 ・API 個別算出式 ＝ FETCH 算出式

注※

格納レコード用 RD エリア数と，シーケンシャルインデクス用 RD エリア数と，二次インデクス用 RD エリア数の総和

(3)　HiRDB/SD データベースアクセスユティリティ（pdsdbexe）実行時のメモリ所要量

HiRDB/SD データベースアクセスユティリティ（pdsdbexe）実行時のメモリ容量の計算式を次の表に示します。

表 3-41　メモリ所要量（pdsdbexe）

項目	メモリ所要量の計算式（キロバイト）
コマンドプロセス（pdsdbexe）が使用するプロセス固有領域	255 ＋ 66×l_cmd ＋ 0.2×n_sdb ＋ 30.3×n_rec ＋ 0.1×n_cmp ＋ dbs
コマンドプロセス（pdsdbexe）が使用する仮想メモリサイズ	41,400

計算式中で使用している変数の値および意味を次に示します。

変数	変数の値および意味
l_cmd	{pdsdbexe 操作コマンドの文字列長の最大長（バイト）÷65,536} ＋ 1
n_sdb	SDB データベース数
n_rec	SDB データベース内のレコード数
n_cmp	SDB データベース内の構成要素数（集団項目を含んだ値）
dbs	SDB データベース情報長（1SDB データベース情報のサイズ）（単位：キロバイト） -d オプションに SD を指定した場合に加算します。 SDB データベース情報長の求め方については，「3.4.2(2)(a)　SDB ディレクトリ情報長の計算式」を参照してください。

(4)　データベース複写ユティリティ（pdcopy）実行時のメモリ所要量

データベース複写ユティリティ実行時のメモリ所要量については，マニュアル「HiRDB システム導入・設計ガイド」の「データベース複写ユティリティ（pdcopy）実行時のメモリ容量」を参照してください。

(5)　データベース回復ユティリティ（pdrstr）実行時のメモリ所要量

データベース回復ユティリティ（pdrstr）実行時のメモリ所要量については，マニュアル「HiRDB システム導入・設計ガイド」の「データベース回復ユティリティ（pdrstr）実行時のメモリ容量」を参照してください。

(6)　データベース状態解析ユティリティ（pddbst）実行時のメモリ所要量

データベース状態解析ユティリティ（pddbst）実行時のメモリ所要量については，マニュアル「HiRDB システム導入・設計ガイド」の「データベース状態解析ユティリティ（pddbst）実行時のメモリ所要量」を参照してください。

なお，参照する際，「計算式で使用する変数」については，次の変数を使用してください。

- α（単位：バイト）

  {28000 ＋ A ＋（10592×a）＋（10592×b）＋（10592×c）＋（2264×d）＋（848×e）＋（272×f）＋（432×g）＋（304×h）＋（1024×r)}

- β（単位：バイト）
  {100000 ＋（1024×d）＋（2784×i）＋（2784×j）＋（2784×k）＋（1024×r)}
- r：解析対象の RD エリアがある BES が解析するルートレコード型数

(7) DML プリプロセサ（pdsdbcbl）実行時のメモリ所要量【SD FMB】

DML プリプロセサ（pdsdbcbl）実行時のメモリ容量の計算式を次の表に示します。

表 3-42 メモリ所要量（pdsdbcbl）

項目	メモリ所要量の計算式（キロバイト）
コマンドプロセス（pdsdbcbl）が使用するプロセス固有領域	$300+\sum_{i=1}^{dbn}(dbs(i)+1.5)+(prgn\times 17.0)+\uparrow varn\div 255\uparrow \times 41$
コマンドプロセス（pdsdbcbl）が使用する仮想メモリサイズ	23,000

計算式中で使用している変数の値および意味を次に示します。

変数	変数の値および意味
dbn	SDB データベース節に指定した SDB データベースの数
dbs	SDB データベース節に指定した SDB データベースの SDB データベース情報長 SDB データベース情報長の求め方については，「3.4.2(2)(a) SDB ディレクトリ情報長の計算式」を参照してください。
prgn	UAP ソースファイル内のプログラム数
varn	UAP ソースファイル内の変数の数

3.7　OS のオペレーティングシステムパラメタの見積もり

ここでは，オペレーティングシステムパラメタの見積もりについて説明します。

3.7.1　カーネルパラメタの見積もり

Linux のオペレーティングシステムパラメタ（カーネルパラメタ）の指定値を見積もります。見積もり方法の詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「Linux のオペレーティングシステムパラメタの見積もり」を参照してください。

なお，HiRDB/SD 定義ユティリティ（pdsdbdef），HiRDB/SD データベース作成ユティリティ（pdsdblod），および HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の実行時に使用する通常ファイルのオープン処理に伴い，OS のオペレーティングシステムパラメタ（カーネルパラメタ）の fs.file-max が消費されます。各ユティリティが，通常ファイルとして使用するファイルを次に示します。

- HiRDB/SD 定義ユティリティ（pdsdbdef）が通常ファイルで使用するファイル
  - SDB 制御文ファイル
  - SDB 定義文ファイル
  - 実行結果ファイル
- HiRDB/SD データベース作成ユティリティ（pdsdblod）が通常ファイルで使用するファイル
  - 制御文ファイル
  - 入力データファイル
  - 実行結果ファイル
  - 論理エラー情報ファイル
  - インデクス情報ファイル
  - ソート用ワークファイル
  - 中間ファイル
- HiRDB/SD データベース再編成ユティリティ（pdsdbrog）が通常ファイルで使用するファイル
  - 制御文ファイル
  - アンロードデータファイル
  - アンロードの実行結果ファイル

fs.file-max については，OS のマニュアルを参照してください。

3.7.2　メッセージキューおよびセマフォ所要量の見積もり

メッセージキューおよびセマフォ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「メッセージキューおよびセマフォ所要量の見積もり」の「HiRDB/パラレルサーバの場合の計算式」を参照してください。

3.8　排他資源数の見積もり

SDB データベースを操作する API または DML や，ユティリティを実行するときに必要となる排他資源数の概算式を示します。概算式から最大の排他資源数を算出し，さらに余裕値を加算して，pd_lck_pool_size オペランドにプールサイズを設定してください。pd_lck_pool_size オペランドについては，マニュアル「HiRDB システム定義」の「排他制御に関するオペランド」を参照してください。

ここで示す排他資源数は，トランザクション内で有効となります。1 つのトランザクション内で複数の SDB データベースを操作する API または DML を実行する場合，マニュアル「HiRDB システム定義」の「排他資源数の見積もり」，および「3.8.1　定義系 SQL 実行時の排他資源数」～「3.8.2　SDB データベースを操作する API または DML の実行時の排他資源数」で示す概算式から算出した排他資源数の値の総和が必要です。ただし，すでに排他が掛かっている場合は不要となります。

3.8.1　定義系 SQL 実行時の排他資源数

定義系 SQL 実行時の排他資源数については，マニュアル「HiRDB システム定義」の「排他資源数の見積もり」の「定義系 SQL」を参照してください。

3.8.2　SDB データベースを操作する API または DML の実行時の排他資源数

SDB データベースを操作する API または DML の実行時に，バックエンドサーバで必要とする排他資源数の概算式を次に示します。

なお，SD FMB の SDB データベースを操作する場合，SDB 用 UAP 環境定義の subschema オペランドに指定する SD 排他モードとアクセス目的の指定の組み合わせが次のどれかのときは，ページ（サブページ分割をしている場合はサブページ）に排他が掛かりません。

- SD 排他モードが protected で，アクセス目的が retrieve の場合
- SD 排他モードが exclusive で，アクセス目的が retrieve の場合
- SD 排他モードが exclusive で，アクセス目的が update の場合
- SD 排他モードが nonprotected の場合

上記の条件に該当する場合は，ページの排他資源数は 0 として見積もってください。

(1)　レコードの検索（FETCH），または位置指示子の位置づけ（FIND）

レコードの検索（FETCH），または位置指示子の位置づけ（FIND）をするときの排他資源数の概算式について説明します。

- 4V FMB または 4V AFM の SDB データベースを API で操作する場合

SDB データベースを操作する API の排他モード 2 の指定によって，(a)または(b)の計算式で排他資源数を求めてください。

- 4V FMB または 4V AFM の SDB データベースを HiRDB/SD データベースアクセスユティリティ（pdsdbexe）で操作する場合

  START コマンドの AUTODEQ オペランドに NO を指定しているとき，または AUTODEQ オペランドの指定を省略しているときは，(a)の計算式で排他資源数を求めてください。

  AUTODEQ オペランドに YES を指定しているときは，(b)の計算式で排他資源数を求めてください。

- SD FMB の SDB データベースを DML 文，または HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の DML コマンドで操作する場合

  SDB 用 UAP 環境定義の lockrange オペランドに transaction を指定している場合，または lockrange オペランドを省略している場合は，(a)の計算式で排他資源数を求めてください。

  lockrange オペランドに cursorupdate を指定している場合は，(b)の計算式で排他資源数を求めてください。

(a) SDB データベースを操作する API の排他モード 2 が'C'の場合

検索対象のレコード格納用RDエリア数＋1
＋検索ヒットレコード実現値格納ページ数※1
＋利用するインデクスのインデクス格納用RDエリア数＋1※2
＋検索範囲キー値数※3＋1※4

注※1

- 4V AFM の SDB データベースで仮想ルートレコードの検索が発生する場合（該当データベースキー値に対する 1 回目のアクセス時）は 2，それ以外は 1 を加算します。

  ただし，SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）を指定する場合，または TAM のデータベースの無排他検索機能が動作する場合は 0 を代入します。

- サブページ分割をする場合は，「検索ヒットレコード実現値格納サブページ数」になります。

注※2

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

注※3

4V AFM の SDB データベースで指定の RD エリアを検索範囲にする場合は，検索範囲開始キーから検索範囲終了キーまでのキー値数（検索開始から検索終了までの仮想ルートレコードのレコード実現値数）－1，それ以外は 0 を代入します。

なお，SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）を指定する場合，または 4V TAM の SDB データベースの場合は 0 を代入します。

注※4

SD FMB の SDB データベースで，二次インデクスを使用して子レコードを検索する場合に加算します。

(b)　SDB データベースを操作する API の排他モード 2 が'D'の場合

検索対象のレコード格納用RDエリア数＋1
＋検索ヒットレコード実現値格納ページ数※1
＋利用するインデクスのインデクス格納用RDエリア数＋1※2＋1※3

注※1

- 4V AFM の SDB データベースで仮想ルートレコードの検索が発生する場合（該当データベースキー値に対する 1 回目のアクセス時，または 4V AFM の SDB データベースで指定の RD エリアを検索範囲としたとき）は 2，それ以外は 1 を加算します。
- サブページ分割をする場合は，「検索ヒットレコード実現値格納サブページ数」になります。

注※2

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

注※3

SD FMB の SDB データベースで，二次インデクスを使用して子レコードを検索する場合に加算します。

注意事項

SDB データベースを操作する API または DML の実行時には，不要なページまたはサブページの排他が解除されるため，確保している排他資源の総数を計算する場合には，単に積算しないでください。

また，SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）を指定する場合，TAM のデータベースの無排他検索機能が動作する場合，または SDB 用 UAP 環境定義の subschema オペランドの-e オプションに nonprotected を指定している場合は，排他自動解除機能が適用されないため，「3.8.2(1)(a)　SDB データベースを操作する API の排他モード 2 が'C'の場合」の計算式を参照してください。

排他自動解除機能については，「2.9.4　排他自動解除機能」を参照してください。

(2)　レコードの格納（STORE）

一括削除後にレコードを格納する場合，削除前の一連番号の最大値に達するまでは，内部的にレコードの更新（MODIFY）と同等の変更処理に置き換わります。その場合の排他資源数は「3.8.2(2)(a)　一括削除したあとにレコードを格納する場合」を参照してください。それ以外の場合の排他資源数は「3.8.2(2)(b)　一括削除しないでレコードを追加する場合」を参照してください。

(a)　一括削除したあとにレコードを格納する場合

4＋利用するインデクスのインデクス格納用RDエリア数※1＋1※2＋1※3

注※1

二次インデクスが定義されたレコードの格納の場合は，2 を加算します。

それ以外の場合は，1 を加算します。

注※2

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

注※3

レコードの削除によって，当該データベースキー値を持つ子レコードが 0 件になったあとに実行する，1 件目のレコードの格納の場合に加算します。

(b) 一括削除しないでレコードを追加する場合

格納対象RDエリア数※1＋4＋5※2
＋インデクスが定義された格納レコード実現値数※3＋1※4＋（事前割り当てページ数-1）※5＋1※6

注※1

対象となるレコード型，インデクスが格納されている RD エリアの総数です。

注※2

直前および直後の子レコードがない場合は，その分減算します。また，親レコードがルートレコードの場合は 1 を，ルートレコードの追加の場合は 2 を減算します。

注※3

次の値を代入してください。

- 4V FMB または SD FMB の SDB データベースでルートレコードの格納の場合：1
- 4V FMB の SDB データベースで二次インデクスが定義された子レコードの格納の場合：1
- 4V AFM の SDB データベースで二次インデクスが定義されたレコードの 1 件目の格納の場合：2

  レコードの削除によって，当該データベースキー値を持つ子レコードが 0 件となったあとに実行する 1 件目のレコードの格納を含みます。
- 4V AFM の SDB データベースで二次インデクスが定義されたレコードの 2 件目以降の格納の場合：1
- 4V AFM の SDB データベースで二次インデクスが定義されていないレコードの 1 件目の格納の場合：1

  レコードの削除によって，当該データベースキー値を持つ子レコードが 0 件となったあとに実行する 1 件目のレコードの格納を含みます。
- それ以外の場合：0

注※4

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

注※5

- 該当するファミリに事前ページ割り当て機能を適用している場合，確保予定の事前割り当てページの数を加算します。なお，このレコードの格納でレコード実現値を格納するページ分については，加算済みであるため，1 を減算しています。
- サブページ分割をする場合は，「(事前割り当てサブページ数－1)」になります。

注※6

システム共通定義の pd_lock_uncommited_delete_data オペランドに WAIT を指定し，かつ 4V FMB または SD FMB の SDB データベースの場合に加算します。

(3) レコードの更新（MODIFY）

更新対象RDエリア数※4＋1＋変更レコード実現値格納ページ数※1＋1※2＋1※3＋1※5＋5※6

注※1

- 次に示す値を加算します。
  ・ユーザデータの変更だけの場合：1
  ・USER ポインタの設定だけの場合：2
  ・USER ポインタの設定の解除だけの場合：1
  ・ユーザデータの変更と，USER ポインタの設定または設定の解除を同時に実行する場合：2
- サブページ分割をする場合は，「変更レコード実現値格納サブページ数」になります。

注※2

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

注※3

ユーザデータの変更だけを行う場合，対象レコードがルートレコード以外であれば加算します。

USER ポインタの設定または設定の解除を行う場合，親レコードがルートレコード以外であれば加算します。

注※4

- 4V FMB または 4V AFM の SDB データベースの場合
  1 を加算します。
- SD FMB の SDB データベースの場合
  対象となるレコード型および二次インデクスを格納する RD エリアの総数を加算します。

注※5

SD FMB の SDB データベースで，二次インデクスを定義したレコードを更新する場合に加算します。

注※6

SD FMB の SDB データベースで，キー項目を更新する場合に加算します。

なお，親レコードがルートレコードの場合は，5 ではなく 4 を加算してください。

(4) レコードの削除（ERASE）

削除対象RDエリア数※1＋1＋削除レコード実現値格納ページ数※2＋4※3＋1※4
＋削除レコード実現値数※5＋1※6＋未使用の事前割り当てページ数※7

注※1

対象となるレコード型，インデクスが格納されている RD エリアの総数です。

注※2

サブページ分割をする場合は，「削除レコード実現値格納サブページ数」になります。

注※3

直前および直後の子レコードがない場合は，その分減算します。また，親レコードがルートレコードの場合は 1 を減算します。

注※4

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

注※5

インデクスが定義されたレコード実現値を削除する場合に加算します。

注※6

4V AFM の SDB データベースで，当該データベースキー値を持つ子レコードが 0 件になる場合に加算します。

注※7

- 削除対象レコードがルートレコードであり，該当するファミリを格納するために事前割り当てページを確保している場合，レコードが格納されていない事前割り当てページ数を加算します。
- サブページ分割をする場合は，「未使用の事前割り当てサブページ数」になります。

(5) レコードの一括削除【4V DAM，4V SAM】

削除対象のレコード格納用RDエリア数＋3＋1※
＋利用するインデクスのインデクス格納用RDエリア数

注※

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

(6) 複数レコードの検索（FETCHDB ALL）【4V FMB】

検索対象RDエリア数※1＋1
＋1※2
＋1※3

注※1

前回の複数レコードの検索で取得したファミリの，続きのレコードを取得する場合は 1（レコードを格納する RD エリア分），それ以外は 2（レコードを格納する RD エリアおよびシーケンシャルインデクスを格納する RD エリア分）を加算します。

注※2

前回の複数レコードの検索で取得したファミリの，続きのレコードを取得する場合は 0 とします。

注※3

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

(7) 構成情報取得【4V AFM】

構成情報取得対象のレコード格納用RDエリア数＋1＋2※1
＋利用するインデクスのインデクス格納用RDエリア数＋1※2

注※1

対象の仮想ルートレコードが 1 つしか存在しない場合は，1 となります。

また，SDB データベースを操作する API の排他モード 1 に'N'（無排他モード）を指定する場合，または TAM のデータベースの無排他検索機能が動作する場合は 0 となります。

注※2

システム共通定義の pd_inner_replica_control オペランドを指定している場合に加算します。

3.8.3 HiRDB/SD 定義ユティリティ（pdsdbdef）実行時の排他資源数

HiRDB/SD 定義ユティリティ（pdsdbdef）の実行時には，次の計算式で算出したサーバ当たりの排他資源数が必要になります。

(1) *ENTRY DICTIONARY 文（SCHEMA 指定）の実行時

- ディクショナリサーバで必要とする排他資源数

```
27＋↑DB÷（DICPG-59）↑＋↑DBVW÷（DICPG-59）↑
```

(2) *ENTRY DICTIONARY 文（STORAGE SCHEMA 指定）の実行時

- ディクショナリサーバで必要とする排他資源数

```
18＋↑STDB÷（DICPG-59）↑
```

(3) *ENTRY DIRECTORY 文の実行時

- ディクショナリサーバで必要とする排他資源数

```
36＋↑DBVW÷（DICPG-59）↑×2＋↑DBS÷（DICPG-59）↑
＋↑SUB÷（DICPG-59）↑
```

*CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{RECN} CRTTBL + \sum_{j=1}^{IDXN} CRTIDX$$

*ALTER DICTIONARY 文で変更したデータベース定義に対して，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{ADRECN} CRTTBL + \sum_{j=1}^{ADIDXN} CRTIDX$$

- フロントエンドサーバおよびバックエンドサーバで必要とする排他資源数

  *CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{RECN} CRTTBL + \sum_{j=1}^{IDXN} CRTIDX$$

  *ALTER DICTIONARY 文で変更したデータベース定義に対して，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{ADRECN} CRTTBL + \sum_{j=1}^{ADIDXN} CRTIDX$$

(4) *ALTER DICTIONARY 文（SCHEMA 指定）の実行時【4V FMB，4V AFM】

- ディクショナリサーバで必要とする排他資源数

```
27＋↑DB÷（DICPG-59）↑＋↑DBVW÷（DICPG-59）↑
```

(5) *ALTER DICTIONARY 文（STORAGE SCHEMA 指定）の実行時【4V FMB，4V AFM】

- ディクショナリサーバで必要とする排他資源数

```
18＋↑STDB÷（DICPG-59）↑
```

(6) *ALTER DIRECTORY 文の実行時【4V FMB，4V AFM】

- ディクショナリサーバで必要とする排他資源数

```
36＋↑DBVW÷（DICPG-59）↑×2＋↑DBS÷（DICPG-59）↑
＋↑SUB÷（DICPG-59）↑
```

*ALTER DICTIONARY 文で変更したデータベース定義に対して，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{ADRECN} CRTTBL + \sum_{j=1}^{ADIDXN} CRTIDX$$

- フロントエンドサーバおよびバックエンドサーバで必要とする排他資源数

  *ALTER DICTIONARY 文で変更したデータベース定義に対して，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{ADRECN} CRTTBL + \sum_{j=1}^{ADIDXN} CRTIDX$$

- バックエンドサーバで必要とする排他資源数

  *ALTER DICTIONARY 文で格納 RD エリアの変更をした SDB データベース定義に対して，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{ALRECN} (ALTRDA + PRGTBL)$$

(7) *DELETE DICTIONARY 文（SCHEMA 指定）の実行時

- ディクショナリサーバで必要とする排他資源数

```
15＋↑DB÷（DICPG-59）↑＋↑DBVW÷（DICPG-59）↑
```

削除する SDB ディクショナリ情報に対して，*CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していて，その後，SDB データベース格納定義を削除していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{RECN} DRPTBL$$

- フロントエンドサーバおよびバックエンドサーバで必要とする排他資源数

  削除する SDB ディクショナリ情報に対して，*CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していて，その後，SDB データベース格納定義を削除していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{RECN} DRPTBL$$

(8) *DELETE DICTIONARY 文（STORAGE SCHEMA 指定）の実行時

- ディクショナリサーバで必要とする排他資源数の計算式を次に示します。

```
8＋↑STDB÷（DICPG-59）↑
```

削除する SDB ディクショナリ情報に対して，*CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していて，その後，SDB データベース定義を削除していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{RECN} DRPTBL$$

- フロントエンドサーバおよびバックエンドサーバで必要とする排他資源数

削除する SDB ディクショナリ情報に対して，*CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していて，その後，SDB データベース定義を削除していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{RECN} DRPTBL$$

(9) *DELETE DIRECTORY 文の実行時

- ディクショナリサーバで必要とする排他資源数

```
12＋↑DBS÷（DICPG-59）↑＋↑SUB÷（DICPG-59）↑
```

(10) *CHECK DICTIONARY 文（STORAGE SCHEMA 指定）の実行時

- ディクショナリサーバで必要とする排他資源数

```
23＋↑DB÷（DICPG-59）↑＋↑STDB÷（DICPG-59）↑×2
```

*CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

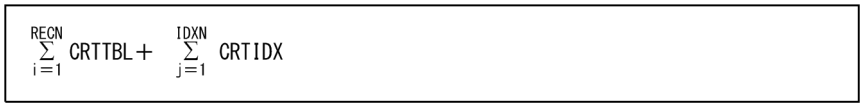
$$\sum_{i=1}^{RECN} CRTTBL + \sum_{j=1}^{IDXN} CRTIDX$$

*ALTER DICTIONARY 文で変更したデータベース定義に対して，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

$$\sum_{i=1}^{ADRECN} CRTTBL + \sum_{j=1}^{ADIDXN} CRTIDX$$

- フロントエンドサーバおよびバックエンドサーバで必要とする排他資源数

*CHECK DICTIONARY 文または*ENTRY DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。



*ALTER DICTIONARY 文で変更したデータベース定義に対して，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文を実行していない場合は，次の計算式で算出した値を加算してください。

(11)　計算式で使用する変数の説明

計算式中で使用している変数の意味を次に示します。

変数	変数の意味
DICPG	データディクショナリ用 RD エリアのページ長（バイト）
DB	SDB データベース定義情報長（バイト） 「表 3-26　計算式 2（SDB データベース定義情報長）」を参照して算出してください。
DBVW	SDB データベースビュー定義情報長（バイト） 「表 3-27　計算式 3（SDB データベースビュー定義情報長）」を参照して算出してください。
STDB	SDB データベース格納定義情報長（バイト） 「表 3-28　計算式 4（SDB データベース格納定義情報長）」を参照して算出してください。
DBS	SDB ディレクトリ情報長（SDB データベース情報長）（バイト） 「3.4.2　共用メモリの見積もり」の，変数 dbs の計算式で算出してください。
SUB	SDB ディレクトリ情報長（SDB データベースビュー情報長）（バイト） 「3.4.2　共用メモリの見積もり」の，変数 sub の計算式で算出してください。
CRTTBL	レコード型定義の排他資源数（個） マニュアル「HiRDB システム定義」の「排他資源数の見積もり」の「定義系 SQL」の「CREATE TABLE」を参照して算出してください。※
CRTIDX	インデクス定義の排他資源数（個） マニュアル「HiRDB システム定義」の「排他資源数の見積もり」の「定義系 SQL」の「CREATE INDEX（プラグインインデクスでない場合）」を参照して算出してください。※
DRPTBL	レコード型定義削除の排他資源数（個） マニュアル「HiRDB システム定義」の「排他資源数の見積もり」の「定義系 SQL」の「DROP TABLE」を参照して算出してください。※
ALTRDA	格納 RD エリア変更の排他資源数 マニュアル「HiRDB システム定義」の「排他資源数の見積もり」の「定義系 SQL」の「ALTER TABLE」を参照して，排他資源数を算出してください。※ その際，次のことに留意してください。 • 「SDB データベース格納定義の WITHIN オペランドに指定した格納 RD エリアの変更」は，「ALTER TABLE 文による RD エリアの変更」に対応しています。 • 「CHANGE RDAREA 横分割表の場合に加算します。」の部分の計算式を加算してください。

変数	変数の意味
PRGTBL	格納条件が変更となる RD エリアの初期化に必要となる排他資源数 マニュアル「HiRDB システム定義」の「排他資源数の見積もり」の「操作系 SQL」の「PURGE TABLE」を参照して，排他資源数を算出してください。※ その際，次のことに留意してください。 •「SDB データベース格納定義の WITHIN オペランドの指定による格納条件の変更をした RD エリア」は，「PURGE TABLE 文の対象 RD エリア」に対応しています。
RECN	SDB データベース内のレコード型数（個）
IDXN	SDB データベース格納定義内のインデクス数（個）
ADRECN	SDB データベースの定義変更で追加したレコード型数（個）
ADIDXN	SDB データベースの定義変更で追加したインデクス数（個）
ALRECN	SDB データベース格納定義で格納 RD エリアを変更したレコード型数

注※

マニュアルを参照する場合，次の用語を読み替えてください。

- 「表」を「レコード型」に読み替えてください。
- 「列」を「構成要素」に読み替えてください。なお，データ属性を持たない構成要素は，構成要素数を数える際の対象にはしないでください。また，SDB データベース定義で OCCURS 句を指定する場合，システム内で生成される構成要素は，構成要素数を数える際の対象にしてください。

3.8.4 HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時の排他資源数

HiRDB/SD データベース作成ユティリティ（pdsdblod）の実行時には，次の計算式で算出したサーバ当たりの排他資源数が必要になります。

・トランザクションを超えた排他制御で使用する資源数

```
X＝（a＋b）×2＋2＋c×d
```

サーバ共通定義の pd_lck_until_disconnect_cnt オペランド値は，X 以上必要です。

変数の説明

- a：格納レコード用 RD エリアの総数
- b：インデクス格納用 RD エリアの総数（シーケンシャルインデクス格納用，二次インデクス格納用）
- c：インデクスの総数（シーケンシャルインデクス，二次インデクス）
- d：レコードの横分割格納 RD エリア数

・トランザクション内の排他制御で使用する資源数

```
Y＝a×2＋1＋ｂ＋ｃ
```

サーバ共通定義の pd_lck_pool_size オペランド値は，↑Y÷4↑キロバイト以上必要です。

変数の説明

- a：使用中セグメント数（すでにデータが格納されているセグメント数）
  environment 文の purge オペランドに yes を指定したときに加算してください。
- b：レコード型を格納する RD エリアの使用セグメント数
- c：インデクスを格納する RD エリアの使用セグメント数
  使用セグメント数は，データを格納する予定のセグメント数としてください。ただし，-d オプションの指定時，すでにデータが格納されているセグメント数（使用中セグメント数）の方が，データを格納する予定のセグメント数より多い場合は，使用中セグメント数で計算してください。

HiRDB/SD データベース作成ユティリティ（pdsdblod）の前処理では，（209 ＋ a）の資源を消費します。そのため，Y は最低でも（209 ＋ a）以上必要です。HiRDB/SD データベース作成ユティリティ（pdsdblod）の前処理で確保した資源は，データロードを開始する前に解放されるため，計算した Y の値が（209 ＋ a）以上の場合は，その値を使用してください。

3.8.5 HiRDB/SD データベース再編成ユティリティ（pdsdbrog）実行時の排他資源数

HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の実行時には，次の計算式で算出したサーバ当たりの排他資源数が必要になります。

・トランザクション内の排他制御で使用する資源数

```
Y＝2×（a＋b）＋1
```

サーバ共通定義，バックエンドサーバ定義の pd_lck_pool_size オペランド値は，↑Y÷4↑キロバイト以上必要です。

変数の説明

- a：1（アンロード対象のデータが格納されている RD エリアの数）
- b：アンロード対象のインデクスが格納されている RD エリアの数

HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の前処理では，（209 ＋ a）の資源を消費します。そのため，Y は最低でも（209 ＋ a）以上必要です。HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の前処理で確保した資源は，アンロードを開始する前に解放されるため，計算した Y の値が（209 ＋ a）以上の場合は，その値を使用してください。

3.8.6　HiRDB/SD データベースアクセスユティリティ（pdsdbexe）実行時の排他資源数

HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の DML コマンドで SDB データベースを操作する際に必要な排他資源数を求める必要があります。DML コマンドで SDB データベースを操作する際に必要な排他資源数は，SDB データベースを操作する API 実行時の排他資源数と同じになります。詳細については，「3.8.2　SDB データベースを操作する API または DML の実行時の排他資源数」を参照してください。

4 HiRDB システムの構築

この章では，HiRDB/SD を使用する際の HiRDB システムの構築について説明します。

4.1　システム構築手順

ここでは，HiRDB を新規導入するときのシステム構築手順，およびほかの製品と連携する場合の環境設定について説明します。

4.1.1　HiRDB を新規導入するときのシステム構築手順

HiRDB を新規導入するときのシステム構築手順を次に示します。

1. HiRDB をインストールします。
   HiRDB のインストール手順については，「4.3　HiRDB のインストール」を参照してください。
2. HiRDB の環境設定をします。
   HiRDB の環境設定の手順については，「4.4　環境設定」を参照してください。
3. SDB データベースを作成します。
   SDB データベースの作成手順については，「4.5　SDB データベースの作成」を参照してください。

4.1.2　ほかの製品と連携する場合の環境設定

ほかの製品と連携する場合の環境設定を次に示します。

(1)　系切り替え機能を使用する場合

系切り替え機能を使用する場合は，クラスタソフトウェアが必要になります。HiRDB/SD では，クラスタソフトウェアとして HA モニタだけが使用できます。系切り替え機能の環境設定方法については，次の個所を参照してください。

高速系切り替え機能を使用する場合

「6.1　高速系切り替え機能の環境設定の流れ」

(2)　インナレプリカ機能を使用する場合

インナレプリカ機能を使用するには，HiRDB Staticizer Option が必要になります。環境設定方法については，「4.6　インナレプリカ機能を使用する場合の環境設定【4V FMB，4V AFM】」を参照してください。

4.2　HiRDB のディレクトリおよびファイル構成

HiRDB のディレクトリおよびファイル構成について説明します。

4.2.1　最初に作成するファイル

(1)　HiRDB 管理者が作成するディレクトリおよびファイル

HiRDB 管理者が作成するディレクトリおよびファイル構成を次の表に示します。

表 4–1　HiRDB 管理者が作成するディレクトリおよびファイル構成

ファイルまたはディレクトリ名	説明
$PDDIR/conf/pdsys	システム共通定義を格納するファイル
$PDDIR/conf/pdutsys	ユニット制御情報定義を格納するファイル
$PDDIR/conf/pdsvrc	サーバ共通定義を格納するファイル
$PDDIR/conf/サーバ名	各サーバ定義を格納するファイル
$PDDIR/conf/pduapenv	UAP 環境定義を格納するディレクトリ
$PDCONFPATH/pdsdbuapenv	SDB 用 UAP 環境定義を格納するディレクトリ

(2)　HiRDB が作成するディレクトリ

HiRDB が作成するディレクトリの構成を次の表に示します。

表 4–2　HiRDB が作成するディレクトリの構成

ディレクトリ名	説明
$PDDIR/bin	HiRDB の運用コマンドおよびユティリティを格納するディレクトリ
$PDDIR/lib	HiRDB の共用ライブラリ，およびメッセージテキストファイルを格納するディレクトリ
$PDDIR/lib/sysconf	HiRDB システム定義の定義解析用ファイルを格納するディレクトリ
$PDDIR/lib/servers	HiRDB サーバの実行ファイルおよびライブラリを格納するディレクトリ

上記の表以外のディレクトリおよびファイルについては，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB が作成するディレクトリおよびファイル」を参照してください。

4.2.2　単調増加ファイル

単調増加ファイルについては，「付録 F　単調増加ファイル」，およびマニュアル「HiRDB システム導入・設計ガイド」の「単調増加ファイル」を参照してください。

4.3　HiRDB のインストール

ここでは，HiRDB のインストールおよびアンインストールについて説明します。

HiRDB をインストールするすべてのサーバマシンで，同じバージョンのプラットフォームを使用してください。

4.3.1　インストール前の準備

インストール前の準備については，マニュアル「HiRDB システム導入・設計ガイド」の「インストール前の作業」を参照してください。

オペレーティングシステムパラメタの見積もりについては，「3.7　OS のオペレーティングシステムパラメタの見積もり」をあわせて参照してください。

4.3.2　HiRDB のインストール手順

ここでは，HiRDB のインストール手順について説明します。

(1)　HiRDB のインストール

HiRDB のインストールについては，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB のインストール」を参照してください。

HiRDB/SD をインストールする場合の注意事項

- HiRDB Structured Data Access Facility をインストールする場合，セットアップ済みの HiRDB Server が存在するときは，HiRDB Server をアンインストールしてから，HiRDB Structured Data Access Facility をインストールしてください。上書きインストールはできません。
- HiRDB Server をインストールする場合，セットアップ済みの HiRDB Structured Data Access Facility が存在するときは，HiRDB Structured Data Access Facility をアンインストールしてください。上書きインストールはできません。
- HiRDB Structured Data Access Facility は HiRDB Server のオプション製品です。ただし，構造型 DB 機能を使用する場合は，HiRDB Structured Data Access Facility だけをインストールしてください。既存のオプション製品と違い，pdopsetup コマンド（付加 PP のセットアップ）ではセットアップできません。
- HiRDB Server と HiRDB Structured Data Access Facility が混在するような複数ユニット構成にはできません。

(2) 付加 PP のインストール

実行者　スーパユーザ

HiRDB/SD に関連する付加 PP を使用する場合は，付加 PP をインストールします。付加 PP の機能と，付加 PP をインストールするサーバマシンを次の表に示します。

表 4-3　付加 PP の機能とインストール先のサーバマシン

製品名	付加 PP を導入すると使用できる機能	インストール先のサーバマシン
HiRDB Staticizer Option	インナレプリカ機能を使用できるようになります。 この付加 PP をインストールする場合は，pdopsetup コマンド（-k オプションに sti を指定）を実行してください。	すべてのサーバマシン
HiRDB Advanced High Availability	使用できる機能については，マニュアル「HiRDB 解説」の「HiRDB Advanced High Availability」を参照してください。ただし，HiRDB/SD ではスタンバイレス型系切り替え機能は使用できません。 この付加 PP をインストールする場合は，pdopsetup コマンド（-k オプションに aha を指定）を実行してください。	
HiRDB Structured Data Access Facility Extension for XDM/SD type	SD FMB の SDB データベースに二次インデクスを定義できるようになります。二次インデクスについては，「2.3.11　二次インデクス【SD FMB】」を参照してください。 この付加 PP をインストールする場合は，pdopsetup コマンド（-k オプションに esd を指定）を実行してください。	

4.3.3　インストール後の作業

HiRDB をインストールしたあとの作業については，マニュアル「HiRDB システム導入・設計ガイド」の「インストール後の作業」を参照してください。

ただし，HiRDB/SD をインストールしたあとの作業には，マニュアル「HiRDB システム導入・設計ガイド」に記載されている内容と差異があります。ここでは，その差異についてだけ説明します。

なお，マニュアル「HiRDB システム導入・設計ガイド」を参照する際，「表やインデクスを定義する」は「SDB データベースを定義する」に読み替えてください。

(1) ワークファイル出力先ディレクトリの作成

実行者　HiRDB 管理者

HiRDB/SD が出力するワークファイルの出力先となるディレクトリを作成してください。

(a) ワークファイル出力先ディレクトリの容量の見積もり

ワークファイル出力中に HiRDB または運用コマンドが異常終了した場合，ワークファイルは削除されません。そのため，pdcspool コマンドを実行する前にディスク容量が不足しないように，ワークファイル出力先ディレクトリの空き領域には十分に余裕のある値を設定してください。

ワークファイル出力先ディレクトリ下の空き領域のサイズについては，マニュアル「HiRDB システム導入・設計ガイド」の「ワークファイル出力先ディレクトリの容量の見積もり」に記載されている計算式に，次の値を加算してください。

- pdsdbdef コマンド実行時の実行結果ファイルの容量
  SDB 定義文ファイルサイズ×1.2
- pdsdblod コマンド実行時の次のファイルの容量
  - 実行結果ファイル
  - ワークファイル
  - インデクス情報ファイル
  - ソート用ワークファイル

  「3.6.1(1) HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量」を参照してください。
- pdsdbrog コマンド実行時の実行結果ファイルの容量
  「3.6.1(2) HiRDB/SD データベース再編成ユティリティ（pdsdbrog）実行時のファイルの容量」を参照してください。

(2) 文字コードの指定

HiRDB/SD で使用する文字コードを指定します。

(a) 文字コードの指定（サーバ側の指定）

使用する文字コードを pdsetup コマンドの -c オプションで指定します。使用できる文字コードを次に示します。

- 4V FMB または 4V AFM の SDB データベースの場合
  シフト JIS 漢字コードまたは Unicode（UTF-8）のどちらかを使用できます。
- SD FMB の SDB データベースの場合
  シフト JIS 漢字コードを使用できます。

pdadmvr -c コマンドで使用している文字コードを確認できます。

(b) 文字コードの指定（クライアント側の指定）

クライアント側では，サーバの文字コードに応じてクライアント環境定義の LANG または PDCLTLANG オペランドに文字コードを指定します。クライアント環境定義は，UAP の作成または実行時に参照されます。サーバの文字コードに応じて指定できるクライアント環境定義については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

なお，SDB データベースを操作する API または DML を発行する場合は，HiRDB/SD で使用する文字コードと同じ文字コードを指定してください。

(3) SDB ディレクトリ情報ファイルの作成

ユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下に SDB ディレクトリ情報ファイルを作成してください。

高速系切り替え機能を使用している場合は，待機系ユニットにも SDB ディレクトリ情報ファイルを作成してください。

(4) リモートシェル実行環境の設定

HiRDB/SD 定義ユティリティ（pdsdbdef）を実行する場合は，ユティリティを実行するマシン自体を対象にリモートシェルの実行を許可する設定が必要です。

例：

- システム共通定義の pd_cmd_rmode オペランドに rsh を指定する場合は，/etc/hosts.equiv または $HOME/.rhosts に自サーバマシンのホスト名を設定してください。
- システム共通定義の pd_cmd_rmode オペランドに ssh を指定する場合は，自サーバマシンの公開鍵を自サーバマシンに登録してください。

4.3.4 HiRDB のアンインストール

HiRDB のアンインストールについては，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB のアンインストール」を参照してください。

実行者　スーパユーザおよび HiRDB 管理者

HiRDB Structured Data Access Facility のアンインストールは，次の場合だけ実施してください。このほかの理由でのアンインストールは推奨しません。

- 今後このサーバマシンで HiRDB Structured Data Access Facility を使用しない場合
- HiRDB Structured Data Access Facility から HiRDB Server に入れ替える場合

4.3.5　HiRDB のバージョンアップ

HiRDB の過去のバージョンから HiRDB Structured Data Access Facility/Parallel Server へのバージョンアップができます。

HiRDB のバージョンアップについては，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB のバージョンアップ」を参照してください。

注意事項

バージョンアップ，または修正版との入れ替え直後に KFPU00244-E メッセージが出力され，システム共通定義 pd_structured_shmpool_dicsize オペランドが指定されている行がエラーとして表示された場合は，HiRDB Structured Data Access Facility/Parallel Server ではなく HiRDB/Parallel Server をインストールしている可能性があります。インストール媒体を確認し，誤っていた場合は再度インストールからし直してください。

4.3.6　修正版 HiRDB Structured Data Access Facility への入れ替え

修正版 HiRDB Structured Data Access Facility への入れ替えができます。

修正版 HiRDB Structured Data Access Facility への入れ替えについては，マニュアル「HiRDB システム導入・設計ガイド」の「修正版 HiRDB への入れ替え」を参照してください。

注意事項

- HiRDB Server から HiRDB Structured Data Access Facility の修正版への入れ替えはできません。
- HiRDB Structured Data Access Facility から HiRDB Server の修正版への入れ替えはできません。

4.4　環境設定

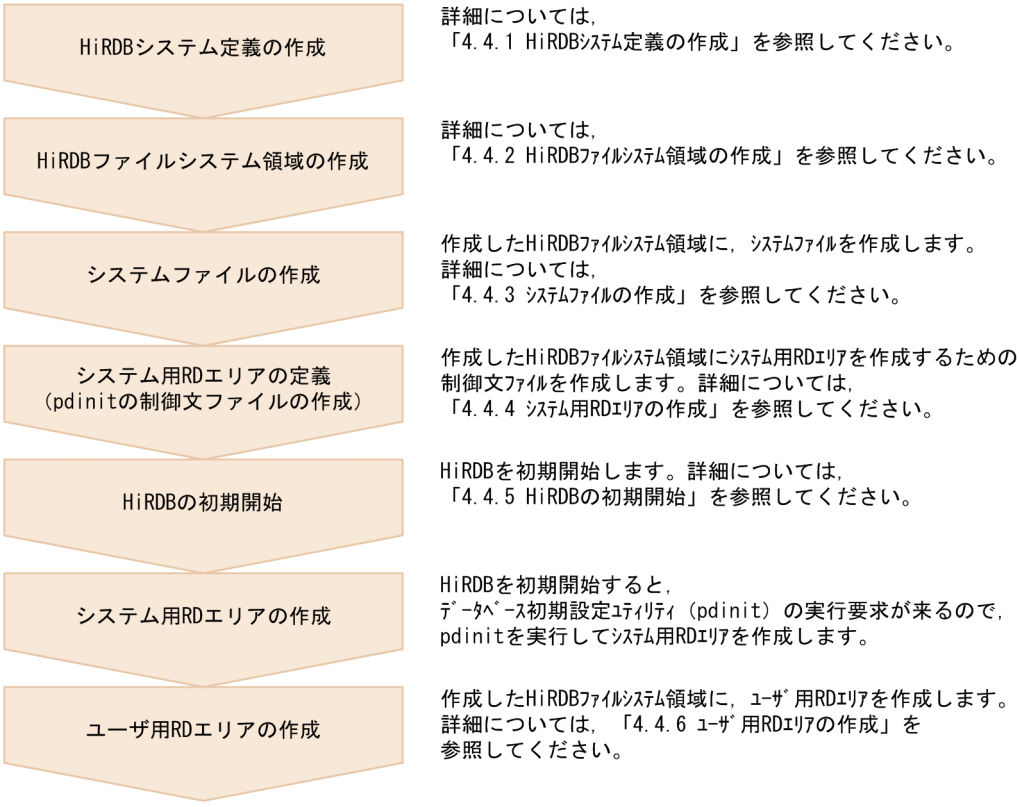
ここでは，構造型 DB 機能を使用する HiRDB の環境を設定する方法について説明します。

「3.1　HiRDB/SD の設計」で設計した内容に従って，構造型 DB 機能を使用する HiRDB の環境を設定してください。

環境設定手順

構造型 DB 機能を使用する HiRDB の環境設定手順を次の図に示します。

図 4-1　コマンドによる構造型 DB 機能を使用する HiRDB の環境設定手順



以降の説明で設定する内容を次に示します。

設定する内容

- データベース初期設定ユティリティ（pdinit）でシステム用 RD エリア（マスタディレクトリ用 RD エリア，データディレクトリ用 RD エリアおよびデータディクショナリ用 RD エリア）を作成して，HiRDB をまず開始できるようにします。
- その後，データベース構成変更ユティリティ（pdmod）で，必要な RD エリア（ユーザ用 RD エリア）を追加します。

なお，ユーザ用 RD エリアは，データベース初期設定ユティリティ（pdinit）でシステム用 RD エリアと一緒に作成することもできます。

4.4.1　HiRDB システム定義の作成

HiRDB のシステム定義の作成については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB システム定義の作成」を参照してください。SDB 用 UAP 環境定義を作成する際は，「4.9　SDB 用 UAP 環境定義ファイルの準備【SD FMB】」を参照してください。

なお，HiRDB システム定義のうち，HiRDB/SD を使用する場合に指定値の見直しが必要なオペランドについては，「9.4　指定値の見直しが必要なオペランド」を参照してください。

また，pdconfchk コマンドでチェックできるオペランドについては，「9.6　pdconfchk コマンドでチェックできるオペランド」もあわせて参照してください。

4.4.2　HiRDB ファイルシステム領域の作成

HiRDB ファイルシステム領域の作成については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB ファイルシステム領域の作成」を参照してください。

4.4.3　システムファイルの作成

システムファイルの作成については，マニュアル「HiRDB システム導入・設計ガイド」の「システムファイルの作成」を参照してください。

システムファイルの設計方法については，「3.1.3　システムファイルの設計」を参照してください。

4.4.4　システム用 RD エリアの作成

システム用 RD エリアの作成については，マニュアル「HiRDB システム導入・設計ガイド」の「システム用 RD エリアの作成」を参照してください。

4.4.5　HiRDB の初期開始

HiRDB の初期開始については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB の初期開始」を参照してください。

4.4.6　ユーザ用 RD エリアの作成

ユーザ用 RD エリアの作成については，マニュアル「HiRDB システム導入・設計ガイド」の「ユーザ用 RD エリアの作成」を参照してください。ただし，HiRDB/SD では「表」は「レコード」に読み替えてください。

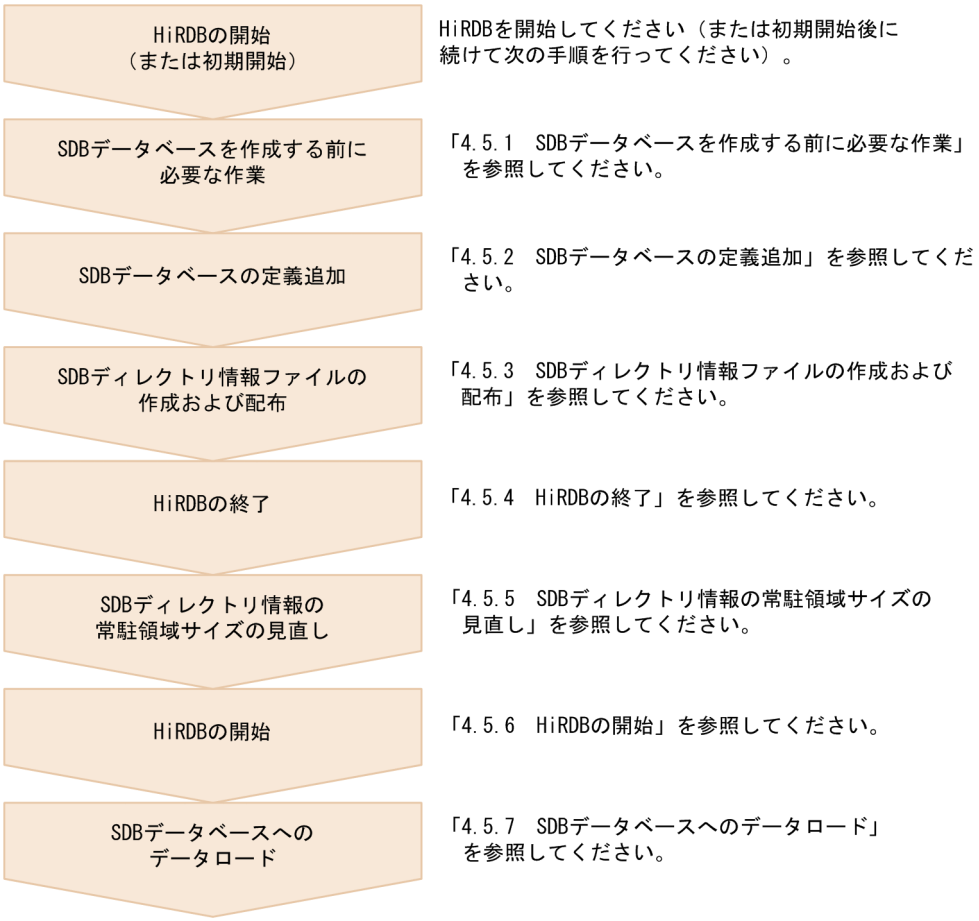
また，HiRDB/SD では次の点に留意してください。

1. ユーザ用 RD エリアに SDB データベースを格納する場合は，create rdarea 文の data model オペランドに structured を指定します。なお，for オペランドには user used by PUBLIC だけを指定できます。
2. ユーザ用 RD エリアに次の表またはインデクスを格納する場合は，create rdarea 文の data model オペランドに relational を指定するか，または data model オペランドの指定を省略してください。
   - 追い付き反映キー対応表
   - 追い付き反映キー対応表のインデクス
   - 追い付き状態管理表
   - 監査証跡表
3. ユーザ用 RD エリア作成時の考慮点については，「3.2.1　ユーザ用 RD エリアを設計する際の考慮点」を参照してください。

4.5　SDB データベースの作成

SDB データベースの作成手順を次の図に示します。

図 4-2　SDB データベースの作成手順



4.5.1　SDB データベースを作成する前に必要な作業

実行者　HiRDB 管理者

SDB データベースを作成する前に必要な作業について説明します。定義系 SQL の実行が必要な場合は，データベース定義ユティリティ（pddef）または HiRDB SQL Executer を使用してください。

(1)　クライアント環境定義の設定

次に示すクライアント環境定義を設定してください。クライアント環境定義の設定方法については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。

- PDHOST
- PDUSER

- PDNAMEPORT

(2) パスワードの変更

HiRDB 管理者用の認可識別子のパスワードが，認可識別子と同じ文字列になっている場合，定義系 SQL の GRANT 文でパスワードを変更してください。GRANT 文の実行例を次に示します。

```
GRANT DBA TO HiRDB管理者用の認可識別子 IDENTIFIED BY 新しいパスワード;
```

(3) HiRDB 管理者以外が SDB データベースを作成する場合

HiRDB 管理者以外が SDB データベースを作成する場合，定義系 SQL の GRANT 文で，SDB データベースを作成するユーザに次のユーザ権限を与える必要があります。

- CONNECT 権限
- スキーマ定義権限

ユーザ権限の種類については「2.15.1　ユーザ権限の種類」を参照してください。

SDB データベースを作成するユーザに，CONNECT 権限，およびスキーマ定義権限を与える場合の実行例を次に示します。

```
GRANT CONNECT TO 権限を与えるユーザの認可識別子 IDENTIFIED BY パスワード;
GRANT SCHEMA TO 権限を与えるユーザの認可識別子;
```

(4) スキーマの定義

定義系 SQL の CREATE SCHEMA でスキーマを定義します。

1 ユーザに対して 1 個のスキーマを定義します。

(5) データの変換方式の設定

メインフレーム上のデータを HiRDB/SD へ移行する際，メインフレームと HiRDB/SD では符号付きパック形式の符号部に仕様差異があるため，メインフレーム上の UAP をそのまま移行できない場合があります。この場合，データの形式を変換する機能（DECIMAL 型の符号正規化機能）を使用してください。このとき，符号部の変換規則をよく確認してください。

DECIMAL 型の符号正規化機能を使用するには，システム共通定義の pd_dec_sign_normalize オペランドに Y を指定します。詳細については，マニュアル「HiRDB システム運用ガイド」の「DECIMAL 型の符号部を変換する方法」を参照してください。

4.5.2　SDB データベースの定義追加

実行者　CONNECT 権限およびスキーマ定義権限保持者

SDB データベースの定義追加の手順を次に示します。

(1)　SDB データベースのデータベース定義を作成する

SDB データベースのデータベース定義を作成します。SDB データベースのデータベース定義には，次の二種類があります。

- SDB データベース定義

  SDB データベースの論理構造（データ操作の対象となるレコード型やレコード間の親子集合など）を定義します。SDB データベース定義については，次の個所を参照してください。

  - 4V FMB の SDB データベースを定義する場合

    「11.5.1　SDB データベース定義【4V FMB】」
  - 4V AFM の SDB データベースを定義する場合

    「11.6.1　SDB データベース定義【4V AFM】」
  - SD FMB の SDB データベースを定義する場合

    「11.7.1　SDB データベース定義【SD FMB】」
- SDB データベース格納定義

  SDB データベースの格納構造（インデクスの構成要素や親子集合を実現するためのポインタなど）を定義します。SDB データベース格納定義については，次の個所を参照してください。

  - 4V FMB の SDB データベースを定義する場合

    「11.5.2　SDB データベース格納定義【4V FMB】」
  - 4V AFM の SDB データベースを定義する場合

    「11.6.2　SDB データベース格納定義【4V AFM】」
  - SD FMB の SDB データベースを定義する場合

    「11.7.2　SDB データベース格納定義【SD FMB】」

SDB データベース定義，SDB データベース格納定義は，SDB 定義文ファイルに記述します。作成した SDB 定義文ファイルを HiRDB/SD 定義ユティリティ（pdsdbdef）の source 文に指定します。

(2)　SDB ディクショナリ情報および SDB ディレクトリ情報を追加する

HiRDB/SD 定義ユティリティ（pdsdbdef）で*ENTRY DICTIONARY 文および*ENTRY DIRECTORY 文を実行し，定義追加する SDB データベースの SDB ディクショナリ情報および SDB ディレクトリ情報をディクショナリ表に追加します。(1)で作成した SDB データベース定義および SDB データベース格納定義を*ENTRY DICTIONARY 文に指定します。

*ENTRY DICTIONARY 文については「11.4.2　*ENTRY DICTIONARY 文（SDB ディクショナリ情報の追加）」を参照してください。また，*ENTRY DIRECTORY 文については「11.4.6　*ENTRY DIRECTORY 文（SDB ディレクトリ情報の追加）」を参照してください。

また，定義追加した SDB データベースのレコード型およびインデクスの定義情報が，次の RD エリアに登録されます。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- ユーザ用 RD エリア

注意事項

*ENTRY DICTIONARY 文および*ENTRY DIRECTORY 文は，1 回の pdsdbdef コマンドで実行してください。

理由：

*ENTRY DIRECTORY 文の実行時には，データベース定義のチェックが実行されます。

*ENTRY DICTIONARY 文および*ENTRY DIRECTORY 文を 1 回の pdsdbdef コマンドで実行した場合，データベース定義の不整合が検知されると，SDB ディクショナリ情報および SDB ディレクトリ情報はディクショナリ表に追加されません。この場合，データベース定義を修正して，pdsdbdef コマンドを再実行することで対処できます。

*ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文を別々に実行した場合，*ENTRY DIRECTORY 文の実行時には，すでに SDB ディクショナリ情報はディクショナリ表に追加されています。そのため，データベース定義の不整合が検知された場合に，データベース定義を修正するには，*DELETE DICTIONARY 文を実行して SDB ディクショナリ情報を削除してから，pdsdbdef コマンドを再実行することになります。

4.5.3　SDB ディレクトリ情報ファイルの作成および配布

実行者　HiRDB 管理者

HiRDB/SD 定義ユティリティ（pdsdbdef）を実行して，「4.5.2(2)　SDB ディクショナリ情報および SDB ディレクトリ情報を追加する」で作成した SDB ディレクトリ情報を SDB ディレクトリ情報ファイルに出力して，全ユニットに配布します。

HiRDB/SD 定義ユティリティ（pdsdbdef）の dirinf 文で，SDB ディレクトリ情報ファイルの出力先を指定します。dirinf 文については，「11.3.5　dirinf 文（SDB ディレクトリ情報ファイルの作成および配布）」を参照してください。

SDB ディレクトリ情報ファイルのファイル名は，pdsdbdir 固定となります。

■SDB ディレクトリ情報ファイルの配布

HiRDB/SD 定義ユティリティ（pdsdbdef）を実行すると，SDB ディレクトリ情報ファイルの作成と同時に，全ユニットに SDB ディレクトリ情報ファイルが配布されます。配布先は，ユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下になります。

高速系切り替え機能を使用している場合は，待機系ユニットにも SDB ディレクトリ情報ファイルが配布されます。

注意事項

ディザスタリカバリシステムのリモートサイトは配布対象外です。リモートサイトに SDB ディレクトリ情報ファイルを配布する方法については，「8.3.1(2) 「6. SDB ディレクトリ情報ファイルをリモートサイトに転送する」の作業」を参照してください。

なお，1 つのサーバマシン内に複数のユニットがある場合，SDB ディレクトリ情報ファイルはサーバマシンに 1 つだけ配布し，各ユニットでその SDB ディレクトリ情報ファイルを共有することを推奨します。

SDB ディレクトリ情報ファイルをユニットごとに用意することもできますが，SDB ディレクトリ情報ファイルを配布したり，SDB ディレクトリ情報の整合性をチェックしたりする際の運用負荷は増えます。

4.5.4 HiRDB の終了

実行者　HiRDB 管理者

pdstop コマンドで HiRDB を正常終了します。必ず正常終了してください。

4.5.5 SDB ディレクトリ情報の常駐領域サイズの見直し

実行者　HiRDB 管理者

次に示す場合は，システム共通定義の pd_structured_shmpool_dicsize オペランド（SDB ディレクトリ情報の常駐領域サイズ）の指定値を見直します。

- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値を超えている場合
- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値に比べてかなり小さい場合

SDB ディレクトリ情報の合計長は，「4.5.3 SDB ディレクトリ情報ファイルの作成および配布」で pdsdbdef コマンドを実行した際に出力される KFPB61231-I メッセージ（size に表示されます）で確認できます。

4.5.6　HiRDB の開始

実行者　HiRDB 管理者

pdstart コマンドで HiRDB を正常開始します。HiRDB の正常開始時，配布した SDB ディレクトリ情報ファイルが読み込まれ，共用メモリ上の常用常駐領域に SDB ディレクトリ情報が常駐化されます。

HiRDB を開始したあとに，pdsdbarc -c コマンドを実行して，SDB ディレクトリ情報の最終更新日時のチェックを行い，全ユニットの SDB ディレクトリ情報が一致していることを確認してください。

4.5.7　SDB データベースへのデータロード

実行者　CONNECT 権限保持者

HiRDB/SD データベース作成ユティリティ（pdsdblod）を使用して，SDB データベースにデータをロードします。

データロードには次の 2 種類の方法があります。

- ユーザが用意した任意の入力データファイルを使用する方法
  入力データファイル中のデータをレコードの形式に変換して，データベースに格納します。
- フォーマットライト機能を使用する方法
  4V DAM の SDB データベースの場合，SDB データベース定義でフォーマットライト機能を使用する指定をしていれば，フォーマットライト機能を使用できます。フォーマットライト機能を使用すると，SDB データベース定義，および SDB データベース格納定義に従って，HiRDB/SD がデータベースにデータを格納します。

どちらの場合でもデータロード時にインデクスが作成されます。

詳細については，「12.　HiRDB/SD データベース作成ユティリティ（pdsdblod）」を参照してください。

4.6　インナレプリカ機能を使用する場合の環境設定【4V FMB，4V AFM】

インナレプリカ機能を使用すると，ノンストップサービスに対応したデータベースシステムを構築できます。インナレプリカ機能を使用するには，HiRDB Staticizer Option が必要になります。

インナレプリカ機能は，4V FMB または 4V AFM の SDB データベースの場合に使用できます。

インナレプリカ機能を使用した更新可能なオンライン再編成を実行できるのは，4V FMB の SDB データベースに限ります。

4.6.1　HiRDB システム定義の指定

インナレプリカ機能を使用する場合の HiRDB システム定義の指定については，マニュアル「HiRDB システム導入・設計ガイド」の「インナレプリカ機能との連携」を参照してください。

4.6.2　環境設定方法

インナレプリカ機能を使用する場合のシステムの環境設定方法については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」を参照してください。

4.6.3　レプリカ RD エリア作成時の留意事項

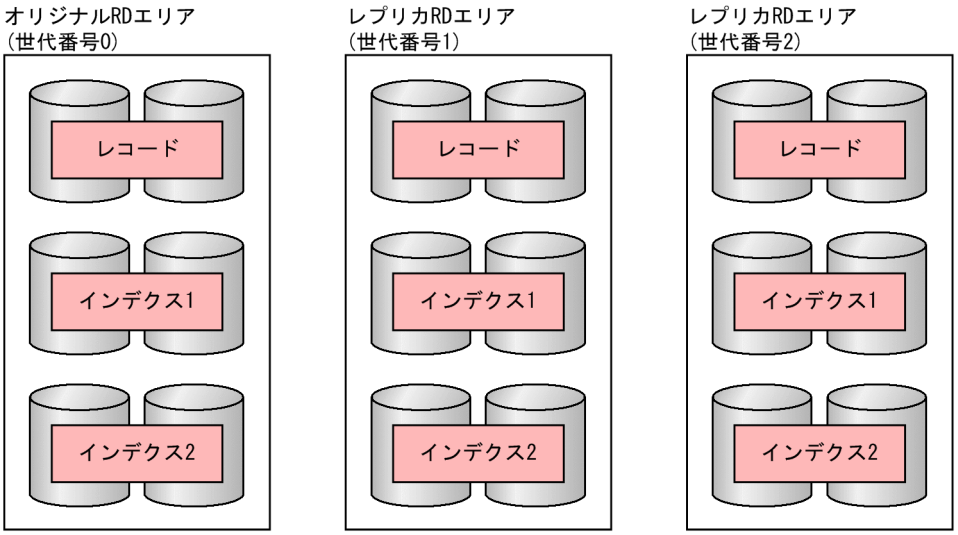
ここでは，レプリカ RD エリア作成時の留意事項について説明します。

(1)　オリジナル RD エリアおよびレプリカ RD エリアの構成

レプリカ RD エリアは，オリジナル RD エリアと同じ構成にしてください。例えば，オリジナル RD エリアでレコードおよびインデクスをそれぞれ 2 つの RD エリアに横分割している場合は，レプリカ RD エリアでもレコードおよびインデクスをそれぞれ 2 つの RD エリアに横分割してください。

オリジナル RD エリアおよびレプリカ RD エリアの正しい構成例を次の図に示します。

図 4-3　オリジナル RD エリアおよびレプリカ RD エリアの正しい構成例



[説明]

レコードおよびインデクスの横分割数を同じにして，オリジナル RD エリアとレプリカ RD エリアの構成を同じにしてください。

(2)　オリジナル RD エリアとレプリカ RD エリアの構成を確認する方法

次に示すコマンドで確認できます。

- pdrdrefls コマンド

  pdrdrefls コマンドの-t オプションにレコード型名を指定すると，そのレコード型に定義されているインデクスの格納 RD エリアの情報が表示されます。

- pddbls コマンド

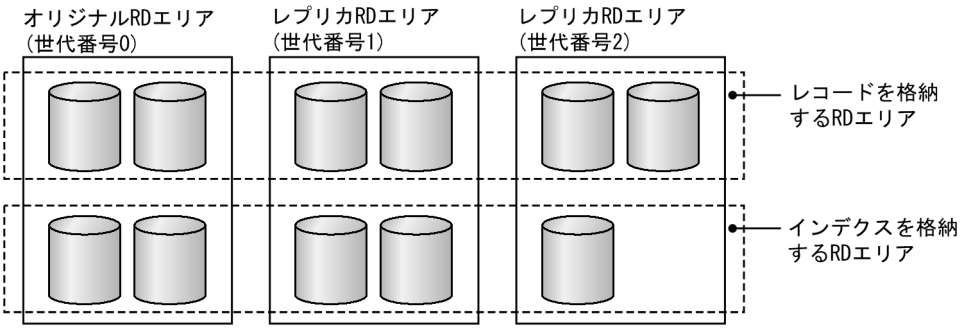
  pddbls コマンドの-q オプションに世代番号を指定すると，その世代番号の RD エリアの情報が表示されます。

(3)　オリジナル RD エリアとレプリカ RD エリアの構成が異なる場合の対処方法

オリジナル RD エリアとレプリカ RD エリアの構成が異なる場合や，世代番号を正しく設定していない場合は，SDB データベースの定義，定義変更，または定義削除ができません（pdsdbdef コマンドが実行できません）。この場合，レプリカ RD エリアを追加するか，または削除してオリジナル RD エリアとレプリカ RD エリアの構成を同じにしてから，データベースの定義などを実行してください。

オリジナル RD エリアとレプリカ RD エリアの構成が異なる場合の対処の例を次の図に示します。

図 4-4　オリジナル RD エリアとレプリカ RD エリアの構成が異なる場合の対処の例



［説明］

上記の図の場合，世代番号 2 のレプリカ RD エリアが 1 つ不足しています。この場合，次に示すどちらかの対処をしないと，データベースの定義，定義変更，または定義削除ができません。

- 世代番号 2 のレプリカ RD エリアを 1 つ追加する
- 世代番号 2 のレプリカ RD エリアを 3 つとも削除する

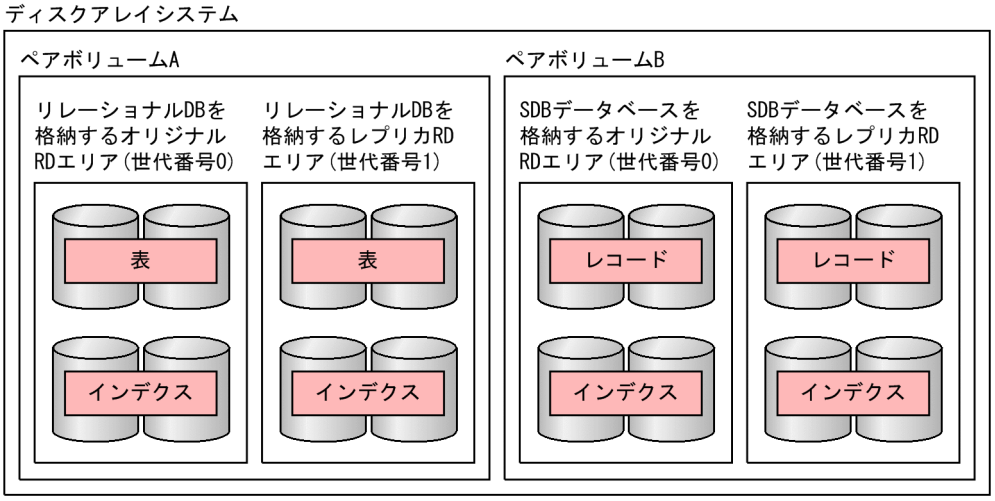
(4)　リレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアの構成【4V FMB】

更新可能なオンライン再編成では，同一 BES 内でリレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアを同じタイミングで再編成できません。pdorbegin コマンドに指定する RD エリアには，1 つの BES でリレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアを混在させることはできません。

そのため，リレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアは別々に再編成できるよう，RD エリアを構成してください。

リレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアを分ける構成の例を次の図に示します。

図 4-5　リレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアを分ける構成の例



4.7 HiRDB Server と HiRDB Structured Data Access Facility の入れ替え手順

ここでは，HiRDB Server と HiRDB Structured Data Access Facility の入れ替え手順について説明します。

なお，以降の説明では，それぞれの製品を次のように表記します。

- HiRDB Server：単体版 HiRDB
- HiRDB Structured Data Access Facility：統合版 HiRDB

4.7.1 入れ替えに関する前提条件

単体版 HiRDB から統合版 HiRDB，および統合版 HiRDB から単体版 HiRDB に入れ替える際の前提条件を次の表に示します。

表 4-4 HiRDB を入れ替える際の前提条件

項番	項目	前提条件
1	プラットフォーム	プラットフォームは次のどれかである必要があります。 • Red Hat Enterprise Linux Server 7（64-bit x86_64） • Red Hat Enterprise Linux Server 8（64-bit x86_64） • Red Hat Enterprise Linux Server 9（64-bit x86_64）
2	HiRDB システムの構成	HiRDB システムの構成が，HiRDB/パラレルサーバの場合だけ入れ替えられます。
3	マルチ HiRDB の構成	単体版 HiRDB と統合版 HiRDB が混在する構成にはできません。
4	使用できる文字コード	HiRDB サーバが使用する文字コードは次のどちらかである必要があります。 • シフト JIS 漢字コード • Unicode（UTF-8） なお，SD FMB の SDB データベースを使用する場合，HiRDB サーバで使用できる文字コードは，シフト JIS 漢字コードだけになります。

4.7.2 単体版 HiRDB から統合版 HiRDB への入れ替え手順

ここでは，単体版 HiRDB を統合版 HiRDB に入れ替える手順について説明します。

(1) 入れ替え作業をする前の考慮点

入れ替え作業をする前の考慮点を次に示します。

(a)　統合版 HiRDB で制限となる機能

単体版 HiRDB で使用できる機能のうち，統合版 HiRDB で制限となる機能を次の表に示します。

表 4-5　統合版 HiRDB で制限となる機能

項番	機能名	制限事項
1	リアルタイム SAN レプリケーション（ディザスタリカバリ）	リアルタイム SAN レプリケーション（ディザスタリカバリ）のリモートサイトへのデータ反映方式は，全非同期方式だけ使用できます。
2	系切り替え機能	系切り替え機能は，高速系切り替え機能だけ使用できます。
3	インデクスキー値の排他制御	インデクスキー値の排他制御は，インデクスキー値無排他だけ使用できます。
4	文字コード変換機能	リレーショナルデータベースの操作と SDB データベースの操作を混在して実行する場合，文字コード変換機能は使用できません。

(b)　HiRDB/SD で制限となる機能

単体版 HiRDB で使用できる機能のうち，HiRDB/SD で制限となる機能があります。詳細については，「付録 A　未サポート機能の一覧」を参照してください。

(c)　修正版 HiRDB のコピーおよび入れ替え

入れ替え前の単体版 HiRDB と修正版の統合版 HiRDB では，次のコマンドは実行できません。

- pdprgcopy コマンド（修正版 HiRDB のコピー）
- pdprgrenew コマンド（修正版 HiRDB の入れ替え）

(2)　単体版 HiRDB から統合版 HiRDB への入れ替え時の注意事項

単体版 HiRDB を統合版 HiRDB に入れ替える場合，HiRDB クライアントも入れ替える必要があります。HiRDB クライアントを入れ替える際の注意事項を次に示します。

- HiRDB Structured Data Access Facility/Run Time をインストールするマシンに，セットアップ済みの HiRDB/Run Time がある場合は，HiRDB/Run Time をアンインストールしてください。そのあとで，HiRDB Structured Data Access Facility/Run Time をインストールしてください。上書きインストールした場合の動作は保証できません。
- HiRDB Structured Data Access Facility/Developer's Kit をインストールするマシンに，セットアップ済みの HiRDB/Developer's Kit がある場合は，HiRDB/Developer's Kit をアンインストールしてください。そのあとで，HiRDB Structured Data Access Facility/Developer's Kit をインストールしてください。上書きインストールした場合の動作は保証できません。

(3)　単体版 HiRDB から統合版 HiRDB への入れ替え手順

運用中の単体版 HiRDB を統合版 HiRDB に入れ替える際に，事前に確認・設計しておくべき項目，および入れ替え手順を次に示します。

なお，単体版 HiRDB のバージョンまたはリビジョン（VV-RR-ZZ の形式で示す HiRDB のバージョン番号のうち，VV または RR）が統合版 HiRDB のバージョンまたはリビジョンより古い場合は，バージョンアップも同時に実施されます。バージョンアップの手順についてはマニュアル「HiRDB システム導入・設計ガイド」の「HiRDB のバージョンアップ」を参照してください。ただし，「HiRDB のバージョンアップ」では上書きインストールの手順が説明されていますが，単体版 HiRDB を統合版 HiRDB に入れ替えてバージョンアップする場合には，アンインストール後にインストールする手順となります。

(a)　入れ替え前に事前に確認・設計しておくべき項目

入れ替え前に事前に確認・設計しておくべき項目を次に示します。

1. 空き領域の確認

   構造型 DB 機能を使用する場合は専用のディクショナリ表を使用します。

   そのため，pddbst コマンドで，データディクショナリ用 RD エリアに必要な空き容量があるかを確認してください。

   データディクショナリ用 RD エリアの見積もりについては，「3.5.4(1)　データディクショナリ用 RD エリアの容量の見積もり」を参照してください。

   空き容量が不足している場合は，データベース構成変更ユティリティ（pdmod）でデータディクショナリ用 RD エリアを拡張してください。詳細については，マニュアル「HiRDB システム運用ガイド」の「RD エリアの容量を大きくする方法（RD エリアの拡張)」を参照してください。

2. HiRDB システム定義の確認

   統合版 HiRDB では単体版 HiRDB で使えない機能があります。そのため，問題がないかを確認してください。詳細については，「(1)(a)　統合版 HiRDB で制限となる機能」を参照してください。

3. メモリ所要量の確認

   構造型 DB 機能を使用する場合に必要なメモリが確保できることを確認してください。

   必要なメモリ所要量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB/パラレルサーバのメモリ所要量の見積もり」を参照してください。なお，メモリ所要量を見積もる際には，「3.4　メモリ所要量の見積もり」で算出した値を加算してください。

4. ファイル容量の確認

   統合版 HiRDB を使用する場合，HiRDB のステータスファイルやシンクポイントダンプファイルの容量が増加することがあります。そのため，必要な容量が確保できることを確認してください。

   必要なファイル容量の見積もりについては，「3.1.3　システムファイルの設計」，およびマニュアル「HiRDB システム導入・設計ガイド」の「ステータスファイルの容量の見積もり」，「シンクポイントダンプファイルの容量の見積もり」を参照してください。

5. OS のオペレーティングシステムパラメタの確認

   統合版 HiRDB に入れ替えることで新たに OS 資源を消費する場合は，統合版 HiRDB をインストールする前に，OS のオペレーティングシステムパラメタ（カーネルパラメタ）の値を変更してください。

オペレーティングシステムパラメタ（カーネルパラメタ）の見積もりについては，「3.7　OS のオペレーティングシステムパラメタの見積もり」，およびマニュアル「HiRDB システム導入・設計ガイド」の「Linux のオペレーティングシステムパラメタの見積もり」を参照してください。

6. システムログファイルの総レコード数の確認

構造型 DB 機能が使用する専用のディクショナリ表を定義する場合に，必要なシステムログ容量があることを確認してください。

システムログファイルの容量の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「システムファイルの設計」の「システムログファイルの設計」を参照してください。なお，システムログ量を見積もる際には，「3.5.1　システムログ量の見積もり」で算出した値を加算してください。

(b)　単体版 HiRDB から統合版 HiRDB への入れ替え手順

単体版 HiRDB から統合版 HiRDB への入れ替え手順を次に示します。

1. RD エリアのバックアップの取得

入れ替えが失敗した場合に備えて，データベース複写ユティリティ（pdcopy）などで，すべての RD エリアのバックアップを取得してください。

バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。

2. HiRDB のオンライン状態の確認

pdls コマンドで HiRDB が正常終了できる状態であることを確認します。すべてのユニットが ACTIVE と表示されているかどうかを確認してください。

3. HiRDB の正常終了

3.の手順で ACTIVE と表示されている場合，pdstop コマンドで HiRDB を正常終了させてください。

HiRDB がすでに終了している場合は，次のどちらかの情報を参照して HiRDB が正常終了しているかどうかを確認してください。

- メッセージログファイル
- syslog ファイル

正常終了していない場合は，pdstart コマンドでいったん HiRDB を開始してから，pdstop コマンドで正常終了させてください。

4. HiRDB の状態確認

pdls -d ust コマンドで HiRDB が正常終了していることを確認してください。

5. ライブラリの共用化の解除

マルチ HiRDB 環境でライブラリを共用している場合は，pdmemsv -d コマンドでライブラリの共用化を解除してください。

入れ替え完了後に，再度 pdmemsv コマンドでライブラリを共用化してください。

6. コマンド，ユティリティ，アプリケーション，および HiRDB と連携している製品の停止

HiRDB Datareplicator，HiRDB Dataextractor，JP1/PFM など，HiRDB にアクセスする連携製品を停止してください。これらを停止しないと，実行形式ファイルや共用ライブラリの削除に失敗するおそれがあります。

7. HiRDB 運用ディレクトリ下のバックアップの取得（HiRDB 運用ディレクトリ数分繰り返す）

入れ替えが失敗した場合に備えて，OS の cp コマンドなどで，次のファイルのバックアップを取得してください。

- HiRDB 運用ディレクトリ下（$PDDIR/conf 下）のファイル
- HiRDB 運用ディレクトリ下のトラブルシュート情報（$PDDIR/spool 下）のファイル

HiRDB 運用ディレクトリのバックアップの取得方法については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB 運用ディレクトリの作成」の「HiRDB 運用ディレクトリのバックアップの取得」を参照してください。

8. 単体版 HiRDB を OS から削除（HiRDB 運用ディレクトリ数分繰り返す）

単体版 HiRDB をアンインストールする前に，単体版 HiRDB の HiRDB 運用ディレクトリ下の HiRDB システムを OS から削除してください。

このとき，単体版 HiRDB の実行に必要なファイルおよびディレクトリも削除してください（pdsetup -d コマンドを実行し，KFPS00036-Q メッセージに y を応答してください）。

pdsetup -d コマンドの終了ステータスが 0 でない場合，障害が起きているおそれがあります。syslog ファイルに出力されたメッセージを参照して，障害原因を取り除いてから，OS からの削除を再度実行してください。

9. 単体版 HiRDB のアンインストール

日立 PP インストーラを使用してサーバマシンごとに単体版 HiRDB をアンインストールしてください。

アンインストールの結果は，日立 PP インストーラの画面で確認してください。アンインストールが成功していない場合は，syslog ファイルに出力されたメッセージを参照して，障害原因を取り除いてから，アンインストールを再度実行してください。

10. 統合版 HiRDB のインストール

日立 PP インストーラを使用して，サーバマシンごとに統合版 HiRDB をインストールしてください。

11. HiRDB システム定義の変更（HiRDB 運用ディレクトリ数分繰り返す）

構造型 DB 機能を使用するために必要なオペランドを，HiRDB 運用ディレクトリ下（$PDDIR/conf 下）の HiRDB のシステム定義ファイルに定義してください。

構造型 DB 機能を使用するために必要なオペランドについては，「9. システム定義」を参照してください。

12. 統合版 HiRDB を OS に登録（HiRDB 運用ディレクトリ数分繰り返す）

統合版 HiRDB をインストールしたあとに，pdsetup コマンドで，統合版 HiRDB の HiRDB 運用ディレクトリ下の HiRDB システムを OS に登録してください。

pdsetup コマンドの-c オプションには，必ず次の条件を満たす文字コードを指定してください。

- 単体版 HiRDB で指定していた文字コード
- 統合版 HiRDB で指定できる文字コード

指定できる文字コードについては，「4.7.1　入れ替えに関する前提条件」を参照してください。

13. HiRDB の再起動

pdstart コマンドで HiRDB を再起動してください。HiRDB の再起動が完了すると，構造型 DB 用の RD エリアの定義や，SDB データベースの定義ができる状態になります。

14. RD エリアのバックアップの取得

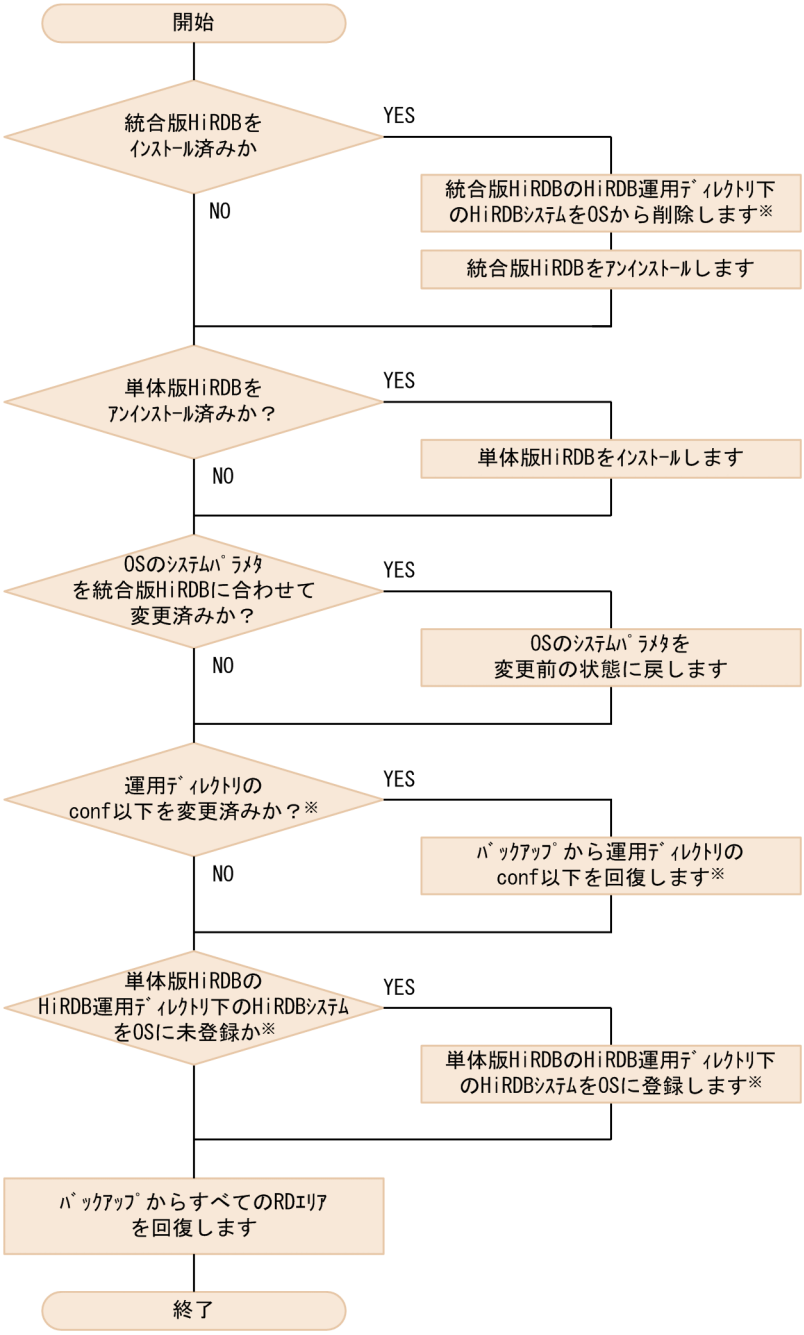
構造型 DB 用の RD エリアの定義や，SDB データベースを定義する前に，構造型 DB 機能が使える状態となったすべての RD エリアのバックアップを，データベース複写ユティリティ（pdcopy）などで取得してください。

(4)　障害時の運用

単体版 HiRDB から統合版 HiRDB への入れ替えに失敗した場合，失敗の原因によっては統合版 HiRDB をいったん単体版 HiRDB に戻して失敗の原因を取り除いてから，再度入れ替える必要があります。

統合版 HiRDB を入れ替え前に戻す手順を次の図に示します。

図 4-6　統合版 HiRDB を単体版 HiRDB に戻す手順（入れ替えに失敗した場合）



注※

HiRDB 運用ディレクトリ数分繰り返します。

4.7.3　統合版 HiRDB から単体版 HiRDB への入れ替え手順

ここでは，統合版 HiRDB を単体版 HiRDB に入れ替える手順について説明します。

(1) 入れ替え作業をする前の考慮点

入れ替え作業をする前の考慮点を次に示します。

(a) 指定値の見直しが必要なオペランド

次のオペランドは統合版 HiRDB 固有のオペランドです。そのため，単体版 HiRDB では指定できないので，オペランドの指定を削除してください。

- システム共通定義
  pd_structured_shmpool_dicsize
  pd_structured_advance_resident
- ユニット制御情報定義
  pd_structured_directory_path
- クライアント環境定義
  PDUAPEXERLOGDMLSZ
  PDUAPEXERLOGDMLDATA
  PDSDBPRMTRC
  PDSDBTRCCNDSIZE
  PDSDBTRCELMSIZE
  PDSDBTRCKEYSIZE
  PDSDBTRCDATASIZE
  PDSDBUAPDIR
  PDSDBUAPFILE

(b) 構想型 DB 機能で使用する資源の削除

構想型 DB 機能で使用する次の資源は，単体版 HiRDB への入れ替え後に削除できなくなります。そのため，単体版 HiRDB への入れ替え前に削除してください。

- SDB データベース定義，SDB データベース格納定義
- SDB データベースを格納する RD エリア
- SDB データベースの操作に関する監査対象イベントの定義
- SDB ディレクトリ情報ファイル
- SDB データベースに関するリソースだけを保有するスキーマ（リレーショナル DB で使用しないスキーマ）
- 追い付き反映キー対応表

また，次の資源については，単体版 HiRDB への入れ替え後に不要となるため，削除してください。

- SDB 用 UAP 環境定義ファイル

資源の削除手順については，「4.7.3(4)　構造型 DB 機能で使用した資源の削除手順」を参照してください。

(c)　マルチ HiRDB の構成

単体版 HiRDB と統合版 HiRDB が混在する構成にはできません。

(d)　修正版 HiRDB のコピーおよび入れ替え

入れ替え前の統合版 HiRDB と修正版の単体版 HiRDB では，次のコマンドは実行できません。

- pdprgcopy コマンド（修正版 HiRDB のコピー）
- pdprgrenew コマンド（修正版 HiRDB の入れ替え）

(2)　統合版 HiRDB から単体版 HiRDB への入れ替え時の注意事項

統合版 HiRDB を単体版 HiRDB に入れ替える場合，HiRDB クライアントも入れ替える必要があります。HiRDB クライアントを入れ替える際の注意事項を次に示します。

- HiRDB/Run Time をインストールするマシンに，セットアップ済みの HiRDB Structured Data Access Facility/Run Time がある場合は，HiRDB Structured Data Access Facility/Run Time をアンインストールしてください。そのあとで，HiRDB/Run Time をインストールしてください。上書きインストールした場合の動作は保証できません。
- HiRDB/Developer's Kit をインストールするマシンに，セットアップ済みの HiRDB Structured Data Access Facility/Developer's Kit がある場合は，HiRDB Structured Data Access Facility/Developer's Kit をアンインストールしてください。そのあとで，HiRDB/Developer's Kit をインストールしてください。上書きインストールした場合の動作は保証できません。

(3)　統合版 HiRDB から単体版 HiRDB への入れ替え手順

運用中の統合版 HiRDB を単体版 HiRDB に入れ替える際に，事前に確認・設計しておくべき項目，および入れ替え手順を次に示します。

なお，統合版 HiRDB のバージョンまたはリビジョン（VV-RR-ZZ の形式で示す HiRDB のバージョン番号のうち，VV または RR）が単体版 HiRDB のバージョンまたはリビジョンより古い場合は，バージョンアップも同時に実施されます。バージョンアップの手順についてはマニュアル「HiRDB システム導入・設計ガイド」の「HiRDB のバージョンアップ」を参照してください。ただし，「HiRDB のバージョンアップ」では上書きインストールの手順が説明されていますが，統合版 HiRDB を単体版 HiRDB に入れ替えてバージョンアップする場合には，アンインストール後にインストールする手順となります。

(a)　入れ替え前に事前に確認・設計しておくべき項目

入れ替え前に事前に確認・設計しておくべき項目を次に示します。

1. OS のオペレーティングシステムパラメタの確認

単体版 HiRDB に入れ替えることで OS 資源の消費量が減る場合は，単体版 HiRDB をインストールする前に，OS のオペレーティングシステムパラメタ（カーネルパラメタ）の値を変更してください。

オペレーティングシステムパラメタ（カーネルパラメタ）の見積もりについては，マニュアル「HiRDB システム導入・設計ガイド」の「Linux のオペレーティングシステムパラメタの見積もり」を参照してください。

(b)　統合版 HiRDB から単体版 HiRDB への入れ替え手順

統合版 HiRDB から単体版 HiRDB への入れ替え手順を次に示します。

1. RD エリアのバックアップ取得

   入れ替えが失敗した場合に備えて，データベース複写ユティリティ（pdcopy）などで，すべての RD エリアのバックアップを取得してください。

   バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。

2. 構造型 DB 機能で使用した資源の削除

   構造型 DB 機能で使用した資源を削除してください。

   資源の削除手順については，「4.7.3(4)　構造型 DB 機能で使用した資源の削除手順」を参照してください。

3. ディクショナリ表の再編成

   2.の手順（構造型 DB 機能で使用した資源の削除）によって発生した，ディクショナリ表中の無効領域を解放するため，データベース再編成ユティリティ（pdrorg）でディクショナリ表を再編成してください。

4. HiRDB のオンライン状態の確認

   pdls コマンドで HiRDB が正常終了できる状態であることを確認します。すべてのユニットが ACTIVE と表示されているかどうかを確認してください。

5. HiRDB の正常終了

   4.の手順で ACTIVE と表示されている場合，pdstop コマンドで HiRDB を正常終了させてください。

   HiRDB がすでに終了している場合は，次のどちらかの情報を参照して HiRDB が正常終了しているかどうかを確認してください。

   - メッセージログファイル
   - syslog ファイル

   正常終了していない場合は，pdstart コマンドでいったん HiRDB を開始してから，pdstop コマンドで正常終了させてください。

6. HiRDB の状態確認

   pdls -d ust コマンドで HiRDB が正常終了していることを確認してください。

7. ライブラリの共用化の解除

マルチ HiRDB 環境でライブラリを共用している場合は，pdmemsv -d コマンドでライブラリの共用化を解除してください。

入れ替え完了後に，再度 pdmemsv コマンドでライブラリを共用化してください。

8. コマンド，ユティリティ，アプリケーション，および HiRDB と連携している製品の停止

HiRDB Datareplicator，HiRDB Dataextractor，JP1/PFM など，HiRDB にアクセスする連携製品を停止してください。これらを停止しないと，実行形式ファイルや共用ライブラリの削除に失敗するおそれがあります。

9. HiRDB 運用ディレクトリ下のファイル，および SDB ディレクトリ情報ファイルのバックアップの取得（HiRDB 運用ディレクトリ数分繰り返す）

入れ替えに失敗した場合に備えて，OS の cp コマンドなどで，次のファイルのバックアップを取得してください。

- HiRDB 運用ディレクトリ下（$PDDIR/conf 下）のファイル
- SDB ディレクトリ情報ファイル
- HiRDB 運用ディレクトリ下のトラブルシュート情報（$PDDIR/spool 下）のファイル

HiRDB 運用ディレクトリのバックアップの取得方法については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB 運用ディレクトリの作成」の「HiRDB 運用ディレクトリのバックアップの取得」を参照してください。

10. SDB ディレクトリ情報ファイルの削除

OS の rm コマンドなどで，SDB ディレクトリ情報ファイルを削除してください。

11. 統合版 HiRDB を OS から削除（HiRDB 運用ディレクトリ数分繰り返す）

統合版 HiRDB をアンインストールする前に，統合版 HiRDB の HiRDB 運用ディレクトリ下の HiRDB システムを OS から削除してください。

このとき，統合版 HiRDB の実行に必要なファイルおよびディレクトリも削除してください（pdsetup -d コマンドを実行し，KFPS00036-Q メッセージに y を応答してください）。

pdsetup -d コマンドの終了ステータスが 0 でない場合，障害が起きているおそれがあります。syslog ファイルに出力されたメッセージを参照して，障害原因を取り除いてから，OS からの削除を再度実行してください。

12. 統合版 HiRDB のアンインストール

日立 PP インストーラを使用してサーバマシンごとに統合版 HiRDB をアンインストールしてください。

アンインストールの結果は，日立 PP インストーラの画面で確認してください。アンインストールが成功していない場合は，syslog ファイルに出力されたメッセージを参照して，障害原因を取り除いてから，アンインストールを再度実行してください。

13. 単体版 HiRDB のインストール

日立 PP インストーラを使用して，サーバマシンごとに単体版 HiRDB をインストールしてください。

14. HiRDB システム定義の変更（HiRDB 運用ディレクトリ数分繰り返す）

構造型 DB 機能を使用するために指定していたオペランドを，HiRDB 運用ディレクトリ下（$PDDIR/conf 下）の HiRDB のシステム定義ファイルから削除してください。

構造型 DB 機能を使用するために必要なオペランドについては，「9.　システム定義」を参照してください。

15. 単体版 HiRDB を OS に登録（HiRDB 運用ディレクトリ数分繰り返す）

単体版 HiRDB をインストールしたあとに，pdsetup コマンドで，単体版 HiRDB の HiRDB 運用ディレクトリ下の HiRDB システムを OS に登録してください。

pdsetup コマンドの-c オプションには，必ず統合版 HiRDB で指定していた文字コードを指定してください。

指定できる文字コードについては，「4.7.1　入れ替えに関する前提条件」を参照してください。

16. HiRDB の再起動

pdstart コマンドで HiRDB を再起動してください。HiRDB の再起動が完了すると，構造型 DB 機能が使用できない状態になります。

17. RD エリアのバックアップの取得

入れ替えが完了した時点（構造型 DB 機能が使用できなくなった状態）の，すべての RD エリアのバックアップを，データベース複写ユティリティ（pdcopy）などで取得してください。

(4)　構造型 DB 機能で使用した資源の削除手順

統合版 HiRDB から単体版 HiRDB へ入れ替える際に，事前準備として実施する構造型 DB 機能で使用した資源の削除手順を次に示します。

構造型 DB 機能で使用した資源の削除手順

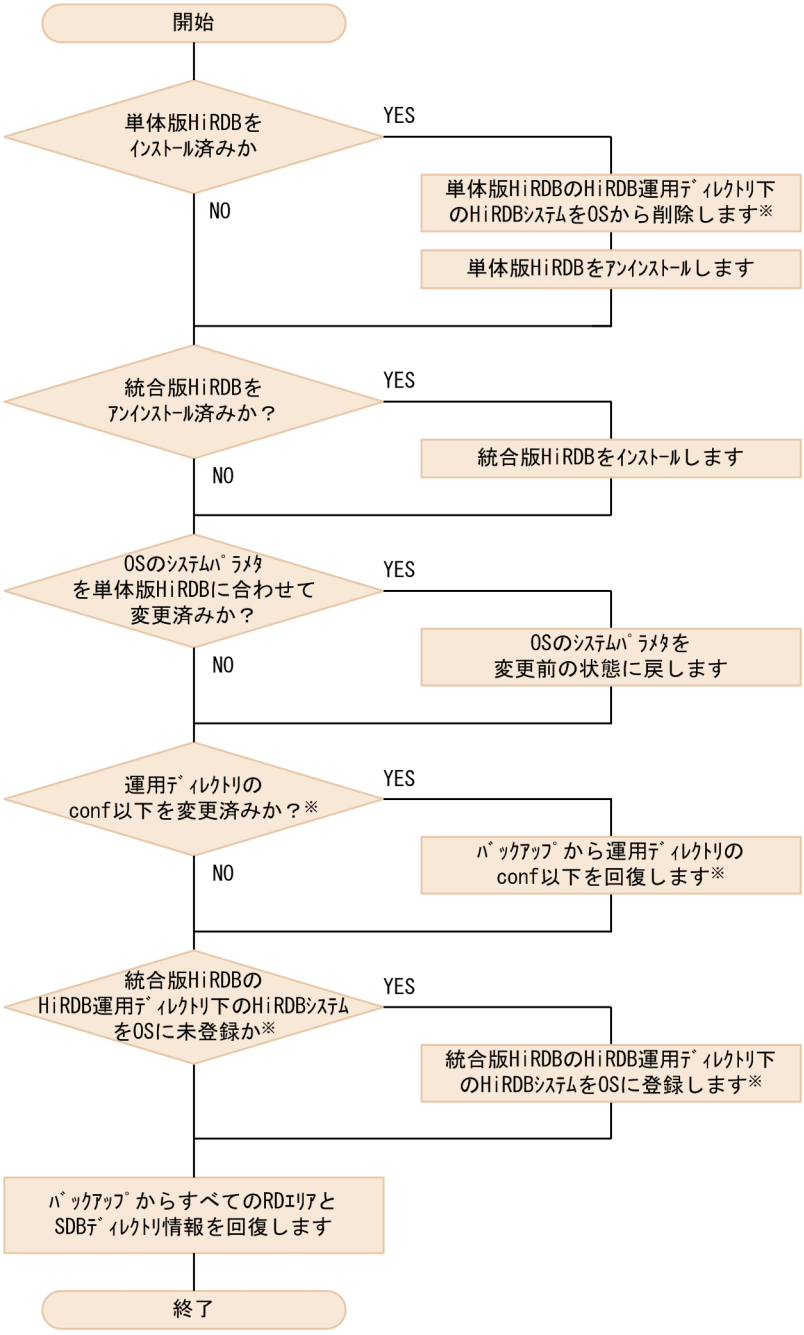
1. pdsdbdef コマンドで，SDB データベースの定義情報（SDB ディクショナリ情報，SDB ディレクトリ情報定義）を削除します。
2. pdmod コマンドで，SDB データベースを格納する RD エリアを削除します。
3. DROP AUDIT 文で，SDB データベース操作に関する監査対象イベントの定義を削除します。
4. DROP SCHEMA 文で，SDB データベースに関するリソースだけを保有するスキーマ（リレーショナル DB で使用しないスキーマ）を削除します。
5. 更新可能なオンライン再編成を使用している場合は，pdsdborcrt -k del コマンドで追い付き反映キー対応表を削除します。
6. SDB 用 UAP 環境定義を使用している場合は，OS の rm コマンドなどで，SDB 用 UAP 環境定義ファイルを格納しているディレクトリ（$PDCONFPATH/pdsdbuapenv）を削除します。

(5)　障害時の運用

統合版 HiRDB から単体版 HiRDB への入れ替えに失敗した場合，失敗の原因によっては単体版 HiRDB をいったん統合版 HiRDB に戻して失敗の原因を取り除いてから，再度入れ替える必要があります。

単体版 HiRDB を入れ替え前に戻す手順を次の図に示す。

図 4-7　単体版 HiRDB を統合版 HiRDB に戻す手順（入れ替えに失敗した場合）



注※

HiRDB 運用ディレクトリ数分繰り返します。

4.8　マルチ HiRDB の設計

マルチ HiRDB のシステム設計については，マニュアル「HiRDB システム導入・設計ガイド」の「マルチ HiRDB の設計」を参照してください。

なお，HiRDB Server と HiRDB Structured Data Access Facility が混在する構成にはできませんので，注意してください。

4.9　SDB 用 UAP 環境定義ファイルの準備【SD FMB】

SDB 用 UAP 環境定義を使用して，SD FMB の SDB データベースにアクセスする場合の準備作業について説明します。

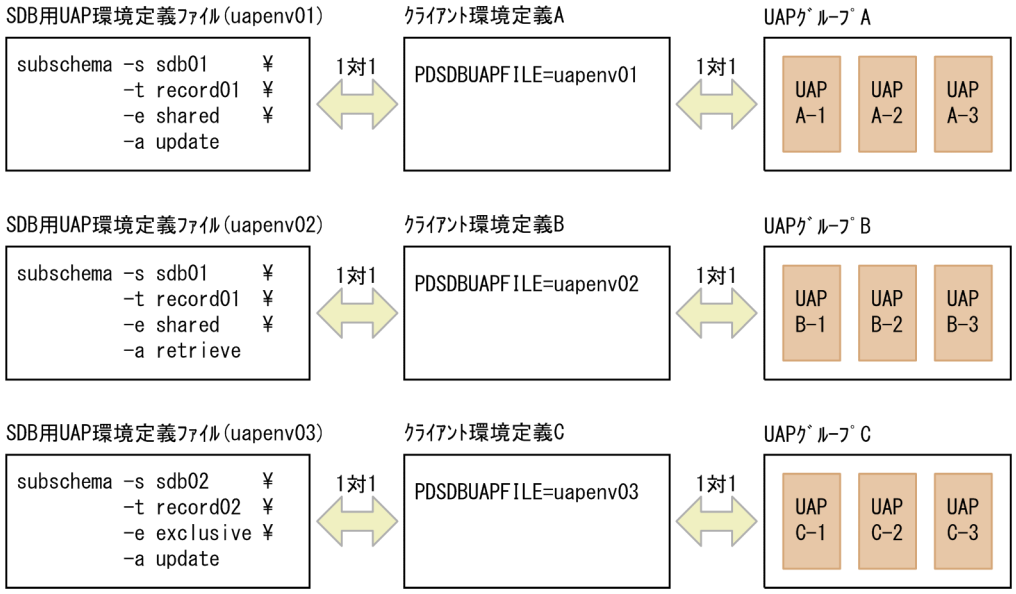
4.9.1　SDB 用 UAP 環境定義ファイルの作成

SDB 用 UAP 環境定義ファイルの作成単位の検討と，作成方法について説明します。

(1)　SDB 用 UAP 環境定義ファイルの作成単位の検討

1 つの UAP グループに対して，SDB 用 UAP 環境定義ファイルを 1 つ作成します。OpenTP1 環境下の場合，サービスグループが UAP グループに該当します。SDB 用 UAP 環境定義ファイルと UAP の関係を次の図に示します。

図 4-8　SDB 用 UAP 環境定義ファイルと UAP の関係



[説明]

SDB 用 UAP 環境定義ファイルは，クライアント環境定義を介して UAP グループと 1 対 1 で対応します。

例えば，UAP グループ A の UAP は，SDB 用 UAP 環境定義ファイル（uapenv01）を使用して SDB データベースにアクセスします。

なお，UAP グループ内で，SDB 用 UAP 環境定義のオペランドの指定値を変えたい場合は，UAP グループを分割してください。

(2)　SDB 用 UAP 環境定義ファイルの作成方法

SDB 用 UAP 環境定義ファイルを OS のテキストエディタで作成してください。SDB 用 UAP 環境定義の指定内容については，「9.3　SDB 用 UAP 環境定義【SD FMB】」を参照してください。

SDB 用 UAP 環境定義ファイルのファイル名は，先頭がアルファベットの英数字の文字列にしてください。また，ファイル名は最大 8 文字としてください。

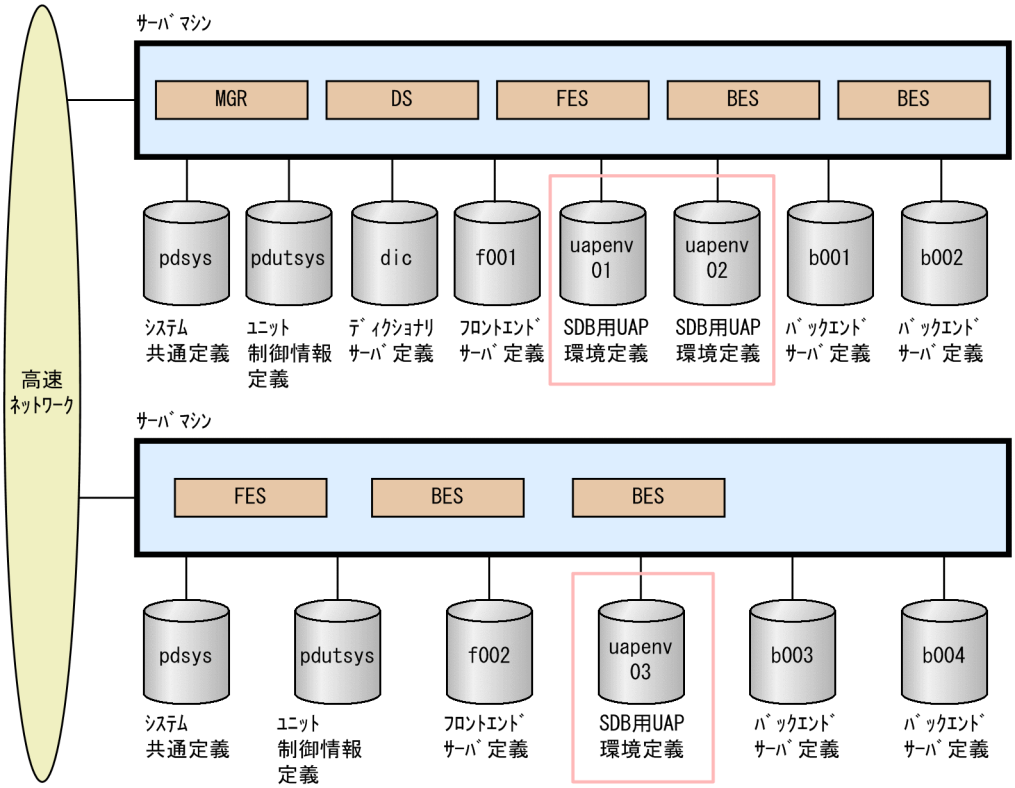
4.9.2　SDB 用 UAP 環境定義ファイルを配置するサーバマシンの検討

SDB 用 UAP 環境定義ファイルは，フロントエンドサーバがあるサーバマシンに配置します。マルチフロントエンドサーバの場合は，次に示すサーバマシンに SDB 用 UAP 環境定義ファイルを配置してください。

- SDB 用 UAP 環境定義ファイルを使用する UAP が，接続する可能性があるすべてのフロントエンドサーバのサーバマシン

SDB 用 UAP 環境定義ファイルの配置例を次の図に示します。

図 4-9　SDB 用 UAP 環境定義ファイルの配置例



参考

SDB 用 UAP 環境定義ファイルを使用する UAP が接続する可能性があるフロントエンドサーバが特定できない場合は，各フロントエンドサーバのサーバマシンにすべての SDB 用 UAP 環境定義ファイルを配置してください。

OS のファイル共用機能（NFS）を使用すると，SDB 用 UAP 環境定義ファイルを 1 つのサーバマシンで管理できます。NFS を使用した HiRDB システム定義ファイルの共用化については，マニュアル「HiRDB システム定義」の「HiRDB システム定義ファイルの共用化」を参照してください。

4.9.3　SDB 用 UAP 環境定義ファイルを格納するディレクトリの作成

(1)　$PDCONFPATH/pdsdbuapenv ディレクトリの作成

「4.9.2　SDB 用 UAP 環境定義ファイルを配置するサーバマシンの検討」で決定した SDB 用 UAP 環境定義ファイルを配置するサーバマシンの$PDCONFPATH 下に，pdsdbuapenv ディレクトリを作成してください。$PDCONFPATH/pdsdbuapenv 下に，SDB 用 UAP 環境定義ファイルを格納します。

注意事項

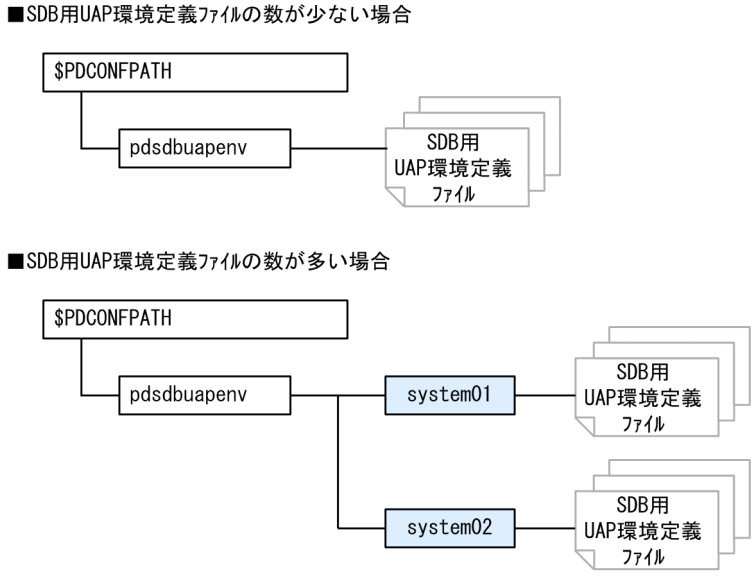
$PDCONFPATH/pdsdbuapenv ディレクトリのパーミッションは，755 に設定して維持してください。

■SDB 用 UAP 環境定義ファイルの数が多い場合

$PDCONFPATH/pdsdbuapenv 下に格納する SDB 用 UAP 環境定義ファイルの数が多くなると，HiRDB/SD がファイルサーチする際の処理時間が長くなります。そのため，SDB 用 UAP 環境定義ファイルの数が多くなる場合は，業務システムなどの単位で，$PDCONFPATH/pdsdbuapenv 下にさらにディレクトリを作成することを推奨します。作成したディレクトリ下に SDB 用 UAP 環境定義ファイルを格納します。

SDB 用 UAP 環境定義ファイルを格納するディレクトリの構成例を次の図に示します。

図 4-10　SDB 用 UAP 環境定義ファイルを格納するディレクトリの構成例



[説明]

上記の例の場合，$PDCONFPATH/pdsdbuapenv 下に system01 と system02 のディレクトリを作成しています。system01 と system02 のディレクトリ下に，それぞれ SDB 用 UAP 環境定義ファイルを格納しています。

注意事項

- ディレクトリ名称に使用する文字は，先頭が英字の英数字（最大 8 文字）にしてください。
- ここで作成したディレクトリ（上記の例の system01 と system02 のディレクトリ）のパーミッションは，755 に設定して維持してください。
- SDB 用 UAP 環境定義ファイルを格納できるディレクトリの階層は，$PDCONFPATH/pdsdbuapenv 下 1 階層までです。例えば，$PDCONFPATH/pdsdbuapenv/system01/subdir01 ディレクトリを作成し，そのディレクトリに SDB 用 UAP 環境定義ファイルを格納することはできません。

(2)　SDB 用 UAP 環境定義ファイルの配置

SDB 用 UAP 環境定義ファイルを，「4.9.2　SDB 用 UAP 環境定義ファイルを配置するサーバマシンの検討」で決定したサーバマシンに配置してください。「4.9.3(1)　$PDCONFPATH/pdsdbuapenv ディレクトリの作成」で作成したディレクトリ下に SDB 用 UAP 環境定義ファイルを格納します。

注意事項

SDB 用 UAP 環境定義ファイルに対するパーミッションは，644 に設定して維持してください。

参考

UAP が HiRDB サーバに接続した際，HiRDB/SD は SDB 用 UAP 環境定義ファイルを読み込んで，解析結果をプロセス固有メモリに保持します。その結果，DML の実行時に，SDB 用 UAP 環境定義ファイルに指定された機能が適用されます。

4.9.4 クライアント環境定義の作成

次に示すオペランドを指定したクライアント環境定義を，UAP グループごとに作成してください。

- PDSDBUAPFILE

  このオペランドには，SDB 用 UAP 環境定義ファイルの名称を指定します。

  このオペランドの指定は必須となります。

- PDSDBUAPDIR

  このオペランドには，SDB 用 UAP 環境定義ファイルを格納するディレクトリ名を指定します。$PDCONFPATH/pdsdbuapenv 下に作成したディレクトリ名を指定します。

  「4.9.3(1) $PDCONFPATH/pdsdbuapenv ディレクトリの作成」で，$PDCONFPATH/pdsdbuapenv 下にディレクトリを作成した場合に，このオペランドを指定してください。

PDSDBUAPFILE オペランドおよび PDSDBUAPDIR オペランドの詳細については，「16.3.1(4) SDB データベースを操作する API または DML の実行時にだけ指定できるクライアント環境定義」を参照してください。

クライアント環境定義の PDSDBUAPFILE オペランドおよび PDSDBUAPDIR オペランドの指定例を次に示します。

- 例 1

  $PDCONFPATH/pdsdbuapenv 下に格納している SDB 用 UAP 環境定義ファイル（uapenv01）を UAP に適用する場合の，クライアント環境定義の指定例を次に示します。

```
PDSDBUAPFILE=uapenv01
```

- 例 2

  $PDCONFPATH/pdsdbuapenv/system01 下に格納している SDB 用 UAP 環境定義ファイル（uapenv01）を UAP に適用する場合の，クライアント環境定義の指定例を次に示します。

```
PDSDBUAPFILE=uapenv01       ...1
PDSDBUAPDIR=system01        ...2
```

  [説明]

  1. SDB 用 UAP 環境定義ファイルのファイル名を指定します。

2. SDB 用 UAP 環境定義ファイルを格納したディレクトリ名を指定します。$PDCONFPATH/pdsdbuapenv 下のディレクトリ名を指定します。

■24 時間連続稼働のシステムの場合

OpenTP1 環境下の場合，クライアント環境定義を変更すると，ユーザサーバの再起動が必要になります。そのため，24 時間連続稼働するシステムの場合は，0 バイトの SDB 用 UAP 環境定義ファイルを作成しておくか，またはコメントだけを記述した SDB 用 UAP 環境定義ファイルを作成しておくことを推奨します。このあらかじめ作成しておいた SDB 用 UAP 環境定義ファイルの名称と格納ディレクトリを，クライアント環境定義の PDSDBUAPFILE と PDSDBUAPDIR に指定しておいてください。

このように準備しておくと，システムの稼働後に SDB 用 UAP 環境定義を使用することになっても，クライアント環境定義の変更が不要になります。そのため，ユーザサーバの再起動が不要になります。

SDB 用 UAP 環境定義の変更方法については，「5.10.1　SDB 用 UAP 環境定義の変更方法【SDFMB】」を参照してください。

5 運用

この章では，構造型 DB 機能を使用する HiRDB の運用方法について説明します。

5.1　運用項目の一覧

構造型 DB 機能を使用する HiRDB の主な運用項目の一覧を次の表に示します。

表 5-1　HiRDB の主な運用項目の一覧

項番	運用項目		参照先
1	通常・定期運用	HiRDB の開始，終了	「5.2　HiRDB の開始および終了」
2		システムの状態監視	「5.3　システムの状態監視」
3		統計情報の取得	「5.4　統計情報の取得」
4		データベースのバックアップの取得	「5.5　データベースのバックアップの取得」
5		システムログファイルの運用	「HiRDB システム運用ガイド」の「システムログファイルの運用」
6		RD エリアの空き容量の確認	「5.14.1　RD エリアの空き容量の確認」
7		一連番号の監視	「5.25　一連番号の監視」
8	保守運用	SDB データベースの定義追加，定義変更，または定義削除	• 「5.6　SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合）」 • 「5.7　SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合）」
9		SDB ディレクトリ情報ファイルの配布	「5.8　SDB ディレクトリ情報ファイルの配布方法」
10		常用常駐領域の SDB ディレクトリ情報の確認	「5.9　全ユニットの SDB ディレクトリ情報が一致しているか確認する方法」
11		• HiRDB システム定義の変更 • SDB 用 UAP 環境定義の変更	「5.10　HiRDB システム定義の変更」
12		クライアント環境定義の変更	「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」
13		SDB データベースへのデータロード	「5.19　SDB データベースへのデータロード」
14		インデクスの再作成	「5.20　インデクスの再作成」
15		SDB データベースの再編成	• 「5.11　SDB データベースの再編成（インナレプリカ機能を使用しない場合）」 • 「5.12　SDB データベースの再編成（インナレプリカ機能を使用したオンライン再編成）【4V FMB，4V AFM】」 • 「5.13　SDB データベースの再編成（インナレプリカ機能を使用した更新可能なオンライン再編成）【4V FMB】」

項番	運用項目		参照先
16		RD エリアの追加，削除，容量拡張，設定変更（自動増分の設定など）	「5.14　RD エリアの運用」
17		HiRDB ファイルシステム領域の作成，削除，バックアップ取得，回復	「HiRDB システム運用ガイド」の「HiRDB ファイルシステム領域の運用」
18		SDB データベースの横分割数の変更	「5.15　SDB データベースの横分割数を増やす方法」
19		ユーザ権限の設定	「5.16.1　構造型 DB 機能に関するユーザ権限の設定」
20		スキーマの追加，削除	「5.17　スキーマの追加，削除」
21		OS の時刻の変更	「5.24　OS の時刻を変更する方法」
22		システム構成の変更	「5.27　システム構成を変更する方法」
23	トラブルシュート	データベースの回復	「5.18　データベースの回復」
24		障害発生時の対処	「HiRDB システム運用ガイド」の「障害が発生したときの対処方法」
25		障害発生時の対処（pdsdblod コマンドの異常終了時）	「5.22　障害が発生したときの対処方法（pdsdblod コマンドの異常終了時）」
26		障害発生時の対処（更新可能なオンライン再編成）	「5.23　障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】」
27		PRF トレース機能の使用	「5.26　PRF トレース機能」
28	その他	ログレスモードでの運用（ログを取得しないときの運用）	「HiRDB システム運用ガイド」の「ログレスモードで UAP またはユティリティを実行するときの運用」
29		高速系切り替え機能使用時の運用	「6.　高速系切り替え機能の設定と運用」
30		セキュリティ監査機能使用時の運用	「7.　セキュリティ監査機能の設定と運用」
31		ディザスタリカバリシステム使用時の運用	「8.　ディザスタリカバリシステムの構築と運用【4V FMB，4V AFM】」

5.2　HiRDB の開始および終了

pdstart コマンドで HiRDB を開始し，pdstop コマンドで HiRDB を終了します。pdstart コマンドおよび pdstop コマンドは，システムマネジャがあるサーバマシンで実行してください。

開始方法および終了方法の詳細については，マニュアル「HiRDB システム運用ガイド」の「HiRDB の開始および終了方法」を参照してください。

pdstart コマンドおよび pdstop コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

■高速系切り替え機能を使用している場合

高速系切り替え機能を使用している場合の HiRDB の開始方法および終了方法については，「6.11.1 HiRDB の開始」を参照してください。

なお，HiRDB を開始したあとに，pdsdbarc -c コマンドを実行して，SDB ディレクトリ情報の最終更新日時のチェックを行い，全ユニットの SDB ディレクトリ情報が一致していることを確認してください。

また，高速系切り替え機能を使用している場合は，実行系ユニットと待機系ユニットの SDB ディレクトリ情報が一致していることを確認してください。確認方法については，「5.9.2　常用常駐領域の SDB ディレクトリ情報の確認方法（待機系ユニットの場合）」を参照してください。

5.3　システムの状態監視

HiRDB の稼働中は次に示す項目を定期的に監視してください。

- 出力されたメッセージ
- システムログファイルの状態

メッセージの参照方法については，マニュアル「HiRDB システム運用ガイド」の「メッセージログでシステムの実行状況を確認する」を参照してください。

システムログファイルの監視方法については，マニュアル「HiRDB システム運用ガイド」の「システムログファイルの運用」を参照してください。なお，メッセージおよびシステムログファイル以外にも，監視する項目があります。詳細については，マニュアル「HiRDB システム運用ガイド」の「システムの稼働環境を知る方法（システムの状態監視）」を参照してください。

5.4　統計情報の取得

統計情報の取得については，マニュアル「HiRDB システム運用ガイド」の「システムの稼働環境を知る方法」を参照してください。SDB データベースに関する情報は，次に示す統計情報にだけ出力されます。

- システムの稼働に関する統計情報
- UAP に関する統計情報
- グローバルバッファプールに関する統計情報
- インデクスに関する統計情報
- データベース操作に関する HiRDB ファイルの統計情報
- デファードライト処理に関する統計情報
- データベースの入出力に関する統計情報

5.5　データベースのバックアップの取得

pdcopy コマンドで定期的にデータベースのバックアップを取得します。バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。pdcopy コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

同時にバックアップを取得する必要がある RD エリア

バックアップを取得する際，ユーザ用 RD エリアのバックアップだけを取得するのではなく，システム用 RD エリアのバックアップも取得する必要があるケースがあります。バックアップを取得する際は，次に示す表に記載されている RD エリアのバックアップを同時に取得するようにしてください。

表 5-2　同時にバックアップを取得する必要がある RD エリア

前回のバックアップ取得以降に実行した操作		バックアップを取得する RD エリアの種類			
		MST	DIR	DIC	USR
レコードの格納					○
レコードの更新					○
レコードの削除					○
レコードの一括削除					○
HiRDB/SD 定義ユティリティ※	• *ENTRY DICTIONARY 文（SDB ディクショナリ情報の登録） • *ALTER DICTIONARY 文（SDB ディクショナリ情報の変更）			○	
	• *DELETE DICTIONARY 文（SDB ディクショナリ情報の削除） • *CHECK DICTIONARY 文（データベース定義のチェック） • *ENTRY DIRECTORY 文（SDB ディレクトリ情報の追加） • *ALTER DIRECTORY 文（SDB ディレクトリ情報の変更） • *DELETE DIRECTORY 文（SDB ディレクトリ情報の削除）	○	○	○	○
HiRDB/SD データベース作成ユティリティ					○

（凡例）

○：同時にバックアップを取得する必要がある RD エリア

空白：バックアップを取得する必要がないことを意味しています。

MST：マスタディレクトリ用 RD エリア

DIR：データディレクトリ用 RD エリア

DIC：データディクショナリ用 RD エリア

USR：ユーザ用 RD エリア

注

- 上記の表の見方について説明します。例えば，SDB ディレクトリ情報の追加を実行した場合，その操作のあとでマスタディレクトリ用 RD エリア，データディレクトリ用 RD エリア，データディクショナリ用 RD エリア，およびユーザ用 RD エリアのバックアップを同時に取得する必要があるということを示しています。
- マニュアル「HiRDB システム運用ガイド」の「同時にバックアップを取得する必要がある RD エリア」にも，同時にバックアップを取得する必要がある RD エリアの記載があります。そちらもあわせて参照してください。

注※

SDB ディレクトリ情報ファイルを作成しただけの場合は，バックアップを取得する必要はありません。

インナレプリカ機能を使用したバックアップの取得方法

インナレプリカ機能を使用したバックアップの取得方法については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「インナレプリカグループ内の RD エリアのバックアップと回復」を参照してください。

5.6　SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合）

ここでは，HiRDB の再起動を必要とする SDB データベースの定義追加，定義変更，または定義削除の手順について説明します。

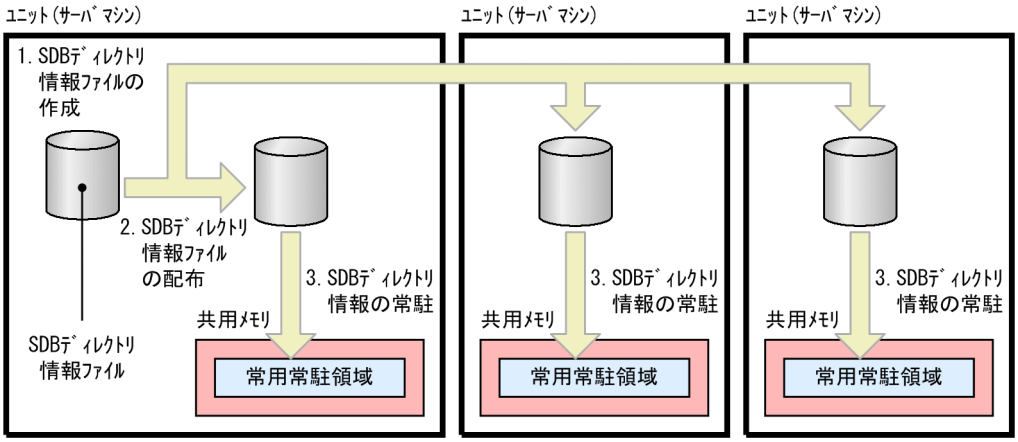
HiRDB の再起動を必要としない SDB データベースの定義追加，定義変更の手順については，「5.7　SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合）」を参照してください。

なお，SDB データベースの定義削除，または分割格納条件の変更機能を使って，格納 RD エリアの削除や，分割条件を変更して格納 RD エリアの分割，統合をする場合は，定義削除や変更後に HiRDB の再起動を必要とするため，ここで説明している手順を実施してください。

5.6.1　SDB ディレクトリ情報の変更の流れ

SDB データベースの定義追加，定義変更，または定義削除したあとに，SDB ディレクトリ情報を変更する必要があります。SDB ディレクトリ情報を変更する際の流れを次の図に示します。

図 5-1　SDB ディレクトリ情報を変更する際の流れ



［説明］

1. pdsdbdef コマンドを実行して SDB データベースの定義追加，定義変更，または定義削除したあとに，SDB ディレクトリ情報ファイルを作成します。
2. 1.で作成した SDB ディレクトリ情報ファイルを各ユニットに配布します。
3. HiRDB を再起動すると，SDB ディレクトリ情報が常用常駐領域に常駐されます。

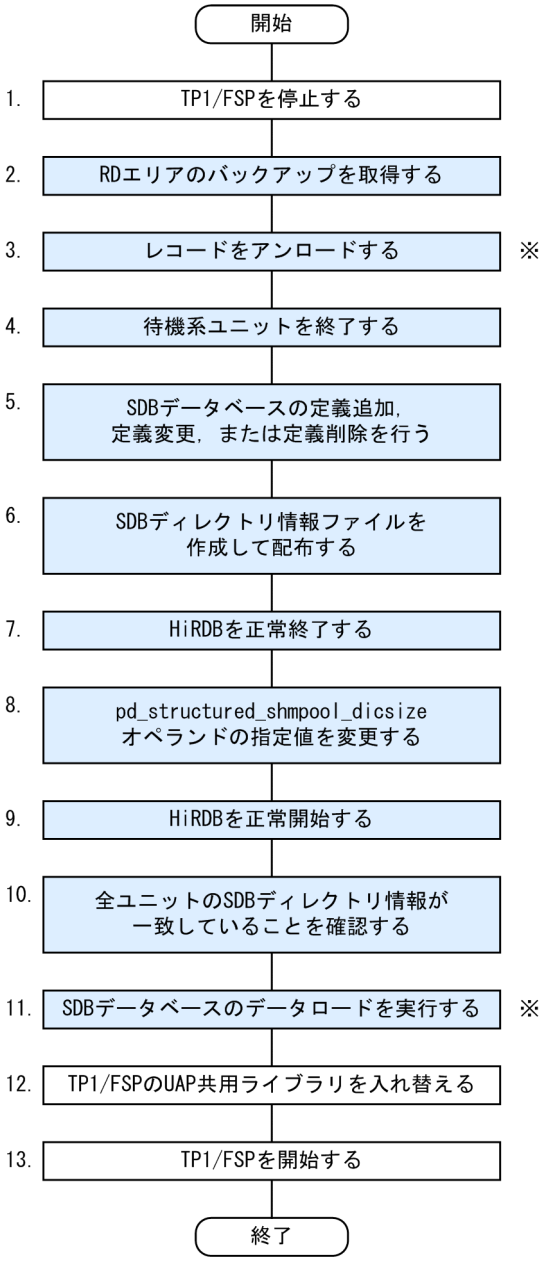
上記の流れで，変更後の SDB ディレクトリ情報を共用メモリに常駐します。

なお，1.と 2.は，1 回の pdsdbdef コマンドで実行できます。

5.6.2　SDB データベースの定義追加，定義変更，または定義削除の手順（概要）

SDB データベースの定義追加，定義変更，または定義削除する手順（概要）を次の図に示します。

図 5-2　SDB データベースの定義追加，定義変更，または定義削除する手順（概要）



注※

「5.6.5(1)　SDB データベースの定義変更をする方法」で格納 RD エリアを分割する場合，「5.6.5(2) SDB データベースを再定義する方法」で SDB データベースの定義変更をする場合に必要な手順です。

注意事項

- 4V FMB または 4V AFM の SDB データベースの定義追加，定義変更，または定義削除をする場合は，上記の 1.～13.の操作を実行してください。
- SD FMB の SDB データベースの定義追加，定義変更，または定義削除をする場合は，上記の図の色の付いている個所（2.～11.）の操作を実行してください。

図中の項番は，「5.6.3　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）」の（　）の番号と対応しています。

■上記の手順の途中で障害が発生した場合

次に示すどちらかの方法で対処してください。

- SDB データベースの定義追加，定義変更，または定義削除後の状態にする場合

  障害発生時に実行していた手順を再実行し，以降の手順を続行してください。
- SDB データベースの定義追加，定義変更，または定義削除前の状態に戻す場合

  項番 2 で取得したバックアップを使用して RD エリアを回復し，そのあとで SDB ディレクトリ情報ファイルを再作成してください。手順については，「5.18.3　例題 2（システム用 RD エリアとユーザ用 RD エリアをバックアップ取得時点に回復する場合）」を参照してください。

  また，項番 12 の TP1/FSP の UAP 共用ライブラリの入れ替え後に障害が発生した場合は，UAP 共用ライブラリを入れ替え前の状態に戻してください。

5.6.3　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）

SDB データベースの定義追加，定義変更，または定義削除をする際の，各手順の詳細について説明します。

注意事項

【4V FMB，4V AFM】の表記がある個所の説明は，4V FMB または 4V AFM の SDB データベースの定義追加，定義変更，または定義削除をする際に実行する操作です。SD FMB の SDB データベースの定義追加，定義変更，または定義削除をする際は，実行する必要がない操作です。

(1)　TP1/FSP を停止する【4V FMB，4V AFM】

TP1/FSP の eesvstop コマンドで TP1/FSP を停止します。この操作によってオンラインが停止状態になります。

(2)　RD エリアのバックアップを取得する

pdcopy コマンドで，次に示す RD エリアのバックアップを取得します。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 処理対象の SDB データベースを格納しているユーザ用 RD エリア（レコードおよびインデクスを格納しているユーザ用 RD エリア）

バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。

(3) レコードをアンロードする

SDB データベースの定義変更をする場合は，pdsdbrog コマンドでレコードのアンロードを実行します。ただし，*ALTER DICTIONARY 文で最下位のレコード型とその親子集合の追加の定義変更の場合は，アンロードは不要です。

注意事項

SD FMB の SDB データベースの定義変更は，SDB データベースの再定義となります。そのため，SD FMB の SDB データベースの定義変更をする場合は，レコードのアンロードを必ず実行してください。

分割格納条件の変更機能を使用して，既存の RD エリアを変更（分割または統合）する場合，pdsdbdef コマンドの実行時に変更（分割または統合）対象の RD エリアに格納されているデータが削除されることがあるため，削除対象のデータが格納されている RD エリアのデータをアンロードしておく必要があります。定義変更と RD エリアに格納されているデータの削除の関係については，「表 11-23　RD エリアの定義変更時のデータの削除有無（格納条件指定の横分割の場合）」または「表 11-24　RD エリアの定義変更時のデータの削除有無（境界値指定の横分割の場合）」を参照してください。

■アンロードが必要となる範囲

定義変更によってアンロードが必要となる範囲について説明します。

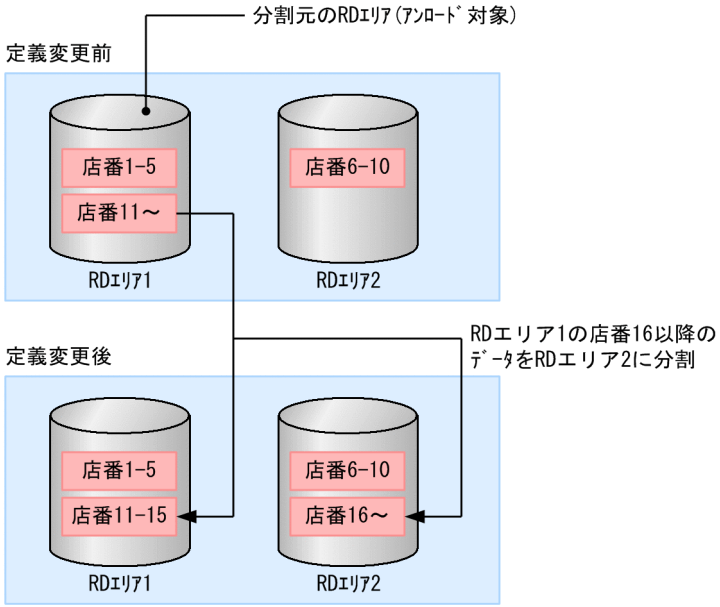
- **SDB データベースを再作成する場合**

  SDB データベースに格納されている全データをアンロードします。

- **データ格納用 RD エリアを分割する場合**

  分割元の RD エリア（分割対象の RD エリア）に格納されているデータをアンロードします。例を次に示します。

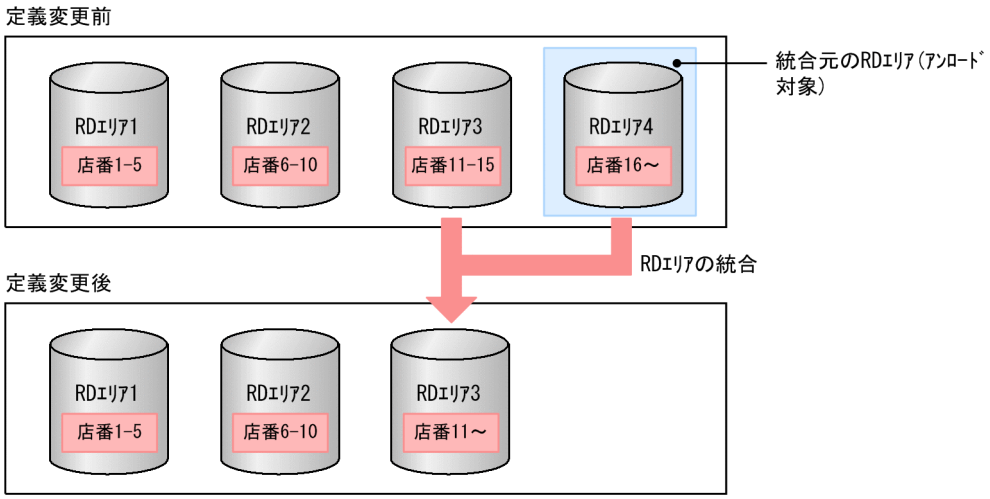
  （例）定義変更によって境界値の範囲の一部を既存の RD エリアに分割する場合



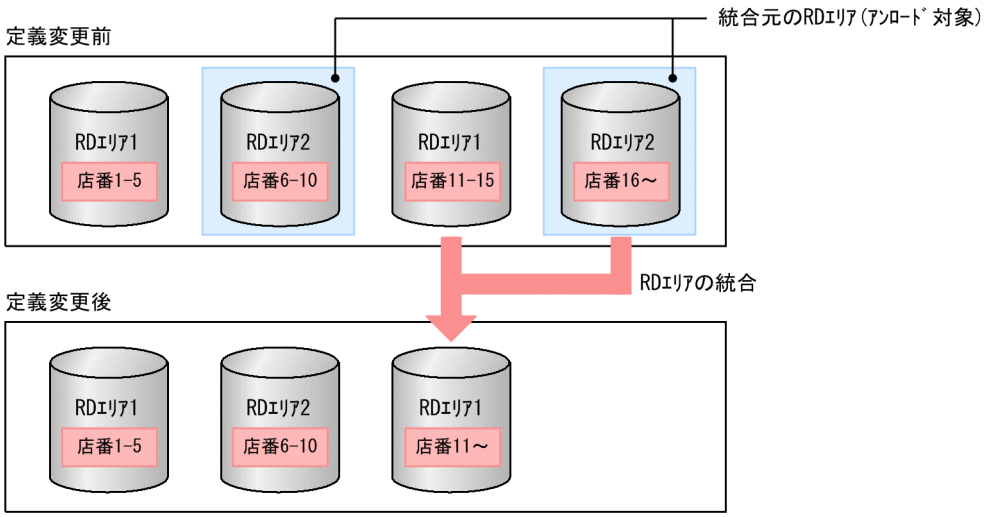
- **データ格納用 RD エリアを統合する場合**

  統合元の RD エリア（削除対象の RD エリア）に格納されているデータをアンロードします。例を次に示します。

  （例 1）定義変更によって削除される RD エリアが，定義変更後には存在しない場合



  （例 2）定義変更によって削除される RD エリアが，定義変更後にも存在する場合



- データ格納用 RD エリアを削除する場合

  削除対象の RD エリアに格納されているデータをアンロードします。

なお，pdsdbrog コマンドを実行する前に pdsdbarc -a コマンドを実行して，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）」を参照してください。

(4)　待機系ユニットを終了する

高速系切り替え機能を使用している場合は，HA モニタの monsbystp コマンドですべての待機系ユニットを終了してください。

(5)　SDB データベースの定義追加，定義変更，または定義削除を行う

- SDB データベースの定義追加をする場合

  「5.6.4　SDB データベースの定義追加」を参照してください。
- SDB データベースの定義変更をする場合

  「5.6.5　SDB データベースの定義変更」を参照してください。
- SDB データベースの定義削除をする場合

  「5.6.6　SDB データベースの定義削除」を参照してください。

(6)　SDB ディレクトリ情報ファイルを作成して配布する

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して全ユニット（高速系切り替え機能を使用している場合は待機系ユニットも含む）に配布してください。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1　SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

(7)　HiRDB を正常終了する

pdstop コマンドで HiRDB を正常終了します。必ず正常終了させてください。

(8)　pd_structured_shmpool_dicsize オペランドの指定値を変更する

次に示す場合は，システム共通定義の pd_structured_shmpool_dicsize オペランドの指定値を変更してください。

- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値を超えている場合
- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値に比べてかなり小さい場合

SDB ディレクトリ情報の合計長は，(6)で pdsdbdef コマンドを実行した際に出力される KFPB61231-I メッセージ（size に表示されます）で確認できます。

(9)　HiRDB を正常開始する

pdstart コマンドで HiRDB を正常開始します。SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報が常用常駐領域に常駐されます。

(4)で待機系ユニットを終了している場合は，pdstart -q コマンドですべての待機系ユニットを開始してください。

(10)　全ユニットの SDB ディレクトリ情報が一致していることを確認する

pdsdbarc -c コマンドを実行して，SDB ディレクトリ情報の最終更新日時のチェックを行い，全ユニットの SDB ディレクトリ情報が一致していることを確認してください。

チェックの結果，問題がある場合は，全ユニットに同じ SDB ディレクトリ情報ファイルが配布されていません。この場合，pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを配布し直してください。そのあとで，HiRDB を再起動してください。

なお，「(11)　SDB データベースのデータロードを実行する」を行う場合は，pdsdbarc -a コマンドを実行して，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）」を参照してください。

(11)　SDB データベースのデータロードを実行する

(3)でレコードをアンロードした場合，pdsdblod コマンドで SDB データベースのデータロードを実行します。構成要素の変更など，レコード型の形式を変更した場合は，(3)で作成したアンロードファイルの内容を加工する必要があります。

■データロードが必要となる範囲

定義変更によってアンロードを行った場合の SDB データベースのデータロードが必要となる範囲について説明します。

- SDB データベースを再作成する場合

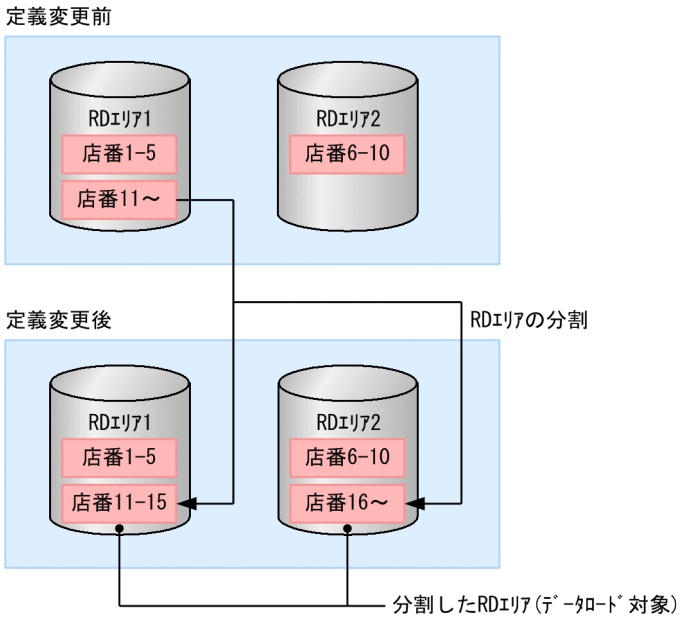
  再作成した SDB データベース全体に対して，pdsdblod コマンドでデータロードを実行します。このとき，(3)で作成したアンロードファイルを使用します。

- データ格納用 RD エリアを分割する場合

  分割した各 RD エリアに対して，pdsdblod コマンドでデータロードを実行してください。データが削除された RD エリアに対しては，初期データロード（purge オペランドに yes を指定）を実行してください。それ以外の RD エリアに対しては，追加データロード（purge オペランドに no を指定）を実行してください。※このとき，(3)で作成したアンロードファイルを使用します。

  例を次に示します。

  （例）定義変更によって境界値の範囲の一部を既存の RD エリアに分割する場合

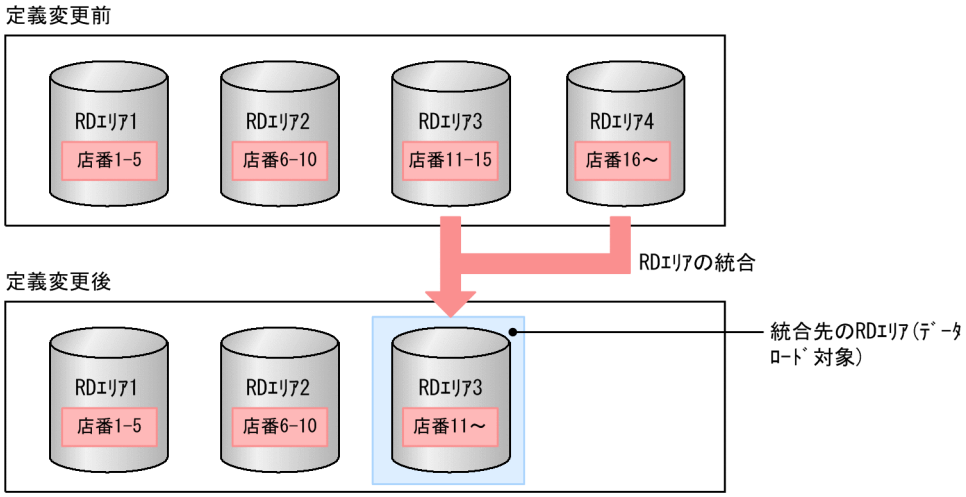


- データ格納用 RD エリアを統合する場合

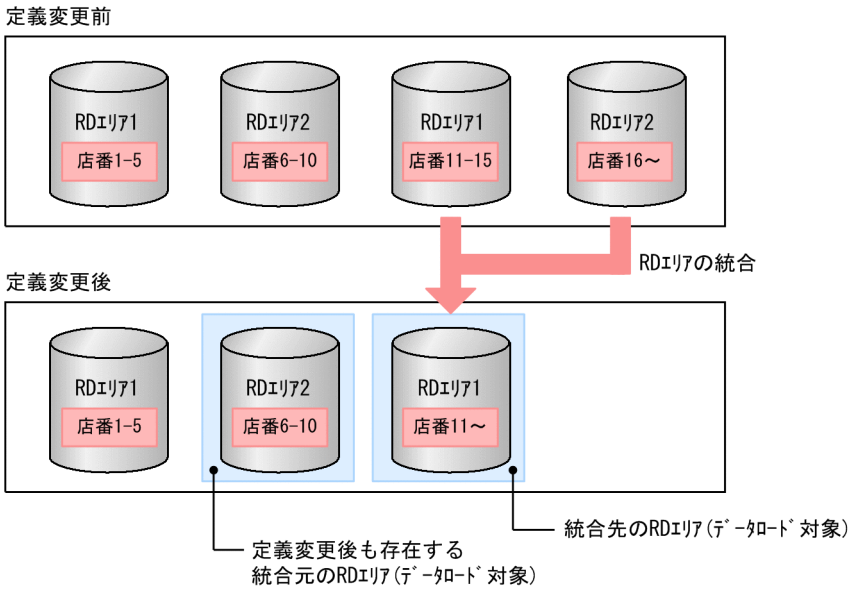
  格納する必要があるデータを pdsdblod コマンドでデータロードしてください。※

  例を次に示します。

  （例 1）定義変更によって削除される RD エリアが，定義変更後には存在しない場合



（例 2）定義変更によって削除される RD エリアが，定義変更後にも存在する場合



＜ 4V FMB の SDB データベースの場合＞

統合先の RD エリアに対して，pdsdblod コマンドで追加データロード（purge オペランドに no を指定）を実行します。このとき，(3)で作成したアンロードファイルを使用します。

また，統合元の RD エリアが定義変更後も存在する場合は，統合元の RD エリアに対して，pdsdblod コマンドで初期データロード（purge オペランドに yes を指定）を実行します。このとき，(3)で作成したアンロードファイルを使用します。

＜ 4V AFM の SDB データベースの場合＞

統合先の RD エリアのデータをアンロードし，(3)でアンロードしたデータとマージします。そのあと，統合先の RD エリアに対して，pdsdblod コマンドで初期データロード（purge オペランドに yes を指定）を実行します。また，統合元の RD エリアが定義変更後も存在する場合は，統合元の RD エリアに対しても，マージしたデータを pdsdblod コマンドで初期データロード（purge オペランドに yes を指定）します。

- **データ格納用 RD エリアを削除する場合**

(3)でレコードをアンロードした場合，データのキー値を，統合先の RD エリアに対応する値に変更し，格納する必要があるデータを pdsdblod コマンドでデータロードしてください。

＜ 4V FMB の SDB データベースの場合＞

キー値を変更したデータを，pdsdblod コマンドで追加データロード（purge オペランドに no を指定）してください。

＜ 4V AFM の SDB データベースの場合＞

統合先の RD エリアに格納されているデータをアンロードし，キー値を変更したデータとマージします。そのあと，統合先の RD エリアに対して，マージしたデータを pdsdblod コマンドで初期データロード（purge オペランドに yes を指定）してください。

注※

アンロードファイル中の格納条件に一致しないレコードはエラーとして扱われます。このため，pdsdblod コマンドの divermsg オペランドには noput を指定して，エラー処理をスキップさせて，各 RD エリアにデータロードを実行してください。

なお，pdsdblod コマンドを実行する前に pdsdbarc -a コマンドを実行して，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）」を参照してください。

(12)　TP1/FSP の UAP 共用ライブラリを入れ替える【4V FMB，4V AFM】

TP1/EE サービス定義の module_dir オペランドに指定した UAP 共用ライブラリ格納ディレクトリに，新しい UAP 共用ライブラリを格納します。または，module_dir オペランドの指定を，新しい UAP 共用ライブラリを格納したディレクトリに変更します。

(13)　TP1/FSP を開始する【4V FMB，4V AFM】

TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

このとき，常用常駐領域中の SDB ディレクトリ情報を入力情報にして，TP1/FSP の SDB 定義情報領域に SDB 定義情報が格納されます。

この作業が完了すると，定義追加，定義変更，または定義削除した SDB データベースにアクセスできるようになります。

5.6.4　SDB データベースの定義追加

SDB データベースの定義追加の手順を次に示します。

手順

1. データベース定義を作成する

定義追加する SDB データベースの SDB データベース定義および SDB データベース格納定義を作成します。SDB データベース定義および SDB データベース格納定義については，次の個所を参照してください。

- 4V FMB の SDB データベースを定義する場合
  「11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】」
- 4V AFM の SDB データベースを定義する場合
  「11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】」
- SD FMB の SDB データベースを定義する場合
  「11.7　SDB データベース定義，SDB データベース格納定義【SD FMB】」

2. SDB ディクショナリ情報および SDB ディレクトリ情報を追加する

pdsdbdef コマンドで*ENTRY DICTIONARY 文および*ENTRY DIRECTORY 文を実行し，定義追加する SDB データベースの SDB ディクショナリ情報および SDB ディレクトリ情報をディクショナリ表に追加します。1.で作成した SDB データベース定義および SDB データベース格納定義を*ENTRY DICTIONARY 文に指定します。

SDB 定義文の参照先を次に示します。

- *ENTRY DICTIONARY 文：「11.4.2　*ENTRY DICTIONARY 文（SDB ディクショナリ情報の追加）」
- *ENTRY DIRECTORY 文：「11.4.6　*ENTRY DIRECTORY 文（SDB ディレクトリ情報の追加）」

注意事項

*ENTRY DICTIONARY 文および*ENTRY DIRECTORY 文は，1 回の pdsdbdef コマンドで実行してください。

理由：

*ENTRY DIRECTORY 文の実行時には，データベース定義のチェックが実行されます。

*ENTRY DICTIONARY 文および*ENTRY DIRECTORY 文を 1 回の pdsdbdef コマンドで実行した場合，データベース定義の不整合が検知されると，SDB ディクショナリ情報および SDB ディレクトリ情報はディクショナリ表に追加されません。この場合，データベース定義を修正して，pdsdbdef コマンドを再実行することで対処できます。

*ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文を別々に実行した場合，*ENTRY DIRECTORY 文の実行時には，すでに SDB ディクショナリ情報はディクショナリ表に追加されています。そのため，データベース定義の不整合が検知された場合に，データベース定義を修正するには，*DELETE DICTIONARY 文を実行して SDB ディクショナリ情報を削除してから，pdsdbdef コマンドを再実行することになります。

5.6.5　SDB データベースの定義変更

SDB データベースの定義変更をするには，次の 2 つの方法があります。

1. SDB データベースの定義変更をする方法
2. SDB データベースを再定義する方法

1.の方法では次の操作ができます。

- 最下位のレコード型とその親子集合の定義追加
- 格納する RD エリアの変更

そのほかの定義変更をする場合は，2.の方法で実施してください。

最下位のレコード型とその親子集合の定義追加の例については，4V FMB の SDB データベースの場合は「11.5.1(4)　*ALTER DICTIONARY 文による SDB データベースの定義変更」を参照してください。4V AFM の SDB データベースの場合は「11.6.1(4)　*ALTER DICTIONARY 文による SDB データベースの定義変更」を参照してください。

注意事項

SD FMB の SDB データベースの定義変更は，「2．SDB データベースを再定義する方法」で実施してください。「1．SDB データベースの定義変更をする方法」は実施できません。

(1)　SDB データベースの定義変更をする方法

*ALTER DICTIONARY 文および*ALTER DIRECTORY 文で，SDB データベースの定義変更ができます。手順を次に示します。

手順

1. データベース定義を作成する

   定義変更する SDB データベースの SDB データベース定義および SDB データベース格納定義を作成します。

   なお，SDB データベース定義および SDB データベース格納定義を作成する際は，変更個所だけを指定するのではなく，変更後の SDB データベース定義および SDB データベース格納定義をすべて指定する必要があります。

   SDB データベース定義および SDB データベース格納定義については，次の個所を参照してください。

   - 4V FMB の SDB データベースの定義を変更する場合

     「11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】」
   - 4V AFM の SDB データベースの定義を変更する場合

     「11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】」

2. レコードをアンロードする

格納 RD エリアの変更をする場合は，pdsdbrog コマンドで，変更対象の RD エリアに格納されたレコードをアンロードします。

3. SDB ディクショナリ情報および SDB ディレクトリ情報を変更する

pdsdbdef コマンドで*ALTER DICTIONARY 文および*ALTER DIRECTORY 文を実行し，ディクショナリ表に登録されている SDB ディクショナリ情報および SDB ディレクトリ情報を変更します。1.で作成した SDB データベース定義および SDB データベース格納定義を*ALTER DICTIONARY 文に指定します。

SDB 定義文の参照先を次に示します。

- *ALTER DICTIONARY 文：「11.4.3　*ALTER DICTIONARY 文（SDB ディクショナリ情報の変更）【4V FMB，4V AFM】」
- *ALTER DIRECTORY 文：「11.4.7　*ALTER DIRECTORY 文（SDB ディレクトリ情報の変更）【4V FMB，4V AFM】」

注意事項

*ALTER DICTIONARY 文および*ALTER DIRECTORY 文は，1 回の pdsdbdef コマンドで実行してください。

理由：

*ALTER DIRECTORY 文の実行時にはデータベース定義のチェックが実行されます。

*ALTER DICTIONARY 文および*ALTER DIRECTORY 文を 1 回の pdsdbdef コマンドで実行した場合，データベース定義の不整合が検知されると，SDB ディクショナリ情報および SDB ディレクトリ情報はディクショナリ表に追加されません。この場合，データベース定義を修正して，pdsdbdef コマンドを再実行することで対処できます。

*ALTER DICTIONARY 文と*ALTER DIRECTORY 文を別々に実行した場合，*ALTER DIRECTORY 文の実行時には，すでに SDB ディクショナリ情報はディクショナリ表に追加されています。そのため，データベース定義の不整合が検知された場合に，データベース定義を修正するには，*DELETE DICTIONARY 文を実行して SDB ディクショナリ情報を削除してから，pdsdbdef コマンドを再実行することになります。

1 回の pdsdbdef コマンドで，*ALTER DICTIONARY 文で追加または削除できる RD エリア数は 15 個までとなります。

4. レコードをデータロードする

格納 RD エリアを変更した場合は，変更した RD エリアに対して pdsdblod コマンドを実行して，2.でアンロードしたレコードをデータロードします。

■SDB データベースの定義変更を誤った場合の対処

名称の誤りなど，SDB データベースの定義変更を誤った場合，次に示す手順で対処してください。

1. pdsdbrog コマンドで，レコードをアンロードします。

2. pdsdbdef コマンドで，定義変更を誤った SDB データベースの定義削除を行います。
3. pdsdbdef コマンドで，SDB データベースの定義追加を行います。
4. pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して，全ユニットに配布します。
5. HiRDB を再起動します。
6. pdsdblod コマンドを使用して，1.でアンロードしたレコードをデータロードします。分割した RD エリアを元に戻した場合は，追加データロードを実行してください。

詳細な手順については，「5.6.3　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）」を参照してください。

参考

上記手順の 2.～4.は，1 回の pdsdbdef コマンドで実行できます。

(2)　SDB データベースを再定義する方法

*DELETE DICTIONARY 文および*DELETE DIRECTORY 文で SDB データベースの定義をいったん削除し，*ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文で SDB データベースを再定義することで，SDB データベースの定義変更を行います。手順を次に示します。

手順

1. データベース定義を作成する

   定義変更する SDB データベースの SDB データベース定義および SDB データベース格納定義を作成します。

2. SDB ディクショナリ情報および SDB ディレクトリ情報をいったん削除し，追加し直す

   SDB ディクショナリ情報および SDB ディレクトリ情報を，*DELETE DICTIONARY 文と*DELETE DIRECTORY 文でいったん削除し，*ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文で再度追加します。1.で作成した SDB データベース定義および SDB データベース格納定義を*ENTRY DICTIONARY 文に指定します。

   なお，*DELETE DICTIONARY 文および*DELETE DIRECTORY 文を実行するために pdsdbdef コマンドを実行する際，-i オプションを指定する必要があります。-i オプションを指定しないと，pdsdbdef コマンドの実行がエラーになります。

注意事項

SDB ディレクトリ情報および SDB ディクショナリ情報を削除する際は，必ず*DELETE DIRECTORY 文，*DELETE DICTIONARY 文の順に SDB 定義文を指定してください。

SDB 定義文の参照先を次に示します。

- *DELETE DICTIONARY 文：「11.4.4　*DELETE DICTIONARY 文（SDB ディクショナリ情報の削除）」
- *DELETE DIRECTORY 文：「11.4.8　*DELETE DIRECTORY 文（SDB ディレクトリ情報の削除）」
- *ENTRY DICTIONARY 文：「11.4.2　*ENTRY DICTIONARY 文（SDB ディクショナリ情報の追加）」
- *ENTRY DIRECTORY 文：「11.4.6　*ENTRY DIRECTORY 文（SDB ディレクトリ情報の追加）」

5.6.6　SDB データベースの定義削除

pdsdbdef コマンドで*DELETE DICTIONARY 文および*DELETE DIRECTORY 文を実行し，SDB データベースの SDB ディクショナリ情報および SDB ディレクトリ情報をディクショナリ表から削除します。*DELETE DICTIONARY 文の実行時には，SDB データベースのレコードおよびインデクスも削除されます。

なお，*DELETE DICTIONARY 文および*DELETE DIRECTORY 文を実行するために pdsdbdef コマンドを実行する際，-i オプションを指定する必要があります。-i オプションを指定しないと，pdsdbdef コマンドの実行がエラーになります。

注意事項

SDB ディレクトリ情報および SDB ディクショナリ情報を削除する際は，必ず*DELETE DIRECTORY 文，*DELETE DICTIONARY 文の順に SDB 定義文を指定してください。

SDB 定義文の参照先を次に示します。

- *DELETE DICTIONARY 文：「11.4.4　*DELETE DICTIONARY 文（SDB ディクショナリ情報の削除）」
- *DELETE DIRECTORY 文：「11.4.8　*DELETE DIRECTORY 文（SDB ディレクトリ情報の削除）」

5.7　SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合）

ここでは，HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更の手順について説明します。

なお，SDB データベースの定義削除をする場合，または分割格納条件の変更機能を使って格納 RD エリアの削除や，分割条件を変更して格納 RD エリアの分割，統合をする場合は，定義削除または定義変更後に HiRDB の再起動を必要とするため，「5.6　SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合）」で説明している手順で実施してください。

注意事項

SD FMB の SDB データベースの場合，HiRDB の再起動を必要としない SDB データベースの定義変更はできません。

前提条件

システム共通定義の pd_structured_advance_resident オペランドに use を指定して，事前常駐領域を使用できるようにしている必要があります。

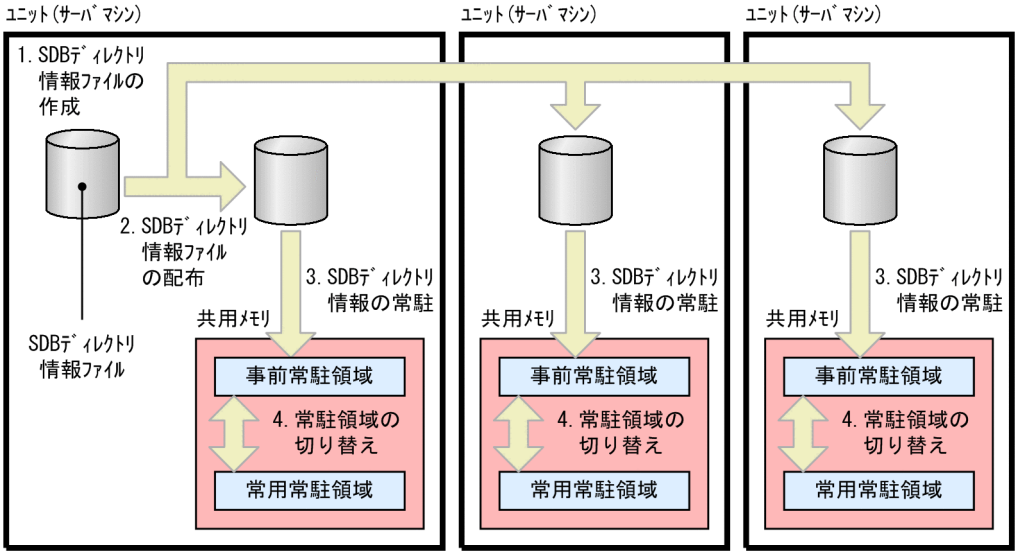
事前準備

- pd_structured_shmpool_dicsize オペランドで指定した SDB ディレクトリ情報の常駐領域サイズに余裕があるか確認してください。SDB データベースの定義追加または定義変更後に作成される SDB ディレクトリ情報の大きさが，pd_structured_shmpool_dicsize オペランドの指定値を超えた場合，SDB ディレクトリ情報を常駐化できません。現在の SDB ディレクトリ情報を作成したときに出力された KFPB61231-I メッセージに，SDB ディレクトリ情報長が出力されています。
- SDB データベースの定義追加をする場合は，SDB データベースを格納する RD エリアを pdmod コマンドで作成しておいてください。また，pdmod コマンドの実行時，create rdarea 文の globalbuffer オペランドを指定して，グローバルバッファを割り当ててください。
- SDB データベースの定義変更で，追加する子レコード型の二次インデクスを新規の RD エリアに格納する場合，および分割格納条件の変更機能を使って RD エリアを追加する場合は，pdmod コマンドで RD エリアを作成しておいてください。また，pdmod コマンドの実行時，create rdarea 文の globalbuffer オペランドでグローバルバッファを割り当ててください。

5.7.1　SDB ディレクトリ情報の変更の流れ

SDB データベースの定義追加または定義変更したあとに，共用メモリに常駐している SDB ディレクトリ情報を変更する必要があります。共用メモリに常駐している SDB ディレクトリ情報を変更する際の流れを次の図に示します。

図 5-3　共用メモリに常駐している SDB ディレクトリ情報を変更する際の流れ



[説明]

1. pdsdbdef コマンドを実行して SDB データベースの定義追加または定義変更したあとに，SDB ディレクトリ情報ファイルを作成します。
2. 1.で作成した SDB ディレクトリ情報ファイルを各ユニットに配布します。
3. pdsdbarc -e コマンドを実行して，SDB ディレクトリ情報を事前常駐領域に常駐します。
4. pdsdbarc -w -q コマンドを実行して，事前常駐領域を常用常駐領域に切り替えます。

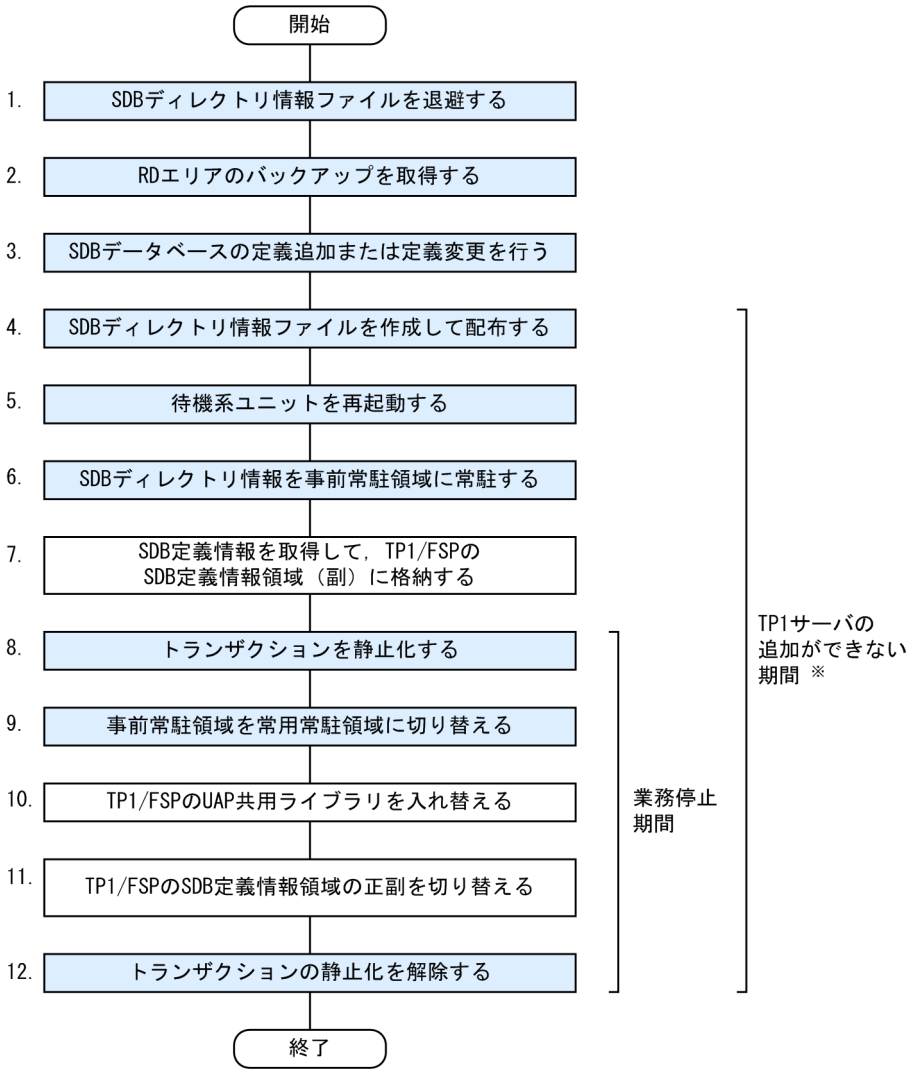
上記の流れで，共用メモリに常駐している SDB ディレクトリ情報を変更します。

なお，1.と 2.は，1 回の pdsdbdef コマンドで実行できます。

5.7.2　SDB データベースの定義追加または定義変更の手順（概要）

SDB データベースの定義追加または定義変更する手順（概要）を次の図に示します。

図 5-4　SDB データベースの定義追加または定義変更する手順（概要）



注※

4V FMB または 4V AFM の SDB データベースの定義追加または定義変更をする場合，この期間は TP1 サーバの追加ができません。

注意事項

- 4V FMB または 4V AFM の SDB データベースの定義追加または定義変更をする場合は，上記の 1.～12.の操作を実行してください。
- SD FMB の SDB データベースの定義追加をする場合は，上記の図の色の付いている個所（1.～6.，8.～9.，12.）の操作を実行してください。

図中の項番は，「5.7.3　SDB データベースの定義追加または定義変更の手順（各手順の詳細）」の（　）の番号と対応しています。

■上記の手順の途中で障害が発生した場合

次に示すどちらかの方法で対処してください。

- SDB データベースの定義追加または定義変更後の状態にする場合
  障害発生時に実行していた手順を再実行し，以降の手順を続行してください。
- SDB データベースの定義追加または定義変更前の状態に戻す場合
  「5.7.7　障害発生時の対処（定義追加または定義変更前の状態に戻す場合）」を参照してください。

■留意事項

HiRDB の再開始処理が完了しないと，SDB データベースの定義追加または定義変更はできません。HiRDB の再開始処理直後に SDB データベースの定義追加または定義変更をする場合は，次に示すどちらかの方法で HiRDB の再開始処理が完了していることを確認してください。

- KFPS02183-I メッセージ（シンクポイントダンプを取得した際に出力されるメッセージ）が出力されている。
- pdls -d trn コマンドを実行して，回復処理中のトランザクションがないことを確認する。

5.7.3　SDB データベースの定義追加または定義変更の手順（各手順の詳細）

SDB データベースの定義追加または定義変更をする際の，各手順の詳細について説明します。

注意事項

【4V FMB，4V AFM】の表記がある個所の説明は，4V FMB または 4V AFM の SDB データベースの定義追加または定義変更をする際に実行する操作です。SD FMB の SDB データベースの定義追加をする際は，実行する必要がない操作です。

(1)　SDB ディレクトリ情報ファイルを退避する

SDB データベースの定義追加または定義変更の作業中に障害が発生した際に備えて，作業前の状態に戻すために，SDB ディレクトリ情報ファイルを退避しておいてください。

(2)　RD エリアのバックアップを取得する

pdcopy コマンドで，次に示す RD エリアのバックアップを取得してください。

- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- 処理対象の SDB データベースを格納しているユーザ用 RD エリア（レコードおよびインデクスを格納しているユーザ用 RD エリア）

バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。

(3) SDB データベースの定義追加または定義変更を行う

- SDB データベースの定義追加をする場合
  「5.7.4 SDB データベースの定義追加」を参照してください。
- SDB データベースの定義変更をする場合
  「5.7.5 SDB データベースの定義変更」を参照してください。

(4) SDB ディレクトリ情報ファイルを作成して配布する

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して全ユニット（高速系切り替え機能を使用している場合は待機系ユニットも含む）に配布してください。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1 SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

pdsdbdef コマンドを実行して SDB ディレクトリ情報ファイルを作成すると，KFPB61231-I メッセージが出力されます。このメッセージに，SDB ディレクトリ情報の長さが表示されます。

（例）

```
KFPB61231-I SDB directory information file assigned, file="/home/user01/pdsdb/pdsdbdir",
num=48, size=577208, time stamp=2012120615420155(2012120615183822)
```

出力内容の説明：

- file：SDB ディレクトリ情報ファイル名
- num：SDB ディレクトリ情報の数
- size：SDB ディレクトリ情報の合計長（単位：バイト）
- time stamp：SDB ディレクトリ情報の最終更新日時
  （ ）内は SDB 定義文の最終更新日時

size に SDB ディレクトリ情報の長さ（単位：バイト）が表示されます。size に表示された値が，pd_structured_shmpool_dicsize オペランド（SDB ディレクトリ情報の常駐領域サイズ）の指定値より大きい場合，SDB ディレクトリ情報を事前常駐領域に常駐できません。この場合，HiRDB を一度停止して pd_structured_shmpool_dicsize オペランドの指定値を大きくする必要があります。

(5) 待機系ユニットを再起動する

高速系切り替え機能を使用している場合は，HA モニタの monsbystp コマンドですべての待機系ユニットをいったん終了し，pdstart -q コマンドですべての待機系ユニットを開始してください。

待機系ユニットを再起動することによって，SDB ディレクトリ情報が常用常駐領域に常駐されます。

待機系ユニットが開始したあと，すべての待機系ユニットで，常用常駐領域に常駐した SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9.2 常用常駐領域の SDB ディレクトリ情報の確認方法（待機系ユニットの場合）」を参照してください。

参考

4V FMB または 4V AFM の SDB データベースの定義追加または定義変更をする場合に，実行系と待機系の全ユニットで SDB ディレクトリ情報を一致させる理由を次に示します。

- 実行系と待機系のユニットのうち，SDB ディレクトリ情報が一致しないユニットが 1 つでもあると，新規に追加した TP1 サーバが一致していない SDB ディレクトリ情報を基に SDB 定義情報領域を作成してしまうおそれがあります。その結果，SDB 定義情報領域中の情報と UAP 共用ライブラリが不整合な状態になり，SDB データベースを操作する API の実行がエラーになります。

(6) SDB ディレクトリ情報を事前常駐領域に常駐する

pdsdbarc -e コマンドで，SDB ディレクトリ情報を事前常駐領域に常駐します。

pdsdbarc -e コマンドの実行時，HiRDB/SD は，事前常駐領域に常駐した SDB ディレクトリ情報の最終更新日時などをチェックします。チェックの結果，最終更新日時の不整合を検知した場合，不整合を検知した SDB ディレクトリ情報を無効状態にします。無効状態の SDB ディレクトリ情報は，(8)の操作で事前常駐領域を常用常駐領域に切り替える際，切り替えの対象外となります。切り替えの対象外となった SDB ディレクトリ情報を使用しているユニット（フロントエンドサーバまたはバックエンドサーバ）を経由する API または DML はエラーになります。そのため，最終更新日時の不整合が検知された場合，出力されたメッセージに従って対処し，事前常駐領域の SDB ディレクトリ情報を正常な状態にしてください。

(7) SDB 定義情報を取得して，TP1/FSP の SDB 定義情報領域（副）に格納する【4V FMB，4V AFM】

TP1/FSP の eesdhchg -b コマンドで SDB 定義情報を取得し，SDB 定義情報領域（副）に格納します（事前常駐領域中の SDB ディレクトリ情報が入力情報になります）。

SDB 定義情報を取得する方法については，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「SDB 定義情報の事前取得」を参照してください。

(8) トランザクションを静止化する

- 4V FMB または 4V AFM の SDB データベースの場合

  TP1/FSP の eesvctl -s コマンドで，トランザクションを静止化します。eesvctl -s コマンドが正常終了して，完全にトランザクションが静止化したことを確認してから次の手順に進んでください。

  注意事項

  - 完全にトランザクションが静止化していない状態で次の手順を行った場合，データベース破壊や UAP の異常終了などが発生するおそれがあります。そのため，eesvctl -s コマンドが正常終了するまで，eesvctl -s を実行し続けてください。

- 「(9)　事前常駐領域を常用常駐領域に切り替える」の操作が完了するまでは，pdsdbdef，pdsdblod，pdsdbrog，および pdsdbexe コマンドを実行しないでください。

- SD FMB の SDB データベースの場合

  サービスを閉塞して，トランザクションを静止化します。サービスを閉塞するには，OpenTP1 のスケジュール閉塞コマンドを実行します。詳細については，マニュアル「OpenTP1 Version 7　分散トランザクション処理機能　OpenTP1 運用と操作」を参照してください。

注意事項

- 完全にトランザクションが静止化していない状態で次の手順を行った場合，データベース破壊や UAP の異常終了などが発生するおそれがあります。
- 「(9)　事前常駐領域を常用常駐領域に切り替える」の操作が完了するまでは，pdsdbdef，pdsdblod，および pdsdbrog コマンドを実行しないでください。

(9)　事前常駐領域を常用常駐領域に切り替える

pdsdbarc -w -q コマンドで，事前常駐領域を常用常駐領域に切り替えます。pdsdbarc -w -q コマンドはユニットごとに実行します。この操作は全ユニットで実行してください。

参考

pdsdbarc -w -q コマンドは実行系ユニットに対してだけ実行できます。待機系ユニットについては，「(5)　待機系ユニットを再起動する」の操作で，常用常駐領域に SDB ディレクトリ情報が常駐されます。

事前常駐領域中の SDB ディレクトリ情報が無効状態の場合，事前常駐領域を常用常駐領域に切り替えることはできません。切り替えられなかった SDB ディレクトリ情報を使用しているユニット（フロントエンドサーバまたはバックエンドサーバ）を経由する API または DML はエラーになるため，無効状態の SDB ディレクトリ情報を次に示すどちらかの方法で正常な状態にしてください。

- 正しい SDB ディレクトリ情報ファイルが配布済みの場合

  事前常駐領域中の SDB ディレクトリ情報が無効状態のユニットに対して，「(6)　SDB ディレクトリ情報を事前常駐領域に常駐する」から作業を再実行してください。

- 正しい SDB ディレクトリ情報ファイルが未配布の場合

  事前常駐領域中の SDB ディレクトリ情報が無効状態のユニットに対して，pdsdbdef コマンドで SDB ディレクトリ情報ファイルを配布し直してください。そのあとで，「(6)　SDB ディレクトリ情報を事前常駐領域に常駐する」から作業を再実行してください。

(10)　TP1/FSP の UAP 共用ライブラリを入れ替える【4V FMB，4V AFM】

TP1/FSP の eechglib コマンドで，TP1/FSP の UAP 共用ライブラリを入れ替えます。

(11)　TP1/FSP の SDB 定義情報領域の正副を切り替える【4V FMB，4V AFM】

TP1/FSP の eesdhchg -c コマンドで，SDB 定義情報領域の正副を切り替えます。この操作によって，定義追加または定義変更した SDB データベースに対して，UAP からアクセスできるようになります。

(12)　トランザクションの静止化を解除する

- 4V FMB または 4V AFM の SDB データベースの場合
  TP1/FSP の eesvctl -r コマンドで，トランザクションの静止化を解除します。
- SD FMB の SDB データベースの場合
  サービスの閉塞を解除して，トランザクションの静止化を解除します。サービスの閉塞を解除するには，OpenTP1 のスケジュール閉塞解除コマンドを実行します。詳細については，マニュアル「OpenTP1 Version 7　分散トランザクション処理機能　OpenTP1 運用と操作」を参照してください。

(12)の作業が完了すると，定義追加または定義変更した SDB データベースにアクセスできるようになります。

5.7.4　SDB データベースの定義追加

SDB データベースの定義追加の手順を次に示します。

(1)　データベース定義を作成する

定義追加する SDB データベースの SDB データベース定義および SDB データベース格納定義を作成します。SDB データベース定義および SDB データベース格納定義については，次の個所を参照してください。

- 4V FMB の SDB データベースを定義する場合
  「11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】」
- 4V AFM の SDB データベースを定義する場合
  「11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】」
- SD FMB の SDB データベースを定義する場合
  「11.7　SDB データベース定義，SDB データベース格納定義【SD FMB】」

(2)　SDB ディクショナリ情報を追加する

pdsdbdef コマンドで*ENTRY DICTIONARY 文を実行し，定義追加する SDB データベースの SDB ディクショナリ情報をディクショナリ表に追加します。*ENTRY DICTIONARY 文には，(1)で作成した SDB データベース定義および SDB データベース格納定義を指定します。

注意事項

- pdsdbdef コマンドを実行する際，SDB 定義文に*CHECK DICTIONARY 文（データベース定義のチェック）も必ずあわせて指定してください。

  *ENTRY DICTIONARY 文と*CHECK DICTIONARY 文を別々に実行した場合，*CHECK DICTIONARY 文の実行時にデータベース定義の不整合が検知されると，不整合が検知された SDB ディクショナリ情報を*DELETE DICTIONARY 文で削除する必要が生じます。しかし，*DELETE DICTIONARY 文を実行した場合，HiRDB を再起動する必要があるため，*ENTRY DICTIONARY 文とあわせて*CHECK DICTIONARY 文を必ず指定するようにしてください。
- pdsdbdef コマンドを実行する際，-i オプションは指定しないでください。-i オプションを指定すると，HiRDB の再起動が必要になります。

*CHECK DICTIONARY 文を指定した場合，データベース定義の指定内容に不整合があると，エラーメッセージが出力されます。また，このとき SDB ディクショナリ情報はディクショナリ表に追加されません。エラーがなくなるまで，(2)の作業を繰り返してください。

なお，1 回の pdsdbdef コマンドで，複数の SDB データベースの SDB ディクショナリ情報を追加できます。

SDB 定義文の参照先を次に示します。

- *ENTRY DICTIONARY 文：「11.4.2　*ENTRY DICTIONARY 文（SDB ディクショナリ情報の追加）」
- *CHECK DICTIONARY 文：「11.4.5　*CHECK DICTIONARY 文（データベース定義のチェック）」

(3)　SDB ディレクトリ情報を追加する

pdsdbdef コマンドで*ENTRY DIRECTORY 文を実行し，SDB ディレクトリ情報を追加します。

注意事項

- (2)の作業で，複数の SDB データベースの SDB ディクショナリ情報を追加した場合，*ENTRY DIRECTORY 文による SDB ディレクトリ情報の追加は，1 回の pdsdbdef コマンドで実行してください。

  pdsdbdef コマンドを複数回実行した場合，何回目かの pdsdbdef コマンドでエラーが発生すると，一部の SDB ディレクトリ情報については作成が完了し，一部の SDB ディレクトリ情報については未作成の状態になります。この場合，未作成状態の SDB ディレクトリ情報を再度作成する必要があります。このため，*ENTRY DIRECTORY 文による SDB ディレクトリ情報の追加は，1 回で行う運用を推奨します。
- pdsdbdef コマンドを実行する際，-i オプションは指定しないでください。-i オプションを指定すると，HiRDB の再起動が必要になります。

*ENTRY DIRECTORY 文については，「11.4.6　*ENTRY DIRECTORY 文（SDB ディレクトリ情報の追加）」を参照してください。

5.7.5　SDB データベースの定義変更

SDB データベースの定義変更をする場合，pdsdbdef コマンドで*ALTER DICTIONARY 文および*ALTER DIRECTORY 文を実行します。

なお，*ALTER DICTIONARY 文および*ALTER DIRECTORY 文による SDB データベースの定義変更では，次の変更だけができます。

- 最下位のレコード型とその親子集合の追加
- 横分割したレコード型の格納 RD エリア追加

最下位のレコード型とその親子集合の追加の例については，4V FMB の SDB データベースの場合は「11.5.1(4)　*ALTER DICTIONARY 文による SDB データベースの定義変更」を参照してください。4V AFM の SDB データベースの場合は「11.6.1(4)　*ALTER DICTIONARY 文による SDB データベースの定義変更」を参照してください。

そのほかの定義変更をする場合は，いったん SDB データベースの定義削除をしたあとに定義変更を行い，再度 SDB データベースを定義追加するため，HiRDB を再起動する必要があります。この場合の手順については，「5.6.5　SDB データベースの定義変更」を参照してください。

注意事項

SD FMB の SDB データベースの場合，HiRDB の再起動を必要としない SDB データベースの定義変更はできません。「5.6.5(2)SDB データベースを再定義する方法」で説明している方法を実施してください。

SDB データベースの定義変更の手順を次に示します。

(1)　データベース定義を作成する

定義変更する SDB データベースの SDB データベース定義および SDB データベース格納定義を作成します。

なお，SDB データベース定義および SDB データベース格納定義を変更する際は，変更個所だけを指定するのではなく，変更後の SDB データベース定義および SDB データベース格納定義をすべて指定する必要があります。

SDB データベース定義および SDB データベース格納定義については，次の個所を参照してください。

- 4V FMB の SDB データベースの定義を変更する場合
  「11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】」

- 4V AFM の SDB データベースの定義を変更する場合
  「11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】」

(2)　SDB ディクショナリ情報を変更する

pdsdbdef コマンドで*ALTER DICTIONARY 文を実行し，SDB データベースの SDB ディクショナリ情報を変更します。*ALTER DICTIONARY 文には，(1)で作成した SDB データベース定義および SDB データベース格納定義を指定します。

注意事項

- pdsdbdef コマンドを実行する際，SDB 定義文に*CHECK DICTIONARY 文（データベース定義のチェック）も必ずあわせて指定してください。

  *ALTER DICTIONARY 文と*CHECK DICTIONARY 文を別々に実行した場合，*CHECK DICTIONARY 文の実行時にデータベース定義の不整合が検知されると，不整合が検知された SDB ディクショナリ情報を*DELETE DICTIONARY 文で削除する必要が生じます。しかし，*DELETE DICTIONARY 文を実行した場合，HiRDB を再起動する必要があるため，*ALTER DICTIONARY 文とあわせて*CHECK DICTIONARY 文を必ず指定するようにしてください。
- pdsdbdef コマンドを実行する際，-i オプションは指定しないでください。-i オプションを指定すると，HiRDB の再起動が必要になります。

*CHECK DICTIONARY 文を指定した場合，データベース定義の指定内容に不整合があると，エラーメッセージが出力されます。また，このとき SDB ディクショナリ情報はディクショナリ表に追加されません。エラーがなくなるまで，(2)の作業を繰り返してください。

SDB 定義文の参照先を次に示します。

- *ALTER DICTIONARY 文：「11.4.3　*ALTER DICTIONARY 文（SDB ディクショナリ情報の変更）【4V FMB，4V AFM】」
- *CHECK DICTIONARY 文：「11.4.5　*CHECK DICTIONARY 文（データベース定義のチェック）」

(3)　SDB ディレクトリ情報を変更する

pdsdbdef コマンドで*ALTER DIRECTORY 文を実行し，SDB ディレクトリ情報を変更します。

注意事項

- (2)の作業で，複数の SDB データベースの SDB ディクショナリ情報を変更した場合，*ALTER DIRECTORY 文による SDB ディレクトリ情報の変更は，1 回の pdsdbdef コマンドで実行してください。

  pdsdbdef コマンドを複数回実行した場合，何回目かの pdsdbdef コマンドでエラーが発生すると，一部の SDB ディレクトリ情報については作成が完了し，一部の SDB ディレクトリ情報に

ついては未作成の状態になります。この場合，未作成状態の SDB ディレクトリ情報を再度作成する必要があります。

一方，1 回の pdsdbdef コマンドで複数の SDB ディレクトリ情報を変更した場合，すべての SDB ディレクトリ情報の作成が完了するか，またはすべての SDB ディレクトリ情報が未作成になるかのどちらかになります。そのため，pdsdbdef コマンドを複数回実行したときに比べて対処が簡単になります。

- pdsdbdef コマンドを実行する際，-i オプションは指定しないでください。-i オプションを指定すると，HiRDB の再起動が必要になります。

*ALTER DIRECTORY 文については，「11.4.7　*ALTER DIRECTORY 文（SDB ディレクトリ情報の変更）【4V FMB，4V AFM】」を参照してください。

■SDB データベースの定義変更を誤った場合の対処

名称の誤りなど，SDB データベースの定義変更を誤った場合，次に示す手順で対処してください。

1. pdsdbrog コマンドで，レコードをアンロードします。
2. pdsdbdef コマンドで，定義変更を誤った SDB データベースの定義削除を行います。
3. pdsdbdef コマンドで，SDB データベースの定義追加を行います。
4. pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して，全ユニットに配布します。
5. HiRDB を再起動します。
6. pdsdblod コマンドを使用して，1.でアンロードしたレコードをデータロードします。

上記の作業の詳細な手順については，「5.6.3　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）」を参照してください。

参考

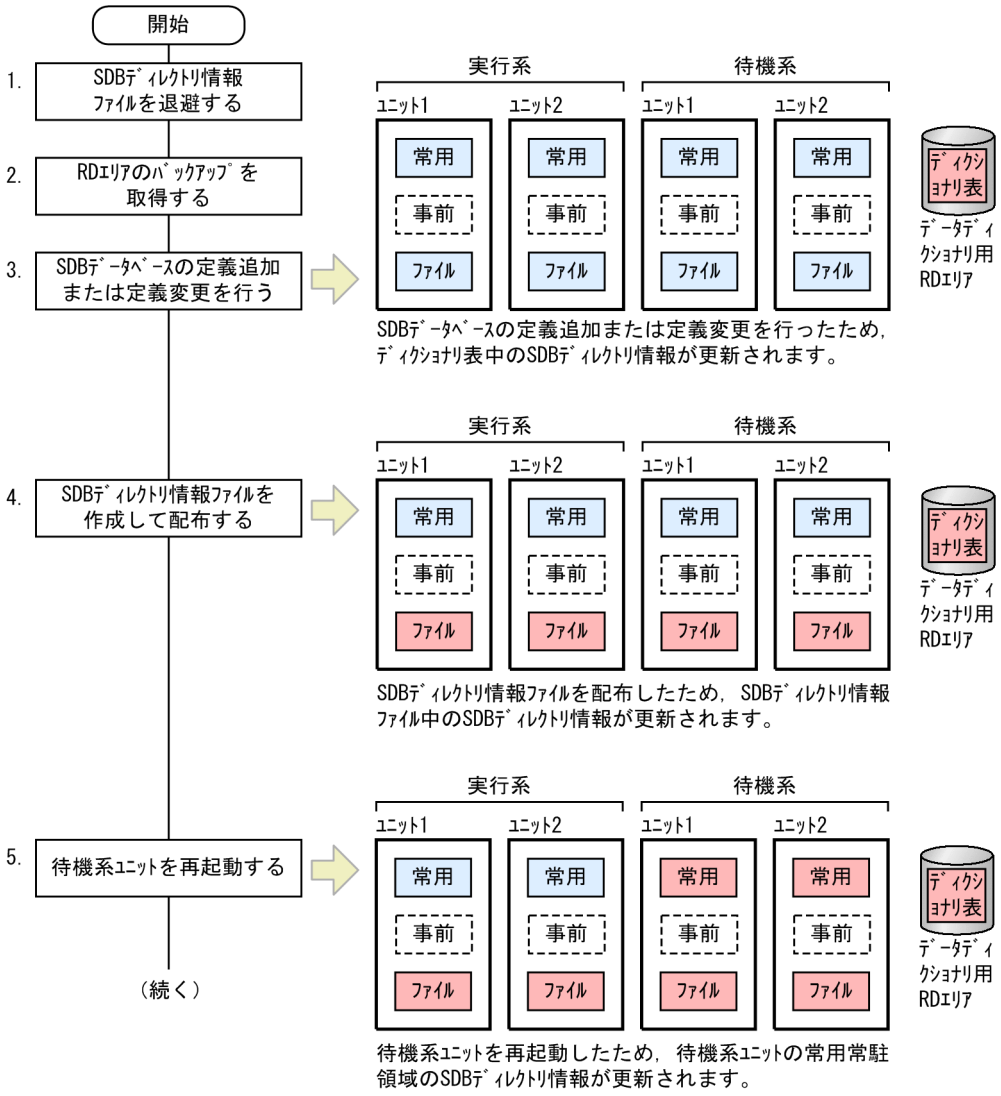
上記手順の 2.～4.は，1 回の pdsdbdef コマンドで実行できます。

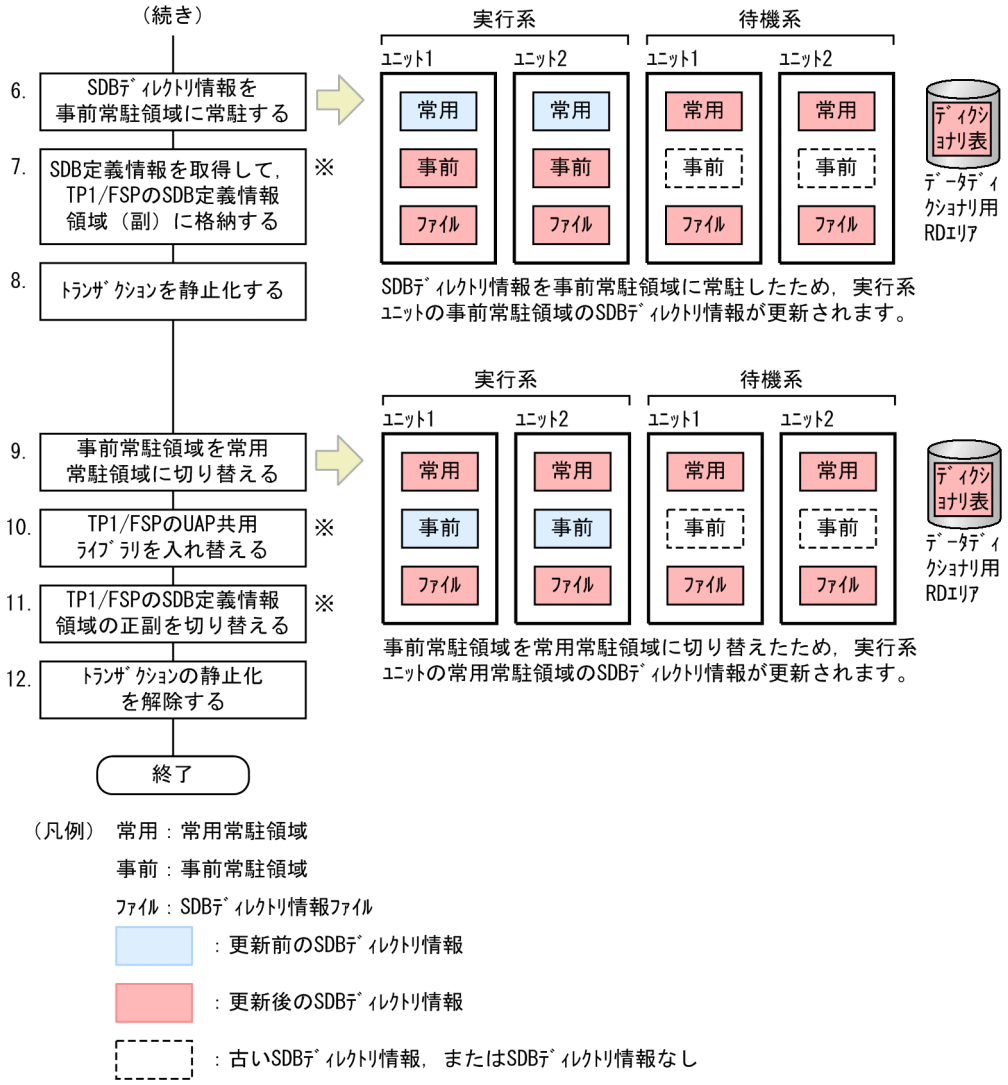
5.7.6　SDB ディレクトリ情報の状態遷移

SDB データベースの定義追加または定義変更時の SDB ディレクトリ情報の状態遷移の例を次の図に示します。

次の図のフローは，「5.7.2　SDB データベースの定義追加または定義変更の手順（概要）」の「図 5-4 SDB データベースの定義追加または定義変更する手順（概要）」に対応しています。

図 5-5　SDB ディレクトリ情報の状態遷移の例





注※

4V FMB または 4V AFM の SDB データベースの定義追加または定義変更をする場合に実行する操作です。

5.7.7　障害発生時の対処（定義追加または定義変更前の状態に戻す場合）

SDB データベースの定義追加または定義変更中に障害が発生し，定義追加または定義変更前の状態に戻す場合の手順を次の図に示します。

なお，障害の発生時点によって（「5.7.3　SDB データベースの定義追加または定義変更の手順（各手順の詳細）」の手順のどの時点で障害が発生したかによって），障害対処の手順の開始位置が異なります。

図 5-6　定義追加または定義変更前の状態に戻す場合の手順（4V FMB または 4V AFM の SDB データベースの場合）

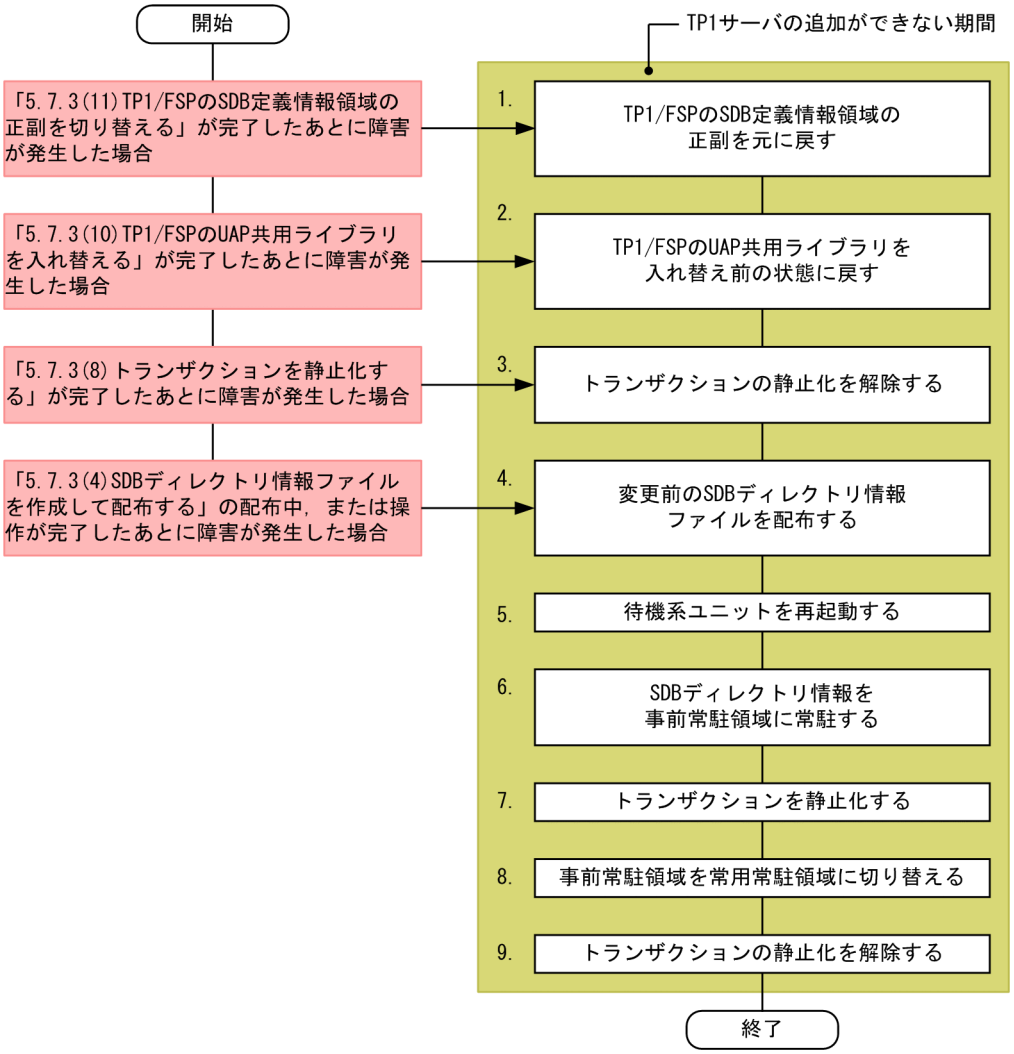
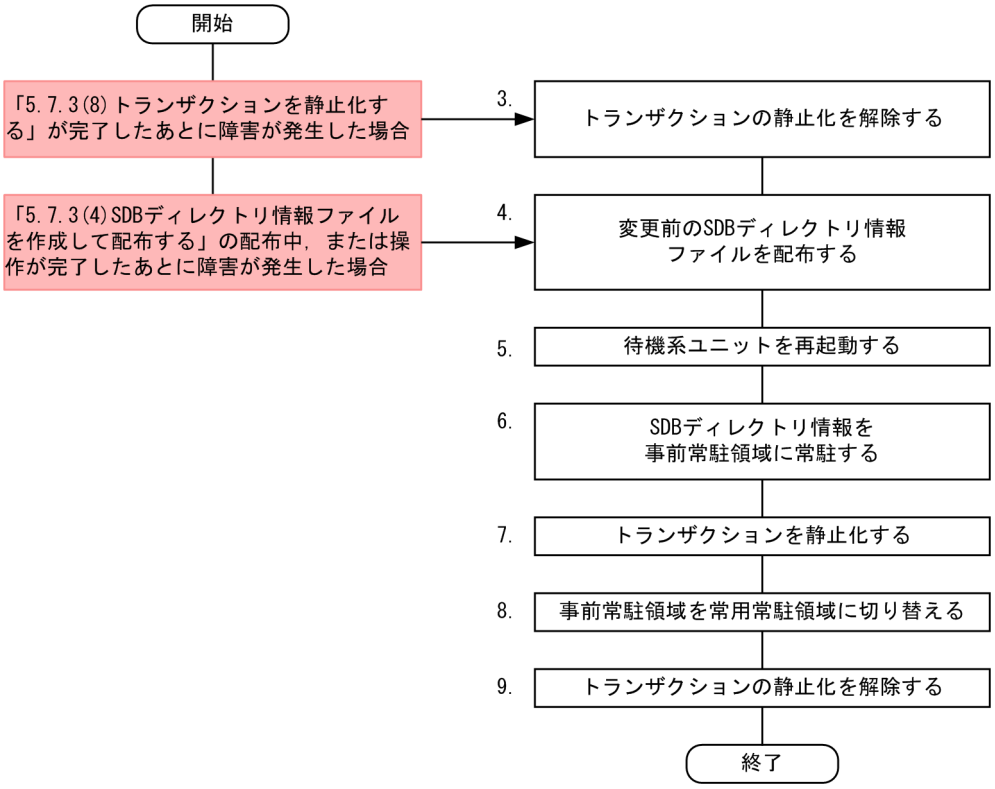


図 5-7　定義追加または定義変更前の状態に戻す場合の手順（SD FMB の SDB データベースの場合）



図中の項番は，以降の説明の（　）の番号と対応しています。

「5.7.3(4)　SDB ディレクトリ情報ファイルを作成して配布する」以前に障害が発生した場合は，UAP の実行環境は変更されていないため，上記の図に示す対処は必要ありません。

注意事項

(1)以降の説明で，【4V FMB，4V AFM】の表記がある個所の説明は，4V FMB または 4V AFM の SDB データベースの定義追加または定義変更をする際に実行する操作です。SD FMB の SDB データベースの定義追加をする際は，実行する必要がない操作です。

(1)　TP1/FSP の SDB 定義情報領域の正副を元に戻す【4V FMB，4V AFM】

TP1/FSP の eesdhchg -c コマンドで，SDB 定義情報領域の正副を切り替えて元の状態（定義追加または定義変更前の状態）に戻します。

(2)　TP1/FSP の UAP 共用ライブラリを入れ替え前の状態に戻す【4V FMB，4V AFM】

TP1/FSP の eechglib コマンドで，TP1/FSP の UAP 共用ライブラリを入れ替え前の状態（定義追加または定義変更前の状態）に戻します。

(3)　トランザクションの静止化を解除する

- 4V FMB または 4V AFM の SDB データベースの場合

  TP1/FSP の eesvctl -r コマンドで，トランザクションの静止化をいったん解除します。

- SD FMB の SDB データベースの場合

  サービスの閉塞を解除して，トランザクションの静止化をいったん解除します。サービスの閉塞を解除するには，OpenTP1 のスケジュール閉塞解除コマンドを実行します。詳細については，マニュアル「OpenTP1 Version 7　分散トランザクション処理機能　OpenTP1 運用と操作」を参照してください。

なお，(3)の操作は，障害対処作業の完了前にとりあえず業務を再開したい場合だけ実施してください。「(7)　トランザクションを静止化する」の操作の前までは，業務を実行できます。

(4)　変更前の SDB ディレクトリ情報ファイルを配布する

変更前の SDB ディレクトリ情報ファイルを，pdsdbdef コマンドで全ユニットに配布してください。

(5)　待機系ユニットを再起動する

高速系切り替え機能を使用している場合は，HA モニタの monsbystp コマンドですべての待機系ユニットをいったん終了し，pdstart -q コマンドですべての待機系ユニットを開始してください。

待機系ユニットを再起動することによって，SDB ディレクトリ情報が常用常駐領域に常駐されます。

(6)　SDB ディレクトリ情報を事前常駐領域に常駐する

pdsdbarc -e コマンドで，SDB ディレクトリ情報を事前常駐領域に常駐します。

(7)　トランザクションを静止化する

- 4V FMB または 4V AFM の SDB データベースの場合

  TP1/FSP の eesvctl -s コマンドで，トランザクションを静止化します。eesvctl -s コマンドが正常終了して，完全にトランザクションが静止化したことを確認してから次の手順に進んでください。

  (3)でトランザクションの静止化を解除した場合に，この操作を実行します。

  注意事項

  完全にトランザクションが静止化していない状態で次の手順を行った場合，データベース破壊や UAP の異常終了などが発生するおそれがあります。そのため，eesvctl -s コマンドが正常終了するまで，eesvctl -s を実行し続けてください。

- SD FMB の SDB データベースの場合

  サービスを閉塞して，トランザクションを静止化します。サービスを閉塞するには，OpenTP1 のスケジュール閉塞コマンドを実行します。詳細については，マニュアル「OpenTP1 Version 7　分散トランザクション処理機能　OpenTP1 運用と操作」を参照してください。

完全にトランザクションが静止化したことを確認してから次の手順に進んでください。

(3)でトランザクションの静止化を解除した場合に，この操作を実行します。

注意事項

完全にトランザクションが静止化していない状態で次の手順を行った場合，データベース破壊や UAP の異常終了などが発生するおそれがあります。

(8)　事前常駐領域を常用常駐領域に切り替える

pdsdbarc -w -q コマンドで，事前常駐領域を常用常駐領域に切り替えます。pdsdbarc -w -q コマンドはユニットごとに実行します。この操作は全ユニットで実行してください。

参考

pdsdbarc -w -q コマンドは実行系ユニットに対してだけ実行できます。待機系ユニットについては，「(5)　待機系ユニットを再起動する」の操作で，常用常駐領域に SDB ディレクトリ情報が常駐されます。

事前常駐領域中の SDB ディレクトリ情報が無効状態の場合，事前常駐領域を常用常駐領域に切り替えることはできません。切り替えられなかった SDB ディレクトリ情報を使用しているユニット（フロントエンドサーバまたはバックエンドサーバ）を経由する API または DML はエラーになるため，無効状態の SDB ディレクトリ情報を次に示すどちらかの方法で正常な状態にしてください。

- 正しい SDB ディレクトリ情報ファイルが配布済みの場合
  「(6)　SDB ディレクトリ情報を事前常駐領域に常駐する」から作業を再実行してください。
- 正しい SDB ディレクトリ情報ファイルが未配布の場合
  「(4)　変更前の SDB ディレクトリ情報ファイルを配布する」から作業を再実行してください。

(9)　トランザクションの静止化を解除する

- 4V FMB または 4V AFM の SDB データベースの場合
  TP1/FSP の eesvctl -r コマンドで，トランザクションの静止化を解除します。
- SD FMB の SDB データベースの場合
  サービスの閉塞を解除して，トランザクションの静止化を解除します。サービスの閉塞を解除するには，OpenTP1 のスケジュール閉塞解除コマンドを実行します。詳細については，マニュアル「OpenTP1 Version 7　分散トランザクション処理機能　OpenTP1 運用と操作」を参照してください。

(9)の操作が完了すると，定義追加または定義変更前の状態に戻ります。

注意事項

SDB ディクショナリ情報および SDB ディレクトリ情報は，ディクショナリ表に追加された状態のままとなっています。そのため，同じ SDB データベースの定義追加または定義変更を再度行う場合は，いったんディクショナリ表に追加されている SDB ディクショナリ情報および SDB ディレクトリ情報を pdsdbdef コマンドの*DELETE DICTIONARY 文および*DELETE DIRECTORY 文で削除する必要があります。

なお，SDB ディクショナリ情報および SDB ディレクトリ情報を削除した場合，削除後に HiRDB を再起動する必要があります。

5.8　SDB ディレクトリ情報ファイルの配布方法

SDB ディレクトリ情報ファイルの配布方法について説明します。

5.8.1　SDB ディレクトリ情報ファイルを作成して配布する場合

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して配布します。手順を例題形式で説明します。

例題

SDB ディレクトリ情報ファイルを作成して，全ユニットに SDB ディレクトリ情報ファイルを配布します。

(1)　pdsdbdef コマンドの SDB 制御文を作成する

```
dirinf /usr/sdb_dir          ...1
       generate=yes          ...2
       transfer=yes          ...3
       checkmode=match       ...4
```

SDB ディレクトリ情報ファイルを作成して配布する場合，SDB 制御文に dirinf 文を指定します。dirinf 文に指定した各オペランドについて説明します。

1. SDB ディレクトリ情報ファイルを出力するディレクトリ名を指定します。SDB ディレクトリ情報ファイルは，ここで指定したディレクトリ下（システムマネジャが定義されているサーバマシンのディレクトリ）に出力されます。
2. SDB ディレクトリ情報ファイルを作成するため，yes を指定します。
3. SDB ディレクトリ情報ファイルを全ユニットに配布するため，yes を指定します。全ユニットのユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下に，SDB ディレクトリ情報ファイルが配布されます。高速系切り替え機能を使用している場合は，待機系ユニットにも SDB ディレクトリ情報ファイルが自動的に配布されます。
4. 配布する SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報の最終更新日時のチェックレベルを指定します。

> **参考**
>
> SDB ディレクトリ情報ファイルの作成および配布だけをする場合は，SDB 制御文には dirinf 文だけを指定します。

(2) pdsdbdef コマンドを実行する

```
pdsdbdef /usr/control_def_file
```

(1)で作成した SDB 制御文ファイル名を pdsdbdef コマンドの引数に指定します。

注意事項

- SDB ディレクトリ情報ファイルのファイル名（pdsdbdir）は変更しないでください。
- ディザスタリカバリシステムのリモートサイトは配布対象外です。リモートサイトに SDB ディレクトリ情報ファイルを配布する方法については，「8.3.1(2) 「6. SDB ディレクトリ情報ファイルをリモートサイトに転送する」の作業」を参照してください。

5.8.2 SDB ディレクトリ情報ファイルの配布だけをする場合

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを配布します。手順を例題形式で説明します。ここでは，SDB ディレクトリ情報ファイルの作成はしないで，配布だけをする手順を説明します。

例題

全ユニットに SDB ディレクトリ情報ファイルを配布します。

(1) pdsdbdef コマンドの SDB 制御文を作成する

```
dirinf /usr/sdb_dir          ...1
       generate=no           ...2
       transfer=yes          ...3
       checkmode=match       ...4
```

SDB ディレクトリ情報ファイルを配布する場合，SDB 制御文に dirinf 文を指定します。dirinf 文に指定した各オペランドについて説明します。

1. 配布対象の SDB ディレクトリ情報ファイルが格納されているディレクトリ名を指定します。システムマネジャが定義されているサーバマシンのディレクトリを指定します。
2. SDB ディレクトリ情報ファイルは作成しないため，no を指定します。
3. SDB ディレクトリ情報ファイルを全ユニットに配布するため，yes を指定します。全ユニットのユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下に，SDB ディレクトリ情報ファイルが配布されます。高速系切り替え機能を使用している場合は，待機系ユニットにも SDB ディレクトリ情報ファイルが自動的に配布されます。
4. 配布する SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報の最終更新日時のチェックレベルを指定します。

参考

SDB ディレクトリ情報ファイルの配布だけをする場合は，SDB 制御文には dirinf 文だけを指定します。

(2) pdsdbdef コマンドを実行する

```
pdsdbdef /usr/control_def_file
```

(1)で作成した SDB 制御文ファイル名を pdsdbdef コマンドの引数に指定します。

注意事項

- SDB ディレクトリ情報ファイルのファイル名（pdsdbdir）は変更しないでください。
- ディザスタリカバリシステムのリモートサイトは配布対象外です。リモートサイトに SDB ディレクトリ情報ファイルを配布する方法については，「8.3.1(2) 「6. SDB ディレクトリ情報ファイルをリモートサイトに転送する」の作業」を参照してください。

5.8.3 特定のユニットに SDB ディレクトリ情報ファイルを配布する場合

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを配布します。手順を例題形式で説明します。ここでは，SDB ディレクトリ情報ファイルの作成はしないで，特定のユニットにだけ配布する手順を説明します。

例題

ユニット（ユニット名：UNT3）に SDB ディレクトリ情報ファイルを配布します。

(1) pdsdbdef コマンドの SDB 制御文を作成する

```
dirinf /usr/sdb_dir          ...1
       generate=no           ...2
       transfer=yes          ...3
       checkmode=match       ...4
       unit=UNT3             ...5
```

SDB ディレクトリ情報ファイルを配布する場合，SDB 制御文に dirinf 文を指定します。dirinf 文に指定した各オペランドについて説明します。

1. 配布対象の SDB ディレクトリ情報ファイルが格納されているディレクトリ名を指定します。システムマネジャが定義されているサーバマシンのディレクトリを指定します。
2. SDB ディレクトリ情報ファイルは作成しないため，no を指定します。

3. SDB ディレクトリ情報ファイルを配布するため，yes を指定します。5.で指定したユニットのユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下に，SDB ディレクトリ情報ファイルが配布されます。高速系切り替え機能を使用している場合は，待機系ユニットにも SDB ディレクトリ情報ファイルが自動的に配布されます。
4. 配布する SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報の最終更新日時のチェックレベルを指定します。
5. 配布対象のユニット名を指定します。

参考

SDB ディレクトリ情報ファイルの配布だけをする場合は，SDB 制御文には dirinf 文だけを指定します。

(2) pdsdbdef コマンドを実行する

```
pdsdbdef /usr/control_def_file
```

(1)で作成した SDB 制御文ファイル名を pdsdbdef コマンドの引数に指定します。

注意事項

- SDB ディレクトリ情報ファイルのファイル名（pdsdbdir）は変更しないでください。
- ディザスタリカバリシステムのリモートサイトは配布対象外です。リモートサイトに SDB ディレクトリ情報ファイルを配布する方法については，「8.3.1(2) 「6. SDB ディレクトリ情報ファイルをリモートサイトに転送する」の作業」を参照してください。

5.9　全ユニットの SDB ディレクトリ情報が一致しているか確認する方法

全ユニットの SDB ディレクトリ情報が一致しているかどうかを確認する方法を説明します。

高速系切り替え機能を使用していない場合は，「5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）」の方法で確認してください。

高速系切り替え機能を使用している場合は，「5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）」および「5.9.2　常用常駐領域の SDB ディレクトリ情報の確認方法（待機系ユニットの場合）」の両方の方法で確認してください。

5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）

HiRDB の全ユニット（ユニット識別子：unt1，unt2）で，常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認する手順を次に示します。

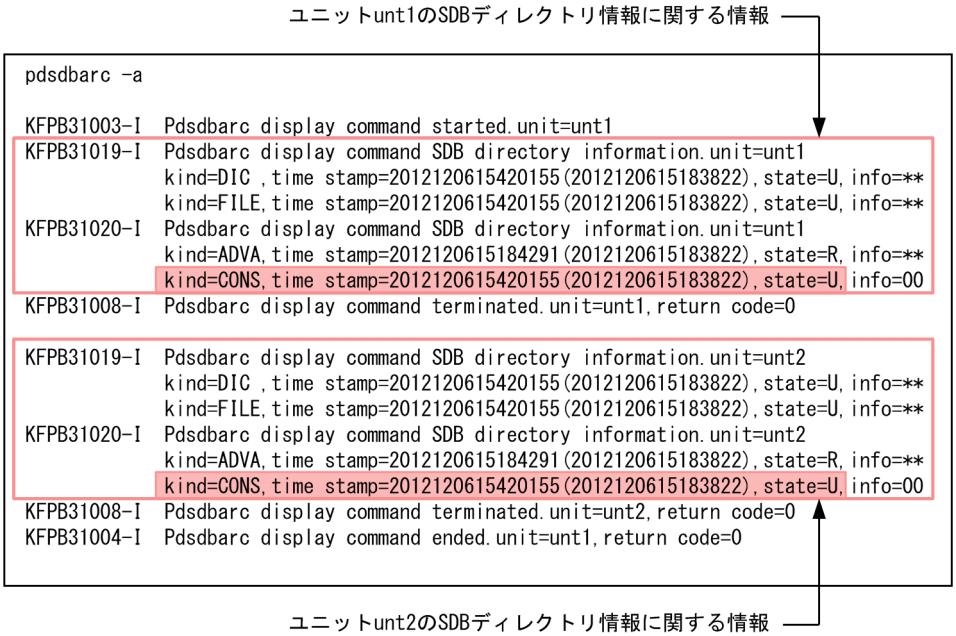
(1)　SDB ディレクトリ情報に関する情報を表示する

```
pdsdbarc -a
```

pdsdbarc -a コマンドを実行して，SDB ディレクトリ情報に関する情報を表示します。

(2)　全ユニットの SDB ディレクトリ情報が一致しているか確認する

pdsdbarc -a コマンドの実行結果を参照して，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。



[説明]

KFPB31019-I および KFPB31020-I メッセージに各ユニットの SDB ディレクトリ情報が表示されます。KFPB31020-I メッセージの kind=CONS の行に，常用常駐領域の SDB ディレクトリ情報に関する情報が表示されます。

- time stamp に表示されている SDB ディレクトリ情報および SDB 定義文の最終更新日時が，全ユニットで一致しているかどうかを確認してください。一致している場合は問題ありません。
- state に U（SDB ディレクトリ情報が有効な状態を意味している）が表示されているかを確認してください。全ユニットで U が表示されている場合は問題ありません。

ユニット間で SDB ディレクトリ情報が不一致の場合は，「5.9.3　常用常駐領域の SDB ディレクトリ情報の不一致が発生したときの対処」を参照して対処してください。

5.9.2　常用常駐領域の SDB ディレクトリ情報の確認方法（待機系ユニットの場合）

pdsdbarc -a コマンドの実行結果には，実行系ユニットの SDB ディレクトリ情報だけが表示されます。待機系ユニットの SDB ディレクトリ情報は表示されません。そのため，待機系ユニットの SDB ディレクトリ情報が一致しているかどうかは，出力されたメッセージで確認します。手順を次に示します。

(1)　KFPB62908-I メッセージの出力を確認する

待機系ユニットの開始処理時に KFPB62908-I メッセージ（SDB ディレクトリ情報が共用メモリの常用常駐領域に常駐化された旨のメッセージ）が出力されます。このメッセージが，全待機系ユニットで出力されていることを確認してください。

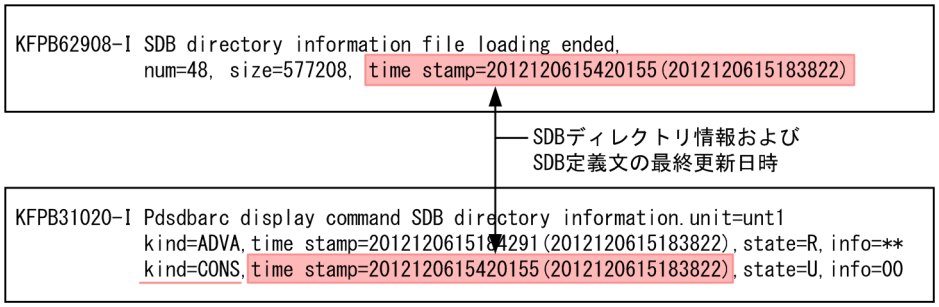
(2)　KFPB31020-I メッセージの出力を確認する

待機系ユニットの開始後，pdsdbarc -a コマンドを実行して，SDB ディレクトリ情報に関する情報を表示してください。実行結果に KFPB31020-I メッセージが出力されたことを確認します。

(3)　KFPB62908-I と KFPB31020-I メッセージの内容を確認する

これらの 2 つのメッセージに出力されている次に示す情報を確認します。すべてが同じ場合は，SDB ディレクトリ情報が一致しています。

- SDB ディレクトリ情報の最終更新日時
- SDB 定義文の最終更新日時



[説明]

- time stamp に表示されている SDB ディレクトリ情報および SDB 定義文の最終更新日時を確認します。（　）内が SDB 定義文の最終更新日時です。
- kind=CONS の行（常用常駐領域の情報）を確認してください。

全待機系ユニットについて確認してください。

SDB ディレクトリ情報が不一致の場合は，「5.9.3　常用常駐領域の SDB ディレクトリ情報の不一致が発生したときの対処」を参照して対処してください。

なお，実行系ユニットの開始後に実行した pdsdbarc -c コマンドの実行結果（KFPB31012-I メッセージ）にも，SDB ディレクトリ情報の最終更新日時と SDB 定義文の最終更新日時が表示されます。KFPB62908-I と KFPB31012-I メッセージでも，実行系ユニットと待機系ユニットの SDB ディレクトリ情報が一致していることを確認できます。

5.9.3　常用常駐領域の SDB ディレクトリ情報の不一致が発生したときの対処

常用常駐領域の SDB ディレクトリ情報の不一致が発生したときの対処方法について説明します。

(1) 実行系ユニットで不一致が発生した場合

次に示すどれかの原因によって，常用常駐領域の SDB ディレクトリ情報が不一致となったおそれがあります。

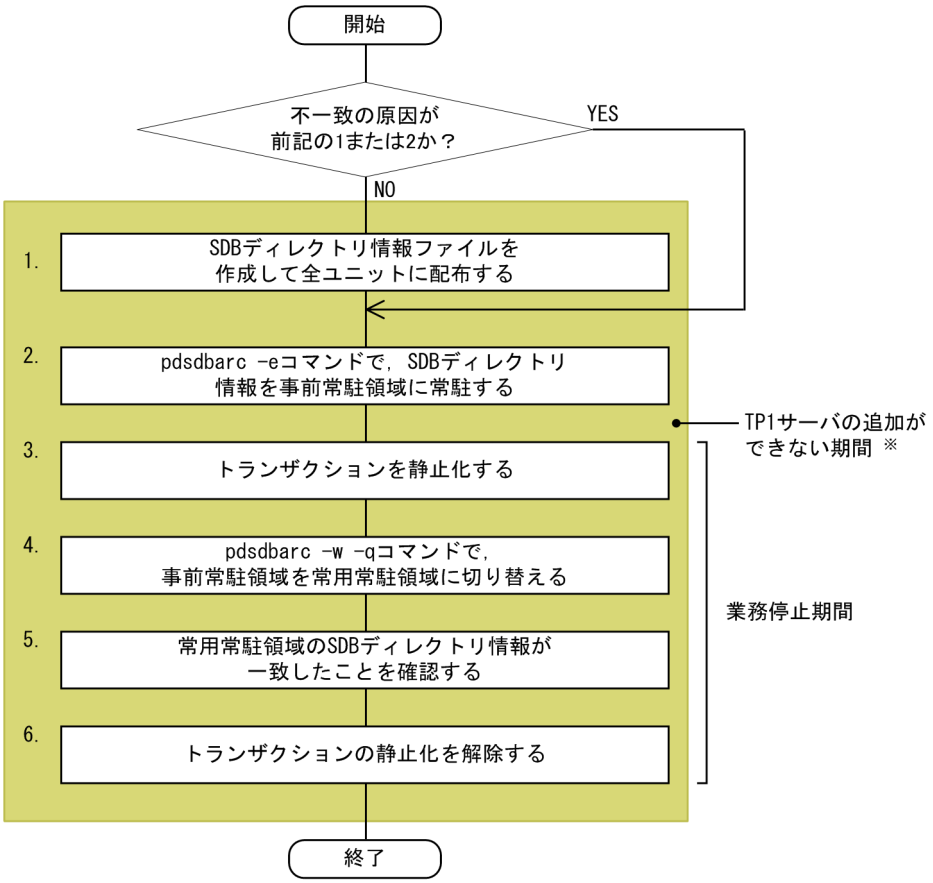
1. SDB ディレクトリ情報を事前常駐領域に常駐していない。
2. 事前常駐領域を常用常駐領域に切り替えていない。
3. 変更後の SDB ディレクトリ情報ファイルを配布していないユニットがある。
4. 配布した SDB ディレクトリ情報ファイルを pd_structured_directory_path オペランドで指定したディレクトリ下に格納していない。

pdsdbarc -a コマンドの実行結果として表示される，KFPB31019-I メッセージの kind=FILE の情報（SDB ディレクトリ情報ファイルに関する情報）を確認してください。

kind=FILE の情報がほかのユニットと同じ場合は，上記の 1.または 2.が原因として考えられます。
kind=FILE の情報がほかのユニットと異なる場合は，上記の 3.または 4.が原因として考えられます。

実行系ユニットで SDB ディレクトリ情報の不一致が発生したときの対処手順を次の図に示します。

図 5-8　実行系ユニットで SDB ディレクトリ情報の不一致が発生したときの対処手順



注　図中の項番は，次の手順の説明の番号と対応しています。

注※

4V FMB または 4V AFM の SDB データベースを使用している場合，この期間は TP1 サーバの追加ができません。

手順

1. pdsdbdef コマンドで SDB ディレクトリ情報ファイルを作成し，全ユニットに配布します。SDB ディレクトリ情報ファイルを作成済みの場合は，pdsdbdef コマンドで全ユニットへの配布だけを行ってください。
2. pdsdbarc -e コマンドで，SDB ディレクトリ情報を事前常駐領域に常駐化します。
3. トランザクションを静止化します。

   ・4V FMB または 4V AFM の SDB データベースの場合

   TP1/FSP の eesvctl -s コマンドで，トランザクションを静止化します。

   eesvctl -s コマンドが正常終了して，完全にトランザクションが静止化したことを確認してから次の手順に進んでください。

   注意事項

   完全にトランザクションが静止化していない状態で次の手順を行った場合，データベース破壊や UAP の異常終了などが発生するおそれがあります。そのため，eesvctl -s コマンドが正常終了するまで，eesvctl -s を実行し続けてください。

   ・SD FMB の SDB データベースの場合

   サービスを閉塞して，トランザクションを静止化します。サービスを閉塞するには，OpenTP1 のスケジュール閉塞コマンドを実行します。詳細については，マニュアル「OpenTP1 Version 7 分散トランザクション処理機能　OpenTP1 運用と操作」を参照してください。

   注意事項

   完全にトランザクションが静止化していない状態で次の手順を行った場合，データベース破壊や UAP の異常終了などが発生するおそれがあります。
4. pdsdbarc -w -q コマンドで，事前常駐領域を常用常駐領域に切り替えます。
5. 常用常駐領域の SDB ディレクトリ情報が一致したことを確認します。

   「5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）」に示す方法で確認してください。
6. トランザクションの静止化を解除します。

   ・4V FMB または 4V AFM の SDB データベースの場合

   TP1/FSP の eesvctl -r コマンドで，トランザクションの静止化を解除します。

   ・SD FMB の SDB データベースの場合

   サービスの閉塞を解除して，トランザクションの静止化を解除します。サービスの閉塞を解除するには，OpenTP1 のスケジュール閉塞解除コマンドを実行します。詳細については，マニュアル

「OpenTP1 Version 7　分散トランザクション処理機能　OpenTP1 運用と操作」を参照してください。

(2)　待機系ユニットで不一致が発生した場合

次に示すどちらかの原因によって，常用常駐領域の SDB ディレクトリ情報が不一致となったおそれがあります。

- 変更後の SDB ディレクトリ情報ファイルを待機系ユニットに配布していない。
- 配布した SDB ディレクトリ情報ファイルを待機系ユニットの pd_structured_directory_path オペランドで指定したディレクトリ下に格納していない。

待機系ユニットで SDB ディレクトリ情報の不一致が発生したときの対処手順を次に示します。

手順

1. pdsdbdef コマンドで，実行系ユニットと同じ SDB ディレクトリ情報ファイルを待機系ユニットに配布します。
2. 不一致が発生した待機系ユニットを再起動します。
3. 待機系ユニットの常用常駐領域の SDB ディレクトリ情報が一致したことを確認します。
   「5.9.2　常用常駐領域の SDB ディレクトリ情報の確認方法（待機系ユニットの場合）」に示す方法で確認してください。

5.10　HiRDB システム定義の変更

HiRDB システム定義を変更する場合は，いったん HiRDB サーバを終了する必要があります。HiRDB システム定義の変更方法については，マニュアル「HiRDB システム運用ガイド」の「HiRDB システム定義を変更する方法」を参照してください。

なお，SDB 用 UAP 環境定義を変更する場合は，HiRDB サーバを終了する必要はありません。「5.10.1 SDB 用 UAP 環境定義の変更方法【SD FMB】」に示す手順で SDB 用 UAP 環境定義を変更してください。

5.10.1　SDB 用 UAP 環境定義の変更方法【SD FMB】

SDB 用 UAP 環境定義の変更手順を次に示します。

手順

1. 変更対象の SDB 用 UAP 環境定義を使用する UAP が実行中でないことを確認します。
   UAP が実行中の場合は，UAP の実行が完了するのを待ってください。
2. SDB 用 UAP 環境定義を変更します。
   変更対象の SDB 用 UAP 環境定義ファイルが複数のサーバマシンにある場合，すべてのファイルを変更してください。

24 時間連続稼働するシステムで，0 バイトまたはコメントだけの SDB 用 UAP 環境定義ファイルを作成している場合も，上記の手順で SDB 用 UAP 環境定義を変更してください。

参考

SDB 用 UAP 環境定義を変更したあとに HiRDB サーバに接続した UAP から，変更後の SDB 用 UAP 環境定義が適用されます。

5.11　SDB データベースの再編成（インナレプリカ機能を使用しない場合）

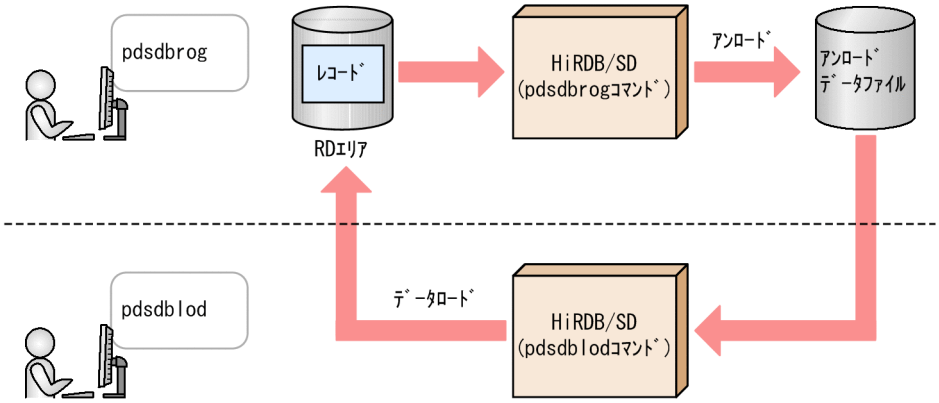
ここでは，インナレプリカ機能を使用しない SDB データベースの再編成について説明します。インナレプリカ機能を使用しない場合，データベースの再編成中に SDB データベースの参照および更新はできません。

インナレプリカ機能を使用した SDB データベースの再編成については，「5.12　SDB データベースの再編成（インナレプリカ機能を使用したオンライン再編成）【4V FMB，4V AFM】」または「5.13　SDB データベースの再編成（インナレプリカ機能を使用した更新可能なオンライン再編成）【4V FMB】」を参照してください。

5.11.1　SDB データベースの再編成の概要

レコードの追加や削除を繰り返すと，レコードの配置が乱れてレコードのアクセス効率や格納効率が低下します。このような場合，pdsdbrog コマンドでレコードをいったんアンロードしてアンロードデータファイルを作成します。そのアンロードデータファイルを入力データファイルにして pdsdblod コマンドでデータロードを行います。データを再格納することによって，レコードの配置が適正化され，レコードのアクセス効率や格納効率が改善されます。この一連の処理を SDB データベースの再編成といいます。SDB データベースの再編成の概要を次の図に示します。

図 5-9　SDB データベースの再編成の概要



なお，pdsdblod コマンドによるデータロードの際，インデクスも再編成されるため，インデクスの格納効率や，インデクスを使用した検索性能も改善されます。

SDB データベースの再編成の実行契機

データの格納効率が悪くなると，次に示すメッセージが出力されます。これらのメッセージが出力された場合が，SDB データベースの再編成の実行契機になります。

- KFPA12300-I
- KFPH00211-I
- KFPH00212-I

5.11.2　SDB データベースの再編成の手順

SDB データベースの再編成手順（pdsdbrog コマンドでアンロードしたレコードを同じ SDB データベースにデータロードする場合の手順）を次の図に示します。

図 5-10　SDB データベースの再編成手順

注意事項

- SDB データベースが横分割されている場合，アンロードとデータロードの実行単位を合わせてください。アンロードは RD エリア単位で実行するため，データロードもそれに合わせて RD エリア単位で実行してください。実行単位を合わせないと，SDB データベースが正しく格納されません。
- レコードをアンロードしてからデータロードするまでの間に，処理対象の SDB データベース定義および SDB データベース格納定義を変更しないでください。変更した場合，レコードが正しく格納されないことがあります。
- レコードをアンロードしてからデータロードするまでの間に，処理対象のデータベースを更新しないでください。データベースを更新した場合，更新したレコード実現値はデータロードの対象外になります。

各手順を以降で説明します。

(1)　全ユニットの SDB ディレクトリ情報が一致しているかどうかを確認する

```
pdsdbarc -a
```

pdsdbarc コマンドで SDB ディレクトリ情報に関する情報を表示し，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9　全ユニットの SDB ディレクトリ情報が一致しているか確認する方法」を参照してください。

確認した結果，SDB ディレクトリ情報が不一致の場合は，pdsdbrog コマンドを実行できません。

(2)　処理対象の RD エリアを閉塞する

```
pdhold -r RDエリア名,RDエリア名 …
```

pdhold コマンドで処理対象の RD エリアを閉塞します。

(3)　アンロードを実行する

```
pdsdbrog 制御文ファイル名
```

pdsdbrog コマンドでレコードをアンロードします。

(4)　プリフィクス部を設定する

プリフィクス部のページ切り替えフラグの指定，または事前割り当てページ数の指定値の変更が必要な場合は，アンロードデータファイルを直接修正して，ページ切り替えフラグまたは事前割り当てページ数を設定してください。ページ切り替えフラグを設定する場合は，「13.8　ページ切り替えフラグの扱い【4V FMB，4V AFM】」を参照してください。

(5)　データロードを実行する

```
pdsdblod 制御文ファイル名
```

pdsdblod コマンドで，(3)でアンロードしたレコードをデータロードします。

- インナレプリカ機能を使用している場合，アンロード対象 RD エリアとデータロード対象 RD エリアの世代番号が同じになるようにしてください。同じにするには，次に示すどちらかの方法でアンロードおよびデータロードを実行してください。
  - pdsdbrog 制御文および pdsdblod 制御文の environment 文の generation オペランドに指定する処理対象 RD エリアの世代番号を同じにする
  - アンロードしてからデータロードするまでの間に，カレント RD エリアを変更しないようにする（generation オペランドを指定しない場合）

  generation オペランドを指定しない場合，処理対象の RD エリアはカレント RD エリアになります。

注意事項

アンロード時とデータロード時で，カレント RD エリアを変更してしまった場合，アンロード対象 RD エリアと異なる RD エリアにデータロードが実行されてしまいます。

- pdsdblod 制御文の各オペランドには，次の表に示す値を指定してください。間違った値を指定すると，レコードが正しく格納されません。

表 5-3　pdsdblod 制御文の各オペランドに指定する値（pdsdbrog コマンドでアンロードしたレコードをデータロードする場合）

項番	オペランド		オペランドの指定値
1	environment 文	schema	pdsdbrog 制御文の environment 文の schema オペランドに指定した SDB データベース名を指定してください。
2		purge	yes を指定してください。
3		generation	pdsdbrog 制御文の environment 文の generation オペランドに指定した世代番号を指定してください。pdsdbrog 制御文の environment 文の generation オペランドを省略した場合は，このオペランドも省略してください。
4		上記以外のオペランド	任意の値を指定してください。
5	load 文	type	4V FMB または SD FMB の SDB データベースの場合は tree を，4V AFM の SDB データベースの場合は single を指定してください。
6		record	pdsdbrog 制御文の unload 文の record オペランドに指定したレコード型名を指定してください。
7		server	pdsdbrog 制御文の unload 文の server オペランドに指定したバックエンドサーバ名を指定してください。pdsdbrog 制御文の unload 文の server オペランドを省略した場合は，省略してください。
8		area	pdsdbrog 制御文の unload 文の area オペランドに指定した RD エリア名を指定してください。pdsdbrog 制御文の unload 文の area オペランドを省略した場合は，省略してください。
9		data	pdsdbrog 制御文の unload 文の unldfile オペランドに指定したアンロードデータファイル名を指定してください。
10		prefix	pdsdbrog 制御文の unload 文の prefix オペランドに指定したプリフィクス長を指定してください。
11		seqkeyck	省略するか，または yes を指定してください。
12		上記以外のオペランド	任意の値を指定してください。
13	dbinf 文※	userpflg	pdsdbrog 制御文の dbinf 文の userpflg オペランドに指定した位置を指定してください。pdsdbrog 制御文の dbinf 文の userpflg オペランドを省略した場合は，省略してください。

項番	オペランド		オペランドの指定値
14		pagenum	pdsdbrog 制御文の dbinf 文の pagenum オペランドに指定した位置を指定してください。pdsdbrog 制御文の dbinf 文の pagenum オペランドを省略した場合は，省略してください。
15		上記以外のオペランド	任意の値を指定してください。
16	idxload 文	すべてのオペランド	任意の値を指定してください。

注※

SD FMB の SDB データベースを再編成する場合は，dbinf 文は省略してください。SD FMB の SDB データベースに対しては，dbinf 文は指定できません。

(6) RD エリアの閉塞を解除する

```
pdrels -r RDエリア名,RDエリア名 …
```

pdrels コマンドで RD エリアの閉塞を解除します。

5.12　SDB データベースの再編成（インナレプリカ機能を使用したオンライン再編成）【4V FMB，4V AFM】

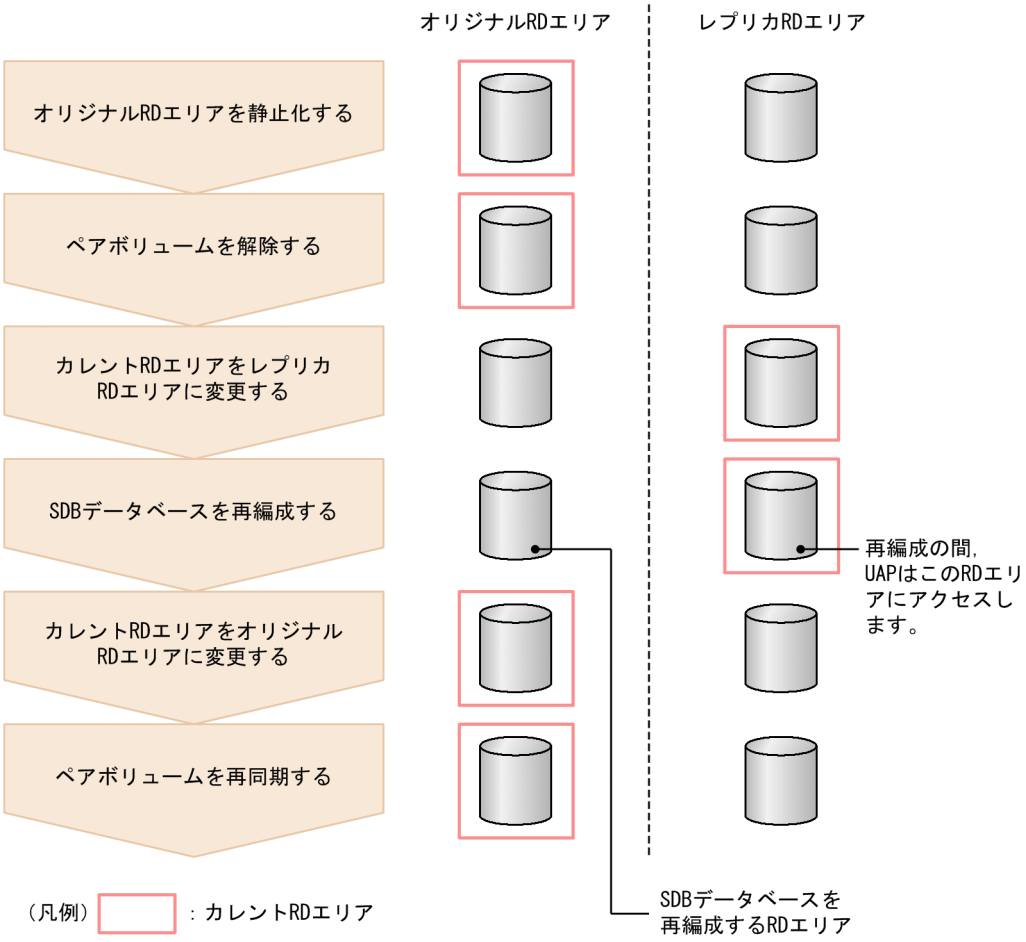
インナレプリカ機能を使用したオンライン再編成の場合，SDB データベースの再編成中にデータベースを参照できます（更新はできません）。ここでは，インナレプリカ機能を使用したオンライン再編成について説明します。

インナレプリカ機能を使用したオンライン再編成については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」も参照してください。

5.12.1　オンライン再編成の流れ

インナレプリカ機能を使用したオンライン再編成の流れを次の図に示します。

図 5-11　インナレプリカ機能を使用したオンライン再編成の流れ



[説明]

- オリジナル RD エリアで SDB データベースの再編成を実行し，レプリカ RD エリアで業務を続行します。ただし，データベースを更新するような業務は実行できません。

- オリジナル RD エリアとレプリカ RD エリアの間でデータの不一致が発生しないように，SDB データベースの再編成が完了するまで UAP がアクセスする RD エリアは更新できない状態にしておく必要があります。
- SDB データベースの再編成対象となる RD エリア（レコードを格納している RD エリア，およびインデクスを格納している RD エリア）は，すべて同じ世代である必要があります。

ポイント

レプリカ RD エリアで SDB データベースの再編成を実行し，オリジナル RD エリアで業務を続行することもできますが，レプリカ RD エリアで SDB データベースを再編成する手順は難しいため，オリジナル RD エリアで SDB データベースを再編成するようにしてください。

注意事項

SDB データベースの再編成中に，「5.7　SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合）」で説明している運用は行わないでください。

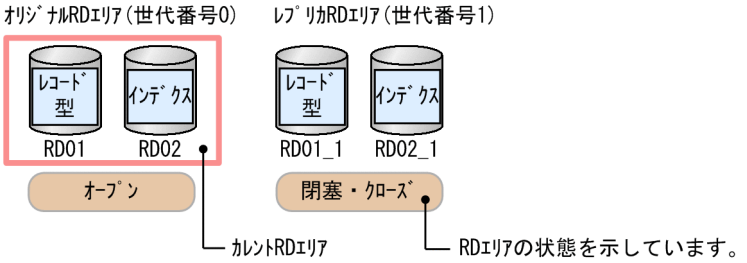
5.12.2　オンライン再編成の手順

インナレプリカ機能を使用したオンライン再編成の手順を例題形式で説明します。

なお，例題の説明は，インナレプリカ機能についての知識があることを前提にしています。インナレプリカ機能については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」を参照してください。

例題

オリジナル RD エリアに格納されている SDB データベースをオンライン再編成します。レコード型およびインデクスを格納している RD エリアは次のとおりです。

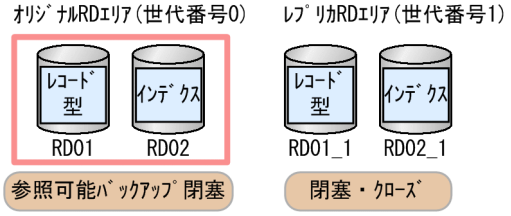


再編成対象のレコード型およびインデクスを格納している RD エリアは，すべて同じ世代である必要があります。上記の構成の場合，RD01_1 と RD02_1 の世代が同じである必要があります。

1. オリジナル RD エリアを参照可能バックアップ閉塞状態にする

```
pdhold -r RD01,RD02 -q 0 -b
```

pdhold コマンドでオリジナル RD エリアを参照可能バックアップ閉塞状態にします（オリジナル RD エリアを静止化します）。



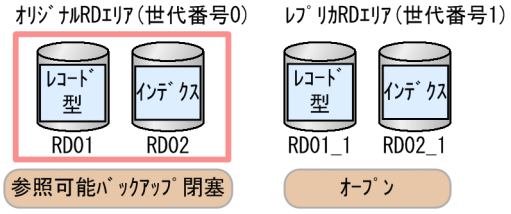
2. ペアボリュームを解除する

ペアボリュームを解除して，ハードウェアまたはソフトウェアのミラーリング機能で，オリジナル RD エリアからレプリカ RD エリアにデータを複写します。

3. レプリカ RD エリアの閉塞を解除してオープン状態にする

```
pdrels -r RD01,RD02 -q 1 -o
```

pdrels コマンドでレプリカ RD エリアの閉塞を解除してオープン状態にします。

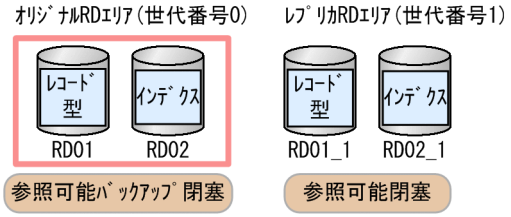


4. レプリカ RD エリアを参照可能閉塞状態にする

```
pdhold -r RD01,RD02 -q 1 -i
```

pdhold コマンドでレプリカ RD エリアを参照可能閉塞状態にします。

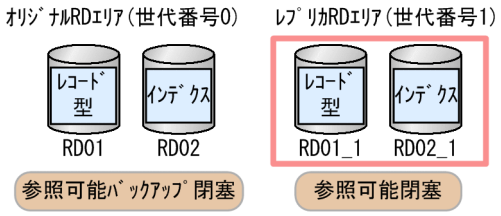
SDB データベースの再編成が完了するまでの間，オリジナル RD エリアとレプリカ RD エリアの間でデータの不一致が発生しないように，業務で利用するレプリカ RD エリアを更新抑止状態にする必要があるため，レプリカ RD エリアを参照可能閉塞状態にします。



5. カレント RD エリアをレプリカ RD エリアに変更する

```
pddbchg -r RD01,RD02 -q 1
```

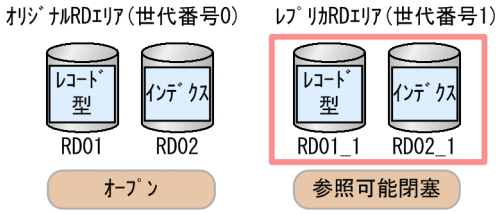
pddbchg コマンドで，カレント RD エリアをオリジナル RD エリアからレプリカ RD エリアに変更します。これによって，UAP はこのレプリカ RD エリアにアクセス（参照だけ）します。



6. オリジナル RD エリアの閉塞を解除する

```
pdrels -r RD01,RD02 -q 0
```

pdrels コマンドでオリジナル RD エリアの閉塞を解除します。



7. 全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認する

```
pdsdbarc -a
```

pdsdbrog コマンドを実行する前に pdsdbarc コマンドを実行して，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9.1　常用常駐領域の SDB ディレクトリ情報の確認方法（実行系ユニットの場合）」を参照してください。

8. オリジナル RD エリアのレコードをアンロードする

```
pdsdbrog /usr/control_rog_file
```

pdsdbrog コマンドで，オリジナル RD エリアのレコードをアンロードします。

pdsdbrog 制御文の environment 文の generation オペランドに 0 を指定（オリジナル RD エリアを処理対象にする指定）してください。

4V FMB の SDB データベースの場合の pdsdbrog 制御文の指定例については，「13.10.2　4V FMB の SDB データベースのアンロードおよびデータロード（インナレプリカ機能を使用したオンライン再編成の場合）」を参照してください。

ページ切り替えフラグの指定，または事前割り当てページ数の指定値の変更が必要な場合は，アンロードの完了後，アンロードデータファイルを直接修正して，ページ切り替えフラグまたは事前割り当てページ数を設定してください。ページ切り替えフラグを設定する場合は，「13.8　ページ切り替えフラグの扱い【4V FMB，4V AFM】」を参照してください。

9. オリジナル RD エリアにレコードをデータロードする

```
pdsdblod /usr/control_lod_file
```

pdsdblod コマンドで，オリジナル RD エリアにレコードをデータロードします。

pdsdblod 制御文の environment 文の generation オペランドに 0 を指定（オリジナル RD エリアを処理対象にする指定）してください。

4V FMB の SDB データベースの場合の pdsdblod 制御文の指定例については，「13.10.2　4V FMB の SDB データベースのアンロードおよびデータロード（インナレプリカ機能を使用したオンライン再編成の場合）」を参照してください。

10. オリジナル RD エリアのバックアップを取得する

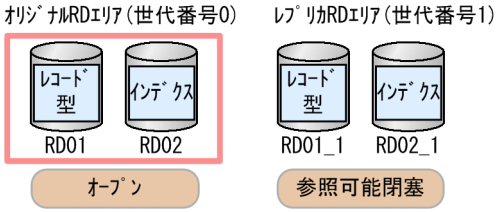
```
pdcopy -m /hirdb/rdarea/rdmast/rdmast01 -M -r RD01,RD02 -q 0
       -b /usr/hirdbsd/backup01 -p /usr/hirdbsd/list01
```

pdcopy コマンドでオリジナル RD エリアのバックアップを取得します。

11. カレント RD エリアをオリジナル RD エリアに変更する

```
pddbchg -r RD01,RD02 -q 0
```

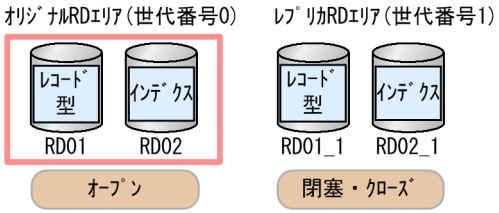
pddbchg コマンドで，カレント RD エリアをレプリカ RD エリアからオリジナル RD エリアに変更します。



12. レプリカ RD エリアを閉塞かつクローズ状態にする

```
pdhold -r RD01,RD02 -q 1 -c
```

pdhold コマンドでレプリカ RD エリアを閉塞かつクローズ状態にします。



13. ペアボリュームを生成する

オリジナル RD エリアを正ファイルとして，レプリカ RD エリアとのペアボリュームを生成します。

これでオリジナル RD エリアおよびレプリカ RD エリアの状態が最初の状態に戻ります。

5.13 SDB データベースの再編成（インナレプリカ機能を使用した更新可能なオンライン再編成）【4V FMB】

ここでは，インナレプリカ機能を使用した更新可能なオンライン再編成について説明します。インナレプリカ機能を使用した更新可能なオンライン再編成の場合，SDB データベースの再編成中にデータベースを参照および更新できます。

インナレプリカ機能を使用した更新可能なオンライン再編成については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「更新可能なオンライン再編成の運用方法」も参照してください。

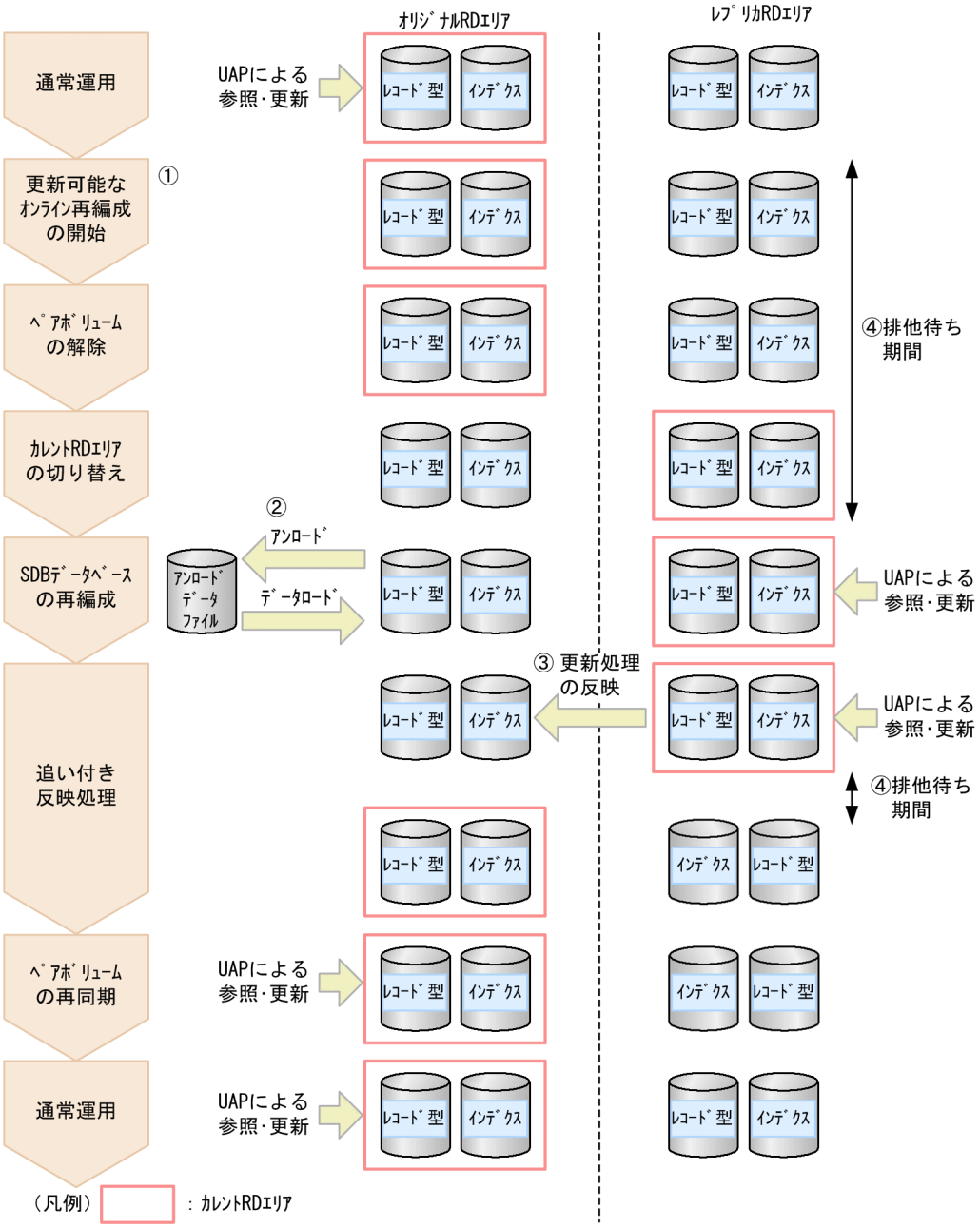
5.13.1 更新可能なオンライン再編成の概要

(1) 更新可能なオンライン再編成の流れ

更新可能なオンライン再編成の場合，オリジナル RD エリアで SDB データベースの再編成処理を行い，レプリカ RD エリアで UAP からの参照および更新処理を行います。オリジナル RD エリアでの再編成処理が完了したあとに，レプリカ RD エリアに対して実行された更新処理の内容をオリジナル RD エリアに反映します。

更新可能なオンライン再編成の処理の流れを次の図に示します。

図 5-12　更新可能なオンライン再編成の処理の流れ



[説明]

図中の丸付き数字は，次の番号に対応しています。

1. pdorbegin コマンドを実行した時点が，更新可能なオンライン再編成の開始時点となります。pdorbegin コマンドを実行すると，オリジナル RD エリアおよびレプリカ RD エリアの状態がオンライン再編成閉塞になります。
2. オリジナル RD エリアの SDB データベースに対して，再編成処理（アンロード処理およびデータロード処理）を実行します。UAP からの参照および更新処理は，カレント RD エリアで実行されます。
3. pdorend コマンドを実行すると，レプリカ RD エリアに対して実行された更新処理の内容を，オリジナル RD エリアに反映します。この処理を追い付き反映処理といいます。

4. 次の期間は，UAP からの参照および更新処理が排他待ち状態になります。

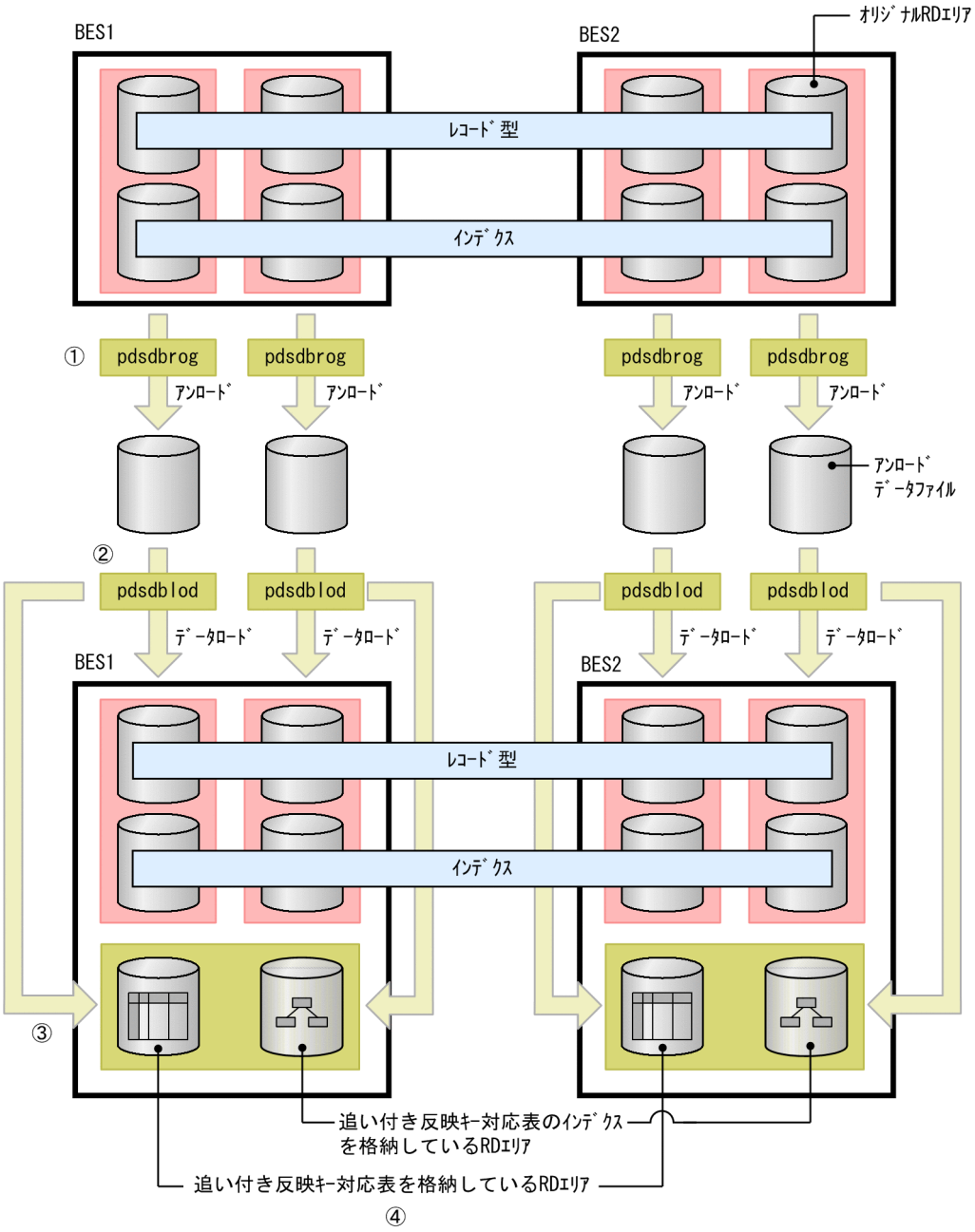
・更新可能なオンライン再編成の開始時点からカレント RD エリアの切り替え完了後

・追い付き反映処理中のオリジナル RD エリアとカレント RD エリアの同期を取得する期間

(2) SDB データベースの再編成処理の概要

SDB データベースの再編成は，インナレプリカ機能を使用しないときと同様に，pdsdbrog コマンドでいったんレコードをアンロードし，その後 pdsdblod コマンドでデータロードすることで実現します。SDB データベースの再編成処理の概要を次の図に示します。

図 5-13 SDB データベースの再編成処理の概要



［説明］

図中の丸付き数字は，次の番号に対応しています。

1. pdsdbrog コマンドを実行して，オリジナル RD エリアのレコードをアンロードします。この例では，SDB データベースを横分割しているため，RD エリア単位にアンロードを実行しています。
2. pdsdblod コマンドを実行して，オリジナル RD エリアにデータロードします。この例では，SDB データベースを横分割しているため，RD エリア単位にデータロードを実行しています。
3. データロードの際，オリジナル RD エリアとレプリカ RD エリアのデータを対応づける情報が，追い付き反映キー対応表に登録されます。ここで登録された情報が追い付き反映処理時に使用されます。また，このとき，追い付き反映キー対応表のインデクスも作成されます。
4. 追い付き反映キー対応表，およびそのインデクスは，BES 単位に作成されます。

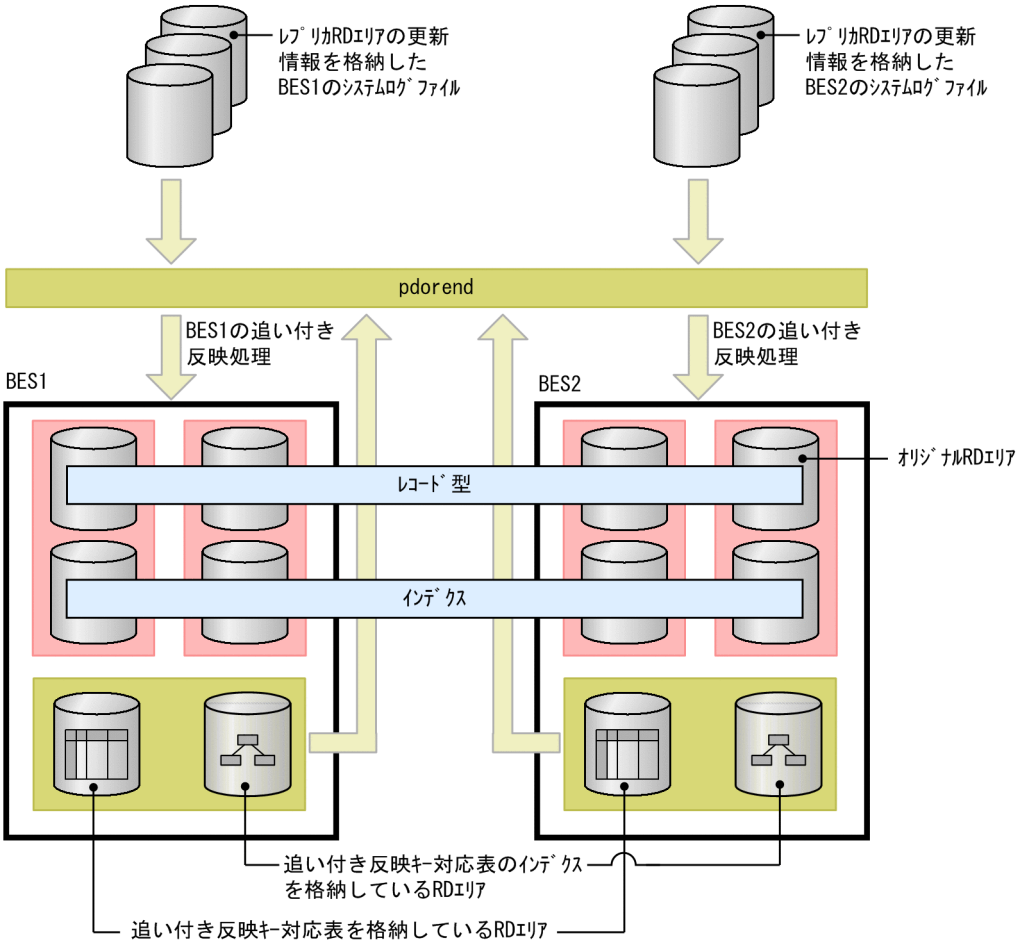
参考

- 追い付き反映キー対応表にユーザはアクセスできません。
- 再編成対象の RD エリアがオンライン再編成閉塞の場合に，更新可能なオンライン再編成として処理されます。

(3) 追い付き反映処理の概要

レプリカ RD エリアの SDB データベースに対して実行された更新処理の内容を，オリジナル RD エリアの SDB データベースに反映します。これを追い付き反映処理といいます。追い付き反映処理の概要を次の図に示します。

図 5-14　追い付き反映処理の概要



[説明]

- pdorend コマンドを実行すると，追い付き反映処理が開始されます。追い付き反映処理は BES 単位に実行されます。
- レプリカ RD エリアの SDB データベースに対する更新情報（システムログ）を入力情報にして，オリジナル RD エリアの SDB データベースを更新します（HiRDB が内部的に API を発行して，オリジナル RD エリアの SDB データベースを更新します）。このとき，追い付き反映キー対応表に登録された情報を使用して，レプリカ RD エリアに対する更新情報とオリジナル RD エリアの反映先データとの対応づけを行います。
- システムログの更新情報の読み込みが完了すると，オリジナル RD エリアとレプリカ RD エリアの同期を取るために，RD エリアにアクセスするトランザクションを排他待ち状態にします。

  このあと，カレント RD エリアをレプリカ RD エリアからオリジナル RD エリアに切り替えて，オンライン再編成閉塞を解除します。
- 追い付き反映処理が完了すると，RD エリアを次の状態にして，オリジナル RD エリアで業務を再開します（RD エリアにアクセスするトランザクションの排他待ち状態を解除します）。

  ・オリジナル RD エリア：オープン状態

  ・レプリカ RD エリア：コマンド閉塞かつクローズ状態

5.13.2　運用前に考慮すること

更新可能なオンライン再編成の運用前に知っておく必要がある制限事項および注意事項について説明します。

(1)　制限事項

- RD エリアの状態がオンライン再編成閉塞の場合，次の表に示すように SDB データベースの操作が制限されます。

表 5-4　制限される SDB データベースの操作

項番	SDB データベースの操作	実行可否	
		オリジナル RD エリア	レプリカ RD エリア
1	SDB データベースを操作する API	×	○※
2	pdsdblod コマンド	○	×
3	pdsdbrog コマンド	○	×
4	pdsdbdef コマンド	×	×
5	pdmod コマンド ただし，次の操作は実行できます。 • RD エリアの再初期化 • RD エリアの拡張 • RD エリアのレプリカ定義 • RD エリアの構成情報複写（複写元）を除く	×	×
6	pddbchg コマンド	×	×

（凡例）

○：実行できます。

×：実行できません。

注※

実行できる SDB データベースを操作する API は，更新ログ取得方式がログ取得モードかログレスモードかで異なります。

ログ取得モードの場合，次の SDB データベースを操作する API だけが実行できます。

・レコードの格納（STORE）

・レコードの削除（ERASE）

ただし，ルートレコードに対しては実行できません。

・レコードの更新（MODIFY）

・レコードの検索（FETCH）

・複数レコードの検索（FETCHDB ALL）

ログレスモードの場合，次の SDB データベースを操作する API で，アクセスモード 1 に'R'（参照モード）を指定するときにだけ実行できます。

・レコードの検索（FETCH）
・複数レコードの検索（FETCHDB ALL）

- クライアント環境定義の PDDBORGUAP オペランドは指定できないため，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「オンライン業務と更新バッチ処理の同時実行」で説明している運用はできません。
- 更新可能なオンライン再編成では，同一 BES 内でリレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアを同じタイミングで再編成できません。pdorbegin コマンドに指定する RD エリアには，1 つの BES でリレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアを混在させることはできません。そのため，リレーショナル DB を格納する RD エリアと SDB データベースを格納する RD エリアは別々に再編成してください。

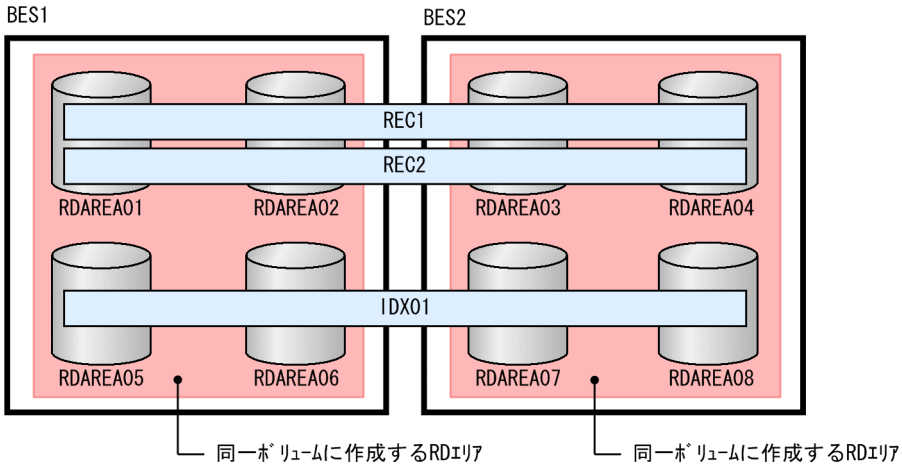
(2) システム構成に関する注意事項

- 更新可能なオンライン再編成は，4V FMB の SDB データベースに対してだけ実行できます。そのため，4V FMB の SDB データベースのインデクスと，4V AFM の SDB データベースのインデクスを同じ RD エリアに格納している場合，その 4V FMB の SDB データベースは更新可能なオンライン再編成ができません。

> **参考**
>
> 4V FMB の SDB データベースのインデクスと，4V AFM の SDB データベースのインデクスが同じ RD エリアに格納されていないかどうかは，pdorcheck コマンドでチェックできます。

- 更新可能なオンライン再編成の対象となる関連 RD エリアは，BES ごとに同一ボリュームに作成してください。関連 RD エリアとは，次の RD エリアのことです。
  - レコード型を格納する RD エリア
  - インデクスを格納する RD エリア

（例）



また，同一ボリュームに作成した関連 RD エリアは，同じタイミングで更新可能なオンライン再編成を実行する必要があります。

参考

関連 RD エリアは pdrdrefls コマンドで確認できます。

(3) 更新可能なオンライン再編成実行時の注意事項

- 更新可能なオンライン再編成を開始すると，追い付き反映処理が終了するまで，同じ BES 内のほかの RD エリアに対して，更新可能なオンライン再編成を実行することはできません。
- 複数の RD エリアに対して SDB データベースの再編成を実行する場合は，全 RD エリアの再編成処理が完了したあとに，追い付き反映処理を実行してください。
- 複数の RD エリアに対して SDB データベースの再編成を実行する場合は，再編成処理に掛かる時間が同じくらいの RD エリアを選択することを推奨します。
- レプリカ RD エリアに対するトランザクション処理が多く，追い付き反映処理が最大待ち時間内に終了しなかった場合，トランザクション処理量を制限してから追い付き反映処理を再実行してください。追い付き反映処理の最大待ち時間は，pdorend コマンドの-w オプションで指定します。
- オンライン再編成閉塞の RD エリアがある場合，HiRDB の正常終了または BES の正常終了ができません（計画停止はできます）。HiRDB の正常終了または BES の正常終了をしたい場合は，次のどちらかの方法で RD エリアのオンライン再編成閉塞を解除してください。
  - pdorend コマンドによる追い付き反映処理を完了して，オンライン再編成閉塞を解除する
  - pdorbegin -u コマンドで RD エリアのオンライン再編成閉塞を解除する

(4) 追い付き反映キー対応表に関する注意事項

(a) 追い付き反映キー対応表を格納する RD エリアに関する注意事項

- 更新可能なオンライン再編成の対象 RD エリアが，複数の BES に存在する場合，各 BES に追い付き反映キー対応表を作成する必要があります。そのため，追い付き反映キー対応表を格納する RD エリアを各 BES に作成しておく必要があります。
- 追い付き反映キー対応表へのデータ登録時に，追い付き反映キー対応表を格納する RD エリアの容量不足が発生すると，データロードまたは追い付き反映処理がエラーになります。そのため，RD エリアの容量見積もりを必ず実施してください。RD エリアの容量見積もり方法については，「3.5.6 ユーザ用 RD エリア（追い付き反映キー対応表を格納する）の容量の見積もり【4V FMB】」を参照してください。

  RD エリアの容量不足が発生した場合，追い付き反映キー対応表を初期化する必要があります。その場合，BES 内の更新可能なオンライン再編成の対象 RD エリアの情報がすべて削除されます。そのため，データロードが完了している RD エリアも，データロードを再実行する必要があります（BES が異なる場合は，データロードを再実行する必要はありません）。

(b)　データロードの再実行前に追い付き反映キー対応表を初期化するときの注意事項

障害などの発生によって，データロードの再実行が必要となった場合は，データロードを再実行する前に，追い付き反映キー対応表を pdsdborcrt コマンドで初期化する必要があります。初期化しない場合（更新可能なオンライン再編成の対象 RD エリアに関する情報が，追い付き反映キー対応表に登録されたままの場合），データロードの再実行がエラーになります。

ただし，追い付き反映キー対応表を初期化すると，BES 内の更新可能なオンライン再編成の対象 RD エリアの情報がすべて削除されます。そのため，データロードが完了している RD エリアも，データロードを再実行する必要があります（BES が異なる場合は，データロードを再実行する必要はありません）。

なお，更新可能なオンライン再編成の実行中に，追い付き反映キー対応表を初期化または削除する必要が生じた場合は，pdsdborcrt コマンドの実行後に警告メッセージと確認メッセージが出力されます。次の流れで応答してください。

1. 警告メッセージ（KFPT82005-W）が出力されます。KFPT82005-W メッセージに出力された BES 名を確認してください。
2. 1.で確認した BES 内に，オンライン再編成閉塞の RD エリアがあるかどうかを pddbls コマンドで確認してください。
3. 確認メッセージ（KFPT82006-Q）が出力されます。追い付き反映キー対応表を初期化または削除する場合は y を，初期化または削除をやめる場合は n を指定してください。

注意事項

次のどちらかの条件を満たす場合は，y を応答してください。

- BES 内にオンライン再編成閉塞の RD エリアがない場合
- データロード時や追い付き反映処理時に障害が発生し，追い付き反映キー対応表の初期化または削除が必要な場合

上記の条件を満たさない場合は，必ず n を応答してください。上記の条件を満たさない場合に y を指定すると，SDB データベースの内容が不整合な状態になるおそれがあります。

(5)　アンロードデータファイルに関する注意事項

(a)　アンロードデータファイルの作成に関する注意事項

アンロード時に作成したアンロードデータファイルを，データロード時の入力データファイルにしてください。ほかの方法で作成したアンロードデータファイルを入力データファイルにした場合，SDB データベースの内容が不整合になります。

(b)　アンロードデータファイルの保管に関する注意事項

更新可能なオンライン再編成の操作が完了するまで，アンロードデータファイルを削除しないでください。アンロードデータファイルを削除してしまうと，追い付き反映キー対応表の障害などによって，データロー

ドから操作を再実行する必要がある場合に，データロードの再実行ができなくなります。この場合，更新可能なオンライン再編成を中止し，レプリカ RD エリアの更新データをオリジナル RD エリアに反映して，オリジナル RD エリアを回復する必要があります。

なお，データロードが正常終了したあと（KFPB63012-I メッセージが出力されたあと）に，データロードの再実行が必要になった場合，アンロードを再実行してアンロードデータファイルを作成しても，アンロードデータファイルの内容はデータロード後の内容になっています。このアンロードデータファイルを入力データファイルに指定した場合，SDB データベースの内容が不整合になります。

(c) アンロードデータファイルの内容を変更する際の注意事項

アンロードデータファイルの内容を変更し，データロード時の入力データファイルにする場合，次の変更だけができます。

- ファミリ単位の追加
- ファミリ単位の削除
- ページ切り替えフラグの設定
- 事前割り当てページ数の設定

上記以外の変更をした場合は，SDB データベースの内容が不整合になるおそれがあります。

また，ファミリ単位の削除を行ったレコードに対して，カレント RD エリアで更新が発生した場合，オリジナル RD エリアでは対応するレコードがないため，追い付き反映処理で実行される API（HiRDB/SD が内部的に発行する API）でエラーが発生します。このエラーをスキップし，追い付き反映処理を続行したい場合は，pdorend コマンドに-f オプションを指定して，追い付き反映制御ファイルに次の指定をしてください。

- reflection 文の dml_error オペランドに skip を指定する

詳細については，「5.13.10　追い付き反映処理時に HiRDB/SD が内部的に発行する API のエラーをスキップしたい場合」を参照してください。

(6) HiRDB の再起動を必要としない SDB データベースの定義追加，定義変更との関連

- 更新可能なオンライン再編成の実行中に，pdsdbarc -w -q コマンドによる事前常駐領域と常用常駐領域の切り替えを行うと，更新可能なオンライン再編成の処理中に整合性エラーが発生します。そのため，次に示す操作の開始から終了までの間は，更新可能なオンライン再編成の対象 SDB データベースに対して，HiRDB の再起動を必要としない SDB データベースの定義追加，定義変更の作業（SDB ディクショナリ情報の作成からトランザクション静止化解除までの作業）を実施しないでください。
  - 「5.13.3(3)　関連 RD エリアの確認（pdrdrefls コマンド）」の操作開始
  - 「5.13.4(9)　追い付き反映処理を実行する（pdorend コマンド）」の操作完了

ただし，更新可能なオンライン再編成の操作より前であれば，HiRDB の再起動を必要としない SDB データベースの定義追加，定義変更の次の作業は実施できます。

- SDB ディクショナリ情報の作成から，TP1/FSP の SDB 定義情報領域（副）に SDB 定義情報を格納するまで

- HiRDB の再起動を必要としない SDB データベースの定義追加によって追加した 4V FMB の SDB データベースに対する更新可能なオンライン再編成は，SDB データベースの定義追加が完了するまで実行できません。追加中の 4V FMB の SDB データベースに対する適用条件のチェック（pdorcheck コマンド）でエラーになります。
- SDB データベースの定義変更でレコード型を追加し，その二次インデクスを格納するために追加した RD エリアは，ディクショナリ表中の SDB ディクショナリ情報が更新されると，pdrdrefls コマンドの実行結果の関連 RD エリアとして表示されます（「5.7.5(2)　SDB ディクショナリ情報を変更する」の操作が終わると，関連 RD エリアとして表示されます）。ただし，この時点では，追加した RD エリアは，pdrdrefls コマンドの実行結果の関連 RD エリアとして表示されますが，更新可能なオンライン再編成の対象にはできません。SDB データベースの定義変更の操作が完了するまで，更新可能なオンライン再編成の対象にはできません。

  各コマンドでの注意事項を次に示します。

  - pdrdrefls コマンド

    二次インデクスを格納するために追加した RD エリアが，関連 RD エリアとして表示されますが，更新可能なオンライン再編成の対象にはできません。

  - pdorcheck コマンド

    二次インデクスを格納するために追加した RD エリアを，pdorcheck コマンドの-r オプションに指定すると，更新可能なオンライン再編成を実行できない旨のチェック結果が返ってきます。二次インデクスを格納するために追加した RD エリアを指定しないで，pdorcheck コマンドを実行してください。

  - pdorbegin コマンド

    -t オプション（レコード型名指定）は指定できません。-r オプション（RD エリア名指定）を指定して，pdorbegin コマンドを実行してください。-r オプションに指定する RD エリアには，二次インデクスを格納するために追加した RD エリアを指定しないでください。

    なお，-t オプションを指定すると，二次インデクスを格納するために追加した RD エリアも，処理対象 RD エリアに含まれるため，更新可能なオンライン再編成の処理が正しく実行されません（途中でエラーになるか，または何も実行されないで正常終了します）。

(7)　系切り替え機能との関連

追い付き反映処理中に系の切り替えが発生した場合，pdorend コマンドが異常終了します。追い付き反映処理が終了しない状態が長く続くと，オンライン側（業務実行側）のレプリカ RD エリアの更新情報がシステムログに蓄積され続けるため，システムログ量が見積もり値を超えるおそれがあります。そのため，系切り替えが発生した場合は，できるだけ早く pdorend コマンドを再実行するようにしてください。

(8)　追い付き反映処理時の格納配置制御

追い付き反映処理時の格納配置制御に関するオプションの適用可否を次の表に示します。

表 5-5　格納配置制御に関するオプションの適用可否（追い付き反映処理時）

追い付き反映処理の種別	格納配置制御に関するオプション		
	ページ切り替えオプション	PCTFREE 有効化オプション	事前割り当てページ数
レコードの格納（STORE）の更新追い付き反映処理	○	○	○
レコードの削除（ERASE）の回復追い付き反映処理	×	×	−

（凡例）

○：レコードの格納（STORE）の実行時（レプリカ RD エリア側の更新時）に指定した値を引き継いで，追い付き反映処理が実行されます。SDB データベースの再編成時に，事前ページ割り当て機能の適用有無を変更した場合の動作については，「2.6.2(4)(d)　事前ページ割り当て機能に関する留意事項」または「2.7.2(4)(d)　事前ページ割り当て機能に関する留意事項」を参照してください。

×：格納配置制御に関するオプションの指定が適用されません。この場合，レコード実現値を格納する際，ページまたはサブページを切り替えません。また，SDB データベース格納定義の PCTFREE オペランドで指定したページ内の未使用領域比率またはサブページ内の未使用領域比率を適用しないで，レコード実現値が格納されます。

−：該当しません。

(9)　更新可能なオンライン再編成中に整合性エラーが発生した場合

更新可能なオンライン再編成中に，誤って pdsdbarc -w -q コマンドを実行したり，系切り替えが発生したりすると，整合性エラーが発生するおそれがあります。整合性エラーが発生した場合，更新可能なオンライン再編成を続行するか，または中止するかを判断してください。

中止する場合は，レプリカ RD エリアの更新データをオリジナル RD エリアに反映し，オリジナル RD エリアで業務を再開します。更新可能なオンライン再編成を中止する場合の手順については，「5.23　障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】」を参照してください。

更新可能なオンライン再編成を続行する場合は，常用常駐領域の SDB ディレクトリ情報を変更前の状態に戻し，そのあとにエラーとなったコマンドを再実行してください。SDB ディレクトリ情報を変更前の状態に戻す方法については，「5.7.7　障害発生時の対処（定義追加または定義変更前の状態に戻す場合）」を参照してください。更新可能なオンライン再編成を続行する際の手順については，「5.23　障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】」を参照してください。

5.13.3　更新可能なオンライン再編成の準備作業

更新可能なオンライン再編成の準備作業の流れを次の表に示します。

表 5-6　更新可能なオンライン再編成の準備作業の流れ

項番	作業項目	実行するコマンド	実行するタイミング
1	RD エリアの物理配置の検討	−	最初に 1 回だけ実行
2	更新可能なオンライン再編成に関するオペランド（システム定義）の追加	−	
3	関連 RD エリアの確認	pdrdrefls コマンド	
4	HiRDB ファイルシステム領域の世代登録	pdmod コマンド	
5	レプリカ RD エリアの定義		
6	次の表を格納する RD エリアの作成 • 追い付き反映キー対応表 • 追い付き状態管理表		
7	追い付き反映キー対応表の作成または初期化	pdsdborcrt コマンド	更新可能なオンライン再編成の実行前に毎回実施
8	適用条件のチェック	pdorcheck コマンド	
9	ペアボリュームの定義	−	最初に 1 回だけ実行
10	追い付き状態管理表の作成	pdorcreate コマンド	
11	SDB ディレクトリ情報の最終更新日時のチェック	pdsdbarc -a コマンド	更新可能なオンライン再編成の実行前に毎回実施

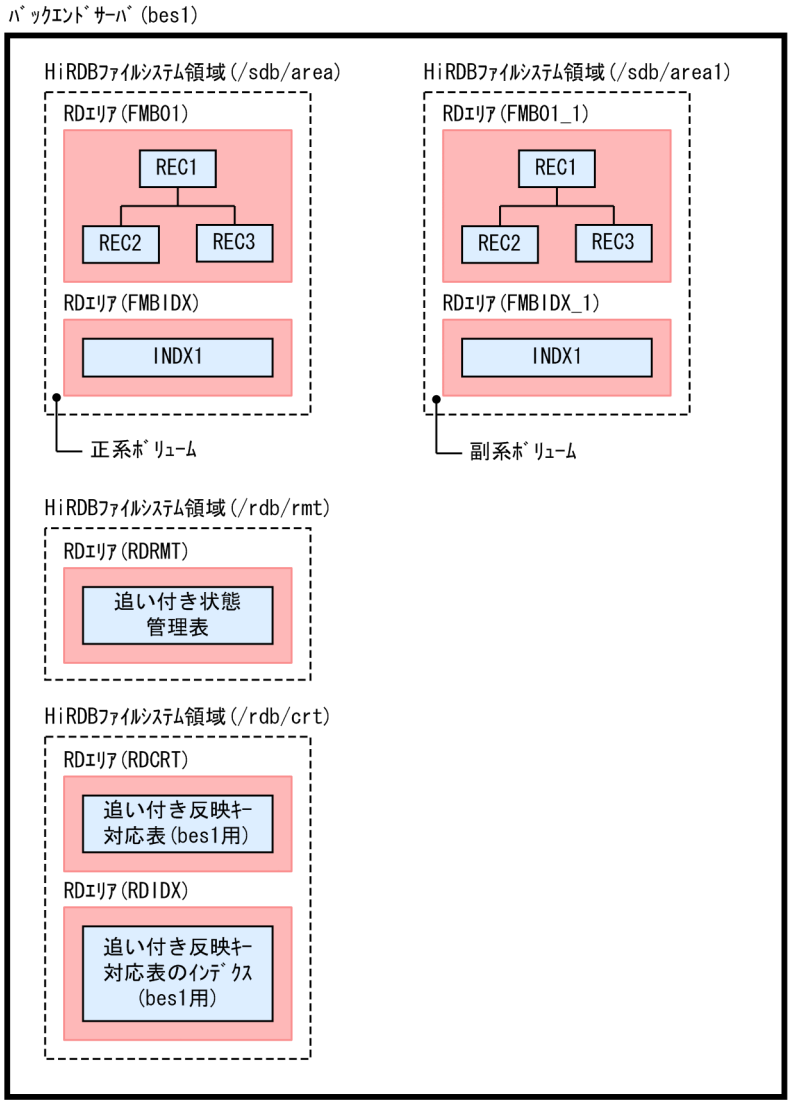
（凡例）

−：なし

更新可能なオンライン再編成の準備作業の手順を例題形式で説明します。なお，例題の説明は，インナレプリカ機能についての知識があることを前提にしています。

例題

SDB データベースを更新可能なオンライン再編成するための準備作業を行います。SDB データベースを格納している RD エリアなどの構成は次のとおりです。



ポイント

- 追い付き状態管理表はシステムで 1 つ必要です。
- 追い付き反映キー対応表は BES ごとに必要です。

(1) RD エリアの物理配置の検討

更新可能なオンライン再編成を実行する RD エリアのボリューム内の物理配置を検討してください。ポイントを次に示します。

- **レプリカ作成ガイドラインを満たしているか**

  更新可能なオンライン再編成ではインナレプリカ機能を使用するため，レプリカ作成ガイドラインも考慮する必要があります。インナレプリカ機能に関する考慮点については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「インナレプリカの運用方法」の「運用前に考慮すること」を参照してください。

- **同じボリュームにインナレプリカ機能単独で使用される RD エリアがないか**

更新可能なオンライン再編成では，マスタ DB とレプリカ DB が一時的に切り替わるため，それとは別のレプリカを使用する作業（インナレプリカ機能を使うが更新可能なオンライン再編成を行わない RD エリア）に運用上悪い影響がないかを考慮する必要があります。

また，次の RD エリアの物理配置も検討してください。

- 追い付き反映キー対応表を格納する RD エリア
- 追い付き反映キー対応表のインデクスを格納する RD エリア
- 追い付き状態管理表を格納する RD エリア

検討のポイントについては，「(6)　追い付き反映キー対応表，追い付き状態管理表を格納する RD エリアの作成（pdmod コマンド）」を参照してください。

(2)　更新可能なオンライン再編成に関するオペランド（システム定義）の追加

次のオペランドを追加してください。各オペランドの詳細については，マニュアル「HiRDB システム定義」を参照してください。

なお，システム定義を変更したあとに，pdconfchk コマンドでシステム定義のチェックを実施してください。

- pd_max_reflect_process_count
  追い付き反映処理で HiRDB が保証する pdorend 追い付き反映プロセスの同時実行数を指定します。このオペランドの留意事項については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「pd_max_reflect_process_count オペランドの留意事項と見積もり」を参照してください。
- pd_log_org_reflected_logpoint
  追い付き反映処理が完了したあと，システムログファイルの状態を変更するかどうかを指定します。障害の発生時にどのように対処するかによって，このオペランドの指定値を決めてください。指定値による障害対策方法の違いについては，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「システムログとオペランドの設定」を参照してください。
  通常は，keep を指定することをお勧めします。
- pd_log_org_no_standby_file_opr
  すべてのシステムログファイルがオンライン再編成上書き禁止状態の場合に，スワップが発生したときの HiRDB の処理を指定します。障害の発生時にどのように対処するかによって，このオペランドの指定値を決めてください。指定値による障害対策方法の違いについては，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「システムログとオペランドの設定」を参照してください。
  通常は，stop を指定することをお勧めします。
- pd_lv_mirror_use = Y
  pd_lv_mirror_use オペランドに Y（レプリカ RD エリアのオープン属性を SCHEDULE にする指定）を指定してください。更新可能なオンライン再編成の運用手順が少なくなります。また，バックエンドサーバ接続保持機能を使用する場合は，必ず Y を指定してください。

指定例

```
set pd_max_reflect_process_count = 8
set pd_log_org_reflected_logpoint = keep
set pd_log_org_no_standby_file_opr = stop
set pd_lv_mirror_use = Y
```

また，次のオペランドの指定値を見積もり直してください。

- pd_lck_pool_size
  次の計算式で求めた値を pd_lck_pool_size オペランドの指定値に加算してください。
  25×a （単位：キロバイト）
  a：更新可能なオンライン再編成時の，pdsdblod コマンドの同時実行数（oreload 文の指定がある pdsdblod コマンド）

(3) 関連 RD エリアの確認（pdrdrefls コマンド）

pdrdrefls コマンドで，更新可能なオンライン再編成の対象となる関連 RD エリアを確認します。

pdrdrefls コマンドの-t オプションにレコード型名を指定すると，指定したレコード型の関連 RD エリアが表示されます。pdrdrefls コマンドの-r オプションに RD エリア名をすると，指定した RD エリアの関連 RD エリアが表示されます。

- レコード型を指定する場合

  ```
  pdrdrefls -e org -t REC1 -l -d ','
  ```

  [オプションの説明]
  -e org：固定のオプションです。
  -t：ルートレコードのレコード型名を指定します。
  -l：出力情報の各情報を改行しないで表示する場合に指定します。
  -d：-l オプション指定時の，各情報の区切り文字を半角 1 文字で指定します。

  pdrdrefls コマンドの実行結果

  ```
  "FMB01","FMBIDX"
  ```

  レコード型 REC1 の関連 RD エリアは，FMB01 と FMBIDX です。

- RD エリアを指定する場合

  ```
  pdrdrefls -e org -r FMB01 -l -d ','
  ```

  [オプションの説明]
  -e org：固定のオプションです。
  -r：ルートレコード（REC1）に対する格納レコード用 RD エリアのオリジナル RD エリア名を指定します。
  -l：出力情報の各情報を改行しないで表示する場合に指定します。

-d：-l オプション指定時の，各情報の区切り文字を半角 1 文字で指定します。

pdrdrefls コマンドの実行結果

```
"FMB01","FMBIDX"
```

RD エリア FMB01 の関連 RD エリアは FMBIDX です。

注意事項

更新可能なオンライン再編成を実行する場合，ここで確認した関連 RD エリアを漏れなくカレント RD エリアの切り替え対象にする必要があります。すべての関連 RD エリアのカレント RD エリアの変更を同じタイミングで実施しないと，オンライン業務のアクセスがエラーになります。

(4) HiRDB ファイルシステム領域の世代登録（pdmod コマンド）

pdmod コマンドの create generation 文で，HiRDB ファイルシステム領域の世代登録を実行します。RD エリアおよび HiRDB ファイルシステム領域の構成を次に示します。

RD エリア名	世代番号	HiRDB ファイルシステム領域	RD エリアの種類	
FMB01	0	/sdb/area	格納レコード用 RD エリア	オリジナル RD エリア
FMBIDX			インデクス格納用 RD エリア	
FMB01_1	1	/sdb/area1	格納レコード用 RD エリア	レプリカ RD エリア
FMBIDX_1			インデクス格納用 RD エリア	

pdmod コマンドの実行例

```
pdmod -a /usr/pdmod/crtgen
```

[オプションの説明]

-a：pdmod コマンドの制御文のパス名を指定します。制御文の内容を次に示します。

```
create generation for HiRDB file system area "/sdb/area1"   ...1
server name bes1                                            ...2
generation number 1                                         ...3
reproduce "/sdb/area";                                      ...4
```

1. レプリカ RD エリアの HiRDB ファイルシステム領域名（/sdb/area1）を指定します。
2. RD エリアがある BES 名を指定します。
3. 1.の HiRDB ファイルシステム領域の世代番号を指定します。
4. オリジナル RD エリアの HiRDB ファイルシステム領域名（/sdb/area）を指定します。

(5) レプリカ RD エリアの定義（pdmod コマンド）

pdmod コマンドの replicate rdarea 文で，レプリカ RD エリアの定義を実行します。「(3) 関連 RD エリアの確認（pdrdrefls コマンド）」で確認した RD エリアのレプリカ RD エリアを漏れなく定義してください。

この例では，レプリカ RD エリア FMB01_1 と FMBIDX_1 を定義します。

pdmod コマンドの実行例

```
pdmod -a /usr/pdmod/reparea
```

[オプションの説明]

-a：pdmod コマンドの制御文のパス名を指定します。制御文の内容を次に示します。

```
replicate rdarea FMB01_1     ...1
   globalbuffer glbuf01      ...2
   reproduce FMB01           ...3
   generation number 1;      ...4
replicate rdarea FMBIDX_1    ...1
   globalbuffer glbuf02      ...2
   reproduce FMBIDX          ...3
   generation number 1;      ...4
```

1. レプリカ RD エリア名（FMB01_1，FMBIDX_1）を指定します。
2. レプリカ RD エリアに割り当てるグローバルバッファを指定します。
3. オリジナル RD エリア名（FMB01，FMBIDX）を指定します。
4. レプリカ RD エリアの HiRDB ファイルシステム領域の世代番号を指定します。

なお，レプリカ RD エリアに割り当てるグローバルバッファは，通常，オリジナル RD エリアと同じものを割り当てます。メモリ資源に余裕がある場合は，オリジナル RD エリアと同じ面数のグローバルバッファを用意して，競合しないようにすることを推奨します。レプリカ RD エリアに別のグローバルバッファを割り当てるためには，HiRDB を正常終了する必要があります。

注意事項

HiRDB の次回開始以降は，ここで指定したグローバルバッファは割り当てられません。そのため，HiRDB の終了後，システム共通定義の pdbuffer オペランドでグローバルバッファを割り当ててください。

(6) 追い付き反映キー対応表，追い付き状態管理表を格納する RD エリアの作成（pdmod コマンド）

pdfmkfs コマンドで HiRDB ファイルシステム領域を作成したあとに，pdmod コマンドで次の RD エリアを作成します。

- RDRMT：追い付き状態管理表を格納する RD エリア（/rdb/rmt）

- RDCRT：追い付き反映キー対応表を格納する RD エリア（/rdb/crt）
- RDIDX：追い付き反映キー対応表のインデクスを格納する RD エリア（/rdb/crt）

（　）内は，HiRDB ファイルシステム領域名です。

次の手順で作成します。

1. HiRDB ファイルシステム領域を作成する

```
pdfmkfs -n 50 -l 10 -k DB -i /rdb/rmt
pdfmkfs -n 50 -l 10 -k DB -i /rdb/crt
```

[オプションの説明]

-n：HiRDB ファイルシステム領域の領域長をメガバイト単位で指定します。

-l：HiRDB ファイルシステム領域内に作成する HiRDB ファイル数の上限値を指定します。

-k：HiRDB ファイルシステム領域の用途を指定します。

-i：HiRDB ファイルシステム領域の全領域を初期化する場合に指定します。

/rdb/rmt：作成する HiRDB ファイルシステム領域の名称を指定します。

/rdb/crt：作成する HiRDB ファイルシステム領域の名称を指定します。

注意事項

追い付き反映キー対応表，およびそのインデクスを格納する RD エリア用の HiRDB ファイルシステム領域（この例では/rdb/crt）には，更新可能なオンライン再編成の対象 RD エリアを配置することはできません。

2. RD エリアを作成する

pdmod コマンドによる RD エリアの作成方法については，マニュアル「HiRDB システム運用ガイド」の「RD エリアを作成する方法（RD エリアの追加）」を参照してください。

注意事項

- pdmod コマンドの create rdarea 文の data model オペランドには，relational を指定してください。
- 追い付き反映処理を実行する BES ごとに，追い付き反映キー対応表を格納する RD エリアを作成し，追い付き反映キー対応表を格納します。
- 追い付き反映処理を実行する BES ごとに，追い付き反映キー対応表のインデクスを格納する RD エリアを作成し，追い付き反映キー対応表のインデクスを格納します。
- 追い付き反映処理を実行する BES ごとに，追い付き状態管理表を格納する RD エリアを作成することを推奨します（必須ではありません）。
- 追い付き反映キー対応表と追い付き状態管理表は，別々の RD エリアに格納することを推奨します。
- RD エリアは公用 RD エリアとして作成してください。

(7) 追い付き反映キー対応表の作成または初期化（pdsdborcrt コマンド）

pdsdborcrt コマンドで，追い付き反映キー対応表を作成します。

```
pdsdborcrt -k crt -r RDCRT -i RDIDX
```

[オプションの説明]

-k crt：追い付き反映キー対応表を作成するときに指定するオプションです。

-r：追い付き反映キー対応表を格納する RD エリアを指定します。

-i：追い付き反映キー対応表のインデクスを格納する RD エリアを指定します。

注意事項

- 更新可能なオンライン再編成の対象 RD エリアが，複数の BES に存在する場合，BES ごとに pdsdborcrt コマンドを実行し，各 BES に追い付き反映キー対応表を作成してください。
- 追い付き反映キー対応表，およびそのインデクスは，更新可能なオンライン再編成の対象 RD エリアとは別の RD エリアに格納してください。

なお，追い付き反映キー対応表は一度作成しておけば，次回の更新可能なオンライン再編成時に再度作成する必要はありません。ただし，この場合，追い付き反映キー対応表を pdsdborcrt コマンドで初期化する必要があります。初期化していない場合，pdorcheck コマンドの実行時にエラーとなります。

初期化する場合のコマンドの実行例

```
pdsdborcrt -k init -s bes1
```

[オプションの説明]

-k init：追い付き反映キー対応表を初期化するときに指定するオプションです。

-s：初期化対象の追い付き反映キー対応表がある BES を指定します。

(8) 適用条件のチェック（pdorcheck コマンド）

pdorcheck コマンドで，更新可能なオンライン再編成が実行できる条件を満たしているかどうかをチェックします。

```
pdorcheck -r ”FMB01”,”FMBIDX”
```

[オプションの説明]

-r：更新可能なオンライン再編成の対象 RD エリア（オリジナル RD エリア）を指定します。「(3) 関連 RD エリアの確認（pdrdrefls コマンド）」で確認した RD エリアを漏れなく指定してください。

注意事項

pdorcheck コマンドによるチェックは必ず実施してください。更新可能なオンライン再編成を適用できない RD エリアがあると，追い付き反映処理がエラーとなり，業務をオリジナル RD エリアに戻すことができなくなります（レプリカ RD エリアで業務を続行することになります）。

(9)　ペアボリュームの定義

更新可能なオンライン再編成の対象となるオリジナル RD エリアとレプリカ RD エリアが，それぞれ定義されているボリューム（HiRDB ファイルシステム領域）をペア化します。オリジナル RD エリアが定義されているボリューム（/sdb/area）への更新が，レプリカ RD エリアが定義されているボリューム（/sdb/area1）に反映される状態にします。

操作方法については，使用しているミラーリング機能のマニュアルを参照してください。日立ディスクアレイサブシステムを使用している場合は，最初にマニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「日立ディスクアレイサブシステム使用時の注意事項および操作手順」を参照してください。

(10)　追い付き状態管理表の作成（pdorcreate コマンド）

pdorcreate コマンドで，追い付き状態管理表を作成します。

```
pdorcreate -o RDRMT
```

［オプションの説明］

-o：「(6)　追い付き反映キー対応表，追い付き状態管理表を格納する RD エリアの作成（pdmod コマンド）」で作成した追い付き状態管理表を格納する RD エリアを指定します。

注意事項

追い付き状態管理表は，更新可能なオンライン再編成の対象 RD エリアとは別の RD エリアに格納してください。更新可能なオンライン再編成の対象 RD エリアに，追い付き状態管理表を格納すると，追い付き反映処理が正常に実行できません。

追い付き反映処理を実行するためには，追い付き反映処理を管理するための表（追い付き状態管理表）を作成する必要があります。

(11)　SDB ディレクトリ情報の最終更新日時のチェック（pdsdbarc -a コマンド）

pdsdbarc -a コマンドで，常用常駐領域中の SDB ディレクトリ情報の最終更新日時をチェックします。

```
pdsdbarc -a
```

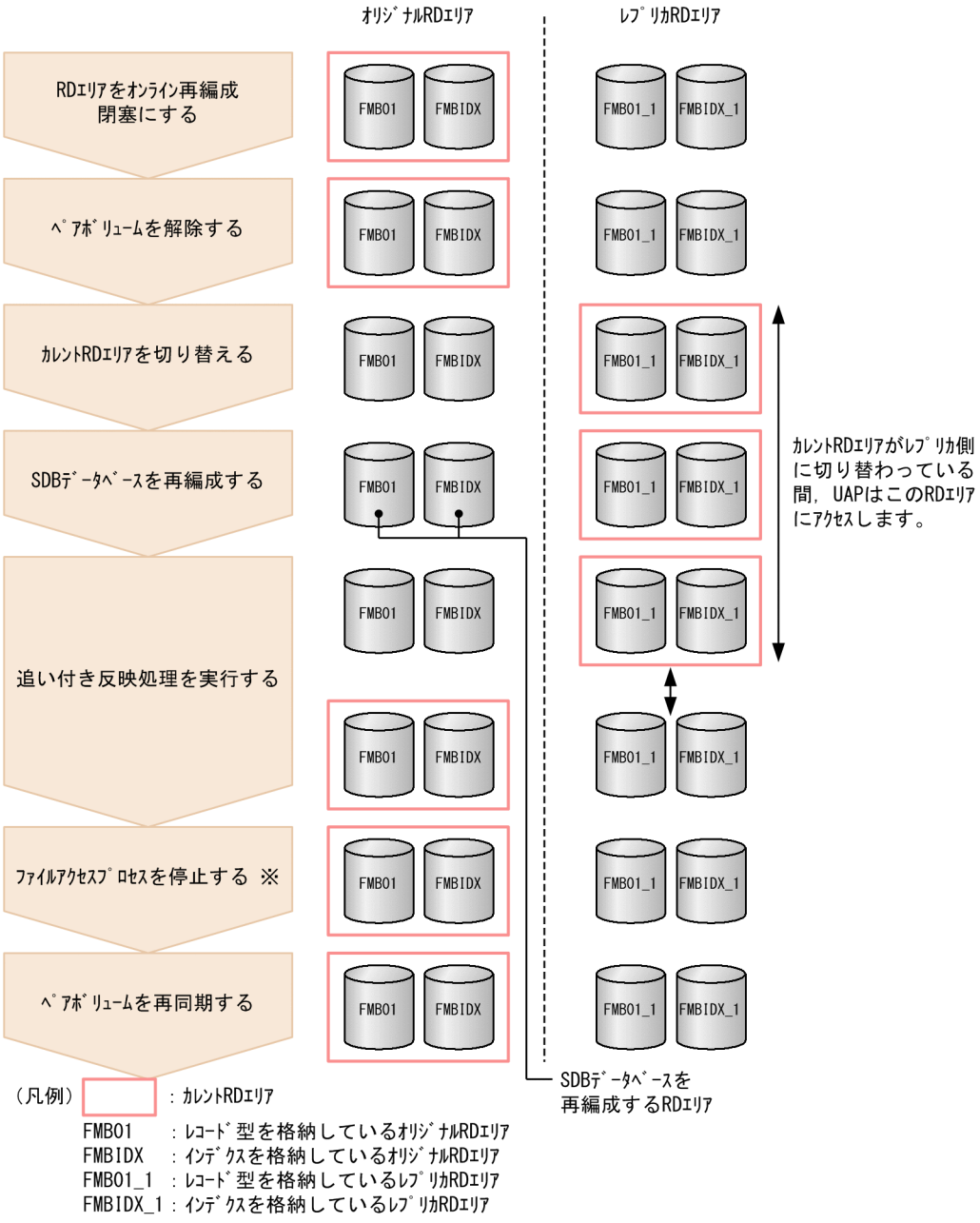
注意事項

pdsdbarc -a コマンドによるチェックは必ず実施してください。SDB ディレクトリ情報の最終更新日時が不整合な状態で更新可能なオンライン再編成を実行すると，データベースの不整合が発生するおそれがあります。

5.13.4　更新可能なオンライン再編成の運用手順

更新可能なオンライン再編成の運用手順の流れを次の図に示します。

図 5-15　更新可能なオンライン再編成の運用手順の流れ



注※

pd_lv_mirror_use オペランドに Y を指定している場合，この操作は必要ありません。

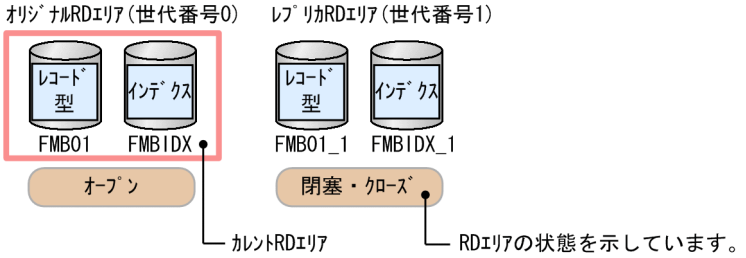
[説明]

オリジナル RD エリアで SDB データベースの再編成を実行し，レプリカ RD エリアで業務（参照および更新処理）を続行します。

更新可能なオンライン再編成の運用手順を例題形式で説明します。なお，例題の説明は，インナレプリカ機能についての知識があることを前提にしています。

例題

オリジナル RD エリアに格納されている SDB データベースに対して，更新可能なオンライン再編成を実施します。レコード型およびインデクスを格納している RD エリアは次のとおりです。



なお，この例では，RD エリアのオープン属性は，オリジナル RD エリアが INITIAL，レプリカ RD エリアが DEFER であるとしています。

(1) RD エリアをオンライン再編成閉塞にする（pdorbegin コマンド）

pdorbegin コマンドで，SDB データベースが格納されている RD エリア（オリジナル RD エリアとレプリカ RD エリアの両方）をオンライン再編成閉塞にします。

```
pdorbegin -t REC1 -q 1 -w 60
```

[オプションの説明]

-t：ルートレコードのレコード型名を指定します。

-q：オンライン再編成閉塞にするレプリカ RD エリアの世代番号を指定します。

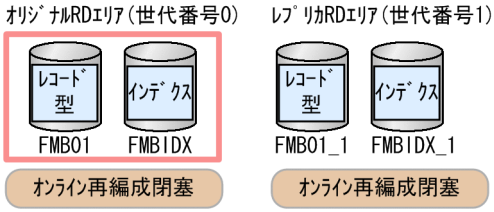
-w：RD エリアの状態をオンライン再編成閉塞にする際の，排他待ち時間を秒単位で指定します。

pdorbegin コマンドを実行すると，次の RD エリアがオンライン再編成閉塞になります。

- オリジナル RD エリア
- -q オプションに指定した世代のレプリカ RD エリア

この例では，FMB01，FMBIDX，FMB01_1，および FMBIDX_1 がオンライン再編成閉塞になります。

このコマンドの実行後から，「(3) カレント RD エリアを切り替える（pdorchg コマンド）」までの間は，オンライン再編成閉塞の RD エリアにアクセスするトランザクションは待ち状態になります。



■RD エリア名指定をする場合

次のように-r オプションに RD エリア名を指定して，更新可能なオンライン再編成の対象 RD エリアをオンライン再編成閉塞にすることもできます。この場合，BES 内の関連 RD エリア（FMB01，FMBIDX）を漏れなく指定してください。

```
pdorbegin -r "FMB01","FMBIDX" -q 1 -w 60
```

[オプションの説明]

-r：オンライン再編成閉塞にするオリジナル RD エリアを指定します。

-q：オンライン再編成閉塞にするレプリカ RD エリアの世代番号を指定します。

-w：RD エリアの状態をオンライン再編成閉塞にする際の，排他待ち時間を秒単位で指定します。

pdorbegin コマンドを実行すると，-r オプションに指定したオリジナル RD エリアと，-q オプションに指定した世代のレプリカ RD エリアが，オンライン再編成閉塞になります。

参考

pdrdrefls コマンドで関連 RD エリアを確認できます。詳細については，「5.13.3(3)　関連 RD エリアの確認（pdrdrefls コマンド）」を参照してください。

(2)　ペアボリュームを解除する

「5.13.3(9)　ペアボリュームの定義」で設定したペアボリュームを解除してください。解除方法については，使用しているミラーリング機能のマニュアルを参照してください。日立ディスクアレイサブシステムを使用している場合は，最初にマニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「日立ディスクアレイサブシステム使用時の注意事項および操作手順」を参照してください。

(3)　カレント RD エリアを切り替える（pdorchg コマンド）

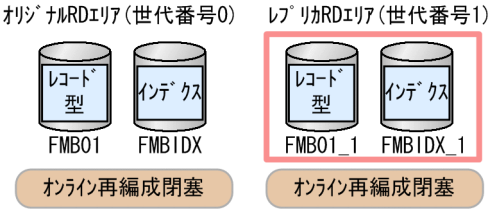
pdorchg コマンドで，カレント RD エリアを切り替えてください。

```
pdorchg -s bes1
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

pdorchg コマンドの完了後，トランザクションの待ち状態は解除され，レプリカ RD エリアにアクセスすることで業務が再開されます。



(4) オリジナル RD エリアのバックアップを取得する（pdcopy コマンド）

pdcopy コマンドで，オリジナル RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r FMB01,FMBIDX -q 0
-b /bkdir/bkup01 -p /bkdir/list01
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：オリジナル RD エリア名を指定します。

-q：バックアップ対象の RD エリアの世代番号を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

この時点でのバックアップの取得は任意ですが，SDB データベースの再編成時の障害に備えてバックアップを取得することをお勧めします。バックアップを取得する場合は，pdcopy コマンドの実行前に，pdlogswap -d sys -w コマンドを実行してシステムログファイルをスワップしてください。

(5) オリジナル RD エリアのレコードをアンロードする（pdsdbrog コマンド）

pdsdbrog コマンドで，オリジナル RD エリアのレコードをアンロードします。

```
pdsdbrog /usr/sdb/control_file1
```

[オプションの説明]

/usr/sdb/control_file1：pdsdbrog 制御文ファイル名を指定します。pdsdbrog 制御文の内容を次に示します。

```
environment
    schema = FMB                    ...1
    generation = 0                  ...2
unload
    server = bes1                   ...3
    record = REC1                   ...4
    unldfile = /usr/sdb/unload1     ...5
    prefix = 5                      ...6
    unldkind = lod                  ...7
dbinf
```

```
    userpflg =(0)                   ...8
    pagenum = (1)                   ...9
```

1. SDB データベース名を指定します。
2. アンロード対象の RD エリアの世代番号を指定します。
3. アンロード対象の RD エリアがある BES を指定します。
4. ルートレコードのレコード型名を指定します。
5. アンロードデータファイル名を指定します。
6. プリフィクス部の長さを指定します。
7. データロードで使用できる入力データファイルの形式でアンロードデータファイルを作成します。
8. USER ポインタフラグの格納位置を指定します。
9. 事前割り当てページ数の格納位置を指定します。

(6) オリジナル RD エリアにレコードをデータロードする（pdsdblod コマンド）

pdsdblod コマンドで，オリジナル RD エリアにレコードをデータロードします。

```
pdsdblod /usr/sdb/control_file2
```

[オプションの説明]

/usr/sdb/control_file2：pdsdblod 制御文ファイル名を指定します。pdsdblod 制御文の内容を次に示します。

```
environment
    schema = FMB                    ...1
    purge = yes                     ...2
    logmode = all                   ...3
    generation = 0                  ...4
load
    server = bes1                   ...5
    type = tree                     ...6
    record = REC1                   ...7
    data = /usr/sdb/unload1         ...8
    prefix = 5                      ...9
    dupkeyck = no                   ...10
    seqkeyck = no                   ...11

dbinf
    userpflg =(0)                   ...12
    pagenum = (1)                   ...13
oreload
    midfile = /usr/sdb/midfile1     ...14
```

1. SDB データベース名を指定します。アンロード時に指定した SDB データベース名を指定します。
2. 既存レコードを削除してデータロードを実行します。必ず yes を指定してください。

3. ログ取得モードでデータロードを実行します。データロードは，ログ取得モードで実行することを推奨します。
4. データロード対象の RD エリアの世代番号を指定します。アンロード時に指定した世代番号を指定します。
5. データロード対象の RD エリアがある BES を指定します。アンロード時に指定した BES 名を指定します。
6. 4V FMB の SDB データベースのデータロードのため，tree を指定します。
7. ルートレコードのレコード型名を指定します。アンロード時に指定したレコード型名を指定します。
8. 「(5)　オリジナル RD エリアのレコードをアンロードする（pdsdbrog コマンド)」で作成したアンロードデータファイルを入力データファイルとして指定します。
9. プリフィクス部の長さを指定します。
10. 入力データと格納済みレコードとのキー重複チェックを実行しない指定をします。
11. 入力データのシーケンシャルインデクスのキー順をチェックしない指定をします。
12. USER ポインタフラグの開始位置を指定します。アンロード時に指定した格納位置を指定します。
13. 事前割り当てページ数の開始位置を指定します。アンロード時に指定した格納位置を指定します。
14. 追い付き反映キー対応表の中間ファイルを格納するディレクトリ名を指定します。

注意事項

- アンロード時に作成したアンロードデータファイルを，データロード時の入力データファイルに指定してください（入力データファイルの指定を間違わないようにしてください)。アンロードデータファイルの内容とデータロードの結果を基に，追い付き反映キー対応表にデータが登録されます。このため，間違ったアンロードデータファイルを入力データファイルに指定した場合，SDB データベースの内容が不整合になります。
- データロードの再実行に備えて，更新可能なオンライン再編成が完了するまで，アンロードデータファイルを削除しないでください。データロードが正常終了したあと（KFPB63012-I メッセージが出力されたあと）に，データロードの再実行が必要になった場合，pdsdbrog コマンドでアンロードを再実行してアンロードデータファイルを作成しても，アンロードデータファイルの内容はデータロード後の内容になっています。

(7)　オリジナル RD エリアのバックアップを取得する（pdcopy コマンド）

pdcopy コマンドで，オリジナル RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r FMB01,FMBIDX -q 0
-b /bkdir/bkup02 -p /bkdir/list02
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：オリジナル RD エリア名を指定します。

-q：バックアップ対象の RD エリアの世代番号を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

pdcopy コマンドの実行前に，pdlogswap -d sys -w コマンドを実行してシステムログファイルをスワップしてください。

(8) 追い付き反映キー対応表を格納している RD エリアのバックアップを取得する（pdcopy コマンド）

pdcopy コマンドで，追い付き反映キー対応表，および追い付き反映キー対応表のインデクスを格納している RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r RDCRT,RDIDX
-b /bkdir/bkup03 -p /bkdir/list03
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：追い付き反映キー対応表，および追い付き反映キー対応表のインデクスを格納している RD エリアを指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

(9) 追い付き反映処理を実行する（pdorend コマンド）

pdorend コマンドで，追い付き反映処理を実行します。

```
pdorend -s bes1
```

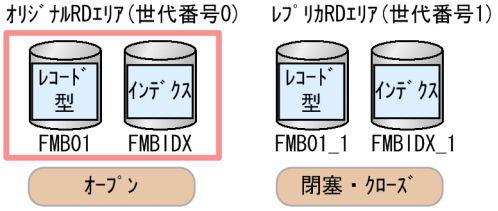
[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

pdorend コマンドを実行すると，次の処理が行われます

- レプリカ RD エリアに対する更新処理が，オリジナル RD エリアに反映されます。
- オリジナル RD エリアとレプリカ RD エリアの状態の最終的な同期を取るため，一時的に該当する RD エリアのレプリカグループ構成に排他を掛けます。このとき，レプリカ RD エリアに対する処理は排他待ちになります。
- カレント RD エリアが，オリジナル RD エリアに切り替わります。

- オリジナル RD エリアおよびレプリカ RD エリアのオンライン再編成閉塞が解除されます。RD エリアの状態は，次のようになります。



(10) ファイルアクセスプロセスを停止する（pdpfresh コマンド）

pd_lv_mirror_use オペランドに Y を指定している場合，(10)および(11)の操作は必要ありません。

pdpfresh コマンドで，ファイルアクセスプロセスを停止します。

```
pdpfresh -s bes1
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

pdpfresh コマンドを実行すると，レプリカ RD エリアの HiRDB ファイルシステム領域にアクセスしているプロセスを非同期に終了させます。強制的にリフレッシュする場合は，-f オプションを指定してください。

なお，更新可能なオンライン再編成の対象 RD エリアのオープン属性が SCHEDULE の場合は，この操作は必要ありません。

(11) レプリカ RD エリアにアクセスしているプロセスが存在しないことを確認する（pdpfresh コマンド）

pdpfresh コマンドで，レプリカ RD エリアの HiRDB ファイルシステム領域にアクセスしているプロセスが存在しないことを確認します。

```
pdpfresh -c
```

[オプションの説明]

-c：サーバプロセスのリフレッシュ状態を表示するオプションです。

pdpfresh コマンドのリターンコードが 0 の場合，レプリカ RD エリアの HiRDB ファイルシステム領域にアクセスしているプロセスは存在していないため，ペアボリュームを再同期できます。

リターンコードが 0 以外の場合，次のコマンドで HiRDB ファイルシステム領域にアクセスしているプロセスを特定してください。

```
pdls -d prc -s bes1 -c
```

［オプションの説明］

-d prc：プロセスの状態を表示するオプションです。

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-c：サーバプロセスによるファイルアクセス状態を表示するオプションです。

pdls コマンドの実行結果

```
HOSTNAME : sdb-bs2000-1 (112622)
STATUS  PID     UID   GID   SVID     TIME    PROGRAM    C-PID  C-GRP
  C     13805   339   300   bes1     999999  UAP1
  L     13806   339   300   bes1     999999
```

［説明］

STATUS 欄に C が表示されているプロセスが，レプリカ RD エリアの HiRDB ファイルシステム領域にアクセスしている可能性があります。このプロセスを強制的にリフレッシュしてよい場合は，pdpfresh -f コマンドを実行してください。このプロセスを強制的にリフレッシュしてはいけない場合は，このプロセスが実行中のジョブの終了まで待つか，またはそのジョブを終了させるような対処をしてください。

(12)　ペアボリュームを再同期する

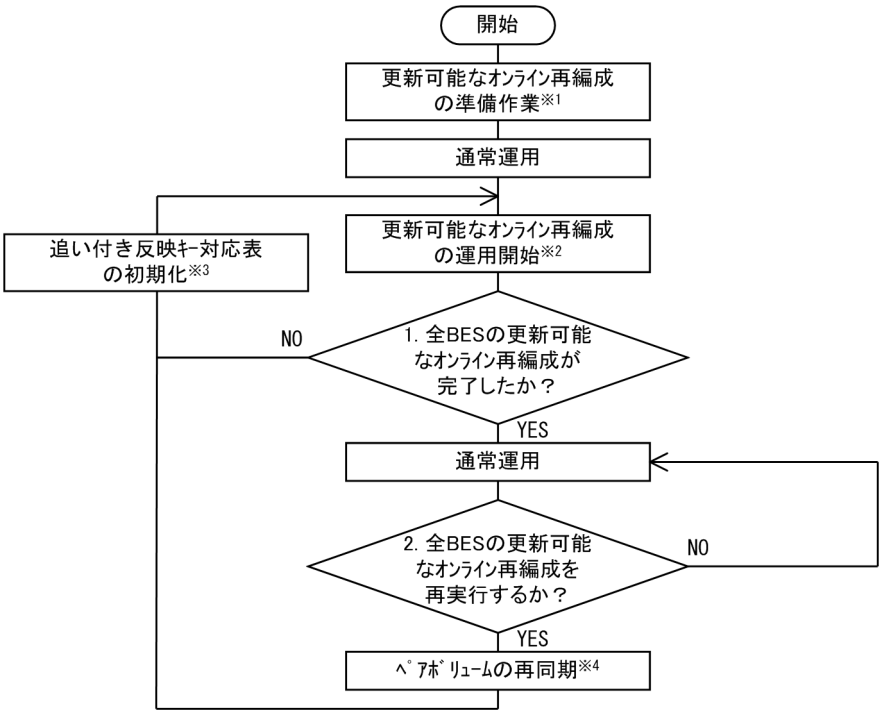
解除したペアボリュームを再同期します。操作方法については，使用しているミラーリング機能のマニュアルを参照してください。日立ディスクアレイサブシステムを使用している場合は，最初にマニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「日立ディスクアレイサブシステム使用時の注意事項および操作手順」を参照してください。

なお，更新可能なオンライン再編成の実行中に障害が発生した場合の対処方法については，「5.23　障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】」を参照してください。

5.13.5　更新可能なオンライン再編成を複数回実行する場合

更新可能なオンライン再編成を複数回実行する場合の手順の流れを次の図に示します。

図 5-16　更新可能なオンライン再編成を複数回実行する場合の手順の流れ



[説明]

全 BES の追い付き反映処理が完了したあとに，ペアボリュームを再同期します（図中の 1.，2.は次の説明に対応しています）。

1. 再編成対象の資源が多く，1 回の再編成運用タイミングで複数回に分けて更新可能なオンライン再編成を実行する場合は繰り返してください。
2. 以前，再編成を実施した際に準備作業は完了しており，追い付き反映キー対応表が作成済みであれば，ペアボリュームを再同期してから手順に従ってください。

注※1

詳細については，「5.13.3　更新可能なオンライン再編成の準備作業」を参照してください。

注※2

詳細については，「5.13.4　更新可能なオンライン再編成の運用手順」を参照してください。ただし，ペアボリュームの再同期は不要です。

注※3

詳細については，「5.13.3(7)　追い付き反映キー対応表の作成または初期化（pdsdborcrt コマンド）」を参照してください。

注※4

詳細については，「5.13.4(12)　ペアボリュームを再同期する」を参照してください。

5.13.6　追い付き反映キー対応表を格納しているRDエリアを確認したい場合

追い付き反映キー対応表を格納しているRDエリアや，追い付き反映キー対応表の作成有無を確認したい場合は，pddbstコマンドの表単位の状態解析を実行してください。

(例)

BES（BES名：bes1）に追い付き反映キー対応表が作成されているかを確認します。また，追い付き反映キー対応表を格納しているRDエリアを確認します。

```
pddbst -t "ORG_USER"."PDORSDB_bes1"
```

[オプションの説明]

-t：追い付き反映キー対応表の表名を指定します。

追い付き反映キー対応表の表名は次の規則に従って決まります。

- 認可識別子は"ORG_USER"固定です。
- 表識別子は"PDORSDB_BES名"になります。

上記のコマンドの指定例では，「bes1」の部分はBES名になります。

5.13.7　更新可能なオンライン再編成の対象RDエリアの状態を確認したい場合

RDエリアの状態はpddblsコマンド，またはpddbstコマンドで確認できます。レプリカRDエリアを含めたデータベースの状態表示については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「データベースの状態表示」を参照してください。

5.13.8　追い付き反映処理の状態を確認したい場合

pdorendコマンドによる追い付き反映処理の状態をpdls -d orgコマンドで確認できます。オンライン再編成の追い付き反映の経過情報の確認方法については，マニュアル「HiRDB コマンドリファレンス」の「pdls【-d org】(オンライン再編成の追い付き反映の経過情報)」を参照してください。

なお，実行結果のSTATUS欄がREADYまたはREFLECの場合，REMAIN_LOG_RECORD_COUNTにはカレントRDエリアの未反映ログレコード数が表示されます。pdorendコマンドの実行前後で，REMAIN_LOG_RECORD_COUNTの値を比較すると，追い付き反映処理と追い付き反映処理中のトランザクション量の関係がわかります。pdorendコマンド実行後のREMAIN_LOG_RECORD_COUNTの値が，実行前に比べて大きくなっている場合は，トランザクションの量を制限して，pdorendコマンドを再実行する必要があります。

5.13.9　更新可能なオンライン再編成の操作を取り消したい場合

pdorchg コマンドによるカレント RD エリアの切り替え前（「5.13.4(3)　カレント RD エリアを切り替える（pdorchg コマンド）」の操作の前）であれば，更新可能なオンライン再編成の操作を取り消すことができます。操作を取り消す場合は，pdorbegin -u コマンドを実行します。コマンドの指定例を次に示します。

- （例 1）更新可能なオンライン再編成をレコード型名指定で開始した場合

```
pdorbegin -t REC1 -u
```

[オプションの説明]

-t：ルートレコードのレコード型名を指定します。

-u：オンライン再編成閉塞を解除するオプションです。

- （例 2）更新可能なオンライン再編成を RD エリア名指定で開始した場合

```
pdorbegin -r FMB01,FMBIDX -u
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-u：オンライン再編成閉塞を解除するオプションです。

注意事項

pdorchg コマンドによるカレント RD エリアの切り替え後（「5.13.4(3)　カレント RD エリアを切り替える（pdorchg コマンド）」の操作のあと）は，更新可能なオンライン再編成の操作を取り消すことはできません。

カレント RD エリアの切り替え後に，更新可能なオンライン再編成の操作を取り消すと，レプリカ RD エリアへの更新処理がオリジナル RD エリアに反映されないため，SDB データベースの内容が不整合な状態になります。

誤って取り消してしまった場合は，データロードを実行しないで，追い付き反映処理だけを実行してください。そうすると，レプリカ RD エリアに対する更新処理をオリジナル RD エリアに反映することができます。

また，この場合，追い付き反映キー対応表の初期化を必ず実施してください。初期化しないで，追い付き反映キー対応表に前回の更新可能なオンライン再編成時のデータが残っていると，SDB データベースの内容が不整合な状態になります。

5.13.10　追い付き反映処理時に HiRDB/SD が内部的に発行する API のエラーをスキップしたい場合

pdorend コマンドによる追い付き反映処理時のルートレコード検索で，HiRDB/SD が内部的に発行する API でエラー（対象レコードが存在しない）が発生した場合，追い付き反映処理がエラー終了します。この API のエラーをスキップして，追い付き反映処理を続行したい場合は，pdorend コマンドに-f オプションを指定して，追い付き反映制御ファイルに次の指定をしてください。

- reflection 文の dml_error オペランドに skip を指定する

エラーとなった API の情報は，追い付き反映制御ファイルの report 文の dml_skip_info オペランドに指定した DML スキップ情報出力ファイルに出力されます。

アンロードデータファイルの内容を変更し，ファミリ単位の削除をした場合に，HiRDB/SD が内部的に発行する API でエラーが発生する可能性があります。ファミリ単位の削除を行ったレコードに対して，カレント RD エリアで更新が発生した場合，オリジナル RD エリアでは対応するレコードがないため，追い付き反映処理での HiRDB/SD が内部的に発行する API でエラーが発生します。

この場合，DML スキップ情報出力ファイルに出力された情報を確認し，スキップ対象となったデータを SDB データベースに反映するかどうかを判断してください。反映が必要な場合は，SDB データベースを操作する API などでデータを追加し，SDB データベースの整合性を回復してください。

DML スキップ情報出力ファイルの出力形式については，「10.10.4(7)　DML スキップ情報出力ファイルの出力形式」を参照してください。

5.14　RD エリアの運用

ここでは，RD エリアの空き容量の確認方法と，RD エリアの追加，容量拡張，削除，および設定変更方法について説明します。

5.14.1　RD エリアの空き容量の確認

レコードの追加および削除が頻繁に行われると，RD エリアの空き容量が少なくなることがあります。そのため，このような RD エリアに対しては，pddbls コマンドで RD エリアの空き容量を定期的に確認するようにしてください。

RD エリアの空き容量の確認方法を，例を使って説明します。

例

RD エリア（RDAREA1）の空き容量を確認します。

```
pddbls -r RDAREA1 -a

STATE OF RDAREA
    RDAREA        ID       STATUS                TYPE
                           OPNMODE
    RDAREA1       4        OPEN                  USER
                           INITIAL
    SEGMENT    75 / 700
```

[説明]

SEGMENT 欄に RD エリアの未使用セグメント数と全セグメント数が表示されます。未使用セグメント数が RD エリアの空き容量になります。この例では，未使用セグメント数が 75 で，全セグメント数が 700 です。未使用セグメント数が残り少なくなったら，RD エリアを拡張するなどの対処を行います。

参考

次に示すメッセージが同じ RD エリアに対して頻繁に出力される場合は，RD エリアの空き容量が少なくなっているおそれがあります。pddbls コマンドで RD エリアの空き容量を確認してください。

- KFPA12300-I
- KFPH00211-I
- KFPH00212-I

5.14.2　RD エリアの追加

pdmod コマンドの create rdarea 文で RD エリアを追加できます。RD エリアの追加手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアを作成する方法（RD エリアの追加)」を参照してください。pdmod コマンドについては，「10.3.1　RD エリアの追加」およびマニュアル「HiRDB コマンドリファレンス」を参照してください。

5.14.3　RD エリアの容量拡張

pdmod コマンドの expand rdarea 文で，RD エリアの容量を拡張できます。RD エリアの容量拡張手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアの容量を大きくする方法（RD エリアの拡張)」を参照してください。pdmod コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

参考

pdmod コマンドの RD エリアの再初期化（initialize rdarea 文）でも，RD エリアの容量を拡張できます。RD エリアの再初期化手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアの容量を大きく，または属性を変更する方法（RD エリアの再初期化)」を参照してください。

5.14.4　RD エリアの削除

不要になった RD エリアを pdmod コマンドの remove rdarea 文で削除できます。RD エリアの削除手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアを削除する方法」を参照してください。pdmod コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

なお，削除対象の RD エリアにレコードまたはインデクスがある場合は，SDB データベースを削除してから RD エリアを削除してください。

5.14.5　RD エリアの設定変更

RD エリアの次に示す設定を pdmod コマンドの initialize rdarea 文で変更できます。

- RD エリアの自動増分の設定
- RD エリアのオープン契機の設定
- RD エリアのページ長およびセグメントサイズ
- RD エリアに割り当てる HiRDB ファイルシステム領域

なお，RD エリアの自動増分の設定とオープン契機の設定は，pdmod コマンドの alter rdarea 文でも変更できます。

RD エリアの設定変更手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアの容量を大きく，または属性を変更する方法（RD エリアの再初期化)」，または「RD エリアのオープン契機を変更する方法（RD エリアの属性変更)」を参照してください。pdmod コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

参考

RD エリア名の変更はできません。

5.15　SDB データベースの横分割数を増やす方法

SDB データベースの横分割数を増やす際の作業の流れを次の図に示します。

図 5-17　SDB データベースの横分割数を増やす際の作業の流れ

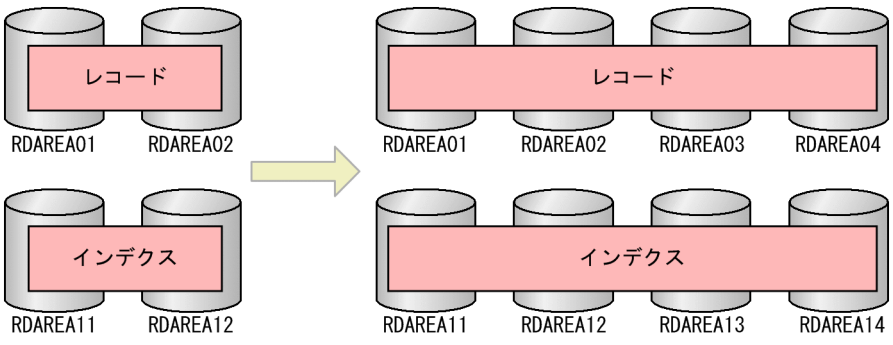
注※
SDBデータベースを再定義する場合や，分割格納条件の変更機能で
RDエリアの格納条件を変更する場合に必要な作業です。

SDB データベースの横分割数を増やす際の手順を例題形式で説明します。

5.15.1　横分割数を増やす手順（再定義をする場合）

例題

次に示す構成の SDB データベースの横分割数を 2 から 4 に増やします。SDB データベースを格納している RD エリアは同一バックエンドサーバ内にあります。



- RDAREA01 に格納していたレコードは，RDAREA01 および RDAREA02 に格納します。
- RDAREA02 に格納していたレコードは，RDAREA03 および RDAREA04 に格納します。

手順

1. RD エリアを追加する

pdmod コマンドの create rdarea 文で RD エリア（RDAREA03，RDAREA04，RDAREA13，および RDAREA14）を追加します。RD エリアの追加手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアを作成する方法（RD エリアの追加）」を参照してください。pdmod コマンドについては，「10.3.1　RD エリアの追加」およびマニュアル「HiRDB コマンドリファレンス」を参照してください。

なお，追加した RD エリアにグローバルバッファを割り当ててください。

2. SDB データベースの定義変更を行う

SDB データベースの横分割数を再定義で変更する場合，いったん SDB データベースの定義削除をしたあとに，新たに SDB データベースの定義追加をする必要があります。その際，HiRDB を再起動する必要があるため，「5.6　SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合）」で説明している手順に従って，SDB データベースの定義削除および定義追加をしてください。

なお，「5.6.3　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）」の手順の説明について，次の点を考慮してください。

- 「5.6.3(3)　レコードをアンロードする」および「5.6.3(11)　SDB データベースのデータロードを実行する」の作業は実施してください。
- 「5.6.3(5)　SDB データベースの定義追加，定義変更，または定義削除を行う」では，「5.6.5(2) SDB データベースを再定義する方法」を参照してください。
- SDB データベース格納定義の変更例を次に示します。

  SDB データベース格納定義の変更例

```
STORAGE SCHEMA SCM01 FOR SCM01
 DBTYPE 4V
 STORAGE RECORD REC01
  SEQUENTIAL INDX01 FOR RECORD
     :
    WITHIN ((RDAREA11),(RDAREA12),(RDAREA13),(RDAREA14))
  DEPENDING ON C1
    WITHIN ((RDAREA01)100,(RDAREA02)200,(RDAREA03)300,(RDAREA04))
      :
END STORAGE SCHEMA
```

  下線部分が RD エリアに関する指定です。

3. RD エリアのバックアップを取得する

pdcopy コマンドで RD エリア（RDAREA01～RDAREA04，RDAREA11～RDAREA14，およびシステム用 RD エリア）のバックアップを取得します。バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。pdcopy コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

5.15.2　横分割数を増やす手順（HiRDB の再起動を必要としない場合）

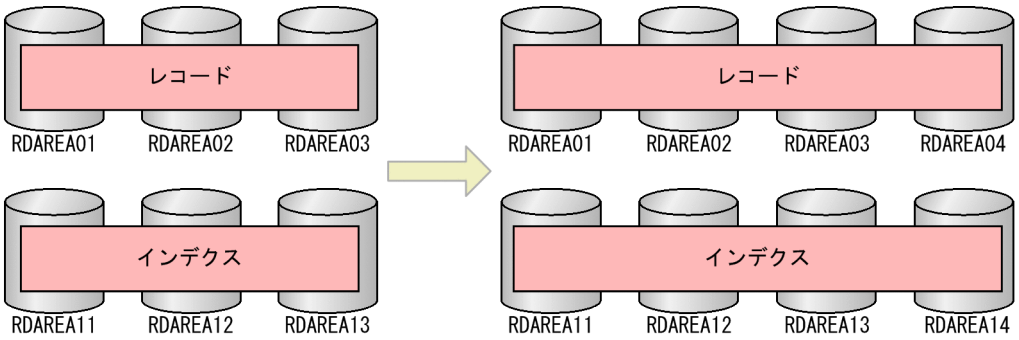
格納条件指定で SDB データベースを横分割しているときに，新たな分割キーを追加する場合に限り，HiRDB の再起動なしに横分割数を増やすことができます。

そのほかの場合は，RD エリアの格納条件を変更することになるため，HiRDB の再起動が必要となります。この場合の手順は，「5.15.1　横分割数を増やす手順（再定義をする場合）」，または「5.15.3　横分割数を増やす手順（HiRDB の再起動を必要とする場合）」を参照してください。

なお，SD FMB の SDB データベースの横分割数を増やす場合には，SDB データベースの再定義が必要になります。この場合の手順については，「5.15.1　横分割数を増やす手順（再定義をする場合）」を参照してください。

例題

次に示す構成の SDB データベースの横分割数を 3 から 4 に増やします。SDB データベースを格納している RD エリアは同一バックエンドサーバ内にあります。



- 既存のレコード格納用 RD エリア RDAREA01～RDAREA03 に，RDAREA04 を追加します。
- RDAREA04 には，新たな分割キー値を持つレコードを格納します。

手順

1. RD エリアを追加する

pdmod コマンドの create rdarea 文で RD エリア（RDAREA04 および RDAREA14）を追加します。RD エリアの追加手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアを作成する方法（RD エリアの追加）」を参照してください。pdmod コマンドについては，「10.3.1　RD エリアの追加」およびマニュアル「HiRDB コマンドリファレンス」を参照してください。

なお，追加した RD エリアにグローバルバッファを割り当ててください。

2. SDB データベースの定義変更を行う

SDB データベースの横分割数を増やす場合は，*ALTER DICTIONARY 文および*ALTER DIRECTORY 文で変更できます。

その際，既存の RD エリア（RDAREA01～RDAREA03）に格納するデータに影響がない格納条件の変更の場合は，HiRDB の再起動は必要ありません。「5.7　SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合）」で説明している手順に従って，SDB データベースの定義変更をしてください。

なお，「5.7.3　SDB データベースの定義追加または定義変更の手順（各手順の詳細）」の手順の説明については，次の点を考慮してください。

- 「5.7.3(8)　トランザクションを静止化する」の手順までは，既存の RD エリア（RDAREA01～RDAREA03）に対して，UAP のアクセスができます。
- 「5.7.3(12)　トランザクションの静止化を解除する」の手順以降から，追加した RD エリア（RDAREA04）へのアクセスができます。

3. RD エリアのバックアップを取得する

pdcopy コマンドで RD エリア（RDAREA01～RDAREA04，RDAREA11～RDAREA14，およびシステム用 RD エリア）のバックアップを取得します。バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。pdcopy コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

5.15.3　横分割数を増やす手順（HiRDB の再起動を必要とする場合）

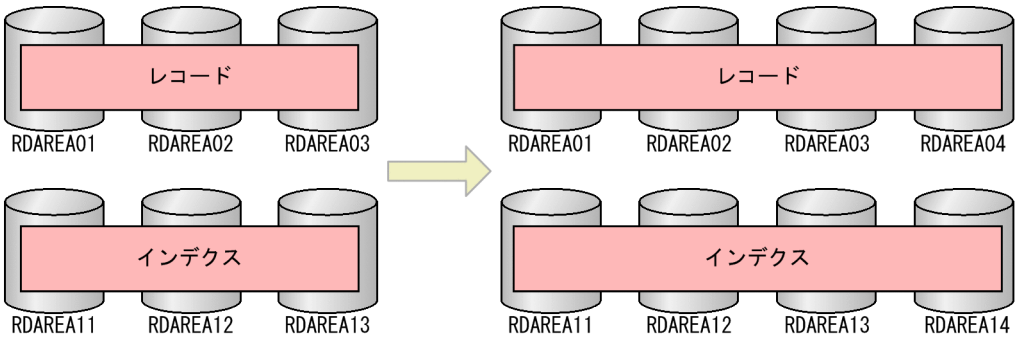
1 つの RD エリアに格納されたデータを複数の RD エリアに分割格納する場合は，HiRDB の再起動が必要となります。

なお，境界値指定で SDB データベースを横分割している場合は，RD エリアの格納条件を必ず変更することになるため，HiRDB の再起動が必要になります。

なお，SD FMB の SDB データベースの横分割数を増やす場合には，SDB データベースの再定義が必要になります。この場合の手順については，「5.15.1　横分割数を増やす手順（再定義をする場合）」を参照してください。

例題

次に示す構成の SDB データベースの横分割数を 3 から 4 に増やします。SDB データベースを格納している RD エリアは同一バックエンドサーバ内にあります。



- RDAREA03 に格納していたレコードを，RDAREA03 および RDAREA04 に格納します。

手順

1. RD エリアを追加する

pdmod コマンドの create rdarea 文で RD エリア（RDAREA04 および RDAREA14）を追加します。RD エリアの追加手順については，マニュアル「HiRDB システム運用ガイド」の「RD エリアを作成する方法（RD エリアの追加）」を参照してください。pdmod コマンドについては，「10.3.1　RD エリアの追加」およびマニュアル「HiRDB コマンドリファレンス」を参照してください。

なお，追加した RD エリアにグローバルバッファを割り当ててください。

2. SDB データベースの定義変更を行う

SDB データベースの横分割数を増やす場合は，*ALTER DICTIONARY 文および*ALTER DIRECTORY 文で変更できます。

その際，既存の RD エリア（RDAREA03）に格納しているデータに影響がある格納条件の変更の場合，HiRDB の再起動が必要となります。そのため，pdsdbdef コマンドの実行時は，-i オプションを指定してください。

「5.6　SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合）」で説明している手順に従って，SDB データベースの定義変更をしてください。

なお，「5.6.3　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）」の手順の説明については，次の点を考慮してください。

- 「5.6.3(3)　レコードをアンロードする」の手順では，格納条件を変更する RD エリア（RDAREA03）だけをアンロードしてください。
- 「5.6.3(11)　SDB データベースのデータロードを実行する」の手順では，格納条件を変更した RD エリア（RDAREA03,RDAREA04）に対してだけデータロードを実行します。このとき，格納条件に合致しないデータは，エラーデータとして格納処理をスキップするため，エラー情報を出力しないように，pdsdblod コマンドの divermsg オペランドに noput を指定してください。

3. RD エリアのバックアップを取得する

pdcopy コマンドで RD エリア（RDAREA01～RDAREA04，RDAREA11～RDAREA14，およびシステム用 RD エリア）のバックアップを取得します。バックアップの取得方法については，マニュアル「HiRDB システム運用ガイド」の「バックアップの取得方法」を参照してください。pdcopy コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

5.16　セキュリティに関する運用

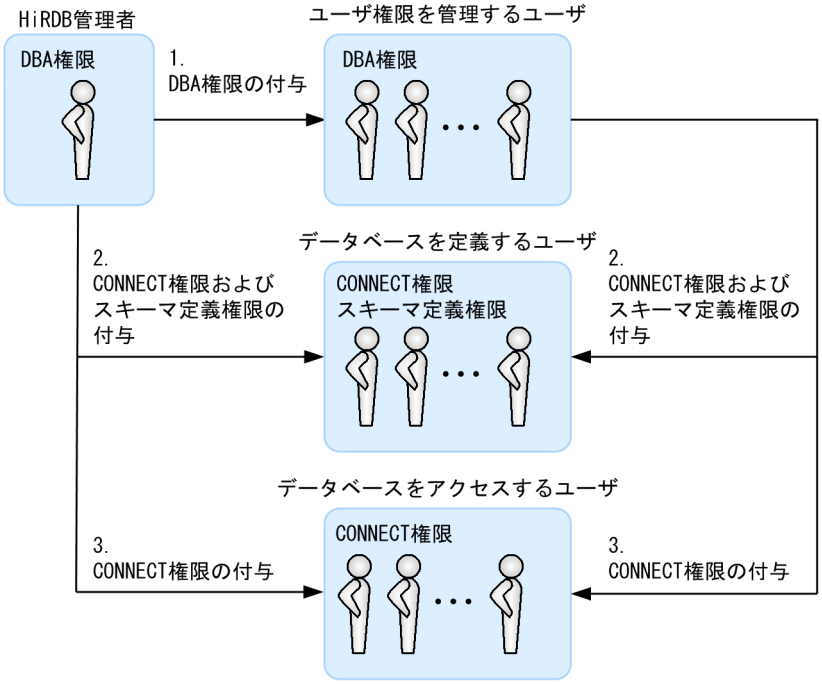
ここでは，機密保護機能を使用したユーザ権限の設定方法と，ディクショナリ表の参照権限の設定方法について説明します。

5.16.1　構造型 DB 機能に関するユーザ権限の設定

(1)　ユーザ権限付与の流れ

HiRDB をインストールすると，HiRDB 管理者が最初の HiRDB のユーザとして自動的に登録されます。HiRDB 管理者には DBA 権限が与えられているため，最初は HiRDB 管理者がほかのユーザに対してユーザ権限を与えます。ユーザ権限付与の流れを次の図に示します。

図 5-18　ユーザ権限付与の流れ



[説明]

1. HiRDB 管理者が，ユーザ権限を管理するユーザに DBA 権限を与えます。HiRDB 管理者がユーザ権限を管理する場合は，この作業は必要ありません。
2. HiRDB 管理者，またはユーザ権限を管理するユーザ（HiRDB 管理者から DBA 権限を与えてもらったユーザ）が，データベースを定義するユーザに CONNECT 権限およびスキーマ定義権限を与えます。
3. HiRDB 管理者，またはユーザ権限を管理するユーザ（HiRDB 管理者から DBA 権限を与えてもらったユーザ）が，データベースをアクセスするユーザに CONNECT 権限を与えます。

参考

マニュアル「HiRDB システム運用ガイド」の「機密保護の設定方法」にユーザ権限についての説明があります。そこでは，次に示す権限についての説明がありますが，HiRDB/SD では次に示す権限を設定する操作をしても無効となります。

- アクセス権限
- RD エリア利用権限

(2) ユーザ権限を管理するユーザに DBA 権限を与える

DBA 権限を与える場合は，定義系 SQL の GRANT 文を実行します。

例題

ユーザ（認可識別子：USER001，パスワード：HIRDB001）に DBA 権限を与えます。

```
GRANT DBA TO USER001 IDENTIFIED BY HIRDB001
```

DBA 権限を削除する場合は，定義系 SQL の REVOKE 文を実行します。

例題

ユーザ（認可識別子：USER001）の DBA 権限を削除します。

```
REVOKE DBA FROM USER001
```

ポイント

- DBA 権限の付与，削除を実行するには，DBA 権限が必要になります。
- 自分の DBA 権限を自分で削除できます。

(3) データベースを定義するユーザに CONNECT 権限およびスキーマ定義権限を与える

CONNECT 権限およびスキーマ定義権限を与える場合は，定義系 SQL の GRANT 文を実行します。

例題

ユーザ（認可識別子：USER002，パスワード：HIRDB002）に，CONNECT 権限およびスキーマ定義権限を与えます。

```
GRANT CONNECT TO USER002 IDENTIFIED BY HIRDB002
GRANT SCHEMA TO USER002
```

ポイント

ユーザ権限は CONNECT 権限，スキーマ定義権限の順に与えてください。逆にした場合，SQL の実行時にエラーとなります。

CONNECT 権限およびスキーマ定義権限を削除する場合は，定義系 SQL の REVOKE 文を実行します。

例題

ユーザ（認可識別子：USER002）の CONNECT 権限およびスキーマ定義権限を取り消します。

```
REVOKE CONNECT FROM USER002
REVOKE SCHEMA FROM USER002
```

ポイント

SDB ディクショナリ情報または SDB ディレクトリ情報があるユーザのスキーマ定義権限は削除できません。スキーマ定義権限を削除する場合は，削除するユーザの SDB ディクショナリ情報および SDB ディレクトリ情報がないことを確認してください。ある場合は，pdsdbdef コマンドの*DELETE DICTIONARY 文および*DELETE DIRECTORY 文で SDB ディクショナリ情報および SDB ディレクトリ情報を削除してください。

(4) データベースにアクセスするユーザに CONNECT 権限を与える

CONNECT 権限を与える場合は，定義系 SQL の GRANT 文を実行します。

例題

ユーザ（認可識別子：USER003，パスワード：HIRDB003）に，CONNECT 権限を与えます。

```
GRANT CONNECT TO USER003 IDENTIFIED BY HIRDB003
```

CONNECT 権限を削除する場合は，定義系 SQL の REVOKE 文を実行します。

例題

ユーザ（認可識別子：USER003）の CONNECT 権限を削除します。

```
REVOKE CONNECT FROM USER003
```

5.16.2 ディクショナリ表の参照権限の設定

ディクショナリ表には，レコード型やインデクスなどの情報が格納されています。デフォルトの設定では，CONNECT 権限を持っていればディクショナリ表（一部のディクショナリ表を除く）の情報を参照できます。ディクショナリ表の参照権限を設定すると，DBA 権限を持っていないユーザからのディクショナリ表の参照を制限できるため，システムのセキュリティを強化できます。

次に示すユティリティの制御文で limited を指定すると，ディクショナリ表の参照権限を設定できます。

- データベース初期設定ユティリティの define system 文の dicinf オペランド
- データベース構成変更ユティリティの alter system 文の dicinf オペランド

ディクショナリ表の参照権限を設定した場合に参照制限が掛かるディクショナリ表を次の表に示します。

表 5-7　ディクショナリ表の参照権限を設定した場合に参照制限が掛かるディクショナリ表

項番	ディクショナリ表の名称	参照権限を設定した場合		参照権限を設定しない場合
		DBA 権限を持つユーザ	DBA 権限を持たないユーザ（一般ユーザ）	全ユーザ
1	SQL_TABLES	○	自分で定義したレコード型の情報だけを参照できます。	○
2	SQL_COLUMNS	○	自分で定義したレコード型の構成要素の情報だけを参照できます。	○
3	SQL_INDEXES	○	自分で定義したレコード型のインデクスの情報だけを参照できます。	○
4	SQL_TABLE_PRIVILEGES	○	自分で定義したレコード型の情報だけを参照できます。	○
5	SQL_DIV_TABLE	○	自分で定義したレコード型の情報だけを参照できます。	○
6	SQL_INDEX_COLINF	○	自分で定義したレコード型のインデクスの情報だけを参照できます。	○
7	SQL_DIV_INDEX	○	自分で定義したレコード型のインデクスの情報だけを参照できます。	○
8	SQL_SDB_DB	○	○	○
9	SQL_SDB_DB_VIEW	○	○	○
10	SQL_SDB_STORAGE_DB	○	○	○
11	SQL_SDB_NAME_CODE	○	○	○
12	SQL_SDB_DIR	○	○	○

（凡例）

○：すべての列を参照できます。

注

上記の表には，一部のディクショナリ表だけを記載しています。そのほかのディクショナリ表の参照権限については，マニュアル「HiRDB システム運用ガイド」の「ディクショナリ表の参照権限を設定するには」を参照してください。

5.17　スキーマの追加，削除

ここでは，スキーマの追加手順と削除手順について説明します。

5.17.1　スキーマの追加

スキーマの追加手順を例題形式で説明します。

例題

新規ユーザ（認可識別子 USER003）を追加し，USER003 のスキーマを定義します。

1. ユーザ（USER003）を追加して権限を与える

```
GRANT CONNECT TO USER003 IDENTIFIED BY HIRDB003
GRANT SCHEMA TO USER003
```

GRANT 文でユーザ（USER003）を追加し，CONNECT 権限およびスキーマ定義権限を与えます。
この作業は DBA 権限を持つユーザが実行します。

2. スキーマを定義する

```
CREATE SCHEMA AUTHORIZATION USER003
```

CREATE SCHEMA 文でスキーマを定義します。
この作業はユーザ（USER003）が実行します。

これでユーザ（USER003）がデータベースを定義できます。

5.17.2　スキーマの削除

スキーマを削除する場合，SDB データベースの定義削除を行う必要があります。定義削除後，HiRDB を再起動する必要があるため，「5.6　SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合）」で説明している手順に従って SDB データベースの定義削除をしてください。

スキーマを削除する場合は，「5.6.3(5)　SDB データベースの定義追加，定義変更，または定義削除を行う」で説明している操作を，次に示すように実施してください。

スキーマの削除手順を例題形式で説明します。

例題

ユーザ（認可識別子 USER003）が所有しているスキーマを削除します。

1. SDB データベースの定義削除を行う

```
pdsdbdef -i /usr/control_def_file
```

pdsdbdef コマンドで*DELETE DIRECTORY 文および*DELETE DICTIONARY 文を実行し，削除対象のスキーマが所有する SDB データベースの SDB ディレクトリ情報および SDB ディクショナリ情報を削除します。

*DELETE DIRECTORY 文および*DELETE DICTIONARY 文を実行する場合は，pdsdbdef コマンドの実行時に-i オプションの指定が必要になります。-i オプションを指定しないと，pdsdbdef コマンドの実行がエラーになります。

注意事項

SDB ディレクトリ情報および SDB ディクショナリ情報を削除する際は，必ず*DELETE DIRECTORY 文を先に実行してください。

2. スキーマを削除する

```
DROP SCHEMA AUTHORIZATION USER003
```

DROP SCHEMA 文でスキーマを削除します。

この作業はユーザ（USER003）が実行するか，または DBA 権限を持つユーザが実行します。

参考

スキーマを削除する際，そのスキーマの SDB ディレクトリ情報と SDB ディクショナリ情報が残っていると，DROP SCHEMA 文がエラーになります。

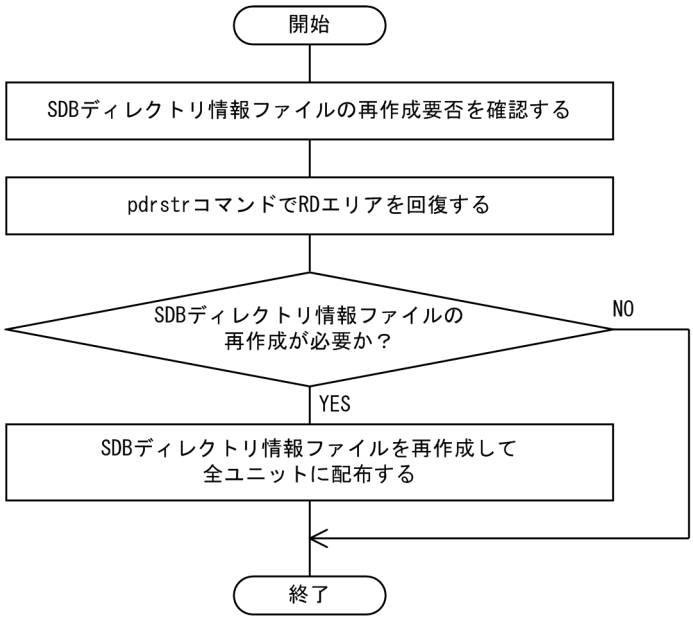
5.18　データベースの回復

RD エリアが障害閉塞した場合，pdrstr コマンドで RD エリアを回復する必要があります。ここでは，障害閉塞した RD エリアを回復して業務を再開するまでの手順について説明します。

5.18.1　データベース回復の流れ

障害閉塞した RD エリアを回復して業務を再開するまでの流れを次の図に示します。

図 5-19　障害閉塞した RD エリアを回復して業務を再開するまでの流れ



(1)　SDB ディレクトリ情報ファイルの再作成要否

次に示す 2 つの条件を満たす場合は，SDB ディレクトリ情報ファイルを再作成する必要があります。

1. 回復対象の RD エリアにシステム用 RD エリア※1 がある
2. バックアップ取得時点から pdrstr コマンドを実行する直前までの間に，次に示すどれかの操作をしている
   - HiRDB/SD 定義ユティリティ（pdsdbdef）の次に示す操作
     - *ENTRY DIRECTORY 文による SDB ディレクトリ情報の追加
     - *ALTER DIRECTORY 文による SDB ディレクトリ情報の変更
     - *DELETE DIRECTORY 文による SDB ディレクトリ情報の削除
   - データベース回復ユティリティ（pdrstr）による RD エリアの回復操作
   - データベース構成変更ユティリティ（pdmod）の次に示す操作
     - expand rdarea 文による RD エリアの拡張
     - initialize rdarea 文による RD エリアの再初期化※2

・alter rdarea 文による RD エリアの属性変更
・replicate rdarea 文による RD エリアのレプリカ定義
・define copy rdarea 文による RD エリアの構成情報複写
・recast rdarea 文による RD エリアの統合

- pddbchg コマンドによるカレント RD エリアの変更操作

注※1

マスタディレクトリ用 RD エリア，データディクショナリ用 RD エリア，またはデータディレクトリ用 RD エリアのどれかを回復した場合が該当します。

注※2

initialize rdarea 文に RD エリア名だけを指定した RD エリアの再初期化の場合（RD エリアの属性やファイル構成を変更していない場合）は，SDB ディレクトリ情報ファイルを再作成する必要はありません。

SDB ディレクトリ情報ファイルのバックアップを取得している場合，そのファイルを配布することもできますが，間違った SDB ディレクトリ情報ファイルを配布するおそれもあるため，SDB ディレクトリ情報ファイルは再作成することを推奨します。

(2) データベース回復の概要

「5.18.2 例題 1（全 RD エリアをバックアップ取得時点に回復する場合）」以降で，代表的なデータベースの回復手順の例をケース別に説明しています。データベースの回復手順の参照先一覧を次の表に示します。この表から該当する例題を選択してください。

表 5-8 データベースの回復手順の参照先一覧

データベース回復時の条件		参照先
データベースをバックアップ取得時点に回復する	全 RD エリアを回復する	「5.18.2 例題 1（全 RD エリアをバックアップ取得時点に回復する場合）」
	回復対象 RD エリアにマスタディレクトリ用 RD エリアを含む	「5.18.3 例題 2（システム用 RD エリアとユーザ用 RD エリアをバックアップ取得時点に回復する場合）」
	回復対象 RD エリアにマスタディレクトリ用 RD エリアを含まない	「5.18.4 例題 3（マスタディレクトリ用 RD エリアを除く RD エリアをバックアップ取得時点に回復する場合）」
データベースを最新の同期点に回復する，または任意の同期点に回復する※	全 RD エリアを回復する	「5.18.5 例題 4（全 RD エリアを最新の同期点に回復する場合）」
	回復対象 RD エリアにマスタディレクトリ用 RD エリアを含む	「5.18.6 例題 5（システム用 RD エリアとユーザ用 RD エリアを最新の同期点に回復する場合）」
	回復対象 RD エリアにマスタディレクトリ用 RD エリアを含まない	「5.18.7 例題 6（マスタディレクトリ用 RD エリアを除く RD エリアを最新の同期点に回復する場合）」

注

- 上記に示す例題は，SDB ディレクトリ情報ファイルの再作成が必要な場合の例題です。SDB ディレクトリ情報ファイルの再作成が必要ない場合のデータベースの回復手順については，マニュアル「HiRDB システム運用ガイド」の「データベースをバックアップ取得時点に回復する方法」または「データベースを最新の同期点に回復する方法」を参照してください。
- 例題の説明は，マニュアル「HiRDB システム運用ガイド」の「データベース回復の概要」を理解していることを前提にしています。

注※

例題では最新の同期点に回復する手順を説明していますが，任意の同期点に回復する場合も手順は同じになります。ただし，任意の同期点に回復する場合は，pdrstr コマンドの-T オプションで回復する時刻を指定します。任意の同期点への回復については，マニュアル「HiRDB システム運用ガイド」の「バックアップ取得時点以降の任意の同期点（範囲指定の回復）」を参照してください。

なお，次に示すデータベースの回復方法の例題は，ここでは説明していません。

- 差分バックアップを使用したデータベースの回復
- pdcopy コマンド以外で取得したバックアップを使用したデータベースの回復

これらのデータベースの回復方法については，マニュアル「HiRDB システム運用ガイド」の「差分バックアップ機能使用時のデータベースの回復方法」または「pdcopy コマンド以外でバックアップを取得した場合の回復方法」を参照してください。

アンロードログファイルを使用してデータベースを回復する場合の注意事項

- アンロードログファイルを使用してデータベースを回復した場合，データベースの回復が完了した RD エリアのバックアップを必ず取得してください。バックアップを取得しないと，そのあとにこの RD エリアで障害が発生したときに RD エリアを回復できなくなります。
- 回復対象の RD エリアにマスタディレクトリ用 RD エリアが含まれる場合，pdstart -r コマンドで HiRDB を開始して，pdrstr コマンドでマスタディレクトリ用 RD エリアを回復する必要があります。最新の同期点または任意の同期点に回復をする場合は，いったんマスタディレクトリ用 RD エリアを単独で回復してください。その後，マスタディレクトリ用 RD エリア以外の RD エリアを回復してください。

インナレプリカ機能を使用したデータベースの回復方法

インナレプリカ機能を使用したデータベースの回復方法については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「インナレプリカグループ内の RD エリアのバックアップと回復」を参照してください。

5.18.2 例題 1（全 RD エリアをバックアップ取得時点に回復する場合）

例題

システム用 RD エリアを含む全 RD エリアをバックアップ取得時点に回復します。

例題中のコマンドおよびコマンドのオプションの詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(1) HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。正常終了できない場合は，pdstop -f コマンドで強制終了してください。

(2) HiRDB を開始する

```
pdstart -r
```

pdstart -r コマンドで HiRDB を開始します。マスタディレクトリ用 RD エリアを回復する場合は，pdstart -r コマンドで HiRDB を開始する必要があります。

系切り替え機能を使用している場合で，HiRDB の開始と同時に共有リソース（共有ディスク，IP アドレスなど）の活性化を行うときは，pdstart -r -t コマンドを実行してください。

(3) RD エリアを回復する

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01 -a
```

pdrstr コマンドで全 RD エリアをバックアップ取得時点に回復します。

［オプションの説明］

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名を指定します。

-b：バックアップファイル名を指定します。

-a：全 RD エリアの回復を指定します。

(4) HiRDB を終了する

```
pdstop
```

pdstop コマンドで HiRDB をいったん終了します。pdstart -r コマンドで HiRDB を開始した場合，pdcopy コマンドおよび pdrstr コマンド以外は実行できません。そのため，HiRDB をいったん終了させる必要があります。

(5) HiRDB を正常開始する

```
pdstart
```

pdstart コマンドで HiRDB を正常開始します。

注意事項

(1)のときに HiRDB が正常終了していない場合は（HiRDB が異常終了した，HiRDB を強制終了したなど），ステータスファイルを初期化してから HiRDB を開始してください。ステータスファイルを初期化すると，HiRDB の開始モードが正常開始になります。

ステータスファイルを初期化するには，pdstsrm コマンドでステータスファイルをいったん削除してから，pdstsinit コマンドでステータスファイルを再作成します。

(6) SDB ディレクトリ情報ファイルを再作成して配布する

```
pdsdbdef /usr/control_def_file
```

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを再作成して全ユニット（高速系切り替え機能を使用している場合は待機系ユニットも含む）に配布します。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1 SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

(7) HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。

(8) pd_structured_shmpool_dicsize オペランドの指定値を変更する

次に示す場合は，システム共通定義の pd_structured_shmpool_dicsize オペランドの指定値を変更してください。

- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値を超えている場合
- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値に比べてかなり小さい場合

SDB ディレクトリ情報の合計長は，(6)で pdsdbdef コマンドを実行した際に出力される KFPB61231-I メッセージ（size に表示されます）で確認できます。

(9) HiRDB を正常開始する

```
pdstart
```

pdstart コマンドで HiRDB を正常開始します。

(10)　全ユニットの SDB ディレクトリ情報が一致していることを確認する

pdsdbarc -c コマンドを実行して，SDB ディレクトリ情報の最終更新日時のチェックを行い，全ユニットの SDB ディレクトリ情報が一致していることを確認してください。

チェックの結果，問題がある場合は，全ユニットに同じ SDB ディレクトリ情報ファイルが配布されていません。この場合，「(6)　SDB ディレクトリ情報ファイルを再作成して配布する」の作業から再実行してください。

5.18.3　例題 2（システム用 RD エリアとユーザ用 RD エリアをバックアップ取得時点に回復する場合）

例題

次に示す RD エリアをバックアップ取得時点に回復します。

- マスタディレクトリ用 RD エリア（rdmast）
- データディレクトリ用 RD エリア（rddir）
- データディクショナリ用 RD エリア（rddic）
- レコードを格納しているユーザ用 RD エリア（rdarea01）
- インデクスを格納しているユーザ用 RD エリア（rdarea02）

> **注意事項**
>
> 「5.5　データベースのバックアップの取得」の「表 5-2　同時にバックアップを取得する必要がある RD エリア」に示す RD エリアについて，同期を取って回復する必要があります。

例題中のコマンドおよびコマンドのオプションの詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(1)　HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。正常終了できない場合は，pdstop -f コマンドで強制終了してください。

(2)　HiRDB を開始する

```
pdstart -r
```

pdstart -r コマンドで HiRDB を開始します。マスタディレクトリ用 RD エリアを回復する場合は，pdstart -r コマンドで HiRDB を開始する必要があります。

系切り替え機能を使用している場合で，HiRDB の開始と同時に共有リソース（共有ディスク，IP アドレスなど）の活性化を行うときは，pdstart -r -t コマンドを実行してください。

(3) RD エリアを回復する

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
-r rdmast,rddir,rddic,rdarea01,rdarea02
```

pdrstr コマンドでシステム用 RD エリア（rdmast，rddir，rddic）とユーザ用 RD エリア（rdarea01，rdarea02）をバックアップ取得時点に回復します。

［オプションの説明］

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名を指定します。

-b：バックアップファイル名を指定します。

-r：回復対象の RD エリア名を指定します。

(4) HiRDB を終了する

```
pdstop
```

pdstop コマンドで HiRDB をいったん終了します。pdstart -r コマンドで HiRDB を開始した場合，pdcopy コマンドおよび pdrstr コマンド以外は実行できません。そのため，HiRDB をいったん終了させる必要があります。

このあと，SDB ディレクトリ情報ファイルを再作成して，全ユニットに配布します。再作成および配布の手順は，「5.18.2　例題 1（全 RD エリアをバックアップ取得時点に回復する場合）」の「5.18.2(5)　HiRDB を正常開始する」以降の手順と同じです。

5.18.4 例題 3（マスタディレクトリ用 RD エリアを除く RD エリアをバックアップ取得時点に回復する場合）

例題

次に示す RD エリアをバックアップ取得時点に回復します。

- データディレクトリ用 RD エリア（rddir）
- データディクショナリ用 RD エリア（rddic）
- レコードを格納しているユーザ用 RD エリア（rdarea01）
- インデクスを格納しているユーザ用 RD エリア（rdarea02）

注意事項

「5.5　データベースのバックアップの取得」の「表 5-2　同時にバックアップを取得する必要がある RD エリア」に示す RD エリアについて，同期を取って回復する必要があります。

例題中のコマンドおよびコマンドのオプションの詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(1)　RD エリアを閉塞，クローズ状態にする

```
pdhold -r rddir,rddic,rdarea01,rdarea02 -c
```

回復対象の RD エリアを pdhold コマンドで閉塞，クローズ状態にします。

(2)　RD エリアを回復する

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01
-r rddir,rddic,rdarea01,rdarea02
```

pdrstr コマンドで RD エリアをバックアップ取得時点に回復します。

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名を指定します。

-b：バックアップファイル名を指定します。

-r：回復対象の RD エリア名を指定します。

(3)　RD エリアの閉塞を解除し，オープン状態にする

```
pdrels -r rddir,rddic,rdarea01,rdarea02 -o
```

pdrels コマンドで，回復した RD エリアの閉塞を解除し，オープン状態にします。

注意事項

以降の作業から回復作業が完了するまで，業務を停止してください。

(4)　SDB ディレクトリ情報ファイルを再作成して配布する

```
pdsdbdef /usr/control_def_file
```

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを再作成して全ユニット（高速系切り替え機能を使用している場合は待機系ユニットも含む）に配布します。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1　SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

(5)　HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。

(6)　pd_structured_shmpool_dicsize オペランドの指定値を変更する

次に示す場合は，システム共通定義の pd_structured_shmpool_dicsize オペランドの指定値を変更してください。

- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値を超えている場合
- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値に比べてかなり小さい場合

SDB ディレクトリ情報の合計長は，(4)で pdsdbdef コマンドを実行した際に出力される KFPB61231-I メッセージ（size に表示されます）で確認できます。

(7)　HiRDB を正常開始する

```
pdstart
```

pdstart コマンドで HiRDB を正常開始します。

(8)　全ユニットの SDB ディレクトリ情報が一致していることを確認する

pdsdbarc -c コマンドを実行して，SDB ディレクトリ情報の最終更新日時のチェックを行い，全ユニットの SDB ディレクトリ情報が一致していることを確認してください。

チェックの結果，問題がある場合は，全ユニットに同じ SDB ディレクトリ情報ファイルが配布されていません。この場合，「(4)　SDB ディレクトリ情報ファイルを再作成して配布する」の作業から再実行してください。

5.18.5　例題 4（全 RD エリアを最新の同期点に回復する場合）

例題

システム用 RD エリアを含む全 RD エリアを障害発生直前の最新の同期点に回復します。

例題中のコマンドおよびコマンドのオプションの詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(1) HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。正常終了できない場合は，pdstop -f コマンドで強制終了してください。

(2) HiRDB を開始する

```
pdstart -r
```

pdstart -r コマンドで HiRDB を開始します。マスタディレクトリ用 RD エリアを回復する場合は，pdstart -r コマンドで HiRDB を開始する必要があります。

系切り替え機能を使用している場合で，HiRDB の開始と同時に共有リソース（共有ディスク，IP アドレスなど）の活性化を行うときは，pdstart -r -t コマンドを実行してください。

(3) 全 RD エリアをバックアップ取得時点に回復する

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01 -a
```

pdrstr コマンドで全 RD エリアをバックアップ取得時点に回復します。

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名を指定します。

-b：バックアップファイル名を指定します。

-a：全 RD エリアの回復を指定します。

(4) 全 RD エリアを最新の同期点に回復する

```
pdrstr -m /rdarea/mast/mast01 -d /unld/unldlog -w /tmp/sortwork -a
```

pdrstr コマンドで，マスタディレクトリ用 RD エリアを最新の同期点に回復します。

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名を指定します。

-d：アンロードログファイルが格納されているディレクトリ名を指定します。

-w：ソート用ワークディレクトリ名を指定します。

-a：全 RD エリアを回復対象とする指定です。

(5) HiRDB を終了する

```
pdstop
```

pdstop コマンドで HiRDB をいったん終了します。pdstart -r コマンドで HiRDB を開始した場合，pdcopy コマンドおよび pdrstr コマンド以外は実行できません。そのため，HiRDB をいったん終了させる必要があります。

(6) HiRDB を正常開始する

```
pdstart
```

pdstart コマンドで HiRDB を正常開始します。

注意事項

(1)のときに HiRDB が正常終了していない場合は（HiRDB が異常終了した，HiRDB を強制終了したなど），ステータスファイルを初期化してから HiRDB を開始してください。ステータスファイルを初期化すると，HiRDB の開始モードが正常開始になります。

ステータスファイルを初期化するには，pdstsrm コマンドでステータスファイルをいったん削除してから，pdstsinit コマンドでステータスファイルを再作成します。

(7) SDB ディレクトリ情報ファイルを再作成して配布する

```
pdsdbdef /usr/control_def_file
```

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを再作成して全ユニット（高速系切り替え機能を使用している場合は待機系ユニットも含む）に配布します。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1 SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

(8) HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。

(9) pd_structured_shmpool_dicsize オペランドの指定値を変更する

次に示す場合は，システム共通定義の pd_structured_shmpool_dicsize オペランドの指定値を変更してください。

- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値を超えている場合

- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値に比べてかなり小さい場合

SDB ディレクトリ情報の合計長は，(10)で pdsdbdef コマンドを実行した際に出力される KFPB61231-I メッセージ（size に表示されます）で確認できます。

(10) HiRDB を正常開始する

```
pdstart
```

pdstart コマンドで HiRDB を正常開始します。

(11) 全ユニットの SDB ディレクトリ情報が一致していることを確認する

pdsdbarc -c コマンドを実行して，SDB ディレクトリ情報の最終更新日時のチェックを行い，全ユニットの SDB ディレクトリ情報が一致していることを確認してください。

チェックの結果，問題がある場合は，全ユニットに同じ SDB ディレクトリ情報ファイルが配布されていません。この場合，「(7) SDB ディレクトリ情報ファイルを再作成して配布する」の作業から再実行してください。

5.18.6 例題 5（システム用 RD エリアとユーザ用 RD エリアを最新の同期点に回復する場合）

例題

次に示す RD エリアを最新の同期点に回復します。

- マスタディレクトリ用 RD エリア（rdmast）
- データディレクトリ用 RD エリア（rddir）
- データディクショナリ用 RD エリア（rddic）
- レコードを格納しているユーザ用 RD エリア（rdarea01）
- インデクスを格納しているユーザ用 RD エリア（rdarea02）

例題中のコマンドおよびコマンドのオプションの詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(1) HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。正常終了できない場合は，pdstop -f コマンドで強制終了してください。

(2) HiRDB を開始する

```
pdstart -r
```

pdstart -r コマンドで HiRDB を開始します。マスタディレクトリ用 RD エリアを回復する場合は，pdstart -r コマンドで HiRDB を開始する必要があります。

系切り替え機能を使用している場合で，HiRDB の開始と同時に共有リソース（共有ディスク，IP アドレスなど）の活性化を行うときは，pdstart -r -t コマンドを実行してください。

(3) RD エリアを最新の同期点に回復する

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01 -d /unld/unldlog
-w /tmp/sortwork -r rdmast,rddir,rddic,rdarea01,rdarea02
```

pdrstr コマンドでマスタディレクトリ用 RD エリアを最新の同期点に回復します。

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名を指定します。

-b：バックアップファイル名を指定します。

-d：アンロードログファイルが格納されているディレクトリ名を指定します。

-w：ソート用ワークディレクトリ名を指定します。

-r：回復対象の RD エリア名を指定します。

(4) HiRDB を終了する

```
pdstop
```

pdstop コマンドで HiRDB をいったん終了します。pdstart -r コマンドで HiRDB を開始した場合，pdcopy コマンドおよび pdrstr コマンド以外は実行できません。そのため，HiRDB をいったん終了させる必要があります。

(5) HiRDB を正常開始する

```
pdstart
```

pdstart コマンドで HiRDB を正常開始します。

注意事項

(1)のときに HiRDB が正常終了していない場合は（HiRDB が異常終了した，HiRDB を強制終了したなど），ステータスファイルを初期化してから HiRDB を開始してください。ステータスファイルを初期化すると，HiRDB の開始モードが正常開始になります。

ステータスファイルを初期化するには，pdstsrm コマンドでステータスファイルをいったん削除してから，pdstsinit コマンドでステータスファイルを再作成します。

このあと，SDB ディレクトリ情報ファイルを再作成して，全ユニットに配布します。再作成および配布の手順は，「5.18.5　例題４（全 RD エリアを最新の同期点に回復する場合）」の「5.18.5(7)　SDB ディレクトリ情報ファイルを再作成して配布する」以降の手順と同じです。

5.18.7　例題６（マスタディレクトリ用 RD エリアを除く RD エリアを最新の同期点に回復する場合）

例題

次に示す RD エリアを最新の同期点に回復します。

- データディレクトリ用 RD エリア（rddir）
- データディクショナリ用 RD エリア（rddic）
- レコードを格納しているユーザ用 RD エリア（rdarea01）
- インデクスを格納しているユーザ用 RD エリア（rdarea02）

例題中のコマンドおよびコマンドのオプションの詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

(1)　RD エリアを閉塞，クローズ状態にする

```
pdhold -r rddir,rddic,rdarea01,rdarea02 -c
```

回復対象の RD エリアを pdhold コマンドで閉塞，クローズ状態にします。

(2)　RD エリアを回復する

```
pdrstr -m /rdarea/mast/mast01 -b /pdcopy/backup01 -d /unld/unldlog
-w /tmp/sortwork -r rddir,rddic,rdarea01,rdarea02
```

pdrstr コマンドで RD エリアを最新の同期点に回復します。

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭の HiRDB ファイル名を指定します。

-b：バックアップファイル名を指定します。

-d：アンロードログファイルが格納されているディレクトリ名を指定します。

-w：ソート用ワークディレクトリ名を指定します。

-r：回復対象の RD エリア名を指定します。

(3) RD エリアの閉塞を解除し，オープン状態にする

```
pdrels -r rddir,rddic,rdarea01,rdarea02 -o
```

pdrels コマンドで，回復した RD エリアの閉塞を解除し，オープン状態にします。

なお，回復作業が完了するまで，業務を停止してください。

(4) SDB ディレクトリ情報ファイルを再作成して配布する

```
pdsdbdef /usr/control_def_file
```

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを再作成して全ユニット（高速系切り替え機能を使用している場合は待機系ユニットも含む）に配布します。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1 SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

(5) HiRDB を正常終了する

```
pdstop
```

pdstop コマンドで HiRDB を正常終了します。

(6) pd_structured_shmpool_dicsize オペランドの指定値を変更する

次に示す場合は，システム共通定義の pd_structured_shmpool_dicsize オペランドの指定値を変更してください。

- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値を超えている場合
- SDB ディレクトリ情報の合計長が，pd_structured_shmpool_dicsize オペランドの指定値に比べてかなり小さい場合

SDB ディレクトリ情報の合計長は，(4)で pdsdbdef コマンドを実行した際に出力される KFPB61231-I メッセージ（size に表示されます）で確認できます。

(7) HiRDB を正常開始する

```
pdstart
```

pdstart コマンドで HiRDB を正常開始します。

(8)　全ユニットの SDB ディレクトリ情報が一致していることを確認する

pdsdbarc -c コマンドを実行して，SDB ディレクトリ情報の最終更新日時のチェックを行い，全ユニットの SDB ディレクトリ情報が一致していることを確認してください。

チェックの結果，問題がある場合は，全ユニットに同じ SDB ディレクトリ情報ファイルが配布されていません。この場合，「(4)　SDB ディレクトリ情報ファイルを再作成して配布する」の作業から再実行してください。

5.19　SDB データベースへのデータロード

5.19.1　SDB データベースへのデータロード手順

ここでは，pdsdblod コマンドを使用して，SDB データベースにデータロードする手順について説明します。

(1)　全ユニットの SDB ディレクトリ情報が一致しているかどうかを確認する

```
pdsdbarc -a
```

pdsdbarc コマンドで SDB ディレクトリ情報に関する情報を表示し，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9　全ユニットの SDB ディレクトリ情報が一致しているか確認する方法」を参照してください。

確認した結果，SDB ディレクトリ情報が不一致の場合は，pdsdblod コマンドを実行できません。

(2)　レコード格納用 RD エリア，インデクス格納用 RD エリアをコマンド閉塞，クローズ状態にする

pdhold -c コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアをコマンド閉塞かつクローズ状態にします。

```
pdhold -r FMB01,FMBIDX -c
```

[オプションの説明]

-r：対象の RD エリア名を指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

> **参考**
>
> RD エリアの状態は，pddbls コマンドで確認できます。

なお，初期データロードの場合は，RD エリアのバックアップを取得する必要はありません。そのため，(2)～(4)の作業は必要ありません。

(3)　レコード格納用 RD エリア，インデクス格納用 RD エリアのバックアップを取得する

pdcopy コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r FMB01,FMBIDX
-b /bkdir/bup01 -p /bkdir/list01
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：対象の RD エリア名を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

(4) レコード格納用 RD エリア，インデクス格納用 RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：対象 RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(5) データロードを実行する

pdsdblod コマンドでデータロードを実行します。

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文を指定した制御文ファイルの名称を絶対パスまたは相対パスで指定します。初期データロード，追加データロードの制御文については，「12.3　pdsdblod 制御文」で説明します。

(6) レコード格納用 RD エリア，インデクス格納用 RD エリアをコマンド閉塞，クローズ状態にする

pdhold -c コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアをコマンド閉塞かつクローズ状態にします。

```
pdhold -r FMB01,FMBIDX -c
```

[オプションの説明]

-r：対象の RD エリア名を指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

(7) レコード格納用 RD エリア，インデクス格納用 RD エリアのバックアップを取得する

pdcopy コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r FMB01,FMBIDX
-b /bkdir/bup02 -p /bkdir/list02
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：対象の RD エリア名を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

(8) レコード格納用 RD エリア，インデクス格納用 RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：対象 RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

5.20　インデクスの再作成

5.20.1　インデクスの再作成手順

ここでは，pdsdblod コマンドを使用して，インデクスを再作成する手順について説明します。

(1)　全ユニットの SDB ディレクトリ情報が一致しているかどうかを確認する

```
pdsdbarc -a
```

pdsdbarc コマンドで SDB ディレクトリ情報に関する情報を表示し，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。確認方法については，「5.9　全ユニットの SDB ディレクトリ情報が一致しているか確認する方法」を参照してください。

確認した結果，SDB ディレクトリ情報が不一致の場合は，pdsdblod コマンドを実行できません。

(2)　レコード格納用 RD エリア，インデクス格納用 RD エリアをコマンド閉塞，クローズ状態にする

pdhold -c コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアをコマンド閉塞かつクローズ状態にします。

```
pdhold -r FMB01,FMBIDX -c
```

[オプションの説明]

-r：対象の RD エリア名を指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

> **参考**
>
> RD エリアの状態は，pddbls コマンドで確認できます。

(3)　レコード格納用 RD エリア，インデクス格納用 RD エリアのバックアップを取得する

pdcopy コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r FMB01,FMBIDX
-b /bkdir/bup01 -p /bkdir/list01
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：対象の RD エリア名を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

(4) レコード格納用 RD エリア，インデクス格納用 RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：対象の RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(5) インデクスを再作成する

pdsdblod コマンドでインデクスを再作成します。

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文を指定した制御文ファイルの名称を絶対パスまたは相対パスで指定します。インデクスの再作成の制御文については，「12.3　pdsdblod 制御文」で説明します。

(6) レコード格納用 RD エリア，インデクス格納用 RD エリアをコマンド閉塞，クローズ状態にする

pdhold -c コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアをコマンド閉塞かつクローズ状態にします。

```
pdhold -r FMB01,FMBIDX -c
```

[オプションの説明]

-r：対象の RD エリア名を指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

(7) レコード格納用 RD エリア，インデクス格納用 RD エリアのバックアップを取得する

pdcopy コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r FMB01,FMBIDX
-b /bkdir/bup02 -p /bkdir/list02
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：対象の RD エリア名を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

(8) レコード格納用 RD エリア，インデクス格納用 RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，レコード格納用 RD エリアおよびインデクス格納用 RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：対象 RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

5.21　障害が発生したときの対処方法

障害が発生したときの対処方法については，マニュアル「HiRDB システム運用ガイド」の「障害が発生したときの対処方法」を参照してください。

更新可能なオンライン再編成の実行時に障害が発生したときの対処方法については，「5.23　障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】」を参照してください。

5.22　障害が発生したときの対処方法（pdsdblod コマンドの異常終了時）

ここでは，pdsdblod コマンドが異常終了した場合の対処方法について説明します。

5.22.1　エラー発生位置と対処コード

(1)　初期データロードの場合（更新可能なオンライン再編成の場合）【4V FMB】

初期データロード時にエラーが発生した場合の対処を次の表に示します。

表 5-9　初期データロード時にエラーが発生した場合の対処（更新可能なオンライン再編成の場合）

項番	エラー発生直前に出力されたメッセージ		リターンコード 8 の場合	リターンコード 12 の場合			
				RD エリアが閉塞していない場合		RD エリアが閉塞状態の場合	
				インデクス一括作成モードの場合	インデクス更新モードの場合	インデクス一括作成モードの場合	インデクス更新モードの場合
1	KFPB63000-I	pdsdblod コマンドの開始	手順 2	－	－	手順 7	手順 7
2	KFPB63009-I	削除処理の終了	－	手順 3	手順 3	－	－
3	KFPB63010-I	データ格納処理の開始	－	手順 3	手順 3	手順 7	手順 7
4	KFPB63012-I	データ格納処理の終了	－	手順 3	手順 3	手順 7	手順 7
5	KFPB63036-I	インデクス一括作成の開始	－	手順 3	－	手順 7	－
6	KFPB63038-I	インデクス一括作成の開始※	－	手順 3	－	手順 7	－
7	KFPB63039-I	インデクス一括作成の終了※	－	手順 3	－	手順 7	－
8	KFPB63037-I	インデクス一括作成の終了	－	手順 3	－	手順 7	－
9	KFPB63042-I	追い付き反映キー対応表の登録処理の開始	－	手順 6	手順 6	手順 7	手順 7

項番	エラー発生直前に出力されたメッセージ		リターンコード 8 の場合	リターンコード 12 の場合			
				RD エリアが閉塞していない場合		RD エリアが閉塞状態の場合	
				インデクス一括作成モードの場合	インデクス更新モードの場合	インデクス一括作成モードの場合	インデクス更新モードの場合
10	KFPB63043-I	追い付き反映キー対応表の登録処理の終了	－	再実行不要	再実行不要	再実行不要	再実行不要

（凡例）－：なし

注※ 「インデクス数×RD エリア数」分，メッセージが出力されます。

(2) 初期データロードの場合（更新可能なオンライン再編成以外の場合）

初期データロード時にエラーが発生した場合の対処を次の表に示します。

表 5-10 初期データロード時にエラーが発生した場合の対処（更新可能なオンライン再編成以外の場合）

項番	エラー発生直前に出力されたメッセージ		リターンコード 4 の場合	リターンコード 8 の場合	リターンコード 12 の場合			
					RD エリアが閉塞していない場合		RD エリアが閉塞状態の場合	
					インデクス一括作成モードの場合	インデクス更新モードの場合	インデクス一括作成モードの場合	インデクス更新モードの場合
1	KFPB63000-I	pdsdblod コマンドの開始	－	手順 2	－	－	手順 4	手順 4
2	KFPB63009-I	削除処理の終了	－	－	手順 3	手順 3	－	－
3	KFPB63010-I	データ格納処理の開始	－	－	手順 3	手順 3	手順 4	手順 4
4	KFPB63012-I	データ格納処理の終了	手順 1	－	手順 5	再実行不要	－	再実行不要
5	KFPB63036-I	インデクス一括作成の開始	－	－	－	－	－	－
6	KFPB63038-I	インデクス一括作成の開始※	－	－	手順 5	－	手順 4	－
7	KFPB63039-I	インデクス一括作成の終了※	－	－	手順 5	－	手順 4	－

項番	エラー発生直前に出力されたメッセージ		リターンコード 4 の場合	リターンコード 8 の場合	リターンコード 12 の場合			
					RD エリアが閉塞していない場合		RD エリアが閉塞状態の場合	
					インデクス一括作成モードの場合	インデクス更新モードの場合	インデクス一括作成モードの場合	インデクス更新モードの場合
8	KFPB63037-I	インデクス一括作成の終了	手順 1	−	再実行不要	−	再実行不要	−
9	KFPB63042-I	追い付き反映キー対応表の登録処理の開始	−	−	−	−	−	−
10	KFPB63043-I	追い付き反映キー対応表の登録処理の終了	−	−	−	−	−	−

（凡例）−：なし

注※ 「インデクス数×RD エリア数」分，メッセージが出力されます。

(3) 追加データロードの場合【4V FMB，SD FMB】

追加データロード時にエラーが発生した場合の対処を次の表に示します。

表 5-11 追加データロード時にエラーが発生した場合の対処

項番	エラー発生直前に出力されたメッセージ		リターンコード 4 の場合	リターンコード 8 の場合	リターンコード 12 の場合			
					RD エリアが閉塞していない場合※1		RD エリアが閉塞状態の場合	
					インデクス一括作成モードの場合	インデクス更新モードの場合	インデクス一括作成モードの場合	インデクス更新モードの場合
1	KFPB63000-I	pdsdblod コマンドの開始	−	手順 2	−	−	−	−
2	KFPB63010-I	データ格納処理の開始	−	−	手順 8	手順 8	手順 9	手順 9
3	KFPB63044-I	インデクスの抽出開始	−	−	手順 8	−	手順 9	手順 9
4	KFPB63046-I	インデクスの抽出開始※2	−	−	手順 8	−	手順 9	手順 9
5	KFPB63047-I	インデクスの抽出終了※2	−	−	手順 8	−	手順 9	手順 9

項番	エラー発生直前に出力されたメッセージ		リターンコード 4 の場合	リターンコード 8 の場合	リターンコード 12 の場合			
					RD エリアが閉塞していない場合※1		RD エリアが閉塞状態の場合	
					インデクス一括作成モードの場合	インデクス更新モードの場合	インデクス一括作成モードの場合	インデクス更新モードの場合
6	KFPB63045-I	インデクスの抽出終了	－	－	手順 8	－	手順 9	手順 9
7	KFPB63012-I	データ格納処理の終了	手順 10	－	手順 10	再実行不要	－	再実行不要
8	KFPB63036-I	インデクス一括作成の開始	－	－	手順 10	－	－	－
9	KFPB63038-I	インデクス一括作成の開始※2	－	－	手順 10	－	手順 9	－
10	KFPB63039-I	インデクス一括作成の終了※2	－	－	手順 10	－	手順 9	－
11	KFPB63037-I	インデクス一括作成の終了	手順 10	－	再実行不要	－	再実行不要	－

（凡例）－：なし

注※1

ロールバックしたあとに追加データロード前の状態に戻っていることを確認します。

注※2

「インデクス数×RD エリア数」分，メッセージが出力されます。

(4)　インデクスの再作成の場合

インデクスの再作成時にエラーが発生した場合の対処を次の表に示します。

表 5-12　インデクスの再作成時にエラーが発生した場合の対処

項番	エラー発生直前に出力されたメッセージ		リターンコード 4 の場合	リターンコード 8 の場合	リターンコード 12 の場合	
					RD エリアが閉塞していない場合	RD エリアが閉塞状態の場合
1	KFPB63000-I	pdsdblod コマンドの開始	－	手順 2	－	－
2	KFPB63044-I	インデクスの抽出開始	－	－	手順 8	手順 9
3	KFPB63046-I	インデクスの抽出開始※	－	－	手順 8	手順 9

項番	エラー発生直前に出力されたメッセージ		リターンコード 4 の場合	リターンコード 8 の場合	リターンコード 12 の場合	
					RD エリアが閉塞していない場合	RD エリアが閉塞状態の場合
4	KFPB63047-I	インデクスの抽出終了※	－	－	手順 8	手順 9
5	KFPB63045-I	インデクスの抽出終了	－	－	手順 8	手順 9
6	KFPB63036-I	インデクス一括作成の開始	－	－	手順 8	－
7	KFPB63038-I	インデクス一括作成の開始※	－	－	手順 8	手順 4
8	KFPB63039-I	インデクス一括作成の終了※	－	－	手順 8	手順 4
9	KFPB63037-I	インデクス一括作成の終了	－	－	再実行不要	再実行不要

（凡例）－：なし

注※ 「インデクス数×RD エリア数」分，メッセージが出力されます。

(5) フォーマットライトの場合【4V DAM】

フォーマットライト時にエラーが発生した場合の対処を次の表に示します。

表 5-13 フォーマットライト時にエラーが発生した場合の対処

項番	エラー発生直前に出力されたメッセージ		リターンコード 4 の場合	リターンコード 8 の場合	リターンコード 12 の場合	
					RD エリアが閉塞していない場合	RD エリアが閉塞状態の場合
1	KFPB63000-I	pdsdblod コマンドの開始	－	手順 2	－	手順 4
2	KFPB63009-I	削除処理の終了	－	－	手順 3	－
3	KFPB63010-I	データ格納処理の開始	－	－	手順 3	手順 4
4	KFPB63012-I	データ格納処理の終了	－	－	再実行不要	再実行不要

（凡例）－：なし

5.22.2 エラーの対処方法

pdsdblod コマンドが異常終了した場合の対処手順を説明します。

(1) 手順 1 の対処方法

不正な入力データを修正し，pdsdblod コマンドを再実行します。

手順

1. 入力データを修正します。
2. environment 文の purge オペランドに yes を指定して，pdsdblod コマンドを再実行します。

(2) 手順 2 の対処方法

エラーとなった制御文を修正し，pdsdblod コマンドを再実行します。

手順

1. エラーとなった制御文を修正します。
2. pdsdblod コマンドを再実行します。

(3) 手順 3 の対処方法

出力されたメッセージを参照し，エラー原因を取り除いたあとに pdsdblod コマンドを再実行します。

手順

1. 出力されたメッセージを参照し，エラー原因を取り除いてください。
2. environment 文の purge オペランドに yes を指定して，pdsdblod コマンドを再実行します。

(4) 手順 4 の対処方法

RD エリアを再初期化してから pdsdblod コマンドを再実行します。

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. pdclose コマンドで，RD エリアをクローズします。
3. pdmod コマンドで，RD エリアを再初期化します。
4. pdrels コマンドで，RD エリアの閉塞を解除し，オープンします。
5. pdsdblod コマンドを再実行します。

(5) 手順 5 の対処方法

次のどちらかの手順を実行してください。

(a) 格納済みのデータを削除して pdsdblod コマンドを再実行する

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. environment 文の purge オペランドに yes を指定して，pdsdblod コマンドを再実行します。

(b) インデクスを再作成する

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. エラーが発生したインデクスに対して，インデクスの再作成の制御文を作成してください。
3. pdsdblod コマンドを実行します。

上記の手順の 1.～3.を，エラーが発生した以降のインデクス数分繰り返し実行してください。

(6) 手順 6 の対処方法

追い付き反映キー対応表を初期化してから，pdsdblod コマンドを再実行します。

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. 追い付き反映キー対応表を pdsdborcrt コマンドで初期化します。
3. environment 文の purge オペランドに yes を指定して，pdsdblod コマンドを再実行します。

更新可能なオンライン再編成の場合，エラーが発生した RD エリアがあるバックエンドサーバ内の，更新可能なオンライン再編成の対象となった全 RD エリアに対して，上記の手順の 3.を繰り返してください。

(7) 手順 7 の対処方法

RD エリアを再初期化してから，pdsdblod コマンドを再実行します。

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. 追い付き反映キー対応表を pdsdborcrt コマンドで初期化します。
3. pdclose コマンドで RD エリアをクローズします。
4. pdmod コマンドで RD エリアを再初期化します。
5. pdrels コマンドで，RD エリアの閉塞を解除し，オープン状態にします。
6. pdsdblod コマンドを再実行します。

更新可能なオンライン再編成の場合，エラーが発生したRDエリアがあるバックエンドサーバ内の，更新可能なオンライン再編成の対象となった全RDエリアに対して，上記の手順の3.～6.を繰り返してください。

(8) 手順8の対処方法

出力されたメッセージを参照して，エラー原因を取り除いてから，pdsdblodコマンドを再実行します。

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. pdsdblodコマンドを再実行します。

(9) 手順9の対処方法

RDエリアを追加データロード実行直前の状態に回復してから，pdsdblodコマンドを再実行します。

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. pdcloseコマンドで，障害閉塞しているRDエリアをクローズします。
3. pdrstrコマンドでRDエリアを回復します。
4. pdrelsコマンドで，RDエリアの閉塞を解除し，オープン状態にします。
5. pdsdblodコマンドを再実行します。

(10) 手順10の対処方法

次のどちらかの手順を実行してください。

ただし，キー値が重複した場合，(b)の方法では対処できないため，(a)の方法で対処してください。

(a) RDエリアを追加データロード実行直前の状態に回復してからpdsdblodコマンドを再実行する

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. pdholdコマンドで，RDエリアを閉塞し，クローズします。
3. pdrstrコマンドでRDエリアを回復します。
4. pdrelsコマンドで，RDエリアの閉塞を解除し，オープン状態にします。
5. pdsdblodコマンドを再実行します。

(b) インデクスを再作成する

手順

1. 出力されたメッセージを参照して，エラー原因を取り除いてください。
2. エラーが発生したインデクスに対して，インデクスの再作成の制御文を作成します。
3. pdsdblod コマンドを実行します。

上記の手順の 1.～3.を，エラーが発生した以降のインデクス数分繰り返し実行してください。

5.23　障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】

ここでは，更新可能なオンライン再編成の実行中に障害が発生した場合の対処方法について説明します。

5.23.1　障害発生時の対処の流れ

障害発生時の対処の流れについては，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「更新可能なオンライン再編成実行時の障害」の「障害対策の流れ」を参照してください。

5.23.2　システムログとオペランドの設定

更新可能なオンライン再編成中は，追い付き反映処理を行うためにシステムログに情報を残しています。詳細については，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「システムログとオペランドの設定」を参照してください。

5.23.3　システムログファイルの障害回復

システムログファイルに障害が発生した場合は，マニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「システムログファイルの障害回復」を参照して対処してください。

5.23.4　更新可能なオンライン再編成の再編成状態の確認

障害が発生した場合，更新可能なオンライン再編成の再編成状態を次の手順で確認してください。

手順

1. pdls -d org コマンドを実行して，追い付き反映処理の状態を確認してください。詳細については，「5.13.8　追い付き反映処理の状態を確認したい場合」を参照してください。
2. pdls -d org コマンドの実行結果の STATUS 欄に表示された追い付き反映処理の状態を確認してください。READY の場合は，次の手順に進んでください。READY 以外の場合は，5.に進んでください。
3. pddbls コマンドを実行して，カレント RD エリアを確認してください。
4. ペアボリュームの状態を確認してください。日立ディスクアレイサブシステムを使用している場合は，pairdisplay コマンドで確認します。そのほかの製品のミラーリング機能を使用している場合，使用しているミラーリング機能のマニュアルを参照して，ペア状態を確認してください。

5.「表 5-14　更新可能なオンライン再編成状態の確認方法」の判定条件を参照して，該当する再編成状態を確認してください。

6.「5.23.5　障害発生時の対処方法」を参照して，該当する障害の対処を実施してください。

表 5-14　更新可能なオンライン再編成状態の確認方法

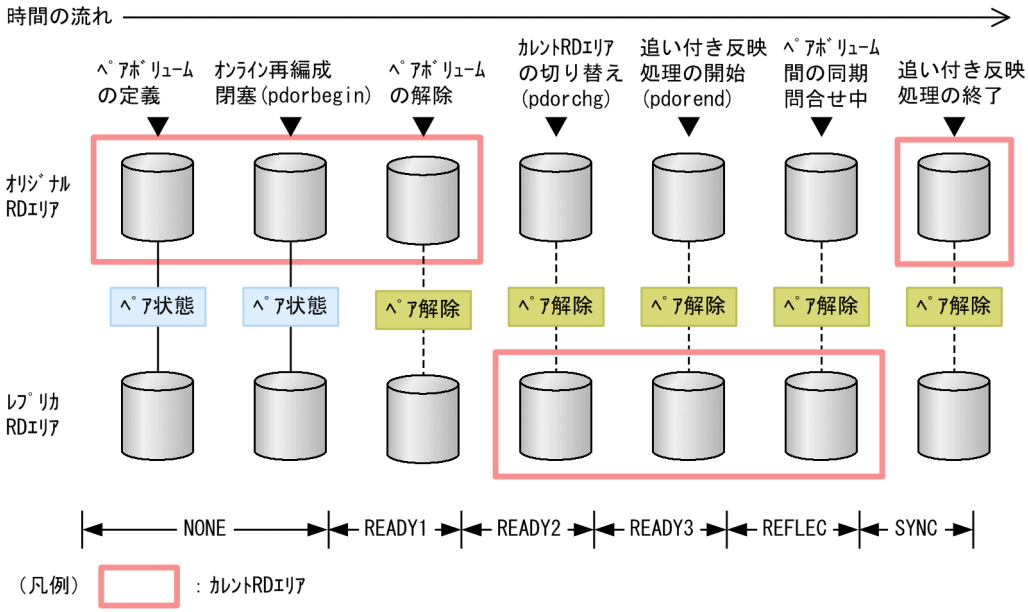
判定要素					判定結果
オリジナル RD エリアの状態 (pddbls)	レプリカ RD エリアの状態 (pddbls)	カレント RD エリア (pddbls)	追い付き状態 (pdls)	ペアボリューム状態	再編成状態
OPEN	HOLD(CMD)CLOSE	マスタ DB	NONE	ペア状態	NONE（pdorbegin コマンドの失敗，または未実施）
HOLD(ORG)	HOLD(ORG)	マスタ DB	READY	ペア状態	READY1（ボリュームペアの解除失敗，または未実施）
HOLD(ORG)	HOLD(ORG)	マスタ DB	READY	ペア解除状態	READY2（pdorchg コマンドの失敗，または未実施）
HOLD(ORG)	HOLD(ORG)	レプリカ DB	READY	ペア解除状態	READY3（再編成の失敗，または pdorend コマンド未実施）
HOLD(ORG)	HOLD(ORG)	レプリカ DB	REFLEC	ペア解除状態	REFLEC（追い付き処理の失敗）
HOLD(ORG)	HOLD(ORG)	レプリカ DB	SYNC	ペア解除状態	SYNC（レプリカからオリジナル RD エリアへの業務切り替え処理の失敗）

注

オリジナルまたはレプリカのボリューム障害時は，障害側が HOLD になります。

判定結果の再編成状態から障害発生時点が次の図のように特定できます。

図 5-20　更新可能なオンライン再編成の障害発生時点



5.23.5　障害発生時の対処方法

更新可能なオンライン再編成中に障害が発生したときの対処方法について，障害のケースごとに説明します。説明中に出てくる再編成状態とは，「表 5-14　更新可能なオンライン再編成状態の確認方法」で確認した再編成状態のことです。

なお，再編成状態が，NONE，READY1，または READY2 の場合は，pdorbegin -u コマンドで更新可能なオンライン再編成の操作を取り消すことができます。

また，障害発生の原因がシステムログファイル障害の場合は，システムログファイルの障害を回復したあとに，ここで説明している対処を実施してください。

(1)　コマンドまたはユニットが異常終了した場合

コマンドまたはユニットが異常終了した場合の対処方法を次の表に示します。再編成状態によって対処方法が異なります。

表 5-15　コマンドまたはユニットが異常終了した場合の対処方法

項番	再編成状態	対処方法
1	NONE	更新可能なオンライン再編成を続行する場合は，障害原因を取り除いたあとに，次の操作から再実行してください。 • 「5.13.4(1)　RD エリアをオンライン再編成閉塞にする（pdorbegin コマンド）」
2	READY1	更新可能なオンライン再編成を続行する場合は，使用しているミラーリング機能のマニュアルを参照して，対処してください。

項番	再編成状態	対処方法
3	READY2	更新可能なオンライン再編成を続行する場合は，障害原因を取り除いたあとに，次の操作から再実行してください。 • 「5.13.4(3)　カレント RD エリアを切り替える（pdorchg コマンド）」
4	READY3	• 更新可能なオンライン再編成を続行する場合 pdsdbrog コマンドが異常終了した場合は，障害原因を取り除いたあとに，次の操作から再実行してください。 ・「5.13.4(5)　オリジナル RD エリアのレコードをアンロードする（pdsdbrog コマンド）」 注意事項 • pdsdblod コマンドが異常終了した場合は，障害原因を取り除いたあとに，異常終了時に指定していたアンロードデータファイルを入力として，次の操作から再実行してください。 ・「5.13.4(6)　オリジナル RD エリアにレコードをデータロードする（pdsdblod コマンド）」 ただし，異常終了時に指定していたアンロードデータファイルを削除してしまった場合は，データロードの再実行ができないため，項番 5 または項番 6 の対処方法を実施してください。 • すでに追い付き反映キー対応表へのデータ登録処理が開始されている pdsdblod コマンドがある場合（KFPB63042-I メッセージが出力されている場合），pdsdblod コマンドの再実行前に追い付き反映キー対応表を初期化してください。この場合，初期化した追い付き反映キー対応表がある BES 内の，更新可能なオンライン再編成の対象 RD エリアに対して，pdsdblod コマンドを再実行する必要があります。
5		• 更新可能なオンライン再編成を中止する場合（HiRDB を停止できない場合） レプリカ RD エリアの更新データをオリジナル RD エリアに反映し，オリジナル RD エリアで通常運用を再開します。HiRDB は停止しませんが，トランザクションを制限する必要があります。手順については，「5.23.6　対処の詳細手順（その 1）」を参照してください。
6		• 更新可能なオンライン再編成を中止する場合（HiRDB を停止できる場合） レプリカ RD エリアの更新データをオリジナル RD エリアに反映し，オリジナル RD エリアで業務を再開します。手順については，「5.23.7　対処の詳細手順（その 2）」を参照してください。
7	REFLEC	• 更新可能なオンライン再編成を続行する場合 障害原因を取り除くか，またはカレント RD エリアへのアクセスを制限するかしたあとに，次の操作から再実行してください。 ・「5.13.4(9)　追い付き反映処理を実行する（pdorend コマンド）」 • 更新可能なオンライン再編成を中止する場合（HiRDB を停止できない場合） 対処方法は，項番 5 と同じです。 • 更新可能なオンライン再編成を中止する場合（HiRDB を停止できる場合） 対処方法は，項番 6 と同じです。
8	SYNC	• 更新可能なオンライン再編成を続行する場合 障害原因を取り除いたあとに，次の操作から再実行してください。 ・「5.13.4(9)　追い付き反映処理を実行する（pdorend コマンド）」 • 更新可能なオンライン再編成を中止する場合（HiRDB を停止できない場合） 対処方法は，項番 5 と同じです。 • 更新可能なオンライン再編成を中止する場合（HiRDB を停止できる場合） 対処方法は，項番 6 と同じです。

(2)　正系ボリューム障害が発生した場合

正系ボリューム障害が発生した場合の対処方法を次の表に示します。再編成状態によって対処方法が異なります。

表 5-16　正系ボリューム障害が発生した場合の対処方法

項番	再編成状態	対処方法
1	NONE	更新可能なオンライン再編成を続行する場合の対処手順については，「5.23.8　対処の詳細手順（その 3)」を参照してください。
2	READY1	更新可能なオンライン再編成を続行する場合は，使用しているミラーリング機能のマニュアルを参照して，対処してください。
3	READY2	更新可能なオンライン再編成を続行する場合の対処手順については，「5.23.8　対処の詳細手順（その 3)」を参照してください。
4	READY3	更新可能なオンライン再編成を続行する場合の対処手順については，「5.23.9　対処の詳細手順（その 4)」を参照してください。
5	REFLEC	• 更新可能なオンライン再編成を続行する場合 対処手順については，「5.23.10　対処の詳細手順（その 5)」を参照してください。 この場合，追い付き反映処理は最初から反映し直しとなります。 また，pd_log_org_reflected_logpoint=keep，pd_log_org_no_standby_file_opr=stop を指定し，システムログ反映開始ポイントを消去していないことが前提条件となります。 • 更新可能なオンライン再編成を中止し，レプリカ RD エリアで通常運用を行う場合 pdorend -u コマンドを実行して，追い付き反映処理を中止してください。
6	SYNC	更新可能なオンライン再編成を中止し，レプリカ RD エリアで通常運用を行うことになります。pdorend -u コマンドを実行して，追い付き反映処理を中止してください。

(3)　副系ボリューム障害が発生した場合

副系ボリューム障害が発生した場合の対処方法を次の表に示します。再編成状態によって対処方法が異なります。

表 5-17　副系ボリューム障害が発生した場合の対処方法

項番	再編成状態	対処方法
1	NONE	更新可能なオンライン再編成を続行する場合は，使用しているミラーリング機能のマニュアルを参照して，対処してください。障害原因を取り除いたあとに，次の操作から再実行してください。 • 「5.13.4(1)　RD エリアをオンライン再編成閉塞にする（pdorbegin コマンド)」
2	READY1	更新可能なオンライン再編成を続行する場合は，次の手順で対処してください。 1. pdorbegin -u コマンドで更新可能なオンライン再編成の操作を取り消します。詳細については，「5.13.9　更新可能なオンライン再編成の操作を取り消したい場合」を参照してください。 2. 副系ボリュームを回復します。回復方法については，使用しているミラーリング機能のマニュアルを参照してください。 3. 次の操作から，更新可能なオンライン再編成を再実行してください。
3	READY2	

項番	再編成状態	対処方法
		・「5.13.4(1)　RD エリアをオンライン再編成閉塞にする（pdorbegin コマンド）」
4	READY3	更新可能なオンライン再編成をいったん中止します。レプリカ RD エリアの更新データをオリジナル RD エリアに反映し，オリジナル RD エリアで通常運用を再開します。手順については，「5.23.11　対処の詳細手順（その 6）」を参照してください。
5	REFLEC	
6	SYNC	障害原因を取り除いたあとに，次の操作から再実行してください。 •「5.13.4(9)　追い付き反映処理を実行する（pdorend コマンド）」

(4)　構成情報の複写漏れによってディレクトリ管理情報が不正となった場合

構成情報の複写漏れによってディレクトリ管理情報が不正となった場合の対処方法を次の表に示します。再編成状態によって対処方法が異なります。

表 5-18　構成情報の複写漏れによってディレクトリ管理情報が不正となった場合の対処方法

項番	再編成状態	対処方法
1	READY1	更新可能なオンライン再編成を続行する場合は，次の手順で対処してください。 1. pdorbegin -u コマンドで更新可能なオンライン再編成の操作を取り消します。詳細については，「5.13.9　更新可能なオンライン再編成の操作を取り消したい場合」を参照してください。 2. レプリカ RD エリアを pdhold -c コマンドで閉塞かつクローズ状態にします。 3. pdmod コマンドの define copy rdarea 文で，レプリカ RD エリアにオリジナル RD エリアの構成情報を複写します。 4. 次の操作から，更新可能なオンライン再編成を再実行してください。 ・「5.13.4(1)　RD エリアをオンライン再編成閉塞にする（pdorbegin コマンド）」
2	READY2	
3	READY3	更新可能なオンライン再編成を続行する場合の対処手順については，「5.23.12　対処の詳細手順（その 7）」を参照してください。

(5)　追い付き反映キー対応表に障害が発生した場合

追い付き反映キー対応表に障害が発生した場合の対処方法を次の表に示します。

表 5-19　追い付き反映キー対応表に障害が発生した場合の対処方法

項番	障害発生のタイミング	障害原因	対処方法
1	pdsdblod コマンドの実行中	追い付き反映キー対応表を格納する RD エリアで容量不足が発生した場合	更新可能なオンライン再編成を続行する場合は，次の手順で対処してください。 なお，説明中の RD エリアとは，追い付き反映キー対応表を格納している RD エリアのことです。 1. pdrels -o コマンドで RD エリアの閉塞を解除し，オープン状態にします。 2. pdmod コマンドで RD エリアを再初期化して容量を拡張します。 3. pdsdborcrt -k init コマンドで，追い付き反映キー対応表を初期化します。

項番	障害発生のタイミング	障害原因	対処方法
			4. 追い付き反映キー対応表の中間ファイルを削除します。 5. 再初期化した RD エリアがあるサーバの更新可能なオンライン再編成は，データロードから再実行する必要があります。サーバ内の更新可能なオンライン再編成の対象 RD エリアに対して，pdsdblod コマンドでデータロードを実行してください。 なお，pdsdblod コマンドの入力データファイルとするアンロードデータファイルを削除してしまった場合は，データロードを実行できません。この場合，項番 2 の対処を実施してください。 このあと，次の操作から再実行してください。 • 「5.13.4(7)　オリジナル RD エリアのバックアップを取得する（pdcopy コマンド）」
2			更新可能なオンライン再編成を中止する場合は，レプリカ RD エリアのデータをオリジナル RD エリアに反映し，オリジナル RD エリアで業務を再開します。手順については，「表 5-15　コマンドまたはユニットが異常終了した場合の対処方法」の項番 5 または項番 6 と同じです。
3		上記以外の追い付き反映キー対応表へのアクセス障害が発生した場合	更新可能なオンライン再編成を続行することはできません。レプリカ RD エリアの更新情報をオリジナル RD エリアに反映し，オリジナル RD エリアで業務を再開します。手順については，「表 5-15　コマンドまたはユニットが異常終了した場合の対処方法」の項番 5 または項番 6 と同じです。
4	pdorend コマンドの実行中	追い付き反映キー対応表を格納する RD エリアで容量不足が発生した場合	更新可能なオンライン再編成を続行する場合は，次の手順で対処してください。 なお，説明中の RD エリアとは，追い付き反映キー対応表を格納している RD エリアのことです。 1. pdrels -o コマンドで RD エリアの閉塞を解除し，オープン状態にします。 2. pdmod コマンドで RD エリアを拡張します。 このあと，次の操作から再実行してください。 • 「5.13.4(9)　追い付き反映処理を実行する（pdorend コマンド）」
5			更新可能なオンライン再編成を中止する場合は，レプリカ RD エリアの更新情報をオリジナル RD エリアに反映し，オリジナル RD エリアで業務を再開します。手順については，「表 5-15　コマンドまたはユニットが異常終了した場合の対処方法」の項番 5 または項番 6 と同じです。
6		上記以外の追い付き反映キー対応表へのアクセス障害が発生した場合	更新可能なオンライン再編成を続行することはできません。レプリカ RD エリアの更新情報をオリジナル RD エリアに反映し，オリジナル RD エリアで業務を再開しま

項番	障害発生のタイミング	障害原因	対処方法
			す。手順については，「表 5-15　コマンドまたはユニットが異常終了した場合の対処方法」の項番 5 または項番 6 と同じです。

5.23.6　対処の詳細手順（その 1）

対処手順を次に示します。

(1)　追い付き反映処理を取り消す

pdorend -u コマンドで，追い付き反映処理を取り消します。

```
pdorend -s bes1 -u
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-u：追い付き反映処理を取り消す場合に指定するオプションです。

(2)　レプリカ RD エリアを参照可能バックアップ閉塞にする

pdhold -b コマンドで，レプリカ RD エリアを参照可能バックアップ閉塞にします。

```
pdhold -r FMB01_1,FMBIDX_1 -b
```

[オプションの説明]

-r：更新可能なオンライン再編成の対象 RD エリア（BES 内のレプリカ RD エリア）を指定します。

-b：RD エリアの状態を参照可能バックアップ閉塞にする場合に指定するオプションです。

この例では，RD エリアの構成は次のようになっています。

- FMB01_1：レコード型を格納しているレプリカ RD エリア
- FMBIDX_1：インデクスを格納しているレプリカ RD エリア

(3)　レプリカ RD エリアのバックアップを取得する

pdcopy コマンドで，レプリカ RD エリアのバックアップを取得します。

```
pdcopy -m /HiRDB/rdarea/rdmast/rdmast00 -M r -r FMB01,FMBIDX
-q 1 -b /bkdir/bkup04 -p /bkdir/list04
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-M：バックアップ取得モードを指定します。

-r：オリジナル RD エリア名を指定します。

-q：バックアップ対象のレプリカ RD エリアの世代番号を指定します。

-b：バックアップファイル名を指定します。

-p：pdcopy コマンドの処理結果リストの出力先を指定します。

(4) オリジナル RD エリアをクローズ状態にする

- オリジナル RD エリアの状態が障害閉塞の場合

  pdclose コマンドで，オリジナル RD エリアをクローズ状態にします。

```
pdclose -r FMB01,FMBIDX
```

  [オプションの説明]

  -r：オリジナル RD エリア名を指定します。

- オリジナル RD エリアの状態がオープン状態の場合

  pdhold -c コマンドで，オリジナル RD エリアをコマンド閉塞かつクローズ状態にします。

```
pdhold -r FMB01,FMBIDX -c
```

  [オプションの説明]

  -r：オリジナル RD エリア名を指定します。

  -c：RD エリアをクローズ状態にする場合に指定するオプションです。

> **参考**
>
> RD エリアの状態は，pddbls コマンドで確認できます。

(5) オリジナル RD エリアを回復する

pdrstr コマンドで，オリジナル RD エリアを回復します。

```
pdrstr -m /HiRDB/rdarea/rdmast/rdmast00 -b /bkdir/bkup04
-r FMB01,FMBIDX -q 0
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-b：レプリカ RD エリアのバックアップファイル※を指定します。

-r：オリジナル RD エリア名を指定します。

-q：回復対象の RD エリアの世代番号を指定します。

注※

「5.23.6(3)　レプリカ RD エリアのバックアップを取得する」で取得したバックアップファイルを使用します。

(6)　オリジナル RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，オリジナル RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(7)　カレント RD エリアを切り替える

pddbchg コマンドで，カレント RD エリアをレプリカ RD エリアからオリジナル RD エリアに切り替えます。

```
pddbchg -r FMB01,FMBIDX -q 0
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-q：オリジナル RD エリアの世代番号を指定します。

(8)　レプリカ RD エリアの参照可能バックアップ閉塞状態を解除し，閉塞かつクローズ状態にする

pdrels コマンドで，レプリカ RD エリアの参照可能バックアップ閉塞状態を解除します。

```
pdrels -r FMB01,FMBIDX -q 1
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-q：レプリカ RD エリアの世代番号を指定します。

そのあとに，pdhold -c コマンドで，レプリカ RD エリアをコマンド閉塞状態かつクローズ状態にします。

```
pdhold -r FMB01_1,FMBIDX_1 -c
```

[オプションの説明]

-r：レプリカ RD エリアを指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

この例では，RD エリアの構成は次のようになっています。

- FMB01_1：レコード型を格納しているレプリカ RD エリア
- FMBIDX_1：インデクスを格納しているレプリカ RD エリア

5.23.7　対処の詳細手順（その 2）

対処手順を次に示します。

(1)　追い付き反映処理を取り消す

pdorend -u コマンドで，追い付き反映処理を取り消します。

```
pdorend -s bes1 -u
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-u：追い付き反映処理を取り消す場合に指定するオプションです。

(2)　レプリカ RD エリアを参照可能バックアップ閉塞にする

pdhold -b コマンドで，レプリカ RD エリアを参照可能バックアップ閉塞にします。

```
pdhold -r FMB01_1,FMBIDX_1 -b
```

[オプションの説明]

-r：更新可能なオンライン再編成の対象 RD エリア（BES 内のレプリカ RD エリア）を指定します。

-b：RD エリアの状態を参照可能バックアップ閉塞にする場合に指定するオプションです。

この例では，RD エリアの構成は次のようになっています。

- FMB01_1：レコード型を格納しているレプリカ RD エリア
- FMBIDX_1：インデクスを格納しているレプリカ RD エリア

(3)　カレント RD エリアを切り替える

pddbchg コマンドで，カレント RD エリアをレプリカ RD エリアからオリジナル RD エリアに切り替えます。

```
pddbchg -r FMB01,FMBIDX -q 0
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-q：オリジナル RD エリアの世代番号を指定します。

(4) HiRDB を終了する

pdstop コマンドで HiRDB を終了します。

```
pdstop
```

(5) ペアボリュームを再同期する

ペアボリュームを再同期して，レプリカ RD エリアの内容をオリジナル RD エリアに反映します。操作方法については，使用しているミラーリング機能のマニュアルを参照してください。日立ディスクアレイサブシステムを使用している場合は，最初にマニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「日立ディスクアレイサブシステム使用時の注意事項および操作手順」を参照してください。

(6) HiRDB を開始する

pdstart コマンドで HiRDB を開始します。

```
pdstart
```

5.23.8 対処の詳細手順（その 3）

対処手順を次に示します。

(1) 障害が発生したボリュームを交換する

障害が発生したボリュームを交換してください。

(2) システムログをアンロードする

pdlogunld コマンドで，システムログをアンロードします。

```
pdlogunld -d sys -s bes1 -g logfg01 -o /uldir/ulog01
```

[オプションの説明]

-d：システムログファイルに対する操作のため，sys を指定します。

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-g：ログファイルグループ名を指定します。

-o：アンロードログファイル名を指定します。

(3)　オリジナル RD エリアを閉塞かつクローズ状態にする

pdhold -c コマンドで，オリジナル RD エリアを閉塞かつクローズ状態にします。

```
pdhold -r FMB01,FMBIDX -c
```

[オプションの説明]

-r：オリジナル RD エリアを指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

この例では，RD エリアの構成は次のようになっています。

- FMB01：レコード型を格納しているオリジナル RD エリア
- FMBIDX：インデクスを格納しているオリジナル RD エリア

(4)　オリジナル RD エリアを回復する

pdrstr コマンドで，オリジナル RD エリアを回復します。

```
pdrstr -m /HiRDB/rdarea/rdmast/rdmast00 -b /bkdir/bkup00
-l /uldir/ulog01 -r FMB01,FMBIDX -q 0
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-b：アンロードする前に取得したバックアップファイル※を指定します。

-l：アンロードログファイルを指定します。

-r：オリジナル RD エリア名を指定します。

-q：回復対象の RD エリアの世代番号を指定します。

注※

「5.13.4(4)　オリジナル RD エリアのバックアップを取得する（pdcopy コマンド）」で取得したバックアップファイルを使用します。

(5)　オリジナル RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，オリジナル RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(6)　更新可能なオンライン再編成の操作を再開する

更新可能なオンライン再編成の操作を再開します。次の操作から再開してください。

- 「5.13.4(1)　RD エリアをオンライン再編成閉塞にする（pdorbegin コマンド）」

なお，再編成状態が READY2 の場合（「表 5-16　正系ボリューム障害が発生した場合の対処方法」の項番 3 の場合）は，「5.13.4(1)　RD エリアをオンライン再編成閉塞にする（pdorbegin コマンド）」の操作を実施したあとのペアボリュームの解除は必要ありません。

5.23.9　対処の詳細手順（その 4）

対処手順を次に示します。

(1)　障害が発生したボリュームを交換する

障害が発生したボリュームを交換してください。

(2)　システムログをアンロードする

pdlogunld コマンドで，システムログをアンロードします。

```
pdlogunld -d sys -s bes1 -g logfg01 -o /uldir/ulog01
```

[オプションの説明]

-d：システムログファイルに対する操作のため，sys を指定します。

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-g：ログファイルグループ名を指定します。

-o：アンロードログファイル名を指定します。

(3)　オリジナル RD エリアをクローズ状態にする

pdclose コマンドで，オリジナル RD エリアをクローズ状態にします。

```
pdclose -r FMB01,FMBIDX
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

この例では，RD エリアの構成は次のようになっています。

- FMB01：レコード型を格納しているオリジナル RD エリア
- FMBIDX：インデクスを格納しているオリジナル RD エリア

(4) オリジナル RD エリアを回復する

pdrstr コマンドで，オリジナル RD エリアを回復します。

```
pdrstr -m /HiRDB/rdarea/rdmast/rdmast00 -b /bkdir/bkup01
-l /uldir/ulog01 -r FMB01,FMBIDX -q 0
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-b：アンロードする前に取得したバックアップファイル※を指定します。

-l：アンロードログファイルを指定します。

-r：オリジナル RD エリア名を指定します。

-q：回復対象の RD エリアの世代番号を指定します。

注※

「5.13.4(4) オリジナル RD エリアのバックアップを取得する（pdcopy コマンド）」で取得したバックアップを使用します。

(5) オリジナル RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，オリジナル RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(6) 更新可能なオンライン再編成の操作を再開する

更新可能なオンライン再編成の操作を再開します。次の操作から再開してください。

- 「5.13.4(5) オリジナル RD エリアのレコードをアンロードする（pdsdbrog コマンド）」

5.23.10 対処の詳細手順（その 5）

対処手順を次に示します。

(1) 障害が発生したボリュームを交換する

障害が発生したボリュームを交換してください。

(2)　RD エリアを閉塞かつクローズ状態にする

pdhold -c コマンドで，次の RD エリアを閉塞かつクローズ状態にします。

- オリジナル RD エリア
- 追い付き反映キー対応表を格納している RD エリア
- 追い付き反映キー対応表のインデクスを格納している RD エリア

```
pdhold -r FMB01,FMBIDX,RDCRT,RDIDX -c
```

[オプションの説明]

-r：閉塞かつクローズ状態にする RD エリアを指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

この例では，RD エリアの構成は次のようになっています。

- FMB01：レコード型を格納しているオリジナル RD エリア
- FMBIDX：インデクスを格納しているオリジナル RD エリア
- RDCRT：追い付き反映キー対応表を格納している RD エリア
- RDIDX：追い付き反映キー対応表のインデクスを格納している RD エリア

(3)　オリジナル RD エリアを回復する

pdrstr コマンドで，オリジナル RD エリアを回復します。

```
pdrstr -m /HiRDB/rdarea/rdmast/rdmast00 -b /bkdir/bkup02
-r FMB01,FMBIDX -q 0
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-b：データロード後に取得したバックアップファイル※を指定します。

-r：オリジナル RD エリア名を指定します。

-q：回復対象の RD エリアの世代番号を指定します。

注※

「5.13.4(7)　オリジナル RD エリアのバックアップを取得する（pdcopy コマンド）」で取得したバックアップを使用します。

(4)　追い付き反映キー対応表を格納している RD エリアを回復する

pdrstr コマンドで，次の RD エリアを回復します。

- 追い付き反映キー対応表を格納している RD エリア

- 追い付き反映キー対応表のインデクスを格納している RD エリア

```
pdrstr -m /HiRDB/rdarea/rdmast/rdmast00 -b /bkdir/bkup03
-r RDCRT,RDIDX
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-b：データロード後に取得したバックアップファイル※を指定します。

-r：回復対象の RD エリア名を指定します。

注※

「5.13.4(8)　追い付き反映キー対応表を格納している RD エリアのバックアップを取得する（pdcopy コマンド）」で取得したバックアップを使用します。

(5)　RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，次の RD エリアの閉塞を解除し，オープン状態にします。

- オリジナル RD エリア
- 追い付き反映キー対応表を格納している RD エリア
- 追い付き反映キー対応表のインデクスを格納している RD エリア

```
pdrels -r FMB01,FMBIDX,RDCRT,RDIDX -o
```

[オプションの説明]

-r：閉塞を解除し，オープン状態にする RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(6)　追い付き反映処理を実行する（pdorend コマンド）

pdorend コマンドで，追い付き反映処理を実行します。

```
pdorend -s bes1 -z
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-z：システムログ反映開始ポイントから追い付き反映処理を再実行する場合に指定するオプションです。

以降，次の操作から，更新可能なオンライン再編成の操作を再開してください。

- 「5.13.4(12)　ペアボリュームを再同期する」

なお，pd_lv_mirror_use オペランドに Y を指定していない場合は，次の操作から，更新可能なオンライン再編成の操作を再開してください。

- 「5.13.4(10)　ファイルアクセスプロセスを停止する（pdpfresh コマンド）」

5.23.11　対処の詳細手順（その 6）

対処手順を次に示します。

(1)　追い付き反映処理を取り消す

pdorend -u コマンドで，追い付き反映処理を取り消します。

```
pdorend -s bes1 -u
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-u：追い付き反映処理を取り消す場合に指定するオプションです。

(2)　オリジナル RD エリアを閉塞かつクローズ状態にする

pdhold -c コマンドで，オリジナル RD エリアを閉塞かつクローズ状態にします。

```
pdhold -r FMB01,FMBIDX -c
```

[オプションの説明]

-r：オリジナル RD エリアを指定します。

-c：RD エリアをクローズ状態にする場合に指定するオプションです。

この例では，RD エリアの構成は次のようになっています。

- FMB01：レコード型を格納しているオリジナル RD エリア
- FMBIDX：インデクスを格納しているオリジナル RD エリア

(3)　システムログをアンロードする

pdlogunld コマンドで，システムログをアンロードします。

```
pdlogunld -d sys -s bes1 -g logfg02 -o /uldir/ulog01
```

[オプションの説明]

-d：システムログファイルに対する操作のため，sys を指定します。

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

-g：ログファイルグループ名を指定します。

-o：アンロードログファイル名を指定します。

(4) オリジナル RD エリアを回復する

pdrstr コマンドで，オリジナル RD エリアを回復します。

アンロードする前に取得したバックアップファイルとレプリカ RD エリアのシステムログを使用して，オリジナル RD エリアを障害発生直前の状態に回復します。

```
pdrstr -m /HiRDB/rdarea/rdmast/rdmast00 -b /bkdir/bkup01
-l /uldir/ulog01 -r FMB01,FMBIDX -q 0
```

[オプションの説明]

-m：マスタディレクトリ用 RD エリアの先頭 HiRDB ファイル名を指定します。

-b：アンロードする前に取得したバックアップファイル※を指定します。

-l：アンロードログファイルを指定します。

-r：オリジナル RD エリア名を指定します。

-q：回復対象の RD エリアの世代番号を指定します。

注※

「5.13.4(4) オリジナル RD エリアのバックアップを取得する（pdcopy コマンド）」で取得したバックアップファイルを使用します。

(5) オリジナル RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，オリジナル RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels -r FMB01,FMBIDX -o
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(6) カレント RD エリアを切り替える

pddbchg コマンドで，カレント RD エリアをレプリカ RD エリアからオリジナル RD エリアに切り替えます。

```
pddbchg -r FMB01,FMBIDX -q 0
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-q：オリジナル RD エリアの世代番号を指定します。

5.23.12　対処の詳細手順（その 7）

対処手順を次に示します。

(1)　レプリカ RD エリアをクローズ状態にする

レプリカ RD エリアがクローズ状態でない場合，pdclose コマンドでレプリカ RD エリアをクローズ状態にします。

```
pdclose -r FMB01,FMBIDX -q 1
```

[オプションの説明]

-r：オリジナル RD エリア名を指定します。

-q：レプリカ RD エリアの世代番号を指定します。

> **参考**
>
> RD エリアの状態は，pddbls コマンドで確認できます。

この例では，RD エリアの構成は次のようになっています。

- FMB01：レコード型を格納しているオリジナル RD エリア
- FMBIDX：インデクスを格納しているオリジナル RD エリア
- FMB01_1：レコード型を格納しているレプリカ RD エリア
- FMBIDX_1：インデクスを格納しているレプリカ RD エリア

(2)　レプリカ RD エリアを再初期化する

pdmod コマンドの initialize rdarea 文で，レプリカ RD エリアを再初期化します。

```
pdmod -a /usr/pdmod/initialize
```

[オプションの説明]

-a：pdmod コマンドの制御文ファイル名を指定します。制御文の内容を次に示します。

```
initialize rdarea FMB01_1           ...1
    with reconstruction             ...2
    file name ”/sdb/area0”          ...3
            initial 100 segments
    file name ”/sdb/area1”          ...3
            initial 100 segments;
```

1. レプリカ RD エリアを再初期化する指定です。
2. オリジナル RD エリアの構成と合わせるために指定します。

3. 再初期化対象のレプリカ RD エリアの HiRDB ファイルシステム領域の名称と HiRDB ファイルの名称を指定します。

レプリカ RD エリアの構成情報が，オリジナル RD エリアの構成情報と一致するように制御文を作成してください。

(3) ファイルアクセスプロセスを停止する

pd_lv_mirror_use オペランドに Y を指定している場合，(3)の操作は必要ありません。

pdpfresh コマンドで，ファイルアクセスプロセスを停止します。

```
pdpfresh -s bes1
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

(4) ペアボリュームを再同期する

ペアボリュームを再同期して，正系ボリュームの内容を副系ボリュームにコピーします。その後，ペア状態を解除します。

操作方法については，使用しているミラーリング機能のマニュアルを参照してください。日立ディスクアレイサブシステムを使用している場合は，最初にマニュアル「インナレプリカ機能 HiRDB Staticizer Option」の「日立ディスクアレイサブシステム使用時の注意事項および操作手順」を参照してください。

(5) カレント RD エリアを切り替える（pdorchg コマンド）

pdorchg コマンドで，カレント RD エリアを切り替えてください。

```
pdorchg -s bes1
```

[オプションの説明]

-s：更新可能なオンライン再編成の対象 RD エリアがある BES を指定します。

(6) レプリカ RD エリアの閉塞を解除し，オープン状態にする

pdrels -o コマンドで，レプリカ RD エリアの閉塞を解除し，オープン状態にします。

```
pdrels - r  FMB01_1,FMBIDX_1 -o
```

[オプションの説明]

-r：レプリカ RD エリア名を指定します。

-o：RD エリアをオープン状態にする場合に指定するオプションです。

(7) 更新可能なオンライン再編成の操作を再開する

更新可能なオンライン再編成の操作を再開します。次の操作から再開してください。

- 「5.13.4(5) オリジナル RD エリアのレコードをアンロードする（pdsdbrog コマンド）」

5.24　OS の時刻を変更する方法

OS の時刻を変更する方法については，マニュアル「HiRDB システム運用ガイド」の「OS の時刻を変更する方法」を参照してください。

なお，OS の時刻を遅らせた場合，次に示す手順で SDB ディレクトリ情報ファイルを再作成し，全ユニットに再配布してください。

手順

1. HiRDB を再起動します。
2. SDB ディレクトリ情報ファイルを再作成し，全ユニットに再配布します。
3. HiRDB を再起動します。

なお，上記の操作の間は，トランザクションを静止化しておいてください。

注意事項

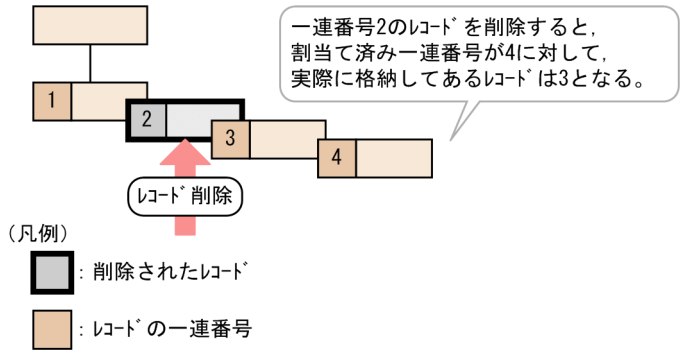
OS の時刻を遅らせた場合，それ以前の SDB ディレクトリ情報ファイルは使用しないでください。

5.25　一連番号の監視

一連番号の監視とは，あらかじめ決めておいた監視値まで一連番号を割り当てたときに，一連番号が枯渇していることをメッセージで通知する機能です。

レコードの格納時に割り当てられる一連番号は，レコードの格納と削除を繰り返すうちに利用されない一連番号ができ，レコード数が少なくても割り当てられる一連番号の最大値に達して，レコードが格納できなくなってしまいます。割り当て済み一連番号と実際のレコード件数の関係を次の図に示します。

図 5-21　割り当て済み一連番号と実際のレコード件数の関係

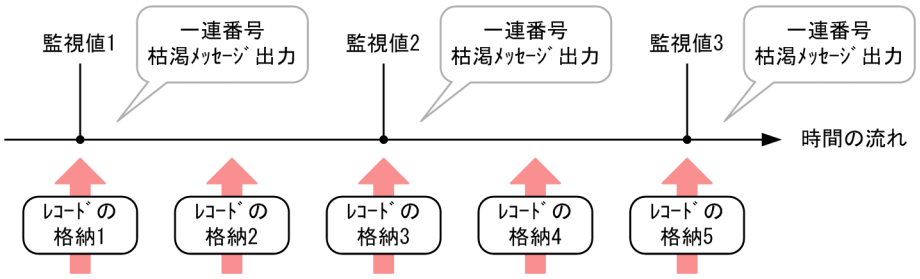


一連番号を再利用することもできますが（SDB データベース定義の SETOPTION 句下の REUSE 句に YES を指定），この場合，一連番号が再利用されるのは最後尾のレコードを削除した場合だけであり，それ以外のレコードを削除した場合は有効とはなりません。

レコードの格納と削除の繰り返しによって一連番号が枯渇した場合は，SDB データベースを再編成することで解消できます。割り当て済みの一連番号を監視することで，SDB データベースの再編成契機を判断できます。

一連番号の監視の概要を次の図に示します。

図 5-22　一連番号の監視の概要



[説明]

一連番号の監視は，SDB データベース定義の SETOPTION 句下の WARNING 句，または SDBOPTION 句下の OCCURRENCE WARNING 句で指定します。

一連番号の監視では，監視値（監視する一連番号使用比率）を 3 つまで指定できます。ここでは，監視する一連番号使用比率の小さい順に監視値 1，監視値 2，監視値 3 とします。

監視値に達するごとに一連番号の枯渇メッセージが出力されます。以降，一連番号の枯渇メッセージが出力されたかどうかの状態が内部的に管理されます。これを一連番号の枯渇メッセージの出力状態といいます。一連番号の枯渇メッセージの出力状態が内部的に管理されるため，レコードが格納されても，次の監視値に達するまでメッセージは出力されません。

一度に複数の監視値を超えた場合は，対応するメッセージがすべて出力されます。

なお，例えば監視値 2 までメッセージが出力されている状態でも，次に示す場合には，再度，監視値 1（監視する一連番号使用比率のうち，いちばん小さい値）から監視されます。

- HiRDB を正常開始，再開始，または強制開始した場合
- HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更を行って，定義を更新した場合
- 次に示す RD エリアの場合

  ・クローズ状態の RD エリアを pdopen コマンド，または pdrels -o コマンドでオープンした場合，その RD エリア

  ・HiRDB/SD データベース作成ユティリティ（pdsdblod）の environment 文の purge オペランドに yes を指定して実行した RD エリア（インナレプリカ機能を使用したオンライン再編成の場合も含む）

  ・HiRDB をユニット単位に正常開始，再開始，または強制開始した場合，該当ユニット内の BES が管理するすべての RD エリア

5.25.1 注意事項

- レコードの一括削除またはレコードの削除を行っても，一連番号の枯渇メッセージの出力状態は変わりません。そのため，再度同一の一連番号で，一連番号の枯渇メッセージが出力されることはありません。
- 一連番号の監視に対する排他処理は行わないので，複数のトランザクションから同一 RD エリアに対して，レコードの格納が同時に実行されると，同一のメッセージが複数回出力されることがあります。

5.25.2 レプリカ RD エリアを使用する場合の運用

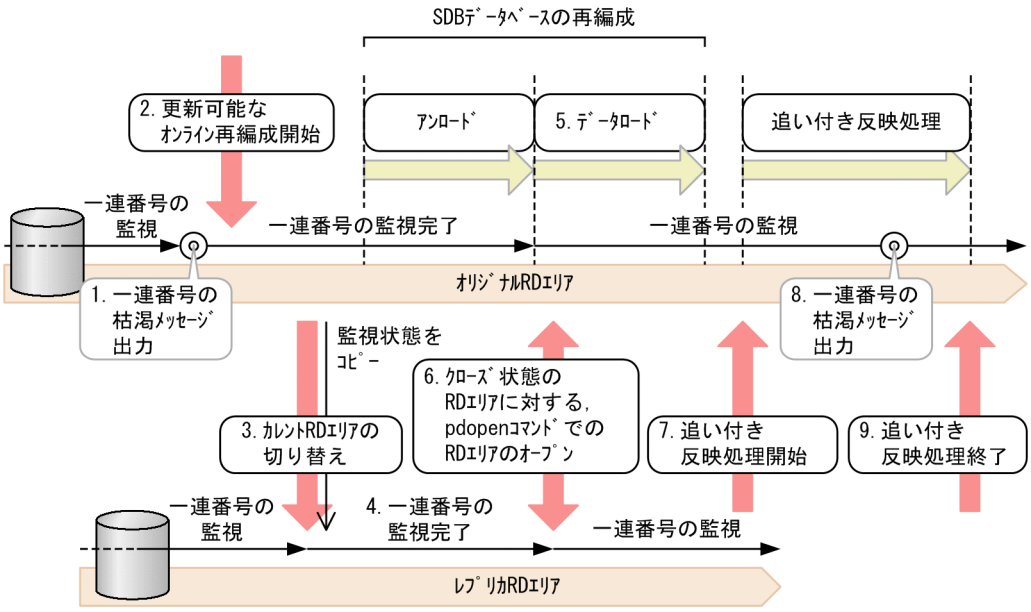
レプリカ RD エリアを使用する場合の運用を次に示します。

(1) 更新可能なオンライン再編成でレプリカ RD エリアを使用する場合【4V FMB】

更新可能なオンライン再編成では，レプリカ RD エリアでの一連番号の枯渇を防ぐため，オリジナル RD エリアとレプリカ RD エリアで，それぞれ一連番号の監視を行います。

更新可能なオンライン再編成でレプリカ RD エリアを使用する場合の，一連番号の監視の運用について，次の図に示します。

図 5-23　更新可能なオンライン再編成でレプリカ RD エリアを使用する場合の，一連番号の監視の運用



[説明]

1. オンライン運用中に一連番号を監視します。指定した監視値のうち，一連番号使用比率のいちばん大きい値に対応する一連番号の枯渇メッセージまで出力されると，以降，同一 RD エリア（オリジナル RD エリア）ではメッセージは出力されません（一連番号の監視完了となります）。
2. pdorbegin コマンドで更新可能なオンライン再編成を開始します。
3. pdorchg コマンドでカレント RD エリアをオリジナル RD エリアからレプリカ RD エリアに切り替える直前，一連番号の枯渇メッセージの出力状態がオリジナル RD エリアからレプリカ RD エリアにコピーされます。
4. この例の場合，3.の直前でオリジナル RD エリアに対する一連番号の枯渇メッセージの出力が完了しています（一連番号の監視完了の状態となっています）。レプリカ RD エリアには，その状態が引き継がれるため，メッセージは出力されません。
5. SDB データベースの再編成をすると，一連番号が再利用されるようになります。このため，データロードが完了した RD エリアは，HiRDB/SD データベース作成ユティリティ（pdsdblod）の終了時に，指定した監視値のうち，一連番号使用比率のいちばん小さい値から監視される状態になります。
6. RD エリアがクローズ状態の場合，pdopen コマンドを入力してオープン状態にすると，オリジナル RD エリア，レプリカ RD エリアそれぞれ別に，指定した監視値のうち，一連番号使用比率のいちばん小さい値から監視される状態になります。
7. pdorend コマンドで追い付き反映処理を開始します。
8. 追い付き反映処理で割り当て済み一連番号が監視値を超えた場合，一連番号の枯渇メッセージが出力されます。この場合，SDB データベース定義の SET 句下の OCCURENCE NUMBER 句，および SDBOPTION 句下の OCCURRENCE WARNING 句の指定値に問題がないか見直してから，SDB データベースを再作成する必要があります。

9. 追い付き反映処理が終了します。カレント RD エリアがレプリカ RD エリアからオリジナル RD エリアに切り替わります。このとき，一連番号の枯渇メッセージの出力状態は変わりません（レプリカ RD エリアでの出力状態は引き継がれません）。

> **参考**
>
> 一連番号の監視の概要については，「図 5-22　一連番号の監視の概要」の説明を参照してください。

(2)　更新可能なオンライン再編成以外でレプリカ RD エリアを使用する場合【4V FMB，4V AFM】

pddbchg コマンドまたは環境変数 PDDBACCS を指定してレプリカ RD エリアを使用した場合，一連番号の枯渇メッセージは出力されません。

5.26　PRF トレース機能

PRF トレース機能とは，HiRDB の一連の処理に対して出力される，トラブルシュート用のトレース情報を取得する機能です。

ここでは，HiRDB/SD に関する PRF トレース情報についてだけ説明します。PRF トレース情報を取得するための設定や取得内容，PRF トレース情報の参照手順など，「PRF トレース機能」の詳細については，マニュアル「HiRDB システム運用ガイド」の「PRF トレース機能」を参照してください。

5.26.1　PRF トレース情報の詳細と取得ポイント

HiRDB/SD に関する PRF トレース情報のイベント名とそのイベントに関連する情報を次の表に示します。また，PRF トレース情報の主な取得ポイントを「図 5-24　PRF トレースの取得ポイントの概念図 (HiRDB/SD の場合)」に示します。

なお，PRF トレース情報には次に示す情報以外に，HiRDB の内部情報を出力することがあります。

表 5-20　PRF トレース情報の主な取得ポイント一覧（HiRDB/SD）

項番	イベント名	イベント ID	取得ポイント (図中の番号) ※1	主なプロセス	イベント種別※2	出力レベル
1	参照バッファ読み込み要求開始	0xD205D401	8	pdbes	8	00000004
2	参照バッファ読み込み要求完了	0xD206D401	8	pdbes	8	00000004
3	更新バッファ読み込み要求開始	0xD205D402	8	pdbes	8	00000004
4	更新バッファ読み込み要求完了	0xD206D402	8	pdbes	8	00000004
5	バッファ書き込み要求開始	0xD205D403	8	pdbes	8	00000004
6	バッファ書き込み要求完了	0xD206D403	8	pdbes	8	00000004
7	バッファ解放要求開始	0xD205D404	8	pdbes	8	00000004
8	バッファ解放要求完了	0xD206D404	8	pdbes	8	00000004
9	最終セグメント内の使用中ページ確保要求開始	0xD205D405	8	pdbes	8	00000004
10	最終セグメント内の使用中ページ確保要求完了	0xD206D405	8	pdbes	8	00000004
11	ページ満杯設定要求開始	0xD205D406	8	pdbes	8	00000004
12	ページ満杯設定要求完了	0xD206D406	8	pdbes	8	00000004
13	ページ満杯解除要求開始	0xD205D407	8	pdbes	8	00000004
14	ページ満杯解除要求完了	0xD206D407	8	pdbes	8	00000004

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	出力レベル
15	ページ解放要求開始	0xD205D408	8	pdbes	8	00000004
16	ページ解放要求完了	0xD206D408	8	pdbes	8	00000004
17	データ検索要求開始	0xD205D409	8	pdbes	8	00000004
18	データ検索要求終了	0xD206D409	8	pdbes	8	00000004
19	データ検索用メモリ解放要求開始	0xD205D40A	8	pdbes	8	00000004
20	データ検索用メモリ解放要求終了	0xD206D40A	8	pdbes	8	00000004
21	一意性制約チェック要求開始	0xD205D40B	8	pdbes	8	00000004
22	一意性制約チェック要求終了	0xD206D40B	8	pdbes	8	00000004
23	インデクスメンテナンス要求開始	0xD205D40C	8	pdbes	8	00000004
24	インデクスメンテナンス要求完了	0xD206D40C	8	pdbes	8	00000004
25	ログ出力要求開始	0xD205D40E	8	pdbes	8	00000004
26	ログ出力要求完了	0xD206D40E	8	pdbes	8	00000004
27	指定ページの近傍空きページ確保要求開始	0xD205D40F	8	pdbes	8	00000004
28	指定ページの近傍空きページ確保要求完了	0xD206D40F	8	pdbes	8	00000004
29	最終セグメント内の空きページ確保要求開始	0xD205D410	8	pdbes	8	00000004
30	最終セグメント内の空きページ確保要求完了	0xD206D410	8	pdbes	8	00000004
31	データロード時の空きページ確保要求開始	0xD205D411	−	pdbes	8	00000004
32	データロード時の空きページ確保要求完了	0xD206D411	−	pdbes	8	00000004
33	未使用ページ読み込み要求開始	0xD205D412	8	pdbes	8	00000004
34	未使用ページ読み込み要求終了	0xD206D412	8	pdbes	8	00000004
35	全インデクスページ解放要求開始	0xD205D413	−	pdbes	8	00000004
36	全インデクスページ解放要求完了	0xD206D413	−	pdbes	8	00000004

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	出力レベル
37	インデクスのルートページ情報更新要求開始	0xD205D414	−	pdbes	8	00000004
38	インデクスのルートページ情報更新要求完了	0xD206D414	−	pdbes	8	00000004
39	インデクス一括作成日時設定要求開始	0xD205D415	−	pdbes	8	00000004
40	インデクス一括作成日時設定要求完了	0xD206D415	−	pdbes	8	00000004
41	指定ページの近傍使用中ページ確保要求開始	0xD205D416	8	pdbes	8	00000004
42	指定ページの近傍使用中ページ確保要求完了	0xD206D416	8	pdbes	8	00000004
43	表検索用バッファ読み込み要求開始	0xD205D417	−	pdbes	8	00000004
44	表検索用バッファ読み込み要求終了	0xD206D417	−	pdbes	8	00000004
45	次セグメント内の使用中ページ番号の取得要求開始	0xD205D418	−	pdbes	8	00000004
46	次セグメント内の使用中ページ番号の取得要求完了	0xD206D418	−	pdbes	8	00000004
47	先頭使用中ページ番号の取得要求開始	0xD205D419	−	pdbes	8	00000004
48	先頭使用中ページ番号の取得要求完了	0xD206D419	−	pdbes	8	00000004
49	指定セグメント内の空きページ確保要求開始	0xD205D41A	8	pdbes	8	00000004
50	指定セグメント内の空きページ確保要求完了	0xD206D41A	8	pdbes	8	00000004
51	指定セグメント内の使用中ページ確保要求開始	0xD205D41B	8	pdbes	8	00000004
52	指定セグメント内の使用中ページ確保要求完了	0xD206D41B	8	pdbes	8	00000004
53	後続セグメント内の空きページサーチ要求開始	0xD205D41C	8	pdbes	8	00000004
54	後続セグメント内の空きページサーチ要求完了	0xD206D41C	8	pdbes	8	00000004

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	出力レベル
55	SDB 用 UAP 環境定義ファイルのオープン開始	0x1105D41D	10	pdfes	8	00000004
56	SDB 用 UAP 環境定義ファイルのオープン完了	0x1106D41D	10	pdfes	8	00000004
57	SDB 用 UAP 環境定義ファイルの読み込み開始	0x1105D41E	10	pdfes	8	00000004
58	SDB 用 UAP 環境定義ファイルの読み込み完了	0x1106D41E	10	pdfes	8	00000004
59	SDB 用 UAP 環境定義ファイルのクローズ開始	0x1105D41F	10	pdfes	8	00000004
60	SDB 用 UAP 環境定義ファイルのクローズ完了	0x1106D41F	10	pdfes	8	00000004
61	ルートレコード格納ページの排他の再確保処理開始	0xD22FD420	8	pdbes	8	00000004
62	ルートレコード格納ページの排他の再確保処理終了	0xD22FD421	8	pdbes	8	00000004
63	ページ ID とテーブル ID の整合性チェック要求開始	0xD205D422	8	pdbes	8	00000004
64	ページ ID とテーブル ID の整合性チェック要求終了	0xD206D422	8	pdbes	8	00000004
65	データベース名称管理テーブルの確保開始	0xD103D101	8	pdbes	8	00000004
66	データベース名称管理テーブルの確保終了	0xD104D101	8	pdbes	8	00000004
67	HiRDB/SD 処理開始	0xD003D200	1，3	pdbes, pdfes	8	00000004
68	HiRDB/SD 処理終了	0xD004D200	4，6	pdbes, pdfes	8	00000004
69	個別開始要求処理開始	0xD105D201	8	pdbes	8	00000004
70	個別開始要求処理終了	0xD106D201	8	pdbes	8	00000004
71	更新可能なオンライン再編成時の追い付きログの出力開始	0xD205D202	8	pdbes	8	00000004
72	更新可能なオンライン再編成時の追い付きログの出力完了	0xD206D202	8	pdbes	8	00000004
73	FES-BES 間通信処理開始	0xD005D203	2	pdfes	8	00000004
74	FES-BES 間通信処理終了	0xD006D203	5	pdfes	8	00000004

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	出力レベル
75	FES-BES 間通信処理	0xD003D204	3，4	pdbes	8	00000004
76	レコードの検索要求処理開始	0xD205D210	8	pdbes	8	00000004
77	レコードの検索要求処理終了	0xD206D210	8	pdbes	8	00000004
78	レコードの格納要求処理開始	0xD205D211	8	pdbes	8	00000004
79	レコードの格納要求処理終了	0xD206D211	8	pdbes	8	00000004
80	レコードの変更要求処理開始	0xD205D212	8	pdbes	8	00000004
81	レコードの変更要求処理終了	0xD206D212	8	pdbes	8	00000004
82	レコードの削除要求処理開始	0xD205D213	8	pdbes	8	00000004
83	レコードの削除要求処理終了	0xD206D213	8	pdbes	8	00000004
84	レコードの一括削除要求処理開始	0xD205D214	8	pdbes	8	00000004
85	レコードの一括削除要求処理終了	0xD206D214	8	pdbes	8	00000004
86	容量情報取得要求処理開始	0xD205D215	8	pdbes	8	00000004
87	容量情報取得要求処理終了	0xD206D215	8	pdbes	8	00000004
88	複数レコードの検索要求処理開始	0xD205D216	8	pdbes	8	00000004
89	複数レコードの検索要求処理終了	0xD206D216	8	pdbes	8	00000004
90	レコードの取得要求処理開始	0xD205D217	8	pdbes	8	00000004
91	レコードの取得要求処理終了	0xD206D217	8	pdbes	8	00000004
92	個別開始実行時の管理領域確保の開始	0xD105D300	8	pdbes	8	00000004
93	個別開始実行時の管理領域確保の完了	0xD106D301	8	pdbes	8	00000004
94	レコードの検索実行時のレコード取得の開始	0xD205D310 0xD205D312 0xD205D314	8	pdbes	8	00000004
95	レコードの検索実行時のレコード取得の完了	0xD206D311 0xD206D313 0xD206D315	8	pdbes	8	00000004
96	レコードの格納実行時の挿入位置検索の開始	0xD205D320 0xD205D322	8	pdbes	8	00000004

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	出力レベル
		0xD205D328				
97	レコードの格納実行時の挿入位置検索の完了	0xD206D321 0xD206D323 0xD206D329	8	pdbes	8	00000004
98	レコードの格納実行時のレコード格納の開始	0xD205D324	8	pdbes	8	00000004
99	レコードの格納実行時のレコード格納の完了	0xD206D325	8	pdbes	8	00000004
100	レコードの格納実行時の親子集合接続の開始	0xD205D326	8	pdbes	8	00000004
101	レコードの格納実行時の親子集合接続の完了	0xD206D327	8	pdbes	8	00000004
102	レコードの更新実行時の更新対象レコード取得の開始	0xD205D330	8	pdbes	8	00000004
103	レコードの更新実行時の更新対象レコード取得の完了	0xD206D331	8	pdbes	8	00000004
104	レコードの更新実行時のレコード変更の開始	0xD205D332	8	pdbes	8	00000004
105	レコードの更新実行時のレコード変更の完了	0xD206D333	8	pdbes	8	00000004
106	レコードの更新実行時のUSER ポインタ設定の開始	0xD205D334	8	pdbes	8	00000004
107	レコードの更新実行時のUSER ポインタ設定の完了	0xD206D335	8	pdbes	8	00000004
108	レコードの更新実行時のUSER ポインタ解除の開始	0xD205D336	8	pdbes	8	00000004
109	レコードの更新実行時のUSER ポインタ解除の完了	0xD206D337	8	pdbes	8	00000004
110	レコードの削除実行時の削除対象レコード取得の開始	0xD205D340	8	pdbes	8	00000004
111	レコードの削除実行時の削除対象レコード取得の完了	0xD206D341	8	pdbes	8	00000004
112	レコードの削除実行時のレコード削除の開始	0xD205D342 0xD205D346 0xD205D348	8	pdbes	8	00000004

項番	イベント名	イベント ID	取得ポイント（図中の番号）※1	主なプロセス	イベント種別※2	出力レベル
113	レコードの削除実行時のレコード削除の完了	0xD206D343 0xD206D347 0xD206D349	8	pdbes	8	00000004
114	レコードの削除実行時の親子集合切り離しの開始	0xD205D344	8	pdbes	8	00000004
115	レコードの削除実行時の親子集合切り離しの完了	0xD206D345	8	pdbes	8	00000004
116	レコード検索実行時の排他解除の開始	0xD205D350	8	pdbes	8	00000004
117	レコード検索実行時の排他解除の完了	0xD206D351	8	pdbes	8	00000004
118	UAP 統計レポート取得時の管理領域確保の開始	0xD205D352	9	pdfes	8	00000004
119	UAP 統計レポート取得時の管理領域確保の完了	0xD206D353	9	pdfes	8	00000004
120	UAP 統計レポート取得時の情報書き込みの開始	0xD205D354	9	pdfes	8	00000004
121	UAP 統計レポート取得時の情報書き込みの完了	0xD206D355	9	pdfes	8	00000004
122	複数レコードの検索実行時のレコード取得の開始	0xD205D360	8	pdbes	8	00000004
123	複数レコードの検索実行時のレコード取得の完了	0xD206D361	8	pdbes	8	00000004
124	複数レコードの検索実行時の排他解除の開始	0xD205D362	8	pdbes	8	00000004
125	複数レコード検索実行時の排他解除の完了	0xD206D363	8	pdbes	8	00000004
126	位置指示子への空値設定実行時の排他解除の開始	0xD205D370	8	pdbes	8	00000004
127	位置指示子への空値設定実行時の排他解除の完了	0xD206D371	8	pdbes	8	00000004
128	レコードの取得実行時のレコード取得の開始	0xD205D380	8	pdbes	8	00000004
129	レコードの取得実行時のレコード取得の完了	0xD206D381	8	pdbes	8	00000004

（凡例）

－：SDB データベースへのアクセスや，COMMIT の延長で動作しないイベントです。

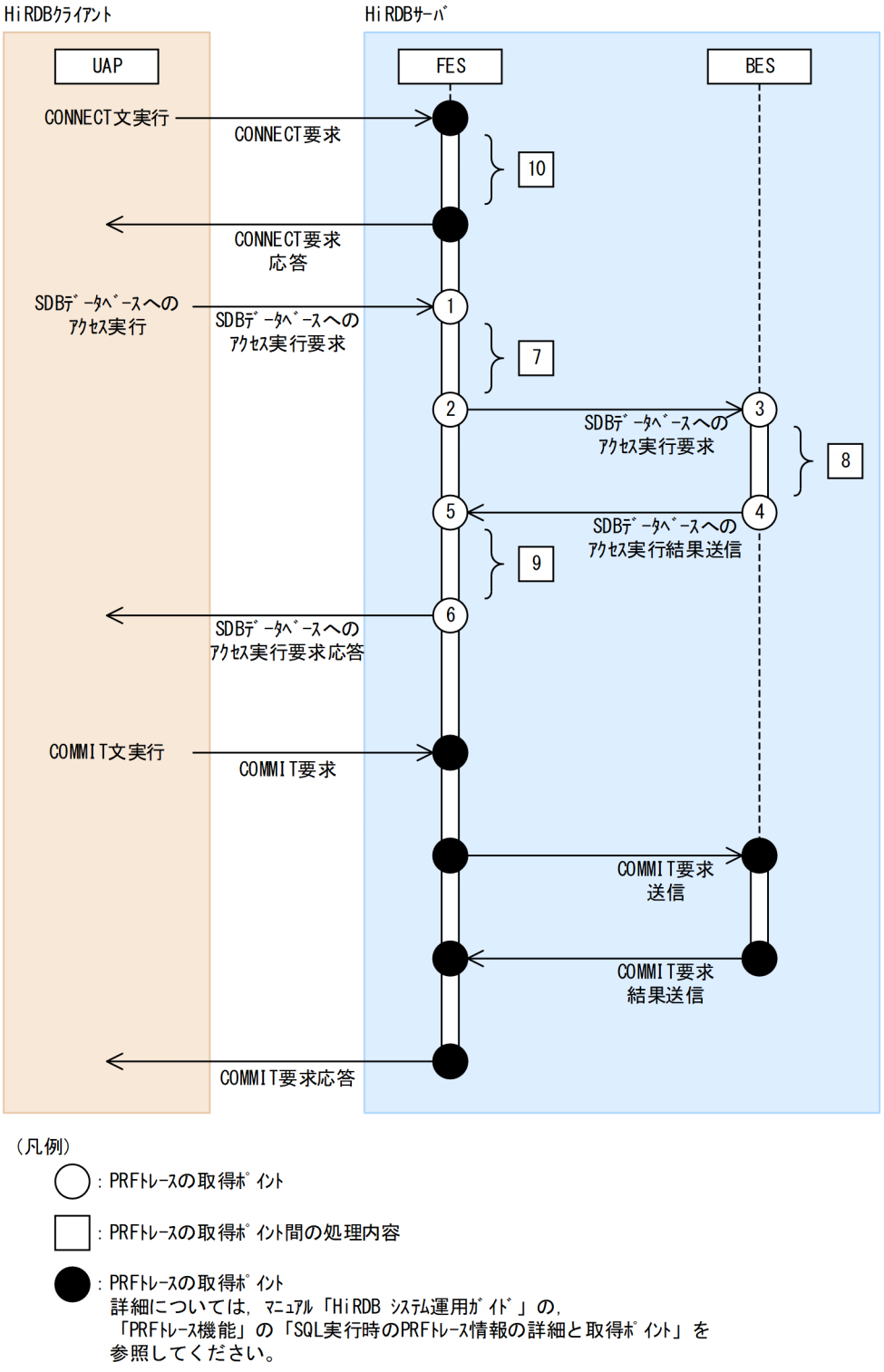
注※1

「図 5-24　PRF トレースの取得ポイントの概念図（HiRDB/SD の場合）」の番号と対応しています。図中の対応する番号のポイントで取得されることを意味します。

注※2

イベント種別は「8：その他」です。そのほかのイベント種別については，マニュアル「HiRDB システム運用ガイド」の「PRF トレース機能」を参照してください。

図 5-24　PRF トレースの取得ポイントの概念図（HiRDB/SD の場合）



（凡例）

○：PRFトレースの取得ポイント

□：PRFトレースの取得ポイント間の処理内容

●：PRFトレースの取得ポイント
詳細については，マニュアル「HiRDB システム運用ガイド」の，
「PRFトレース機能」の「SQL実行時のPRFトレース情報の詳細と取得ポイント」を
参照してください。

5.27　システム構成を変更する方法

ユニット構成やサーバ構成を変更する方法については，マニュアル「HiRDB システム運用ガイド」の「システム構成を変更する方法」を参照してください。

ただし，マニュアル「HiRDB システム運用ガイド」に記載されている内容とは仕様差があります。ここでは，その仕様差についてだけ説明します。

仕様差のある個所

マニュアル「HiRDB システム運用ガイド」の「システム構成を変更する方法」の次の個所。

- 「ユニットの追加」

  「HiRDB の稼働中にユニットを追加する場合」，および「HiRDB を終了してユニットを追加する場合」の「新規サーバマシンを準備します」の手順

- 「ユニットの移動」

  「HiRDB の稼働中にユニットを追加する場合」，および「HiRDB を終了してユニットを追加する場合」の「新規サーバマシンを準備します」の手順

HiRDB/SD を使用する場合の注意点（仕様差）

HiRDB/SD を使用する場合は，上記個所に記載されている「既存のユニットに合わせること」に加えて，次の点に注意してください。

- 既存のユニットが HiRDB/SD である必要があります。
- HiRDB/SD のバージョンを既存のユニットと合わせる必要があります。

6 高速系切り替え機能の設定と運用

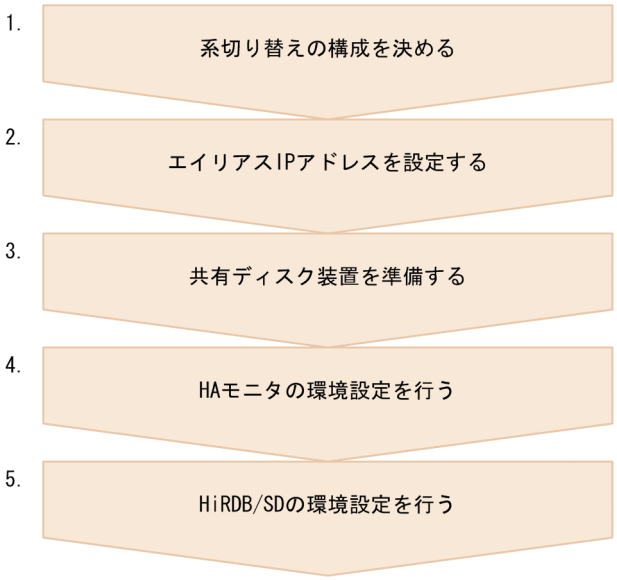
この章では，高速系切り替え機能の設定方法と運用方法について説明します。

なお，この章の説明は，HA モニタについての知識があることを前提にしています。

6.1　高速系切り替え機能の環境設定の流れ

高速系切り替え機能の環境設定の流れを次の図に示します。

図 6-1　高速系切り替え機能の環境設定の流れ



［説明］

1. 系切り替えの構成を決めます。1：1 系切り替え構成にするか，相互系切り替え構成にするか，またはそのほかの構成にするかを決めます。詳細については，「6.2　システム構成例」を参照してください。
2. 系切り替え時に使用する IP アドレスをエイリアス IP アドレスとして設定します。詳細については，「6.3　IP アドレス（ホスト名）の構成例」を参照してください。
3. 共有ディスクを準備して，共有ディスク上に HiRDB ファイルシステム領域を作成します。詳細については，「6.4　共有ディスクの準備」を参照してください。
4. HA モニタをインストールして環境設定を行います。詳細については，「6.5　HA モニタの環境設定」を参照してください。
5. HiRDB をインストールして環境設定を行います。詳細については，「6.6　HiRDB の環境設定」～「6.10　停止中のユニットがある状態でもシステムマネジャユニットの系切り替えを実行する設定」を参照してください。

6.2　システム構成例

システム構成例については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の構築」にある「高速系切り替え機能の使用時のシステム構成例」を参照してください。

6.3 IP アドレス（ホスト名）の構成例

IP アドレス（ホスト名）の構成例については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の構築」にある「IP アドレスの構成例」を参照してください。

6.4　共有ディスクの準備

共有ディスクの準備については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の構築」にある「共有ディスク装置の準備」を参照してください。

6.5　HA モニタの環境設定

HA モニタの環境設定については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の構築」にある「HA モニタに関する準備」を参照してください。

6.6　HiRDB の環境設定

ここでは，HiRDB の環境設定をする際の考慮点について説明します。

6.6.1　前提条件および注意事項

(1)　現用系と予備系で一致させること

現用系と予備系で一致させることについては，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の構築」にある「現用系と予備系で一致させること」を参照してください。

(2)　環境設定に関する注意事項

環境設定に関する注意事項については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の構築」にある「環境設定に関する注意事項」を参照してください。

(3)　SDB ディレクトリ情報ファイルに関する注意事項

- 現用系と予備系の HiRDB システム定義の指定内容を同じにするため，SDB ディレクトリ情報ファイルを格納するディレクトリのパスは，現用系と予備系で同じにしてください。
- SDB ディレクトリ情報ファイルを格納するディレクトリ（pd_structured_directory_path オペランドに指定するディレクトリ）を共有ディスク上に作成しないでください。
- 実行系ユニットと待機系ユニットで，同じ SDB ディレクトリ情報ファイルを使用する必要があります。
  待機系ユニットの SDB ディレクトリ情報ファイルを変更した場合，待機系ユニットを開始したあとに，待機系ユニットの常用常駐領域に新しい SDB ディレクトリ情報が常駐されたかどうかを確認してください。確認方法については，「5.9.2　常用常駐領域の SDB ディレクトリ情報の確認方法（待機系ユニットの場合）」を参照してください。
- 1 つのサーバマシンに複数のユニットがある場合（相互系切り替え構成の場合），サーバマシン内の複数のユニットで SDB ディレクトリ情報ファイルを共用できます。

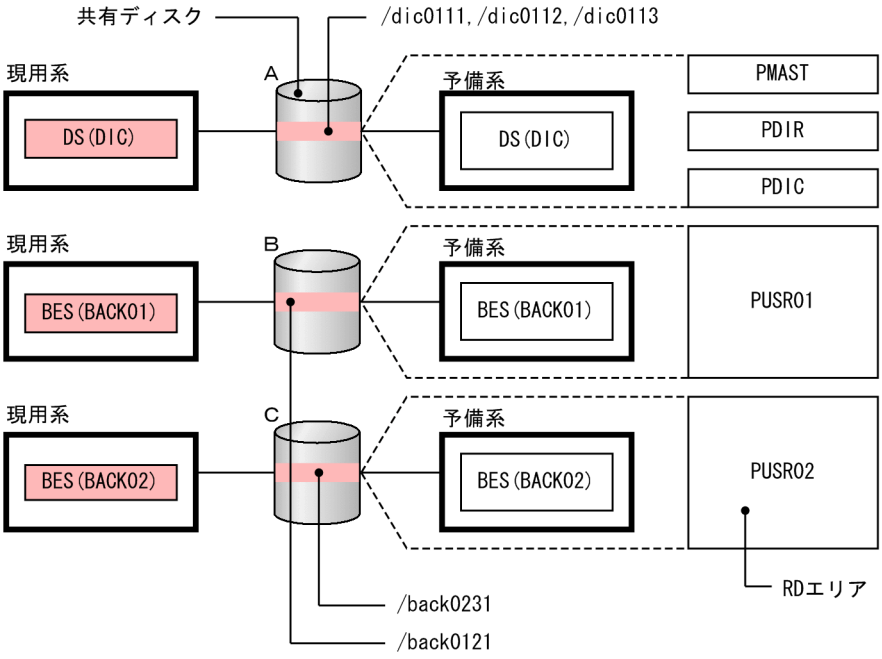
6.6.2　HiRDB システム定義の作成

HiRDB システム定義の作成については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の構築」にある「HiRDB システム定義の作成」を参照してください。

6.6.3　RD エリアの作成

共有ディスクに作成した RD エリア用の HiRDB ファイルシステム領域に RD エリアを作成します。ユーザ用 RD エリアとシステム用 RD エリアをそれぞれ異なる共有ディスクの HiRDB ファイルシステム領域に作成するときの定義例を次に示すシステム構成例を基に説明します。

システム構成例



create rdarea 文の指定例

```
create rdarea PMAST for masterdirectory                ...1
    server name DIC
    file name "/dic0111/prd01"
      initial 10 segments;
create rdarea PDIR for datadirectory                   ...2
    server name DIC
    file name "/dic0112/prd02"
      initial 5 segments;
create rdarea PDIC for datadictionary                  ...3
    server name DIC
    file name "/dic0113/prd03"
      initial 20 segments;
create rdarea PUSR01 for user used by PUBLIC           ...4
    server name BACK01
    data model structured
    file name "/back0121/prd04"
      initial 500 segments;
create rdarea PUSR02 for user used by PUBLIC           ...5
    server name BACK02
    data model structured
    file name "/back0231/prd05"
      initial 500 segments;
```

［説明］

1. マスタディレクトリ用 RD エリア（PMAST）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
2. データディレクトリ用 RD エリア（PDIR）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
3. データディクショナリ用 RD エリア（PDIC）を共有ディスク A の HiRDB ファイルシステム領域に作成します。
4. ユーザ用 RD エリア（PUSR01）を共有ディスク B の HiRDB ファイルシステム領域に作成します。
5. ユーザ用 RD エリア（PUSR02）を共有ディスク C の HiRDB ファイルシステム領域に作成します。

6.6.4 グローバルバッファの定義

マニュアル「HiRDB システム導入・設計ガイド」の「グローバルバッファの割り当て方法」を参照して，作成した RD エリアにグローバルバッファを割り当ててください。

6.6.5 監査証跡ファイルの運用

監査証跡ファイルは HiRDB 管理者が共有ディスクに作成します。HiRDB 管理者，および監査人は，共有ディスク上の監査証跡ファイルを運用してください。

(1) 監査証跡ファイルの作成

監査証跡ファイルは，HiRDB 管理者が共有ディスクに作成します。

(2) 監査証跡ファイルの運用

系切り替えが発生した場合，HiRDB は共有ディスク上の監査証跡ファイルに監査事象を記録します。監査事象の記録に関する監査証跡ファイルの運用については，マニュアル「HiRDB システム運用ガイド」の「監査証跡ファイルの運用」を参照してください。

(3) 監査証跡の取得

系切り替えが発生した場合，監査証跡の取得状態の引き継ぎについては，切り替え元ユニットの停止状態に依存します。切り替え先の系が再開始の場合は系を切り替える前の状態を引き継ぎます。切り替え先の系が正常開始の場合は pd_audit オペランドの指定に従います。

(4) pdload コマンドの実行

HiRDB 管理者は，監査証跡ファイルを入力情報として，pdload コマンドを実行してください（認証は監査人）。ただし，障害などで系が切り替わった場合，HiRDB は切り替わる直前の監査事象を正しく取得しません。このため，pdload コマンドを実行しても切り替え直前のデータを取得できない場合があります。

6.7 系切り替え時のトランザクションエラーを少なくする設定（トランザクションキューイング機能）

トランザクションキューイング機能については，マニュアル「HiRDB システム運用ガイド」の「トランザクションキューイング機能（高速系切り替え機能限定）」を参照してください。

6.8　サーバプロセスの異常終了が多発した場合に系を切り替える設定

サーバプロセスの異常終了が多発した場合に系を切り替える設定については，マニュアル「HiRDB システム運用ガイド」の「サーバ障害以外の障害が発生した場合に系を切り替える方法」にある「サーバプロセスの異常終了が多発した場合に系を切り替える方法」を参照してください。

6.9 RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える設定

RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える設定方法については，マニュアル「HiRDB システム運用ガイド」の「サーバ障害以外の障害が発生した場合に系を切り替える方法」にある「RD エリアの入出力エラー（パス障害）が発生した場合に系を切り替える方法」を参照してください。

6.10 停止中のユニットがある状態でもシステムマネジャユニットの系切り替えを実行する設定

停止中のユニットがある状態でもシステムマネジャユニットの系切り替えを実行する場合の設定方法については，マニュアル「HiRDB システム運用ガイド」の「システムマネジャユニットの系切り替えに関する注意事項」を参照してください。

6.11　高速系切り替え機能を使用している場合の運用

ここでは，高速系切り替え機能を使用している場合の運用方法について説明します。

6.11.1　HiRDB の開始

(1)　HiRDB の開始方法

HiRDB の開始方法については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「HiRDB の開始方法」を参照してください。

pd_structured_directory_path オペランドに指定したディレクトリに SDB ディレクトリ情報ファイルがない場合に HiRDB を開始したときの処理を次の表に示します。この表に示すとおり実行系と待機系で処理が異なることがあるので注意してください。

表 6-1　SDB ディレクトリ情報ファイルがない場合に HiRDB を開始したときの処理

条件		HiRDB を開始したときの処理
対象の系	ユニット中に MGR，FES，または BES の有無	
実行系	あり	開始します。このとき，KFPB62910-W メッセージが出力されます。
	なし	開始します。
待機系	あり	開始処理を中止します。このとき，KFPB62911-E メッセージが出力されます。
	なし	開始します。

(2)　注意事項

HiRDB の開始方法については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「HiRDB の開始方法」を参照してください。

(3)　待機系ユニットの開始処理に時間が掛かる場合

待機系ユニットの開始処理に時間が掛かる場合は，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「待機系の起動に時間が掛かる場合の確認作業」を参照してください。

(4)　ユニットが実行サーバの起動待ち状態になったときの対処

ユニットの開始時に，ユニットが「実行サーバの起動待ち状態」になった場合，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「HA モニタ使用時の注意事項」を参照してください。

6.11.2　HiRDB の終了

HiRDB の終了については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「HiRDB の終了方法」を参照してください。

6.11.3　ユニットの状態を確認する場合

ユニットの状態を確認する方法については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「状態の確認」を参照してください。

6.11.4　統計解析情報の取得

統計解析情報の取得については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「統計ログファイルの運用」を参照してください。

6.11.5　計画系切り替えの手順

計画系切り替えの手順については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「計画系切り替え」を参照してください。

6.11.6　運用上の注意事項

(1)　HiRDB/SD 定義ユティリティ（pdsdbdef）を実行する場合の注意事項

- HiRDB の再起動を必要とする SDB データベースの定義追加，定義変更，または定義削除をする場合

  HA モニタの monsbystp コマンドですべての待機系ユニットを終了してから，HiRDB/SD 定義ユティリティを実行してください。手順については，「5.6　SDB データベースの定義追加，定義変更，または定義削除（HiRDB の再起動を必要とする場合）」を参照してください。

  待機系ユニットを終了しないで HiRDB/SD 定義ユティリティを実行した場合，HiRDB/SD 定義ユティリティの実行後に，すべての待機系ユニットを再起動してください。待機系ユニットを再起動しないと，次に示す場合に待機系ユニットが異常終了します。

- 実行系ユニットの単独再起動（正常終了後の再起動）を行い，そのあとに系切り替えが発生した場合

  待機系ユニットが異常終了した場合は，待機系ユニットを pdstart -u または pdstart -q コマンドで開始してください。

- **HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更をする場合**

  HiRDB/SD 定義ユティリティを実行して，新しい SDB ディレクトリ情報ファイルを待機系に配布したあとに待機系ユニットを再起動してください。手順については，「5.7　SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合）」を参照してください。

  待機系ユニットを再起動しないと，系切り替えが発生したときに待機系ユニットが異常終了します。待機系ユニットが異常終了した場合は，待機系ユニットを pdstart -u または pdstart -q コマンドで開始してください。

(2)　データベース構成変更ユティリティ（pdmod）を実行する場合の注意事項

データベース構成変更ユティリティを実行した場合※，すべての待機系ユニットを再起動してください。すべての待機系ユニットを再起動しないと，系切り替えが発生したときに待機系ユニットが異常終了します。待機系ユニットが異常終了した場合は，待機系ユニットを pdstart -u または pdstart -q コマンドで開始してください。

注※　データベース構成変更ユティリティの次に示す操作を除きます。

- HiRDB ファイルシステム領域の世代登録
- HiRDB ファイルシステム領域の世代削除

(3)　待機系ユニットを再起動する必要がある操作（重要）

次に示す操作をしたあとに，実行系ユニットの単独再起動（正常終了後の再起動）をした場合，すべての待機系ユニットを再起動してください。すべての待機系ユニットを再起動しないと，系切り替えが発生したときに待機系ユニットが異常終了します。待機系ユニットが異常終了した場合は，待機系ユニットを pdstart -u または pdstart -q コマンドで開始してください。

- 定義系 SQL の実行
- 監査証跡表の作成
- pddbchg コマンドの実行
- オンライン再編成の実行（pdorbegin，pdorend コマンドの実行）
- HiRDB システム定義の変更（次に示すオペランドの指定値を変更した場合）
  - グローバルバッファの定義（pdbuffer）
  - RD エリアの最大数（pd_max_rdarea_no）
  - RD エリアを構成する HiRDB ファイルの最大数（pd_max_file_no）
  - インナレプリカグループの最大数（pd_inner_replica_control）
  - 表予約数の最低保証値（pd_assurance_table_no）

- インデクス予約数の最低保証値（pd_assurance_index_no）

(4) コマンド実行時の注意事項

- HiRDB の停止中にコマンドを実行する場合，実行系と待機系の両方の HiRDB が終了している必要があります。ただし，pdstart コマンドを除きます。
- 待機系の HiRDB を開始するための pdstart コマンド以外は，HiRDB の稼働中に待機系の HiRDB でコマンドを実行しないでください。
- HiRDB の強制終了または異常終了後に pdsetup -d コマンドを実行する場合は，pdsetup -d コマンドの応答に Y を指定しないでください。Y を応答すると，そのあと HiRDB を開始できなくなることがあります。

(5) 系が切り替わったあとのホスト名の扱い

高速系切り替え機能では IP アドレスを引き継げないため，現用系と予備系のホスト名が異なります。系が切り替わったあとのホスト名の扱いについて説明します。

- **運用コマンドおよびユティリティに指定するホスト名について**
  - 系が切り替わったあとも，コマンドやユティリティに指定するホスト名またはユニット識別子を変える必要はありません。ホスト名には，システム共通定義の pdunit オペランドの-x オプションに指定した現用系ユニットのホスト名を常に指定します。
  - ユニット識別子を指定する場合は，ホスト名を意識する必要はありません。
  - コマンドの処理結果には現用系ユニットのホスト名が常に表示されます。
- **メッセージに表示されるホスト名について**

  メッセージには常に現用系ユニットのホスト名が表示されます。

(6) RD エリアのオープン契機について

高速系切り替え機能の対象になる待機系ユニットは，待機状態のときに RD エリアをオープンしていません。また，系の切り替え時間を最小限に抑えるため，系切り替えの発生時に全面回復で必要な RD エリアだけをオープンし，そのほかの RD エリアはオープンしません。したがって，待機系の RD エリアのオープン契機は INITIAL 属性になりません。INITIAL 属性の RD エリアは DEFER 属性になります。

RD エリアのオープン契機については，マニュアル「HiRDB システム運用ガイド」の「RD エリアのオープン契機を変更する方法（RD エリアの属性変更)」を参照してください。

(7) 共有ディスク上に作成した HiRDB ファイルにアクセスできない場合

HA モニタの制御によって HiRDB の停止中は，両方の系から共有ディスク上に作成した HiRDB ファイルを操作できなくなることがあります。この場合，OS のコマンドでディスクを活性化してください。

(8)　現用系と待機系の OS の時刻がずれているときの注意事項

現用系と待機系の OS の時刻がずれている（待機系の時刻の遅れが系切り替えに掛かる時間よりも大きい）場合，HiRDB は，時刻を遅らせた場合に行う対処を誤ったときと同じ影響を受けます。そのため，現用系と待機系の OS の時刻を合わせるようにしてください。

OS の時刻を変更する方法については，マニュアル「HiRDB システム運用ガイド」の「OS の時刻を変更する方法」を参照してください。

6.11.7　障害発生時のシステムの処理と HiRDB 管理者の処置

障害発生時のシステムの処理と HiRDB 管理者の処置については，マニュアル「HiRDB システム運用ガイド」の「スタンバイ型系切り替え（サーバモード）の運用」にある「障害発生時の HiRDB 管理者の処置」を参照してください。

7 セキュリティ監査機能の設定と運用

この章では，セキュリティ監査機能の設定方法と運用方法について説明します。

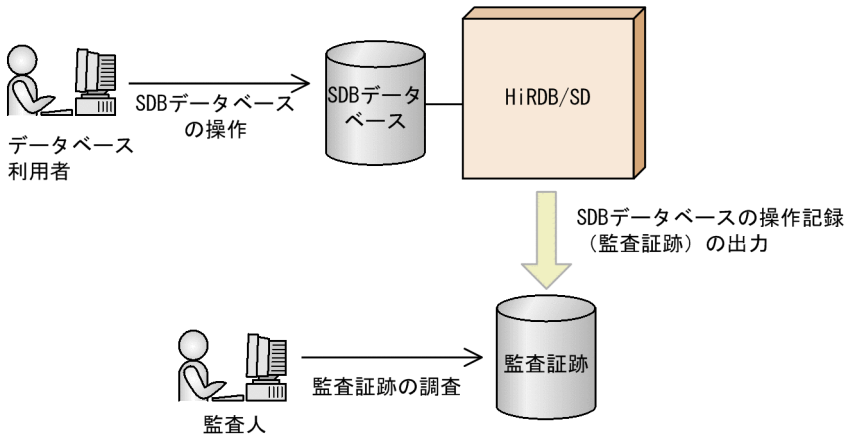
7.1　セキュリティ監査機能の概要

ここでは，セキュリティ監査機能の概要について説明します。

7.1.1　セキュリティ監査機能とは

HiRDB/SD のセキュリティは権限によって守られています。参照できる情報，更新できる情報，および操作できるオブジェクト（SDB データベースなど）を権限によって制限しています。この権限の運用が適切に行われているかどうかをチェックするために，HiRDB/SD では SDB データベースに対する特定の操作を記録できます。この機能をセキュリティ監査機能といい，出力される操作記録を監査証跡といいます。出力された監査証跡を調査して不正なアクセスが行われていないかをチェックできます。このチェックは監査権限を持つユーザ（これを監査人といいます）が行います。セキュリティ監査機能の概要を次の図に示します。

図 7-1　セキュリティ監査機能の概要



監査証跡には，誰がどのような権限を使用して何に対する操作を行ったかという情報が取得されます。どの操作に対して監査証跡を取得するかは，監査人が CREATE AUDIT 文で設定します。監査証跡の取得対象となる操作が実行されると，監査証跡が取得されます。

参考

セキュリティ監査機能はセキュリティを強化する機能ではありません。権限の運用が適切に行われているかどうかをチェックするための操作記録を出力する機能です。

7.1.2　監査証跡の取得範囲

監査証跡の取得範囲を次の表に示します。

表 7-1　監査証跡の取得範囲

項番	種別	コマンド名または機能名	監査証跡の取得可否
1	ユティリティ・コマンド	HiRDB/SD 定義ユティリティ（pdsdbdef コマンド）	○
2		HiRDB/SD データベース作成ユティリティ（pdsdblod コマンド）	○
3		HiRDB/SD データベース再編成ユティリティ（pdsdbrog コマンド）	○
4		HiRDB/SD データベースアクセスユティリティ（pdsdbexe コマンド）	○
5		pdsdbarc コマンド	×
6		pdsdborcrt コマンド	×
7	DB アクセスインタフェース	• レコードの検索（FETCH） • 位置指示子の位置づけ（FIND）※	○
8		レコードの格納（STORE）	○
9		レコードの更新（MODIFY）	○
10		レコードの削除（ERASE）	○
11		個別開始	×
12		個別終了	×
13		構成要素情報取得	×
14		複数レコードの検索（FETCHDB ALL）	○
15		レコードの一括削除	○
16		SDB データベース定義情報取得	×
17		レコードの取得（GET）	○
18	上記以外	マニュアル「HiRDB システム運用ガイド」の「監査対象になるイベント」で説明されているコマンドまたは機能	○

（凡例）

○：監査証跡を取得します。

×：監査証跡を取得しません。

注※

DML の位置指示子の位置づけ（FIND）は，レコードの検索（FETCH）として監査証跡が取得されます。

7.1.3　監査証跡の取得契機

監査証跡の取得契機については，マニュアル「HiRDB システム運用ガイド」の「監査証跡の取得契機」を参照してください。

また，「表 7-1　監査証跡の取得範囲」に示す操作をしたときにも監査証跡が取得されます。

(1)　ユティリティ・コマンド実行時の監査証跡の取得契機

(a)　pdsdbdef コマンド実行時の監査証跡の取得契機

pdsdbdef コマンドを実行した場合，次の表に示すタイミングで監査証跡が取得されます。

表 7-2　pdsdbdef コマンドを実行した場合の監査証跡の取得契機

項番	監査証跡の取得契機	監査対象のイベント	監査証跡の取得単位
1	権限チェック時	pdsdbdef コマンド実行時の CONNECT 権限チェック	pdsdbdef コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。
2	イベントの終了時	SDB ディクショナリ情報の操作	次の SDB 定義文が 1 回実行されると，監査証跡が 1 レコード取得されます。 • *ENTRY DICTIONARY 文 • *ALTER DICTIONARY 文 • *DELETE DICTIONARY 文
3		データベース定義のチェック	次の SDB 定義文が 1 回実行されると，監査証跡が 1 レコード取得されます。 • *CHECK DICTIONARY 文
4		SDB ディレクトリ情報の操作	次の SDB 定義文が 1 回実行されると，監査証跡が 1 レコード取得されます。 • *ENTRY DIRECTORY 文 • *ALTER DIRECTORY 文 • *DELETE DIRECTORY 文
5		SDB ディレクトリ情報ファイルの作成	dirinf 文を指定した pdsdbdef コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。
6		pdsdbdef コマンドの終了	pdsdbdef コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。

次に示す場合は，pdsdbdef コマンドに関する監査証跡は取得されません。

- pdsdbdef コマンド実行時の CONNECT 権限チェックで CONNECT エラーが発生した場合
- pdsdbdef コマンドのオプションまたは制御文の指定に誤りがあり，pdsdbdef コマンドがエラーとなった場合
- 上記の表の項番 2～5 については，該当するリソースにアクセスした場合に限り監査証跡が取得されます。該当するリソースにアクセスする前にエラーが発生した場合は，該当する監査対象イベントの監査証跡は取得されません。

(b)　pdsdblod コマンド実行時の監査証跡の取得契機

pdsdblod コマンドを実行した場合，次の表に示すタイミングで監査証跡が取得されます。

表 7-3　pdsdblod コマンドを実行した場合の監査証跡の取得契機

項番	監査証跡の取得契機	監査対象のイベント	監査証跡の取得単位
1	権限チェック時	pdsdblod コマンド実行時の CONNECT 権限チェック	pdsdblod コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。
2	イベントの終了時	pdsdblod コマンドの終了	

次に示す場合は，pdsdblod コマンドに関する監査証跡は取得されません。

- pdsdblod コマンド実行時の CONNECT 権限チェックで CONNECT エラーが発生した場合
- pdsdblod コマンドのオプションまたは制御文の指定に誤りがあり，pdsdblod コマンドがエラーとなった場合

(c)　pdsdbrog コマンド実行時の監査証跡の取得契機

pdsdbrog コマンドを実行した場合，次の表に示すタイミングで監査証跡が取得されます。

表 7-4　pdsdbrog コマンドを実行した場合の監査証跡の取得契機

項番	監査証跡の取得契機	監査対象のイベント	監査証跡の取得単位
1	権限チェック時	pdsdbrog コマンド実行時の CONNECT 権限チェック	pdsdbrog コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。
2	イベントの終了時	pdsdbrog コマンドの終了	

次に示す場合は，pdsdbrog コマンドに関する監査証跡は取得されません。

- pdsdbrog コマンド実行時の CONNECT 権限チェックで CONNECT エラーが発生した場合
- pdsdbrog コマンドのオプションまたは制御文の指定に誤りがあり，pdsdbrog コマンドがエラーとなった場合

(d)　pdsdbexe コマンド実行時の監査証跡の取得契機【4V FMB，4V AFM】

pdsdbexe コマンドを実行した場合，次の表に示すタイミングで監査証跡が取得されます。

表 7-5　pdsdbexe コマンドを実行した場合の監査証跡の取得契機

項番	監査証跡の取得契機	監査対象のイベント	監査証跡の取得単位
1	権限チェック時	pdsdbexe コマンド実行時の CONNECT 権限チェック	pdsdbexe コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。
2		pdsdbexe コマンドで実行する SQL コマンド	次の SQL コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。 • CONNECT コマンド
3	イベントの終了時※	pdsdbexe コマンド実行時の CONNECT 操作	pdsdbexe コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。

項番	監査証跡の取得契機	監査対象のイベント	監査証跡の取得単位
4		pdsdbexe コマンド終了時の DISCONNECT 操作	DISCONNECT コマンドを実行しないで pdsdbexe コマンドを終了すると，監査証跡が 1 レコード取得されます。
5		pdsdbexe コマンドで実行する SQL コマンド	次の SQL コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。 • CONNECT コマンド • DISCONNECT コマンド
6		pdsdbexe コマンドで実行する DML コマンド	次の DML コマンドが 1 回実行されると，監査証跡が 1 レコード取得されます。 • FETCH コマンド • STORE コマンド • MODIFY コマンド • ERASE コマンド

注※

pdsdbexe コマンド終了時の監査証跡は取得しません。

次に示す場合は，pdsdbexe コマンドに関する監査証跡は取得されません。

- pdsdbexe コマンドのオプションまたは制御文の指定に誤りがあり，pdsdbexe コマンドがエラーとなった場合

(2) データベース操作時の監査証跡の取得契機

HiRDB/SD ではデータベース操作に対する権限を持たないため，実行時には権限チェックは行いません。権限チェックの監査証跡は，CONNECT 要求時に取得されます。データベース操作に対する監査証跡は，イベント終了時だけ取得されます。

なお，次の場合には，監査証跡を取得しないことがあります。

- SDB データベースを操作する API または DML の入力情報不正や，要求順序不正など，データベースアクセス実行前にエラーとなった場合
- FES と BES 間の通信でエラーが発生した場合

データベース操作イベント種別とその監査イベントが発生する操作を次の表に示します。

表 7-6 SDB データベース操作イベント種別と監査イベントが発生する操作

SDB データベース操作イベント種別	権限チェックの監査イベントが発生する操作（AUDITTYPE に PRIVILEGE を指定した場合）	イベントの最終結果の監査証跡を取得する監査イベントが発生する操作（AUDITTYPE に EVENT を指定した場合）
FETCH	権限チェックイベントなし	• レコードの検索（FETCH）実行 • 位置指示子の位置づけ（FIND）実行
STORE	同上	レコードの格納（STORE）実行

SDB データベース操作イベント種別	権限チェックの監査イベントが発生する操作（AUDITTYPE に PRIVILEGE を指定した場合）	イベントの最終結果の監査証跡を取得する監査イベントが発生する操作（AUDITTYPE に EVENT を指定した場合）
MODIFY	同上	レコードの更新（MODIFY）実行
ERASE	同上	レコードの削除（ERASE）実行
CLEAR	同上	レコードの一括削除実行
FETCHDB ALL	同上	複数レコードの検索（FETCHDB ALL）実行
GET	同上	レコードの取得（GET）実行

(3) データベース操作のユーザ要求と監査証跡レコードの関係

HiRDB/SD では，UAP および pdsdbexe コマンドからの SDB データベースを操作する API または DML を，1 つのイベントとして（API または DML に対して 1 対 1 で）監査証跡を取得します。

UAP からの SDB データベースを操作する API をブロック化した複数要求の場合，次のように取得されます。

- ブロック化した個々の API に対して実行したものだけを対象に取得します。
- 監査証跡レコードの「イベント成否」および「SQL コード」項目には複数要求としての最終結果を記録します。
- 監査証跡レコードの「アクセス件数」項目には，実行した要求で操作したレコードの合計値を記録します。
- 上記のため，個々の監査証跡レコードの「SQL データ」項目にレコード操作出力情報として，API 単位の処理結果およびアクセス件数を記録します。レコード削除では要求順序によって前の要求で削除されたレコードに対する要求となることがあり，この場合はレコード操作出力情報のアクセス件数は 0 件として記録します。

また，すべてのレコード操作で，次のレコードの監査証跡は取得しません。

- 4V AFM の SDB データベースの仮想ルートレコード
- 4V FMB または SD FMB の SDB データベースの場合で，親レコード削除に伴って削除される配下の子レコード群
- 4V FMB の SDB データベースの場合で，一括オプション('O')指定で削除する位置づけレコード以外の子レコード群

(4) データベース操作のレコードの検索（FETCH）の取得オプション

CREATE AUDIT 文（監査対象イベントの定義）で SDB データベース操作イベント種別の FETCH に FIRST オプションを指定することで，レコードの検索（FETCH）の監査証跡取得量を削減できます。

このオプションは，4V FMB または SD FMB の SDB データベースを対象にした監査証跡取得オプションのため，4V AFM の SDB データベースに対して指定しても無視されます。FETCH はすべてのレコード検索をイベントの対象としますが，FIRST オプションを指定した場合は，各レコード型に対して検索のイベントが成功した最初のレコードに対してだけ監査証跡を取得し，以降の検索イベントで成功した同じレコード型の監査証跡は取得しません。ただし，同じレコード型であっても，レコード検索のイベントが失敗した場合は，監査証跡が取得されます。

この機能は，個別開始ごとにファミリ単位にルートレコードを含め，その配下の子レコードに対して適用されるため，同じレコード型へのアクセスでもルートレコードのキーが変われば（ルートレコードへの位置づけが発生すれば）イベントの対象として，監査証跡が取得されます。

また，レコード検索処理の途中で，このオプションのイベント定義が有効になった場合は，そのタイミングで同一レコード型に対する 2 回目以降の検索レコードに対する監査証跡は取得されません。

7.1.4　監査証跡として取得する情報

監査証跡として取得する情報については，次の項目を参照してください。

- 「7.2　監査証跡ファイルに出力される情報」
- 「7.8　監査証跡表の列構成」
- 「7.12　監査証跡のレコード項目」

7.1.5　監査証跡の参照

監査証跡は監査証跡ファイルに出力されます。監査証跡ファイル中のデータは，監査証跡表に格納してから参照します。詳細については，「7.5.1　監査証跡の参照方法」を参照してください。

7.1.6　前提となるシステム構成

セキュリティ監査機能を使用する場合，システム構成に関して次に示す前提条件があります。

- サーバがないユニット（システムマネジャだけのユニットを含む）がある場合，セキュリティ監査機能を使用できません。セキュリティ監査機能を使用する場合は，各ユニットにフロントエンドサーバ，ディクショナリサーバ，またはバックエンドサーバのどれかを配置してください。

7.1.7　監査対象になるイベント

監査証跡の取得対象になる操作を監査対象イベントといいます。監査対象イベントについては，マニュアル「HiRDB システム運用ガイド」の「監査対象になるイベント」を参照してください。

さらに，次の表に示す監査対象イベントがあります。

表 7-7　監査対象イベント

イベントの種類	説明および監査対象になるイベント	選択可否
SDB ユティリティ操作イベント	次のユティリティによるオブジェクト操作に関するセキュリティ対象イベントを監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。※ • HiRDB/SD 定義ユティリティ（pdsdbdef コマンド） • HiRDB/SD データベース作成ユティリティ（pdsdblod コマンド） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog コマンド）	選択できます。
SDB データベース操作イベント	次のデータベース操作に関するセキュリティ対象イベントを監査対象にします。次に示すイベントが実行されたときに監査証跡を出力します。 • レコードの検索（FETCH）または位置指示子の位置づけ（FIND） • レコードの格納（STORE） • レコードの更新（MODIFY） • レコードの削除（ERASE） • レコードの一括削除 • 複数レコードの検索（FETCHDB ALL） • レコードの取得（GET）	選択できます。

注※

HiRDB/SD データベースアクセスユティリティ（pdsdbexe コマンド）は，SDB ユティリティ操作イベントとして監査証跡を出力しません。詳細は，「7.1.3(1)(d)　pdsdbexe コマンド実行時の監査証跡の取得契機【4V FMB，4V AFM】」を参照ください。

7.1.8　監査証跡表の自動データロード機能

監査証跡表の自動データロード機能を使用すると，監査証跡ファイル中のデータを監査証跡表に自動的に格納することができます。詳細については，マニュアル「HiRDB システム運用ガイド」の「監査証跡表の自動データロード機能」を参照してください。

7.2　監査証跡ファイルに出力される情報

監査証跡ファイルに出力される情報については，マニュアル「HiRDB システム運用ガイド」の「監査証跡ファイルに出力される情報」を参照してください。

7.3　監査証跡の出力パターン

監査証跡の出力パターンについては，マニュアル「HiRDB システム運用ガイド」の「監査証跡の出力パターン」を参照してください。

ここでは，SDB データベース操作イベントの監査証跡の出力パターン（出力契機の例）について説明します。

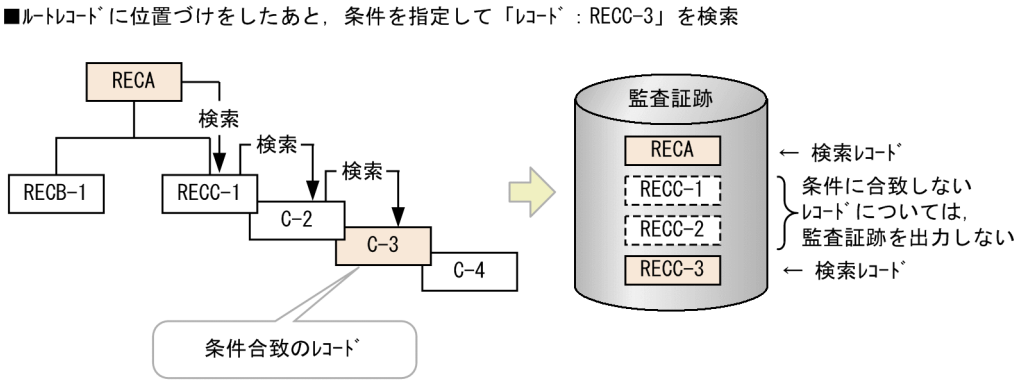
7.3.1　レコードを検索した場合（FETCH）

(1)　4V FMB または SD FMB の SDB データベースの場合

ユーザに返却されたレコード（FIND の場合は位置指示子を位置づけたレコード）の監査証跡だけが出力されます。

4V FMB の SDB データベースのレコードを検索した場合の監査証跡の出力例を次の図に示します。

図 7-2　監査証跡の出力例（4V FMB の SDB データベースのレコードを検索した場合）



■RECC-3の検索時に取得される監査証跡の情報

・イベントタイプ	：SDBアクセス（SDA）
・イベントサブタイプ	：検索（FTC）
・オブジェクト	：レコード名称（RECC）
・要求情報	：レコード操作情報（指定値）
・対象データ	：レコード操作キー情報（一連番号＝3）
・アクセス件数	：1（RECC-3の件数）

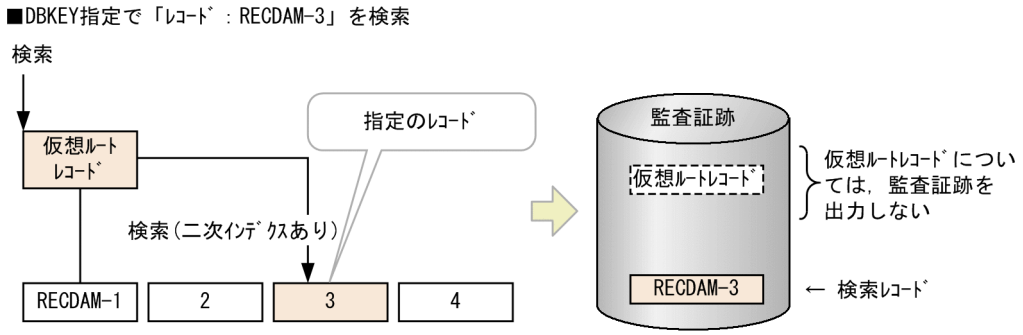
(2)　4V AFM の SDB データベースの場合

次のように監査証跡が出力されます。

- ユーザに返却されたレコードの監査証跡だけが出力されます。
- 仮想ルートレコードの監査証跡は出力されません。

監査証跡の出力例を次の図に示します。

図 7-3　監査証跡の出力例（4V AFM の SDB データベースのレコードを検索した場合）



■RECDAM-3の検索時に取得される監査証跡の情報

・イベントタイプ	：SDBアクセス（SDA）
・イベントサブタイプ	：検索（FTC）
・オブジェクト	：レコード名称（RECDAM）※
・要求情報	：レコード操作情報（指定値）
・対象データ	：レコード操作キー情報（一連番号＝3）
・アクセス件数	：1（RECDAM-3の件数）

注※　ユーザ要求がデータベース名称指定の場合は，データベース名称を設定します。

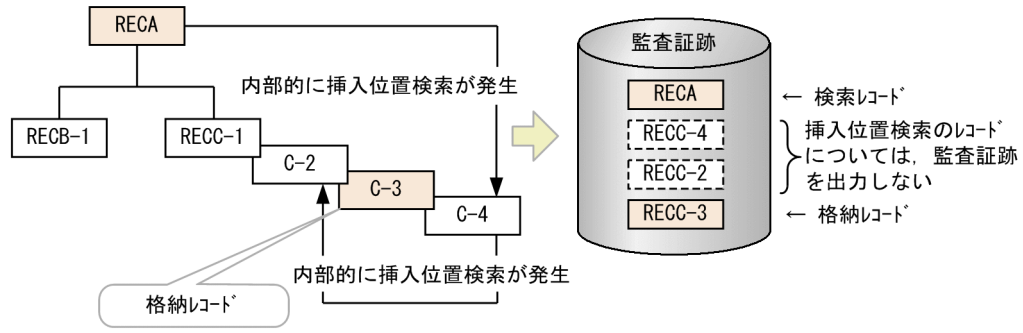
7.3.2　レコードを格納した場合（STORE）

(1)　4V FMB または SD FMB の SDB データベースの場合

格納したレコードの監査証跡だけが出力されます。

4V FMB の SDB データベースのレコードを格納した場合の監査証跡の出力例を次の図に示します。

図 7-4　監査証跡の出力例（4V FMB の SDB データベースのレコードを格納した場合）



■RECC-3の格納時に取得される監査証跡の情報

・イベントタイプ	：SDBアクセス（SDA）
・イベントサブタイプ	：格納（STR）
・オブジェクト	：レコード名称（RECC）
・要求情報	：レコード操作情報（指定値）
・対象データ	：レコード操作キー情報（ユーザキー＝3）
・アクセス件数	：1（RECC-3の件数）

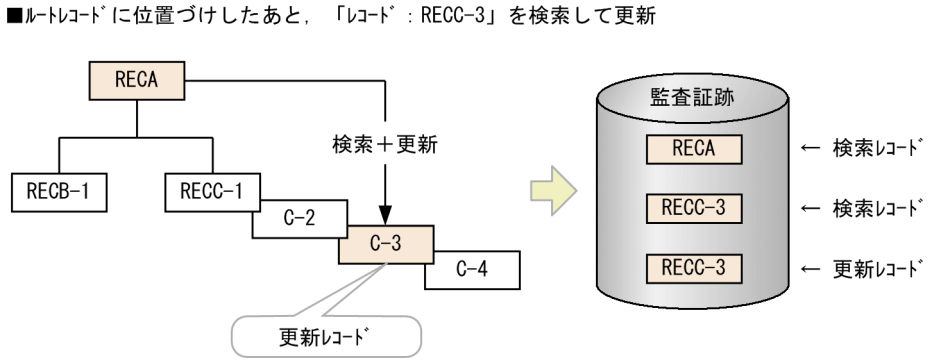
7.3.3　レコードを更新した場合（MODIFY）

(1)　4V FMB または SD FMB の SDB データベースの場合

更新したレコードの監査証跡だけが出力されます。

4V FMB の SDB データベースのレコードを更新した場合の監査証跡の出力例を次の図に示します。

図 7-5　監査証跡の出力例（4V FMB の SDB データベースのレコードを更新した場合）



■RECC-3の更新時に取得される監査証跡の情報

・イベントタイプ	：SDBアクセス（SDA）
・イベントサブタイプ	：更新（MOD）
・オブジェクト	：レコード名称（RECC）
・要求情報	：レコード操作情報（指定値）
・対象データ	：レコード操作キー情報（一連番号＝3）
・アクセス件数	：1（RECC-3の件数）

7.3.4　レコードを削除した場合（ERASE）

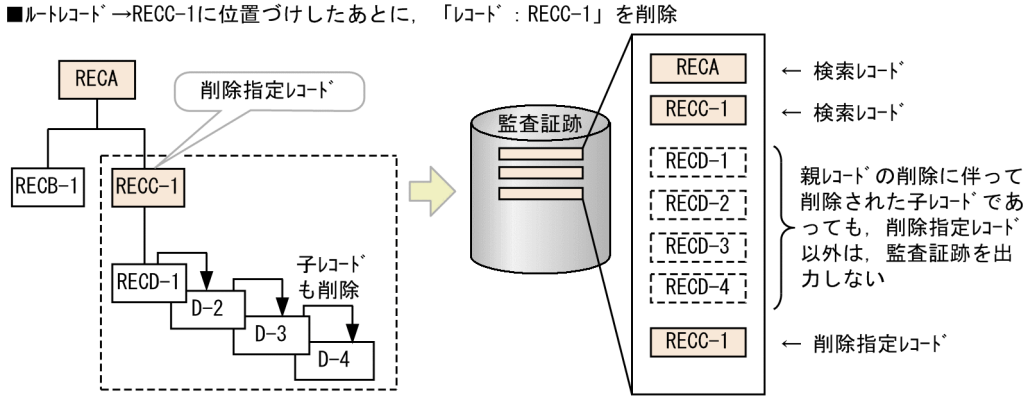
(1)　4V FMB または SD FMB の SDB データベースの親レコードを削除した場合

次のように監査証跡が出力されます。

- 削除のために位置づけたレコード（削除指定レコード）が監査証跡に出力されます。
- 親レコードの削除に伴って削除される配下の子レコード群の監査証跡は出力されません。
- 出力項目の「アクセス件数」には，削除される配下の子レコードの件数が含まれます。

4V FMB または SD FMB の SDB データベースのレコードを削除した場合の監査証跡の出力例を次の図に示します。

図 7-6　監査証跡の出力例（4V FMB または SD FMB の SDB データベースのレコードを削除した場合）



■RECC-1の削除時に取得される監査証跡の情報

・イベントタイプ	：SDBアクセス（SDA）
・イベントサブタイプ	：削除（ERS）
・オブジェクト	：レコード名称（RECC）
・要求情報	：レコード操作情報（シングル：S指定）
・対象データ	：レコード操作キー情報（一連番号＝1）
・アクセス件数	：5（RECC-1とRECD-1～4の件数）

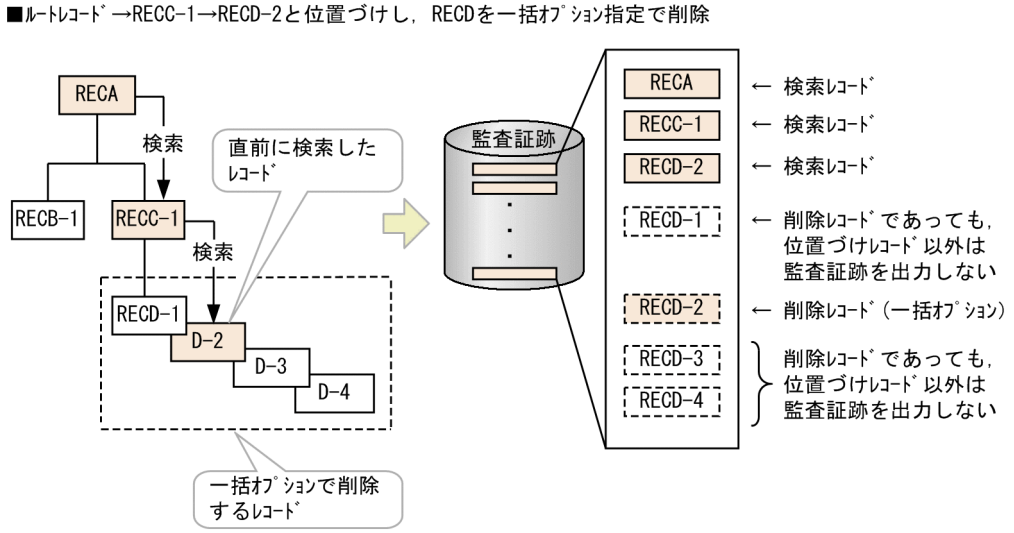
(2)　一括オプション('O')を指定して 4V FMB の SDB データベースのレコードを削除した場合【4V FMB】

次のように監査証跡が出力されます。

- 削除のために直前に位置づけたレコードの監査証跡だけが出力されます。
- 直前に位置づけたレコード以外の削除レコード群の監査証跡は出力されません。
- 出力項目の「アクセス件数」には，削除したすべてのレコード件数が設定されます。

一括オプション('O')を指定して，4V FMB の SDB データベースのレコードを削除した場合の監査証跡の出力例を次の図に示します。

図 7-7　監査証跡の出力例（一括オプション('O')を指定して 4V FMB の SDB データベースのレコードを削除した場合）



■RECDの削除時に取得される監査証跡の情報

・イベントタイプ	：SDBアクセス（SDA）
・イベントサブタイプ	：削除（ERS）
・オブジェクト	：レコード名称（RECD）
・要求情報	：レコード操作情報（一括：0指定）
・対象データ	：レコード操作キー情報（一連番号＝2）※
・アクセス件数	：4（RECD-1～4の件数）

注※　直前に位置づけたﾚｺｰﾄﾞのｷｰを設定します。

7.3.5　一括削除をした場合【4V DAM，4V SAM】

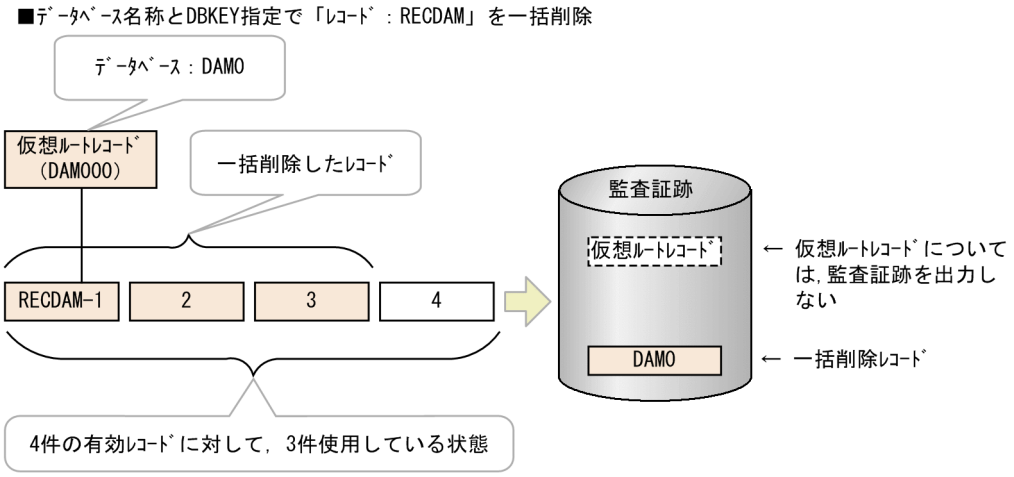
(1)　4V DAM の SDB データベースの場合

次のように監査証跡が出力されます。

- 仮想ルートレコードの管理情報を操作するが，仮想ルートレコードの監査証跡は出力されません。
- ユーザ要求に従って，一括削除対象のデータベースとして監査証跡が出力されます。

監査証跡の出力例を次の図に示します。

図 7-8　監査証跡の出力例（一括削除をした場合）



■一括削除時に取得される監査証跡の情報

・イベントタイプ　　：SDBアクセス（SDA）
・イベントサブタイプ：一括削除（CLR）
・オブジェクト　　：データベース名称（DAMO）
・要求情報　　：レコード操作情報（指定値）
・対象データ　　：レコード操作キー情報（一括削除対象レコードのキー値
　ただし，一連番号は0を設定）※
・アクセス件数：3(RECDAM-1～3の件数)

注※　定義によって，キーの指定なしもできます。

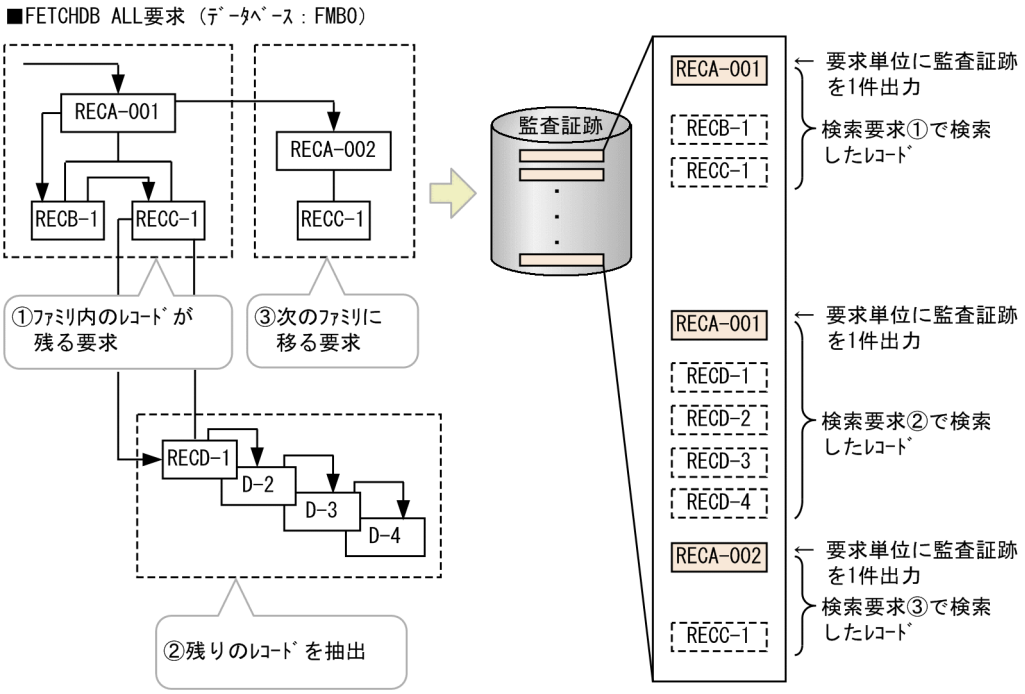
7.3.6　複数レコードを検索した場合（FETCHDB ALL）【4V FMB】

次のように監査証跡が出力されます。

- ルートレコードを含むかどうかに関係なく，各要求に対してルートレコードの監査証跡として出力されます。
- 監査証跡には，ルートレコードのキーと検索したレコード件数が出力されます。

監査証跡の出力例を次の図に示します。

図 7-9　監査証跡の出力例（複数レコードを検索した場合）



■FETCHDB ALL要求時に取得される監査証跡の情報

■①の要求
・イベントタイプ　：SDBアクセス（SDA）
・イベントサブタイプ　：複数レコードの検索（FTA）
・オブジェクト　：データベース名称（FMBO）
・要求情報　：レコード操作情報（指定値）
・対象データ　：レコード操作キー情報（ルートレコードのデータベースキー＝001）
・アクセス件数　：3（RECA-001，RECB-1，RECC-1の件数）

■②の要求
・イベントタイプ　：SDBアクセス（SDA）
・イベントサブタイプ　：複数レコードの検索（FTA）
・オブジェクト　：データベース名称（FMBO）
・要求情報　：レコード操作情報（指定値）
・対象データ　：レコード操作キー情報（ルートレコードのデータベースキー＝001）
・アクセス件数　：4（RECD-1～4の件数）

■③の要求
・イベントタイプ　：SDBアクセス（SDA）
・イベントサブタイプ　：複数レコードの検索（FTA）
・オブジェクト　：データベース名称（FMBO）
・要求情報　：レコード操作情報（指定値）
・対象データ　：レコード操作キー情報（ルートレコードのデータベースキー＝002）
・アクセス件数　：2（RECA-002，RECC-1の件数）

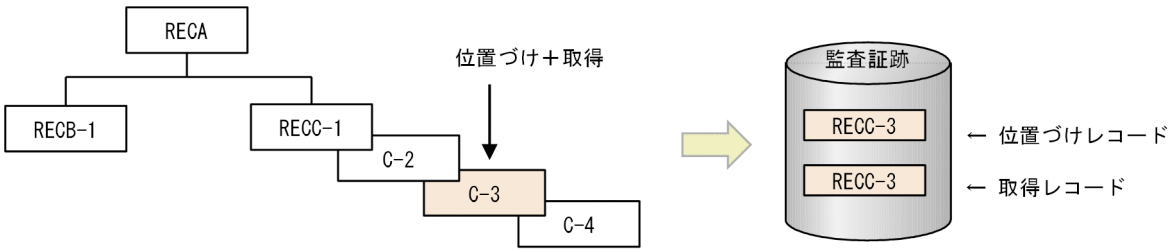
7.3.7　レコードを取得した場合【SD FMB】

取得したレコードの監査証跡だけが出力されます。

SD FMB の SDB データベースのレコードを取得した場合の監査証跡の出力例を次の図に示します。

図 7-10　監査証跡の出力例（SD FMB の SDB データベースのレコードを取得した場合）

■RECC-3に位置づけしたあとに「レコード：RECC-3」を取得



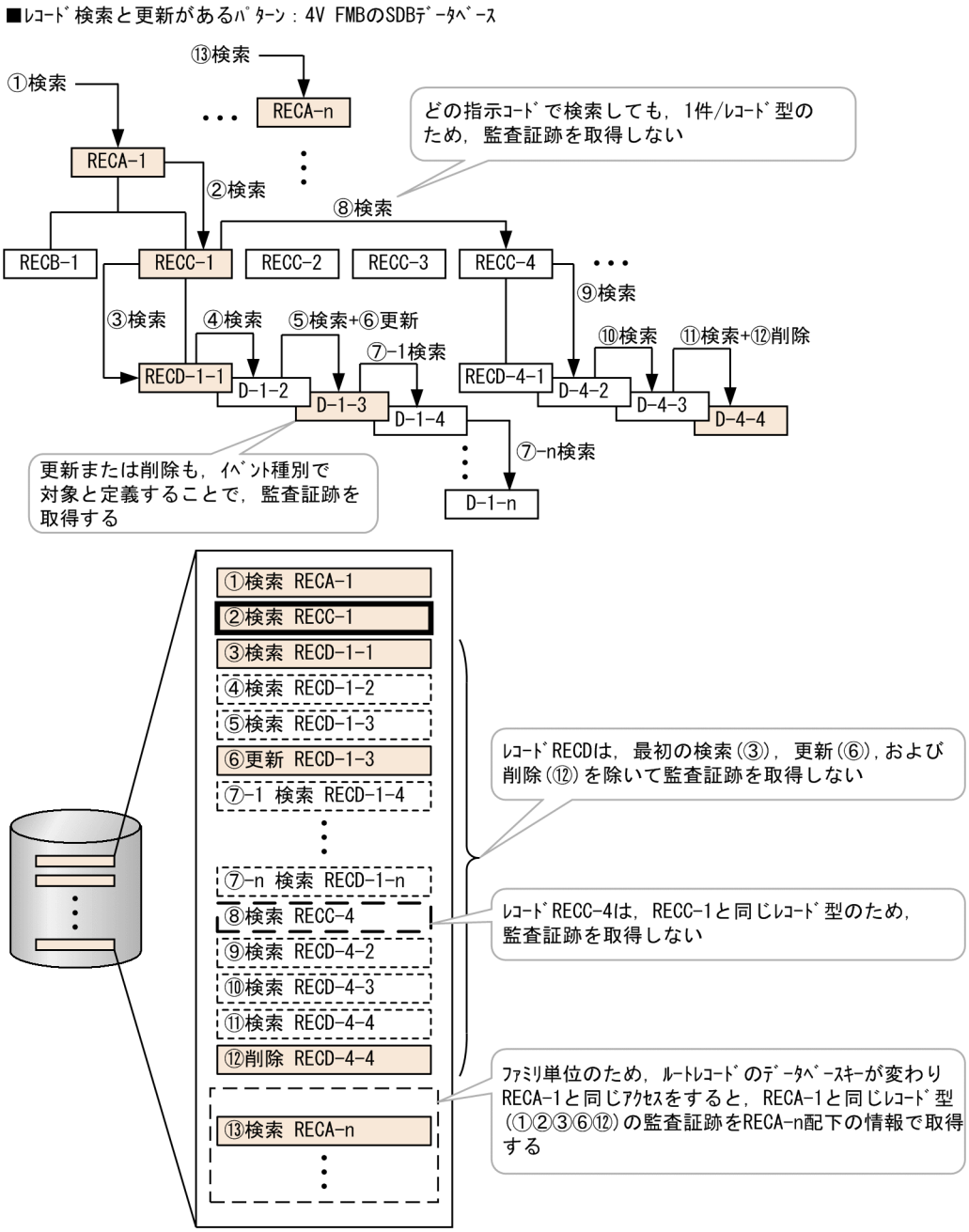
■RECC-3の取得時に取得される監査証跡の情報

・イベントタイプ	：SDBアクセス（SDA）
・イベントサブタイプ	：取得（GET）
・オブジェクト	：レコード名称（RECC）
・要求情報	：レコード操作情報（指定値）
・対象データ	：レコード操作キー情報（一連番号＝3）
・アクセス件数	：1（RECC-3の件数）

7.3.8　FIRST オプション機能適用時のデータベース操作と監査証跡レコードの関係

SDB データベースの操作時に取得される監査証跡の関連を次の図に示します。

図 7-11　SDB データベースの操作時に取得される監査証跡の関連



注　図中の丸付き数字は，アクセス順序を示しています。

7.4　環境設定方法

環境設定方法については，マニュアル「HiRDB システム運用ガイド」の「セキュリティ監査機能の運用」の「環境設定方法」を参照してください。

環境設定をする際，ここで説明する点に注意してください。

7.4.1　監査証跡表を格納する RD エリアの作成

pdmod コマンドの create rdarea 文で，監査証跡表を格納する RD エリアを作成します。このとき，create rdarea 文の data model オペランドに relational を指定するか，または data model オペランドを省略してください。

7.4.2　監査対象イベントの定義

定義系 SQL の CREATE AUDIT 文で監査対象イベントを定義します。CREATE AUDIT 文の指定形式を次に示します。

(1)　指定形式

項番	指定形式	指定要否
1	CREATE AUDIT 〔AUDITTYPE　{PRIVILEGE ｜ EVENT ｜ ANY}〕	○
2	FOR 操作種別	○
3	〔選択オプション〕	○
4	〔WHENEVER　{SUCCESSFUL ｜ UNSUCCESSFUL ｜ ANY}〕	○

■各項目の詳細

項番	指定形式	指定要否
5	操作種別：：＝ {ANY	○
	｜ SESSION〔{セッション種別｜ ANY}〕	×
	｜ PRIVILEGE〔{権限操作種別｜ ANY}〕	×
	｜ DEFINITION〔{オブジェクト定義イベント種別｜ ANY}〕	×
	｜ ACCESS〔{オブジェクト操作イベント種別｜ ANY}〕	×
	｜ UTILITY〔{ユティリティイベント種別｜ ANY}〕	×
	｜ SDB_ACCESS〔{SDB データベース操作イベント種別｜ ANY}〕	○

項番	指定形式	指定要否
	｜SDB_UTILITY〔{SDB ユティリティ操作イベント種別｜ ANY}〕}	○
6	選択オプション：：＝ {ON オブジェクト名 ｜BY AUTHORIZATION 認可識別子}	○※
	オブジェクト名：：＝ {FUNCTION　認可識別子.ルーチン識別子 ｜INDEX　認可識別子.インデクス識別子 ｜LIST　認可識別子.表識別子 ｜PROCEDURE　認可識別子.ルーチン識別子 ｜RDAREA　RD エリア名 ｜SCHEMA　認可識別子 ｜TABLE 〔認可識別子.〕表識別子 ｜TRIGGER　認可識別子.トリガ識別子 ｜TYPE　認可識別子.データ型識別子 ｜VIEW　認可識別子.表識別子 ｜SEQUENCE　認可識別子.順序数生成子識別子}	×
7	セッション種別：：＝ {CONNECT ｜ DISCONNECT ｜ AUTHORIZATION}	×
8	権限操作種別：：＝ {GRANT ｜ REVOKE}	×
9	オブジェクト定義イベント種別：：＝ {CREATE ｜ DROP ｜ ALTER}	×
10	オブジェクト操作イベント種別：：＝ {SELECT ｜ INSERT ｜ UPDATE ｜ DELETE ｜ PURGE ｜ ASSIGN ｜ CALL ｜ LOCK ｜ NEXT VALUE}	×
11	ユティリティイベント種別：：＝ {PDLOAD ｜ PDRORG ｜ PDEXP ｜ PDCONSTCK}	×
12	SDB データベース操作イベント種別：：＝ {FETCH 〔FETCH オプション〕 ｜ STORE ｜ MODIFY ｜ ERASE ｜ CLEAR ｜ FETCHDB ALL ｜ GET}	○
13	FETCH オプション：：＝ FIRST	○
14	SDB ユティリティ操作イベント種別：：＝ {PDSDBDEF ｜ PDSDBLOD ｜ PDSDBROG}	○

（凡例）

○：指定必須のオペランドまたは指定を検討するオペランドです。

×：関係のないオペランドのため，指定不要です。

注※

ON オブジェクト名は指定できません。操作種別に SDB_ACCESS または SDB_UTILITY を指定して ON オブジェクト名を指定した場合，KFPA19680-E メッセージが出力されてエラーとなります。

(2) オペランドの説明

ここでは，指定形式の項番 5，6，12～14 のオペランドについてだけ説明します。そのほかのオペランドについては，マニュアル「HiRDB SQL リファレンス」の「CREATE AUDIT（監査対象イベントの定義)」を参照してください。

5）操作種別：：＝ {ANY

｜ SDB_ACCESS〔{SDB データベース操作イベント種別｜ ANY}〕

｜ SDB_UTILITY〔{SDB ユティリティ操作イベント種別｜ ANY}〕}

監査対象とする操作種別を指定します。個々の操作種別と ANY は，個別に定義，および削除をします。例えば，SDB_UTILITY を定義した状態で，ANY だけを DROP AUDIT 文で削除しても，SDB_UTILITY の定義は残ります（監査対象のままとなります)。

ANY

すべての操作種別を監査対象とします。ただし，次のオプションは無効になります。

- SDB データベース操作イベント種別の FETCH オプション

SDB_ ACCESS〔{SDB データベース操作イベント種別｜ ANY}〕

SDB データベース操作イベントを監査対象に定義する場合に指定します。ANY は，すべての SDB データベース操作イベントを監査対象に定義する場合に指定します。ただし，ANY を指定した場合，FETCH オプションは無効になります。

個々の SDB データベース操作イベント種別と ANY は，個別に定義，および削除をします。例えば，FETCH および ANY を定義した状態で，ANY だけを DROP AUDIT 文で削除しても，FETCH の定義は残ります（監査対象のままとなります)。

SDB_UTILITY〔{SDB ユティリティ操作イベント種別｜ ANY}〕

SDB ユティリティ操作イベントを監査対象に定義する場合に指定します。ANY はすべての SDB ユティリティ操作イベントを監査対象とします。個々の SDB ユティリティ操作イベント種別と ANY は，個別に定義，および削除をします。例えば，PDSDBLOD，PDSDBROG，および ANY を定義した状態で，ANY だけを DROP AUDIT 文で削除しても，PDSDBLOD および PDSDBROG の定義は残ります（監査対象のままとなります)。

6）選択オプション：：＝ {ON オブジェクト名｜ BY AUTHORIZATION 認可識別子}

ON オブジェクト名

このオプションは指定できません。操作種別に SDB_ACCESS または SDB_UTILITY を指定して ON オブジェクト名を指定した場合，KFPA19680-E メッセージが出力されてエラーとなります。

BY AUTHORIZATION 認可識別子

監査証跡を取得するイベント実行者を絞り込む場合に指定します。監査対象とするイベント実行者の認可識別子を指定してください。指定しない場合は，すべてのイベント実行者が対象となります。

SDB ユティリティ操作イベントを指定している場合は，指定できません。指定した場合，KFPA19680-E メッセージが出力されてエラーとなります。

12）SDB データベース操作イベント種別：：＝ {FETCH 〔FETCH オプション〕
| STORE | MODIFY | ERASE | CLEAR | FETCHDB ALL | GET}

SDB データベース操作イベントを監査対象として定義する場合に指定します。SDB データベース操作イベント種別とその監査イベントが発生する操作を次の表に示します。

表 7-8　SDB データベース操作イベント種別とその監査イベントが発生する操作

SDB データベース操作イベント種別	権限チェックの監査イベントが発生する操作（AUDITTYPE に PRIVILEGE を指定した場合）	イベントの最終結果の監査証跡を取得する監査イベントが発生する操作（AUDITTYPE に EVENT を指定した場合）
FETCH	なし	• レコードの検索（FETCH）の実行 • 位置指示子の位置づけ（FIND）の実行
STORE		レコードの格納（STORE）の実行
MODIFY		レコードの更新（MODIFY）の実行
ERASE		レコードの削除（ERASE）の実行
CLEAR		レコードの一括削除の実行
FETCHDB ALL		複数レコードの検索（FETCHDB ALL）の実行
GET		レコードの取得（GET）の実行

13）FETCH オプション：：＝ FIRST

SDB データベース操作イベント種別の FETCH オプションを指定します。

FIRST

4V FMB または SD FMB の SDB データベースを対象にした監査証跡取得オプションです。

FETCH はすべてのレコード検索をイベントの対象としますが，FIRST を指定した場合，各レコード型に対して検索のイベントが成功した最初のレコードの監査証跡だけを取得し，以降の検索イベントで成功した同じレコード型の監査証跡は取得しません。ただし，同じレコード型であっても，レコード検索のイベントが失敗した場合は監査証跡を取得します。このオプションは，個別開始ごとにファミリ単位にルートレコードを含め，その配下の子レコードに対して適用されます。そのため，同じレコード型へのアクセスであっても，ルートレコードのキーが変わった場合（ルートレコードへの位置づけが発生した場合），イベントの対象として監査証跡が取得されます。

また，レコード検索処理の途中でこのオプションのイベント定義が有効になった場合，そのタイミングで同一レコード型に対する 2 回目以降の検索レコードは取得されません。

FIRST は，4V AFM の SDB データベースで指定しても無効になります。

FIRST を指定する場合は，WHENEVER の指定を ANY にする必要があります。操作種別で ANY または SDB_ACCESS で ANY を指定した場合，このオプションの指定は無効になります。また，FETCH と FETCH FIRST の定義が同時に存在した場合は，FETCH の定義として動作します。このため，FETCH FIRST の機能でイベントを取得するためには，操作種別で ANY または SDB_ACCESS で ANY を指定しないで，操作種別のイベントを個々に定義し，かつイベント種別の FETCH は FETCH FIRST だけを定義する必要があります。

14）SDB ユティリティ操作イベント種別：：＝ {PDSDBDEF | PDSDBLOD | PDSDBROG}

SDB ユティリティ操作イベントを監査対象として定義します。SDB ユティリティ操作イベント種別とその監査イベントが発生する操作を次の表に示します。

表 7-9　SDB ユティリティ操作イベント種別とその監査イベントが発生する操作

SDB ユティリティ操作イベント種別	権限チェックの監査イベントが発生する操作（AUDITTYPE に PRIVILEGE を指定した場合）	イベントの最終結果の監査証跡を取得する監査イベントが発生する操作（AUDITTYPE に EVENT を指定した場合）
PDSDBDEF	pdsdbdef コマンドの実行	同左
PDSDBLOD	pdsdblod コマンドの実行	
PDSDBROG	pdsdbrog コマンドの実行	

(3)　SDB データベース操作イベントのオペランドの指定と監査対象イベントの関係

SDB データベース操作のイベント定義では，操作種別とそのほかのオペランドとの組み合わせによっては，監査証跡が出力されないことがあります。そのような定義の指定をした場合は，KFPA19680-E メッセージが出力されます。

イベントタイプ，イベントサブタイプの指定と AUDITTYPE の指定可否を次の表に示します。

表 7-10　イベントタイプ，イベントサブタイプの指定と AUDITTYPE の指定可否

イベントタイプ（操作種別）	イベントサブタイプ（SDB データベース操作イベント種別）	AUDITTYPE の指定可否		
		PRIVILEGE	EVENT	ANY
ANY	該当しません。	△	○	△
SDB_ACCESS	FETCH	×	○	△
	STORE	×	○	△
	MODIFY	×	○	△
	ERASE	×	○	△
	CLEAR	×	○	△
	FETCHDB ALL	×	○	△
	GET	×	○	△
	ANY	×	○	△

（凡例）

○：指定できます。

△：指定できるが，SDB_ACCESS に関する権限チェックの監査証跡は出力されません。

×：指定できません（KFPA19680-E メッセージが出力されてエラーとなります）。

(4) SDB ユティリティ操作イベントのオペランドの指定と監査対象イベントの関係

SDB ユティリティ操作イベントのオペランドの指定と監査対象イベントの関係を次の表に示します。

表 7-11 SDB ユティリティ操作イベントのオペランドの指定と監査対象イベントの関係

項番	AUDITTYPE の指定	操作種別の指定	SDB ユティリティ操作イベント種別の指定	選択オプションの指定	監査対象イベント
1	PRIVILEGE	SDB_UTILITY	PDSDBDEF	なし	pdsdbdef コマンド実行時の CONNECT 権限チェックを監査対象イベントにします。
2				あり	この指定の組み合わせはエラーになります。
3			PDSDBLOD	なし	pdsdblod コマンド実行時の CONNECT 権限チェックを監査対象イベントにします。
4				あり	この指定の組み合わせはエラーになります。
5			PDSDBROG	なし	pdsdbrog コマンド実行時の CONNECT 権限チェックを監査対象イベントにします。
6				あり	この指定の組み合わせはエラーになります。
7			ANY	なし	項番 1，3，5 のすべてを監査対象イベントにします。
8				あり	この指定の組み合わせはエラーになります。
9		ANY	指定なし	なし	項番 1，3，5 のすべてを監査対象イベントにします。
10				あり	この指定をしても，項番 7 の項目は，監査対象イベントにはなりません。
11	EVENT	SDB_UTILITY	PDSDBDEF	なし	pdsdbdef コマンドの次の操作を監査対象イベントにします。 • *ENTRY DICTIONARY 文，*ALTER DICTIONARY 文，*DELETE DICTIONARY 文の実行 • *CHECK DICTIONARY 文の実行 • *ENTRY DIRECTORY 文，ALTER DIRECTORY 文，DELETE DIRECTORY 文の実行

項番	AUDITTYPE の指定	操作種別の指定	SDB ユティリティ操作イベント種別の指定	選択オプションの指定	監査対象イベント
					• SDB ディレクトリ情報ファイルの作成 • pdsdbdef コマンドの終了
12				あり	この指定の組み合わせはエラーになります。
13			PDSDBLOD	なし	pdsdblod コマンドの終了を監査対象イベントにします。
14				あり	この指定の組み合わせはエラーになります。
15			PDSDBROG	なし	pdsdbrog コマンドの終了を監査対象イベントにします。
16				あり	この指定の組み合わせはエラーになります。
17			ANY	なし	項番 11，13，15 のすべてを監査対象イベントにします。
18				あり	この指定の組み合わせはエラーになります。
19		ANY	指定なし	なし	項番 11，13，15 のすべてを監査対象イベントにします。
20				あり	この指定をしても，項番 17 の項目は，監査対象イベントにはなりません。
21	ANY	SDB_UTILITY	PDSDBDEF	なし	項番 1 および 11 を監査対象イベントにします。
22				あり	この指定の組み合わせはエラーになります。
23			PDSDBLOD	なし	項番 3 および 13 を監査対象イベントにします。
24				あり	この指定の組み合わせはエラーになります。
25			PDSDBROG	なし	項番 5 および 15 を監査対象イベントにします。
26				あり	この指定の組み合わせはエラーになります。
27			ANY	なし	項番 21，23，25 のすべてを監査対象イベントにします。
28				あり	この指定の組み合わせはエラーになります。

項番	AUDITTYPE の指定	操作種別の指定	SDB ユティリティ操作イベント種別の指定	選択オプションの指定	監査対象イベント
29		ANY	指定なし	なし	項番 21，23，25 のすべてを監査対象イベントにします。
30				あり	この指定をしても，項番 27 の項目は，監査対象イベントにはなりません。

7.4.3 監査対象イベントの定義の削除

定義系 SQL の DROP AUDIT 文で監査対象イベントの定義を削除します。DROP AUDIT 文の指定形式を次に示します。

(1) 指定形式

項番	指定形式	指定要否
1	DROP AUDIT 〔AUDITTYPE {PRIVILEGE ｜ EVENT ｜ ANY}〕	○
2	FOR 操作種別	○
3	〔選択オプション〕	○
4	〔WHENEVER {SUCCESSFUL ｜ UNSUCCESSFUL ｜ ANY}〕	○

■各項目の詳細

項番	指定形式	指定要否
5	操作種別：：＝ {ANY	○
	｜ SESSION〔{セッション種別｜ ANY}〕	×
	｜ PRIVILEGE〔{権限操作種別｜ ANY}〕	×
	｜ DEFINITION〔{オブジェクト定義イベント種別｜ ANY}〕	×
	｜ ACCESS〔{オブジェクト操作イベント種別｜ ANY}〕	×
	｜ UTILITY〔{ユティリティイベント種別｜ ANY}〕	×
	｜ SDB_ACCESS〔{SDB データベース操作イベント種別｜ ANY}〕	○
	｜ SDB_UTILITY〔{SDB ユティリティ操作イベント種別｜ ANY}〕}	○
6	選択オプション：：＝ {ON オブジェクト名 ｜ BY AUTHORIZATION 認可識別子}	○※
	オブジェクト名：：＝ {FUNCTION 認可識別子.ルーチン識別子	×

項番	指定形式	指定要否
	| INDEX　認可識別子.インデクス識別子 | LIST　認可識別子.表識別子 | PROCEDURE　認可識別子.ルーチン識別子 | RDAREA　RD エリア名 | SCHEMA　認可識別子 | TABLE 〔認可識別子.〕表識別子 | TRIGGER　認可識別子.トリガ識別子 | TYPE　認可識別子.データ型識別子 | VIEW　認可識別子.表識別子 | SEQUENCE　認可識別子.順序数生成子識別子}	
7	セッション種別：：＝ {CONNECT | DISCONNECT | AUTHORIZATION}	×
8	権限操作種別：：＝ {GRANT | REVOKE}	×
9	オブジェクト定義イベント種別：：＝ {CREATE | DROP | ALTER}	×
10	オブジェクト操作イベント種別：：＝ {SELECT | INSERT | UPDATE | DELETE | PURGE | ASSIGN | CALL | LOCK | NEXT VALUE}	×
11	ユティリティイベント種別：：＝ {PDLOAD | PDRORG | PDEXP | PDCONSTCK}	×
12	SDB データベース操作イベント種別：：＝ {FETCH 〔FETCH オプション〕 | STORE | MODIFY | ERASE | CLEAR | FETCHDB ALL | GET}	○
13	FETCH オプション：：＝ FIRST	○
14	SDB ユティリティ操作イベント種別：：＝ {PDSDBDEF | PDSDBLOD | PDSDBROG}	○

（凡例）

○：指定必須のオペランドまたは指定を検討するオペランドです。

×：関係のないオペランドのため，指定不要です。

注※

ON オブジェクト名は指定できません。

(2)　オペランドの説明

ここでは，指定形式の項番 5，6，12～14 のオペランドについてだけ説明します。そのほかのオペランドについては，マニュアル「HiRDB SQL リファレンス」の「CREATE AUDIT（監査対象イベントの定義）」を参照してください。

5）操作種別：：＝ {ANY
| SDB_ACCESS〔{SDB データベース操作イベント種別 | ANY}〕
| SDB_UTILITY〔{SDB ユティリティ操作イベント種別 | ANY}〕}

監査対象から削除する操作種別を指定します。個々の操作種別と ANY は，個別に定義，および削除をします。例えば，SDB_UTILITY を定義した状態で，ANY だけを DROP AUDIT 文で削除しても，SDB_UTILITY の定義は残ります（監査対象のままとなります）。

ANY

すべての操作種別を監査対象とした定義を削除します。

SDB_ACCESS〔{SDB データベース操作イベント種別 | ANY}〕

SDB データベース操作イベントを監査対象から削除する場合に指定します。ANY は，すべての SDB データベース操作イベントを監査対象から削除します。個々の SDB データベース操作イベント種別と ANY は，個別に定義，および削除をします。例えば，FETCH および ANY を定義した状態で，ANY だけを DROP AUDIT 文で削除しても，FETCH の定義は残ります（監査対象のままとなります）。

SDB_UTILITY〔{SDB ユティリティ操作イベント種別 | ANY}〕

SDB ユティリティ操作イベントを監査対象から削除する場合に指定します。ANY はすべての SDB ユティリティ操作イベントを監査対象から削除します。個々の SDB ユティリティ操作イベント種別と ANY は，個別に定義，および削除をします。例えば，PDSDBLOD，PDSDBROG，および ANY を定義した状態で，ANY だけを DROP AUDIT 文で削除しても，PDSDBLOD および PDSDBROG の定義は残ります（監査対象のままとなります）。

6）選択オプション：：＝ {ON オブジェクト名 | BY AUTHORIZATION 認可識別子}

ON オブジェクト名

このオプションは指定できません。

BY AUTHORIZATION 認可識別子

監査対象から削除する，イベント実行者を指定します。

12）SDB データベース操作イベント種別：：＝ {FETCH 〔FETCH オプション〕
| STORE | MODIFY | ERASE | CLEAR | FETCHDB ALL | GET}

削除対象の SDB データベース操作種別を指定します。

13）FETCH オプション：：＝ FIRST

FIRST

削除対象の SDB データベース操作イベント種別の FETCH オプションを指定します。

14）SDB ユティリティ操作イベント種別：：＝ {PDSDBDEF | PDSDBLOD | PDSDBROG}

削除対象の SDB ユティリティ操作イベント種別を指定します。

7.5　運用方法

セキュリティ監査機能の運用方法については，マニュアル「HiRDB システム運用ガイド」の「セキュリティ監査機能の運用」の「運用方法」を参照してください。

ここでは，出力された監査証跡の参照方法についてだけ説明します。

7.5.1　監査証跡の参照方法

監査証跡表に格納された監査証跡を，pdrorg コマンドでアンロードしてファイル出力してください。その際，-W オプションに dat を指定して CSV 形式で出力してください。出力したファイルを入力情報とし，表計算ソフトなどにエクスポートして監査証跡の内容を確認してください。

コマンドの指定例

```
pdrorg -k unld -t 認可識別子.監査証跡表名 -W dat 制御情報ファイル名
```

7.6　監査証跡ファイルの運用

監査証跡ファイルの運用方法については，マニュアル「HiRDB システム運用ガイド」の「監査証跡ファイルの運用」を参照してください。

7.7　監査証跡表へのデータ登録

監査証跡表へのデータ登録方法については，マニュアル「HiRDB システム運用ガイド」の「監査証跡表へのデータ登録」を参照してください。

7.8　監査証跡表の列構成

監査証跡表の列構成については，マニュアル「HiRDB システム運用ガイド」の「監査証跡表の列構成」を参照してください。

7.8.1　監査証跡表の列に格納される情報についての留意事項（SDB ユティリティ操作イベントの場合）

(1)　EVENT_RESULT 列

監査証跡表の EVENT_RESULT 列には，イベントの実行結果が格納されます（権限のチェックまたはイベントが成功したかどうかが格納されます）。

次のコマンド実行時の CONNECT 権限チェックで，CONNECT エラーが発生した場合，その後の監査証跡は取得されないため，CONNECT の成功時に限りイベント成否の値（成功：S）が設定されます。

- pdsdbdef コマンド
- pdsdblod コマンド
- pdsdbrog コマンド

(2)　USED_PRIVILEGE 列

監査証跡表の USED_PRIVILEGE 列には，イベントを実行したときに使用した権限が格納されます。SDB ユティリティ操作イベントが実行された場合，USED_PRIVILEGE 列には CNT（CONNECT 権限）が格納されます。

(3)　UAP_NAME 列

監査証跡表の UAP_NAME 列には，イベント発行元の UAP 名が格納されます。UAP 名が 30 文字未満の場合は，UAP 名の後ろに’△’（空白）が合計 30 文字になるまで設定されます。

そのため，SDB ユティリティ操作イベントが実行された場合，UAP_NAME 列には，次の値が格納されます。

- pdsdbdef コマンドの実行時：'pdsdbdef△…△'
- pdsdblod コマンドの実行時：'pdsdblod△…△'
- pdsdbrog コマンドの実行時：'pdsdbrog△…△'

(4) OBJECT_SCHEMA 列

監査証跡表の OBJECT_SCHEMA 列には，イベントの権限チェックの対象になるオブジェクトの所有者名が格納されますが，SDB ユティリティ操作イベントが実行された場合，ナル値が格納されます。

(5) OBJECT_TYPE 列

監査証跡表の OBJECT_TYPE 列には，イベントの権限チェックの対象になるオブジェクトの種別が格納されます。SDB ユティリティ操作イベントが実行された場合，次のどれかの値が格納されます。

- SCM：SDB データベース（SDB データベース名）
- STS：SDB 格納データベース（SDB データベース格納名）
- DBS：SDB ディレクトリ情報（*ENTRY DIRECTORY 文の DBSCHM オペランドに指定した SDB データベース名）
- DIF：SDB ディレクトリ情報ファイル（ナル値）

（ ）内は，OBJECT_NAME 列に格納されるオブジェクト名です。

(6) ACCESS_COUNT 列

監査証跡表の ACCESS_COUNT 列には，オブジェクトに対するアクセス数が格納されます。次のコマンド実行時に ACCESS_COUNT 列に格納される値を次に示します。

- pdsdbdef コマンドの各イベント
  ナル値が格納されます。
- pdsdblod または pdsdbrog コマンドの終了
  対象イベントが成功した場合はレコード数が格納されます。対象イベントが失敗した場合はナル値が格納されます。ただし，レコード数が INTEGER 型の上限値より大きい場合は，INTEGER 型の上限値（2,147,483,647）が格納されます。

7.8.2 監査証跡表の列に格納される情報についての留意事項（SDB データベース操作イベントの場合）

(1) ACCESS_COUNT 列

監査証跡表の ACCESS_COUNT 列には，SDB データベース操作イベントが成功したときに，オブジェクト（レコード）を検索，格納，更新，または削除したレコード数が，次の表に示すとおりに設定されます。イベントの失敗時には NULL 値が格納されます。

表 7-12　SDB データベース操作イベントのアクセス件数

項番	データベース操作種別		1 回の操作で記録するアクセス件数	
			単一要求	複数要求※
1	SDB データベース操作	• レコードの検索（FETCH） • 位置指示子の位置づけ（FIND）	1（ただし，NOT FOUND の場合は 0）	n Σ（要求ごとの左記件数） i=1
2		レコードの格納（STORE）	1	
3		レコードの更新（MODIFY）		
4		レコードの削除（ERASE）	次のどちらかとなります。 • 削除レコードとそのレコード削除に伴って削除された配下の子レコードの合計数 • 一括オプション('O')指定で削除したレコード数 ただし，INTEGER 型の上限値より大きくなった場合は，INTEGER 型の上限値（2,147,483,647）となります。	n Σ（要求ごとの左記件数） i=1 ただし，INTEGER 型の上限値より大きくなった場合は，INTEGER 型の上限値（2,147,483,647）となります。
5		レコードの一括削除	1 回の要求で削除したレコード数	複数要求はありません。
6		複数レコードの検索（FETCHDB ALL）	1 回の要求で検索したレコード数	
7		レコードの取得（GET）	1	

注※

4V AFM または SD FMB の SDB データベースに対する操作の場合，複数要求はありません。

(2)　SQL_SOURCE 列

監査証跡表の SQL_SOURCE 列には，SDB データベースのレコード操作入力情報が格納されます。記録するレコード操作入力情報の長さを，システム共通定義の pd_aud_sql_source_size オペランドで指定する必要があります。pd_aud_sql_source_size を指定しないか，または 0 を指定した場合，SQL_SOURCE 列には NULL 値が記録されます。また，記録するレコード操作入力情報の長さが，pd_aud_sql_source_size オペランドの指定値よりも長い場合は，レコード操作入力情報の先頭から指定値までの内容が記録され，残りは切り捨てられます。このため，pd_aud_sql_source_size オペランドには，レコード操作入力情報に必要なサイズ以上の値を指定することを推奨します。

レコード操作入力情報に設定する情報を次の表に示します。

表 7-13　レコード操作入力情報に設定する情報（ユーザ指定値）

項番	設定情報	サイズ（バイト）	属性	要求						
				レコードの検索（FETCH）※	レコードの格納（STORE）	レコードの更新（MODIFY）	レコードの削除（ERASE）	レコードの一括削除	複数レコードの検索（FETCH DB ALL）	レコードの取得（GET）
1	要求エントリ番号	2	short	○	○	○	○	○	○	○
2	アクセスモード	1	char	○	○	○	○	○	○	○
3	指示コード	1	char	○	−	−	○	−	○	−
4	検索コード	1	char	○	○	−	−	−	−	−
5	ポインタオプション	1	char	−	○	○	−	−	−	−
6	データ格納形式オプション	1	char	○	−	○	−	−	−	−
7	予備	5	char	−	−	−	−	−	−	−
8	一連番号	4	char	○	−	−	−	−	−	−
9	アクセス条件	可変	char	○	○	−	−	○	○	−

（凡例）

○：ユーザ指定値のデータを設定

−：設定なし（0x00）

注※

位置指示子の位置づけ（FIND）も含まれます。

(a)　設定情報の内容

- 要求エントリ番号（1〜n）

  1 回のアクセスのレコード操作の順番です。要求をブロック化した複数要求の場合は，要求ごとのレコード操作の順番に 1〜n となります。

- アクセスモード

  ＜各要求共通＞

  'R'：参照

  'U'：更新

SD FMB の SDB データベースへのアクセス要求の場合，アクセスモードは SDB 用 UAP 環境定義の指定に従って HiRDB/SD が決定します。そのため，アクセスモードが決定される前にエラーが発生したときは，アクセスモードには X'00'が設定されます。

- 指示コード

  ＜ FETCH または FIND の場合＞

  'F'：FIRST ポインタを使用する検索

  'L'：LAST ポインタを使用する検索

  'U'：USER ポインタを使用する検索

  'N'：NEXT ポインタを使用する検索

  'P'：PRIOR ポインタを使用する検索

  'R'：ROWID 指定検索

  'W'：親子集合型の位置指示子の親レコード検索

  ＜ ERASE の場合＞

  'O'：最下位レベルのレコード型について全レコード削除

  'S'：削除目的のレコードとその下位レベルのレコード削除

  ＜ FETCHDB ALL の場合＞

  'F'：検索範囲の最小キーのルートレコードから検索を開始

  'N'：前回検索したレコードの次のレコードから検索を開始

  'S'：前回の検索したファミリの検索状態に関係なく，次のルートレコードから検索を開始

- 検索コード

  ＜ FETCH または FIND の場合＞

  'N'：USER ポインタから NEXT 方向

  'P'：USER ポインタから PRIOR 方向

  X'00'：指定なし

- ポインタオプション

  ＜ STORE の場合＞

  'U'：USER ポインタにする

  'N'：USER ポインタにしない

  X'00'：USER ポインタにしない

  ＜ MODIFY の場合＞

  'U'：USER ポインタにする

  'C'：USER ポインタを解除

  'N'：USER ポインタにしない

  X'00'：USER ポインタにしない

- データ格納形式オプション

  ＜ FETCH または FIND の場合＞

'D'：ユーザキー以外のキーの構成要素を除いて，データ格納エリアに格納

'E'：構成要素指定エリアで指定した構成要素だけをデータ格納

'K'：キーの構成要素を除かないでデータ格納エリアに格納

'N'：データ格納エリアへの格納をしない

＜ MODIFY の場合＞

'D'：ユーザキー以外のキーの構成要素を除いて，データ格納エリアに格納

'E'：レコードの更新のためのレコードの検索で指定した構成要素だけをデータ格納

'N'：USER ポインタの変更だけのため，データ格納エリアに格納していない

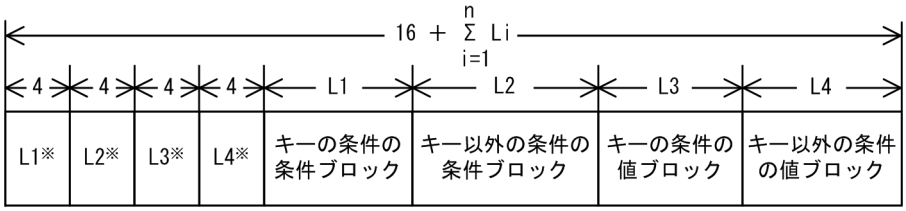
- **一連番号**

  検索要求で指定した一連番号です。

- **アクセス条件**

  次の図に示すフォーマットで格納します。各ブロックは，アクセス要求時のインタフェースエリアの内容をユーザが指定したサイズで設定します。そのため，ユーザが指定したサイズが，次の図に示す有効な条件や，値ブロックよりも大きなサイズで指定されている場合，各ブロックには有効な形式以外にも，ユーザが指定したエリアの内容がそのまま設定されます。

図 7-12　アクセス条件の形式



注※　int型の属性で設定し，サイズ0のときは該当項目は存在しません。

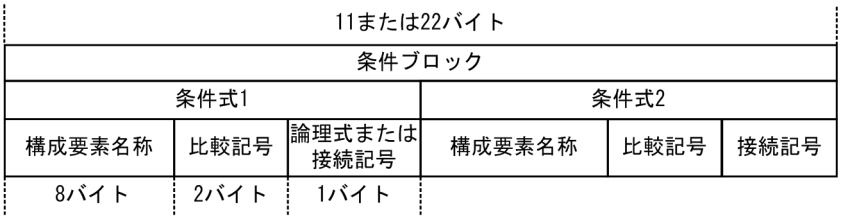
条件式形式オプションに'*FDN'を指定した場合と，'*EFN'を指定した場合で，キーの条件の条件ブロック形式，キー以外の条件の条件ブロック形式，キーの条件の値ブロック形式，およびキー以外の条件の値ブロック形式が異なります。それぞれの形式を説明します。

■条件式形式オプションに'*FDN'を指定した場合

- **キーの条件の条件ブロック形式**

  キーの条件の条件ブロック形式を次の図に示します。

図 7-13　キーの条件の条件ブロックの形式（条件式形式オプションに'*FDN'を指定した場合）



キーの条件の条件ブロック情報を次の表に示します。

表 7-14　キーの条件の条件ブロック情報（条件式形式オプションに'*FDN'を指定した場合）

相対位置（バイト）	長さ（バイト）	属性	内容	備考
0 + 11×(N−1)	8	char	構成要素名称 • 'DBKEY△△△'	なし
8 + 11×(N−1)	2	char	比較記号 • 'GT'または'>△'：大きい • 'GE'または'>='：大きいか等しい • 'LT'または'<△'：小さい • 'LE'または'<='：小さいか等しい • 'EQ'または'△='：等しい	なし
10 + 11×(N−1)	1	char	論理式 • '*'または'&'：論理積 接続記号 • '-'または'_'	次に条件式がある場合は，論理式が格納されます。最後の条件式の場合は，接続記号が格納されます。

（凡例）

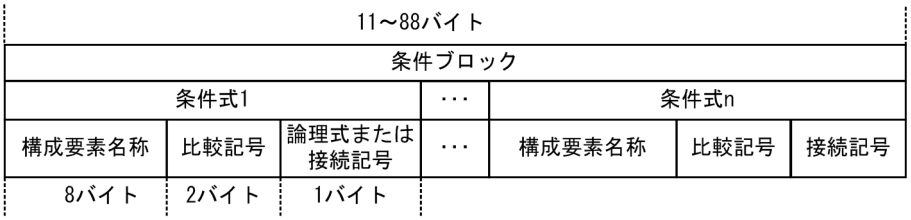
N：先頭からの条件式数

△：半角スペース

- キー以外の条件の条件ブロック形式

キー以外の条件の条件ブロックの形式を次の図に示します。

図 7-14　キー以外の条件の条件ブロックの形式（条件式形式オプションに'*FDN'を指定した場合）



キー以外の条件の条件ブロック情報を次の表に示します。

表 7-15　キー以外の条件の条件ブロック情報（条件式形式オプションに'*FDN'を指定した場合）

相対位置（バイト）	長さ（バイト）	属性	内容	備考
0 + 11×(N−1)	8	char	構成要素名称	なし
8 + 11×(N−1)	2	char	比較記号 • 'GT'または'>△'：大きい • 'GE'または'>='：大きいか等しい • 'LT'または'<△'：小さい	なし

相対位置 （バイト）	長さ （バイト）	属性	内容	備考
			• 'LE'または'<='：小さいか等しい • 'EQ'または'△='：等しい • 'NE'または'^='：等しくない	
10 ＋ 11×(N−1)	1	char	論理式 • '*'または'&'：論理積 • '+'または'|'：論理和 接続記号 • '-'または'_'	次に条件式がある場合は，論理式が格納されます。最後の条件式の場合は，接続記号が格納されます。

（凡例）

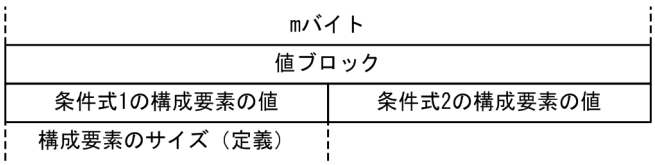
N：先頭からの条件式数

△：半角スペース

- **キーの条件の値ブロック形式**

キーの条件の値ブロックの形式を次の図に示します。

図 7-15　キーの条件の値ブロックの形式（条件式形式オプションに'*FDN'を指定した場合）



キーの条件の値ブロック情報を次の表に示します。

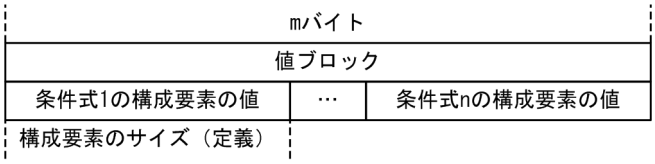
表 7-16　キーの条件の値ブロック情報（条件式形式オプションに'*FDN'を指定した場合）

相対位置 （バイト）	長さ （バイト）	属性	内容
1 つ前の構成要素までのサイズの合計	構成要素のサイズ	キーの条件を構成する各構成要素のデータ型の属性	値 キーの条件式の条件ブロックで指定した構成要素の値

- **キー以外の条件の値ブロック形式**

キー以外の条件の値ブロック形式を次の図に示します。

図 7-16　キー以外の条件の値ブロック形式（条件式形式オプションに'*FDN'を指定した場合）



キー以外の条件の値ブロック情報を次の表に示します。

表 7-17　キー以外の条件の値ブロック情報（条件式形式オプションに'*FDN'を指定した場合）

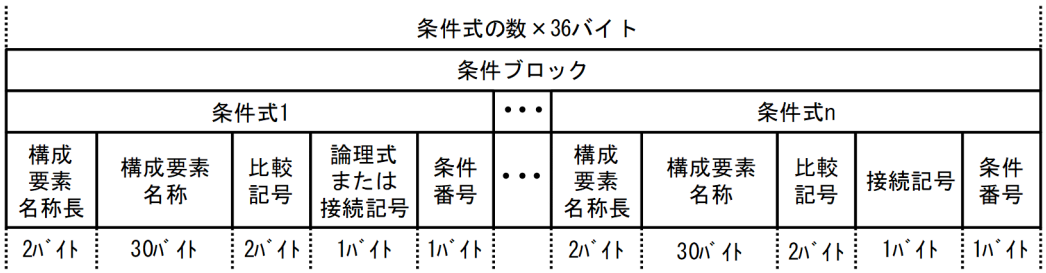
相対位置（バイト）	長さ（バイト）	属性	内容
1 つ前の構成要素までのサイズの合計	構成要素のサイズ	キー以外の条件を構成する各構成要素のデータ型の属性	値 キー以外の条件式の条件ブロックで指定した構成要素の値

■条件式形式オプションに'*EFN'を指定した場合

- キーの条件の条件ブロック形式

  キーの条件の条件ブロック形式を次の図に示します。

図 7-17　キーの条件の条件ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）



キーの条件の条件ブロック情報を次の表に示します。

表 7-18　キーの条件の条件ブロック情報（条件式形式オプションに'*EFN'を指定した場合）

相対位置（バイト）	長さ（バイト）	属性	内容	備考
0 + 36×(N−1)	2	short int	構成要素名称長	なし
2 + 36×(N−1)	30	char	構成要素名称	なし
32 + 36×(N−1)	2	char	比較記号 • 'GT'または'>△'：大きい • 'GE'または'>='：大きいか等しい • 'LT'または'<△'：小さい • 'LE'または'<='：小さいか等しい • 'EQ'または'△='：等しい	なし
34 + 36×(N−1)	1	char	論理式 • '*'または'&'：論理積 接続記号 • '-'または'_'	次に条件式がある場合は，論理式が格納されます。最後の条件式の場合は，接続記号が格納されます。
35 + 36×(N−1)	1	unsigned char	条件番号 • FETCH または FIND の場合	バージョン 10-04 より前の pdsdbcbl コマンドでプリプロセスした

相対位置（バイト）	長さ（バイト）	属性	内容	備考
			WHERE 句で指定した条件の先頭からの順序番号（1～64） • STORE の場合 1	UAP の場合は 0 になります。

（凡例）

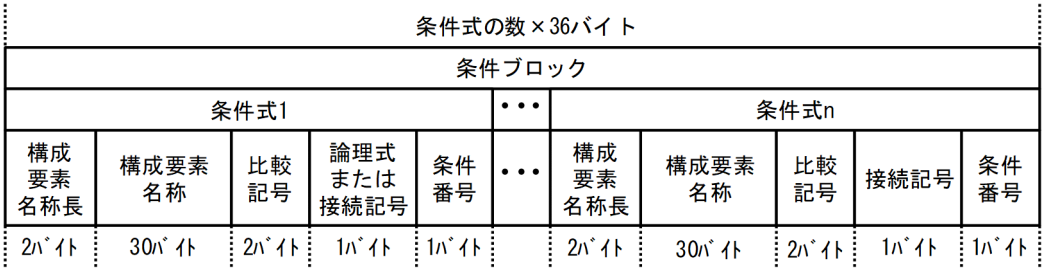
N：先頭からの条件式数

△：半角スペース

- **キー以外の条件の条件ブロック形式**

キー以外の条件の条件ブロックの形式を次の図に示します。

図 7-18　キー以外の条件の条件ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）



キー以外の条件の条件ブロック情報を次の表に示します。

表 7-19　キー以外の条件の条件ブロック情報（条件式形式オプションに'*EFN'を指定した場合）

相対位置（バイト）	長さ（バイト）	属性	内容	備考
0 + 36×(N－1)	2	short int	構成要素名称長	なし
2 + 36×(N－1)	30	char	構成要素名称	なし
32 + 36×(N－1)	2	char	比較記号 • 'GT'または'>△'：大きい • 'GE'または'>='：大きいか等しい • 'LT'または'<△'：小さい • 'LE'または'<='：小さいか等しい • 'EQ'または'△='：等しい	なし
34 + 36×(N－1)	1	char	論理式 • '*'または'&'：論理積 接続記号 • '-'または'_'	次に条件式がある場合は，論理式が格納されます。最後の条件式の場合は，接続記号が格納されます。
35 + 36×(N－1)	1	unsigned char	条件番号	なし

相対位置（バイト）	長さ（バイト）	属性	内容	備考
			FETCH または FIND の，WHERE 句で指定した条件の先頭からの順序番号（1～64）	

（凡例）

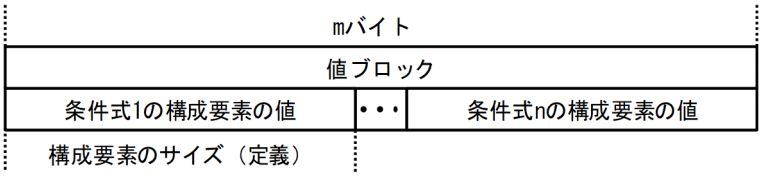
N：先頭からの条件式数

△：半角スペース

- **キーの条件の値ブロック形式**

  キーの条件の値ブロックの形式を次の図に示します。

  図 7-19　キーの条件の値ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）



  キーの条件の値ブロック情報を次の表に示します。

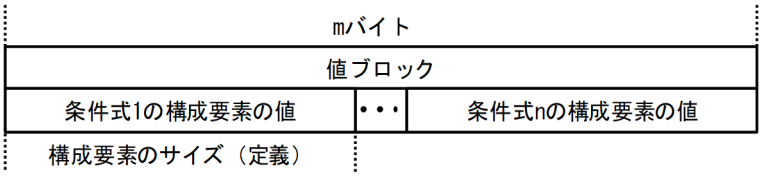
  表 7-20　キーの条件の値ブロック情報（条件式形式オプションに'*EFN'を指定した場合）

相対位置（バイト）	長さ（バイト）	属性	内容
1 つ前の構成要素までのサイズの合計	構成要素のサイズ	char	値 キーの条件式の条件ブロックで指定した構成要素の値

- **キー以外の条件の値ブロック形式**

  キー以外の条件の値ブロックの形式を次の図に示します。

  図 7-20　キー以外の条件の値ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）



  キー以外の条件の値ブロック情報を次の表に示します。

  表 7-21　キー以外の条件の値ブロック情報（条件式形式オプションに'*EFN'を指定した場合）

相対位置（バイト）	長さ（バイト）	属性	内容
1 つ前の構成要素までのサイズの合計	構成要素のサイズ	char	値

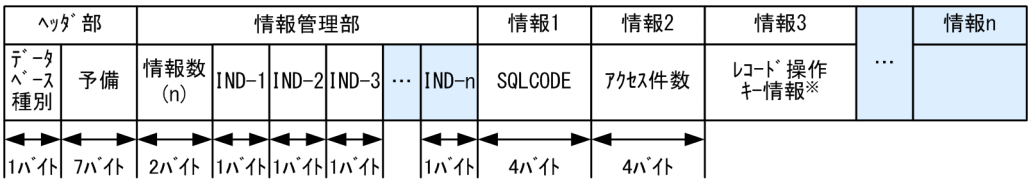
相対位置 （バイト）	長さ （バイト）	属性	内容
			キー以外の条件式の条件ブロックで指定した構成要素の値

(3)　SQL_DATA 列

監査証跡表の SQL_DATA 列には，SDB データベースのレコード操作出力情報が格納されます。記録するレコード操作出力情報の長さを，システム共通定義の pd_aud_sql_data_size オペランドで指定する必要があります。pd_aud_sql_data_size オペランドを省略するか，または 0 を指定した場合は NULL 値が記録されます。また，記録するレコード操作出力情報の長さが，pd_aud_sql_data_size オペランドの指定値よりも長い場合は，レコード操作出力情報の先頭から指定値までの内容が記録され，残りは切り捨てられます。このため，pd_aud_sql_data_size オペランドには，レコード操作出力情報に必要なサイズ以上の値を指定することを推奨します。

レコード操作出力情報の形式を次の図に示します。

図 7-21　レコード操作出力情報の形式



注※

レコード操作キー情報の形式および長さについては，「(a)　レコード操作キー情報」を参照してください。

レコード操作出力情報を次の表に示します。

表 7-22　レコード操作出力情報

相対位置 （バイト）	長さ （バイト）	属性	内容	備考
0	1	char	＜ヘッダ部＞ SDB データベース種別 次の値を設定します。 • 4V FMB または SD FMB の場合：'F' • 4V DAM の場合：'D' • 4V MAM の場合：'M' • 4V TAM の場合：'T' • 4V SAM の場合：'S'	なし
1	7	char	＜ヘッダ部＞ 予備(0x00)	なし

相対位置（バイト）	長さ（バイト）	属性	内容	備考
8	2	short	＜情報管理部＞ 情報数（n）	なし
10	1×n	char	＜情報管理部＞ 情報領域の存在を示すインジケータ（IND）を出力情報数分作成します。 インジケータ（IND）は，情報 1～n にそれぞれ対応していて，各情報領域の存在を管理し，次の値を設定します。 • 情報領域あり：0x00 • 情報領域なし：0xFF	なし
10 ＋(1×n)	4	int	＜情報 1 ＞ SQLCODE SDB データベース操作の処理結果（SQLCODE）を設定します。	なし
10 ＋(1×n)＋SQLCODE のサイズ※	4	int	＜情報 2 ＞ アクセス件数 「表 7-12　SDB データベース操作イベントのアクセス件数」の 1 回の操作で記録するアクセス件数の単一要求の内容で設定します。複数要求の場合も，個々のレコードごとに単一要求の内容で設定します。	エラーの場合（SQLCODE<0）や，情報の取得ができない場合は，領域そのものが存在しません（対応する情報管理部の IND：0xFF）。
10 ＋(1×n)＋SQLCODE のサイズ※＋アクセス件数のサイズ※	「(a)　レコード操作キー情報」を参照		＜情報 3 ＞ レコード操作キー情報を設定します。	

注※

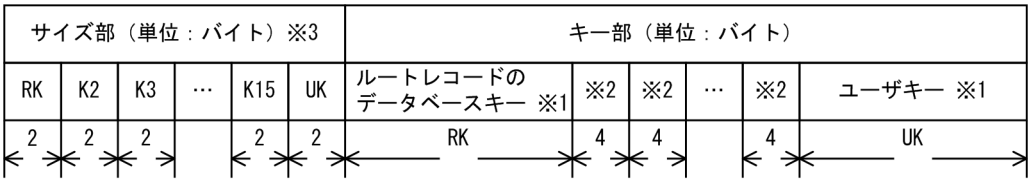
情報管理部の対応するインジケータ（IND）が，情報領域なし（0xFF）の場合は 0 となります。

(a)　レコード操作キー情報

■4V FMB または SD FMB の SDB データベースの場合

レコード操作キー情報の形式を次の図に示します。

図 7-22　レコード操作キー情報の形式（4V FMB または SD FMB の SDB のデータベースの場合）



注※1　構成要素のデータ型の属性で設定します。
注※2　レベル2～15の一連番号でint型の属性で設定します。
注※3　short型の属性で設定します。

[説明]

サイズ部には，すべてのエントリ（RK～UK）が存在し，それぞれキー部の情報と対応します。サイズ部の各エントリに設定したサイズでキー情報が存在します。サイズが 0 の場合，キー情報の領域そのものが存在しません。複数レコードの検索（FETCHDB ALL）では，RK およびルートレコードのデータベースキーだけを設定します。

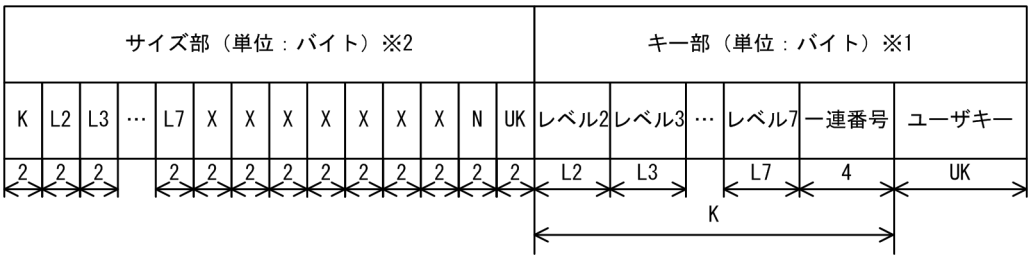
NOT FOUND 時（SQLCODE=100）は，アクセスレコードの階層レベルのキーサイズ（複数レコードの検索（FETCHDB ALL）の場合は RK）および UK に 0 を設定します。

- RK：ルートレコードのデータベースキーのサイズを設定します。
- K2～K15：子レコードの場合，アクセスしたレコードの階層まで一連番号のサイズ 4 を設定し，それ以降は 0 を設定します。
- UK：アクセスレコードがユーザキーを持つ場合，ユーザキーサイズを設定し，持たない場合は 0 を設定します。

■4V AFM の SDB データベースの場合

レコード操作キー情報の形式を次の図に示します。

図 7-23　レコード操作キー情報の形式（4V AFM の SDB データベースの場合）



注※1　一連番号はint型の属性で設定し，各レベルのキーおよびユーザキーは構成要素のデータ型の属性で設定します。
注※2　short型の属性で設定します。

[説明]

サイズ部には，すべてのエントリ（K～UK）が存在し，L2～L7，N はそれぞれキー部の情報と対応します。サイズ部の各エントリに設定したサイズでキー情報が存在します。サイズ 0 の場合，キー情報の領域そのものが存在しません。

NOT FOUND 時（SQLCODE=100）は，サイズ部のエントリに 0 を設定します。

- K：キー全体のサイズ

- L2～L7：各レベルのキー（構成要素名称が DBKEY で一連番号を除く）サイズ
- X：未使用エントリ
  0 を設定します。このエントリに対応するキー情報の領域はありません。
- N：一連番号のエリアサイズ
  4 を設定します。
- UK：アクセスレコードがユーザキーを持つ場合，ユーザキーサイズを設定します。持たない場合は，0 を設定します。

7.8.3 イベントタイプおよびイベントサブタイプの詳細

イベントタイプおよびイベントサブタイプの詳細については，マニュアル「HiRDB システム運用ガイド」の「監査証跡表の列構成」の「イベントタイプおよびイベントサブタイプの詳細」を参照してください。

さらに，次の表に示す SDB ユティリティ操作イベントおよび SDB データベース操作イベントに関するイベントタイプおよびイベントサブタイプがあります。

表 7-23 SDB ユティリティ操作イベントおよび SDB データベース操作イベントに関するイベントタイプおよびイベントサブタイプ

監査イベント	イベントタイプの値	イベントサブタイプの値	詳細
SDB ユティリティ操作イベント	SDU	DEF	HiRDB/SD 定義ユティリティ（pdsdbdef コマンド）の実行
		LOD	HiRDB/SD データベース作成ユティリティ（pdsdblod コマンド）の実行
		ROG	HiRDB/SD データベース再編成ユティリティ（pdsdbrog コマンド）の実行
		ENT	pdsdbdef コマンドの*ENTRY DICTIONARY 文または*ENTRY DIRECTORY 文の実行
		DEL	pdsdbdef コマンドの*DELETE DICTIONARY 文または*DELETE DIRECTORY 文の実行
		ALT	pdsdbdef コマンドの*ALTER DICTIONARY 文または*ALTER DIRECTORY 文の実行
		CHK	pdsdbdef コマンドの*CHECK DICTIONARY 文の実行
		INF	pdsdbdef コマンドの SDB ディレクトリ情報ファイルの作成
SDB データベース操作イベント	SDA	FTC	レコードの検索（FETCH）の実行
		STR	レコードの格納（STORE）の実行

監査イベント	イベントタイプの値	イベントサブタイプの値	詳細
		MOD	レコードの更新（MODIFY）の実行
		ERS	レコードの削除（ERASE）の実行
		CLA	レコードの一括削除の実行
		FTA	複数レコードの検索（FETCHDB ALL）の実行
		GET	レコードの取得（GET）の実行

7.8.4 SQL コードまたは終了コードとイベント成否の詳細

SQL コードまたは終了コードとイベント成否の詳細については，マニュアル「HiRDB システム運用ガイド」の「監査証跡表の列構成」の「SQL コードまたは終了コードとイベント成否の詳細」を参照してください。

SDB ユティリティ操作イベントに関するイベント成否の詳細については，次の表に示します。

表 7-24 SDB ユティリティ操作イベントに関するイベント成否の詳細

イベント名称	終了コードの値	意味	イベント成否の値	備考
pdsdbdef（SDB ディクショナリ情報の操作）	0	成功	S	正常終了
	12	失敗	F	異常終了
pdsdbdef（データベース定義のチェック）	0	成功	S	正常終了
	12	失敗	F	異常終了
pdsdbdef（SDB ディレクトリ情報の操作）	0	成功	S	正常終了
	12	失敗	F	異常終了
pdsdbdef（SDB ディレクトリ情報ファイルの作成）	0	成功	S	正常終了
	12	失敗	F	異常終了
pdsdbdef コマンドの終了	0	成功	S	正常終了
	12	失敗	F	異常終了
pdsdblod コマンドの終了	0	成功	S	正常終了
	4	一部失敗	U	警告レベルのエラー（入力データのエラーを検知）
	8	失敗	F	コマンドの指定誤り
	12	失敗	F	異常終了
pdsdbrog コマンドの終了	0	成功	S	正常終了

イベント名称	終了コードの値	意味	イベント成否の値	備考
	4	一部失敗	U	警告レベルのエラー
	8	失敗	F	コマンドの指定誤り
	12	失敗	F	異常終了

7.9　監査証跡の絞り込み

監査証跡の絞り込みをすると，特定の監査証跡だけを取得できます。監査証跡の絞り込みは，定義系 SQL の CREATE AUDIT 文で条件を定義し，必要に応じて DROP AUDIT 文で削除できます。

詳細については，マニュアル「HiRDB システム運用ガイド」の「監査証跡の絞り込み」を参照してください。

7.10　障害が発生したときの対処方法

監査証跡ファイルに障害が発生したときの対処方法については，マニュアル「HiRDB システム運用ガイド」の「監査証跡ファイルに障害が発生したときの対処方法」を参照してください。

自動データロード機能適用中に障害が発生したときの対処方法については，マニュアル「HiRDB システム運用ガイド」の「自動データロード機能適用中に障害が発生したときの対処方法」を参照してください。

7.11　ほかの機能との関連

- 系切り替え機能との関連
  - 切り替え先の系で監査証跡の取得を引き継ぐかどうかは，切り替え元の HiRDB の終了モードによって決まります。切り替え先の系が再開始の場合は系を切り替える前の状態を引き継ぎます。切り替え先の系が正常開始の場合は pd_audit オペランドの指定に従います。
  - 障害が発生して系が切り替わった場合，切り替わる直前の監査証跡は正しく取得されません。
- インナレプリカ機能との関連

  ペアボリューム化（ディスクの二重化）については監査証跡を取得しません。ペアボリューム化を監査するには OS の監査機能を使用してください。

7.12　監査証跡のレコード項目

監査証跡のレコード項目については，マニュアル「HiRDB システム運用ガイド」の「監査証跡のレコード項目（権限チェック時)」および「監査証跡のレコード項目（イベント終了時)」を参照してください。

ここでは，SDB ユティリティ操作イベントで出力される監査証跡のレコード項目について説明します。

7.12.1　pdsdbdef コマンドの場合

監査証跡のレコード項目（pdsdbdef コマンドの場合）を次の表に示します。

表 7-25　監査証跡のレコード項目（pdsdbdef コマンドの場合）

項番	監査証跡のレコード項目	イベントの種類		
		権限チェック	操作※1	コマンドの終了
1	ユーザ識別子	○	○	○
2	イベント実行日	○	○	○
3	イベント実行時刻	○	○	○
4	イベント実行時刻（マイクロ秒）	○	○	○
5	イベントタイプ	SDU		
6	イベントサブタイプ	DEF	○※2	DEF
7	イベント成否	○※3	○	○
8	使用した権限	CNT	△△△	△△△
9	UAP 名称	pdsdbdef△…△（30 バイト固定）		
10	サービス名称	－	－	－
11	IP アドレス	－	－	－
12	プロセス ID	○	○	○
13	スレッド ID	－	－	－
14	ホスト名	○	○	○
15	ユニット識別子	－	－	－
16	サーバ名称	－	－	－
17	コネクト通番	－	－	－
18	SQL 通番	－	－	－
19	オブジェクトの所有者名	－	－	－
20	オブジェクト名称	－	○※4	－

項番	監査証跡のレコード項目	イベントの種類		
		権限チェック	操作※1	コマンドの終了
21	オブジェクト種別	−	○※5	−
22	付与，削除，変更した権限	−	−	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−	−	−
24	監査関連オペランドの定義値	−	−	−
25	監査証跡種別	−	E	E
26	SQL コードまたは終了コード	−	○	○
27	スワップ元監査証跡ファイル名称	−	−	−
28	スワップ先監査証跡ファイル名称	−	−	−
29	セキュリティ関連変更種別	−	−	−
30	セキュリティ関連変更前設定値	−	−	−
31	セキュリティ関連変更後設定値	−	−	−
32	監査証跡表オプション	−	−	−
33	アクセス件数	−	−	−
34	SQL 文	−	−	−
35	SQL データ	−	−	−
36	ユーザ付加情報 1	−	−	−
37	ユーザ付加情報 2	−	−	−
38	ユーザ付加情報 3	−	−	−
39	関連製品付加情報 1	−	−	−
40	監査証跡表へのデータ登録時刻印	○	○	○
41	監査証跡表へのデータ登録時の通番	○	○	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- 出力されるレコード数については，「表 3-21　出力される監査証跡のレコード数（SDB ユティリティ操作イベントの場合）」を参照してください。
- pdsdbdef コマンドの延長で実行される SQL 文はありません。

- pdsdbdef コマンドを実行したユニットの監査証跡ファイルに出力されます。
- pdsdbdef コマンドの実行時，CONNECT 権限チェックで CONNECT エラーが発生した場合，すべての監査証跡が取得されません。

注※1

次の操作が該当します。

- SDB ディクショナリ情報の操作
- データベース定義のチェック
- SDB ディレクトリ情報の操作
- SDB ディレクトリ情報ファイルの作成

注※2

イベントサブタイプに出力される情報については，「7.8.3　イベントタイプおよびイベントサブタイプの詳細」を参照してください。

注※3

CREATE AUDIT 文の指定内容に関係なく，pdsdbdef コマンド実行時の CONNECT 権限チェックが失敗した場合，監査証跡は取得されません。

注※4

SDB ディレクトリ情報ファイルの作成時は，オブジェクト名称は取得されません。

注※5

オブジェクト種別に出力される情報については，「7.8.1(5)　OBJECT_TYPE 列」を参照してください。

7.12.2　pdsdblod コマンドの場合

監査証跡のレコード項目（pdsdblod コマンドの場合）を次の表に示します。

表 7-26　監査証跡のレコード項目（pdsdblod コマンドの場合）

項番	監査証跡のレコード項目	イベントの種類	
		権限チェック	コマンドの終了
1	ユーザ識別子	○	○
2	イベント実行日	○	○
3	イベント実行時刻	○	○
4	イベント実行時刻（マイクロ秒）	○	○
5	イベントタイプ	SDU	
6	イベントサブタイプ	LOD	
7	イベント成否	○※1	○

項番	監査証跡のレコード項目	イベントの種類	
		権限チェック	コマンドの終了
8	使用した権限	CNT	△△△
9	UAP 名称	pdsdblod△…△（30 バイト固定）	
10	サービス名称	－	－
11	IP アドレス	－	－
12	プロセス ID	○	○
13	スレッド ID	－	－
14	ホスト名	○	○
15	ユニット識別子	－	－
16	サーバ名称	－	－
17	コネクト通番	－	－
18	SQL 通番	－	－
19	オブジェクトの所有者名	－	－
20	オブジェクト名称	－	○
21	オブジェクト種別	－	SCM※2
22	付与，削除，変更した権限	－	－
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	－	－
24	監査関連オペランドの定義値	－	－
25	監査証跡種別	－	E
26	SQL コードまたは終了コード	－	○
27	スワップ元監査証跡ファイル名称	－	－
28	スワップ先監査証跡ファイル名称	－	－
29	セキュリティ関連変更種別	－	－
30	セキュリティ関連変更前設定値	－	－
31	セキュリティ関連変更後設定値	－	－
32	監査証跡表オプション	－	－
33	アクセス件数	－	○
34	SQL 文	－	－
35	SQL データ	－	－
36	ユーザ付加情報 1	－	－

項番	監査証跡のレコード項目	イベントの種類	
		権限チェック	コマンドの終了
37	ユーザ付加情報 2	−	−
38	ユーザ付加情報 3	−	−
39	関連製品付加情報 1	−	−
40	監査証跡表へのデータ登録時刻印	○	○
41	監査証跡表へのデータ登録時の通番	○	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- pdsdblod コマンドが 1 回実行されると 1 レコードが記録されます。
- pdsdblod コマンドの延長で実行される SQL 文はありません。
- pdsdblod コマンドを実行したユニットの監査証跡ファイルに出力されます。
- pdsdblod コマンドの実行時，CONNECT 権限チェックで CONNECT エラーが発生した場合，すべての監査証跡が取得されません。

注※1

CREATE AUDIT 文の指定内容に関係なく，pdsdblod コマンド実行時の CONNECT 権限チェックが失敗した場合，監査証跡は取得されません。

注※2

インデクスの再作成を実行した場合，オブジェクト種別は IDX となります。

7.12.3 pdsdbrog コマンドの場合

監査証跡のレコード項目（pdsdbrog コマンドの場合）を次の表に示します。

表 7-27 監査証跡のレコード項目（pdsdbrog コマンドの場合）

項番	監査証跡のレコード項目	イベントの種類	
		権限チェック	コマンドの終了
1	ユーザ識別子	○	○
2	イベント実行日	○	○
3	イベント実行時刻	○	○

項番	監査証跡のレコード項目	イベントの種類	
		権限チェック	コマンドの終了
4	イベント実行時刻（マイクロ秒）	○	○
5	イベントタイプ	SDU	
6	イベントサブタイプ	ROG	
7	イベント成否	○※	○
8	使用した権限	CNT	△△△
9	UAP 名称	pdsdbrog△…△（30 バイト固定）	
10	サービス名称	−	−
11	IP アドレス	−	−
12	プロセス ID	○	○
13	スレッド ID	−	−
14	ホスト名	○	○
15	ユニット識別子	−	−
16	サーバ名称	−	−
17	コネクト通番	−	−
18	SQL 通番	−	−
19	オブジェクトの所有者名	−	−
20	オブジェクト名称	−	○
21	オブジェクト種別	−	SCM
22	付与，削除，変更した権限	−	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−	−
24	監査関連オペランドの定義値	−	−
25	監査証跡種別	−	E
26	SQL コードまたは終了コード	−	○
27	スワップ元監査証跡ファイル名称	−	−
28	スワップ先監査証跡ファイル名称	−	−
29	セキュリティ関連変更種別	−	−
30	セキュリティ関連変更前設定値	−	−
31	セキュリティ関連変更後設定値	−	−
32	監査証跡表オプション	−	−

項番	監査証跡のレコード項目	イベントの種類	
		権限チェック	コマンドの終了
33	アクセス件数	−	○
34	SQL 文	−	−
35	SQL データ	−	−
36	ユーザ付加情報 1	−	−
37	ユーザ付加情報 2	−	−
38	ユーザ付加情報 3	−	−
39	関連製品付加情報 1	−	−
40	監査証跡表へのデータ登録時刻印	○	○
41	監査証跡表へのデータ登録時の通番	○	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- pdsdbrog コマンドが 1 回実行されると 1 レコードが記録されます。
- pdsdbrog コマンドの延長で実行される SQL 文はありません。
- pdsdbrog コマンドを実行したユニットの監査証跡ファイルに出力されます。
- pdsdbrog コマンドの実行時，CONNECT 権限チェックで CONNECT エラーが発生した場合，すべての監査証跡が取得されません。

注※

CREATE AUDIT 文の指定内容に関係なく，pdsdbrog コマンド実行時の CONNECT 権限チェックが失敗した場合，監査証跡は取得されません。

7.12.4 レコードの検索（FETCH）または位置指示子の位置づけ（FIND）の場合

監査証跡のレコード項目（レコードの検索（FETCH）または位置指示子の位置づけ（FIND）の場合）を次の表に示します。

表 7-28　監査証跡のレコード項目（レコードの検索（FETCH）または位置指示子の位置づけ（FIND）の場合）

項番	監査証跡のレコード項目	イベントの終了
1	ユーザ識別子	○
2	イベント実行日	○
3	イベント実行時刻	○
4	イベント実行時刻（マイクロ秒）	○
5	イベントタイプ	SDA
6	イベントサブタイプ	FTC
7	イベント成否	○
8	使用した権限	△△△
9	UAP 名称	○
10	サービス名称	○
11	IP アドレス	○
12	プロセス ID	○
13	スレッド ID	○
14	ホスト名	○
15	ユニット識別子	○
16	サーバ名称	○
17	コネクト通番	○
18	SQL 通番※1	○
19	オブジェクトの所有者名	データベース名
20	オブジェクト名称※2	○
21	オブジェクト種別※2	SCH または TBL
22	付与，削除，変更した権限	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−
24	監査関連オペランドの定義値	−
25	監査証跡種別	E
26	SQL コードまたは終了コード	○
27	スワップ元監査証跡ファイル名称	−
28	スワップ先監査証跡ファイル名称	−

項番	監査証跡のレコード項目	イベントの終了
29	セキュリティ関連変更種別	−
30	セキュリティ関連変更前設定値	−
31	セキュリティ関連変更後設定値	−
32	監査証跡表オプション	−
33	アクセス件数	○
34	SQL 文	○
35	SQL データ	○
36	ユーザ付加情報 1	○
37	ユーザ付加情報 2	○
38	ユーザ付加情報 3	○
39	関連製品付加情報 1	−
40	監査証跡表へのデータ登録時刻印	○
41	監査証跡表へのデータ登録時の通番	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- SDB データベース操作の要求単位に 1 レコードが記録されます。
- SDB データベース操作の延長で実行される SQL 文はありません。
- 要求を受け付けたフロントエンドサーバがあるユニットの監査証跡ファイルに出力されます。

注※1

SQL 通番は，SDB データベースを操作する API または DML の要求通番と読み替えてください。また，複数要求の場合は，SDB データベース操作の要求単位ごとに同じ通番となります。

注※2

SDB データベースを操作する API または DML でレコード名を指定してアクセスした場合，オブジェクト種別には TBL が，オブジェクト名称にはレコード名が設定されます。また，データベース名を指定してアクセスした場合，オブジェクト種別には SCH が，オブジェクト名称にはデータベース名が設定されます。

7.12.5　レコードの格納（STORE）の場合

監査証跡のレコード項目（レコードの格納（STORE）の場合）を次の表に示します。

表 7-29　監査証跡のレコード項目（レコードの格納（STORE）の場合）

項番	監査証跡のレコード項目	イベントの終了
1	ユーザ識別子	○
2	イベント実行日	○
3	イベント実行時刻	○
4	イベント実行時刻（マイクロ秒）	○
5	イベントタイプ	SDA
6	イベントサブタイプ	STR
7	イベント成否	○
8	使用した権限	△△△
9	UAP 名称	○
10	サービス名称	○
11	IP アドレス	○
12	プロセス ID	○
13	スレッド ID	○
14	ホスト名	○
15	ユニット識別子	○
16	サーバ名称	○
17	コネクト通番	○
18	SQL 通番※1	○
19	オブジェクトの所有者名	データベース名
20	オブジェクト名称※2	○
21	オブジェクト種別※2	TBL
22	付与，削除，変更した権限	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−
24	監査関連オペランドの定義値	−
25	監査証跡種別	E
26	SQL コードまたは終了コード	○

項番	監査証跡のレコード項目	イベントの終了
27	スワップ元監査証跡ファイル名称	−
28	スワップ先監査証跡ファイル名称	−
29	セキュリティ関連変更種別	−
30	セキュリティ関連変更前設定値	−
31	セキュリティ関連変更後設定値	−
32	監査証跡表オプション	−
33	アクセス件数	○
34	SQL 文	○
35	SQL データ	○
36	ユーザ付加情報 1	○
37	ユーザ付加情報 2	○
38	ユーザ付加情報 3	○
39	関連製品付加情報 1	−
40	監査証跡表へのデータ登録時刻印	○
41	監査証跡表へのデータ登録時の通番	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- SDB データベース操作の要求単位に 1 レコードが記録されます。
- SDB データベース操作の延長で実行される SQL 文はありません。
- 要求を受け付けたフロントエンドサーバがあるユニットの監査証跡ファイルに出力されます。

注※1

SQL 通番は，SDB データベースを操作する API または DML の要求通番と読み替えてください。また，複数要求の場合は，SDB データベース操作の要求単位ごとに同じ通番となります。

注※2

オブジェクト種別には TBL が，オブジェクト名称にはレコード名が設定されます。

7.12.6 レコードの更新（MODIFY）の場合

監査証跡のレコード項目（レコードの更新（MODIFY）の場合）を次の表に示します。

表 7-30　監査証跡のレコード項目（レコードの更新（MODIFY）の場合）

項番	監査証跡のレコード項目	イベントの終了
1	ユーザ識別子	○
2	イベント実行日	○
3	イベント実行時刻	○
4	イベント実行時刻（マイクロ秒）	○
5	イベントタイプ	SDA
6	イベントサブタイプ	MOD
7	イベント成否	○
8	使用した権限	△△△
9	UAP 名称	○
10	サービス名称	○
11	IP アドレス	○
12	プロセス ID	○
13	スレッド ID	○
14	ホスト名	○
15	ユニット識別子	○
16	サーバ名称	○
17	コネクト通番	○
18	SQL 通番※1	○
19	オブジェクトの所有者名	データベース名
20	オブジェクト名称※2	○
21	オブジェクト種別※2	TBL
22	付与，削除，変更した権限	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−
24	監査関連オペランドの定義値	−
25	監査証跡種別	E
26	SQL コードまたは終了コード	○
27	スワップ元監査証跡ファイル名称	−
28	スワップ先監査証跡ファイル名称	−
29	セキュリティ関連変更種別	−

項番	監査証跡のレコード項目	イベントの終了
30	セキュリティ関連変更前設定値	−
31	セキュリティ関連変更後設定値	−
32	監査証跡表オプション	−
33	アクセス件数	○
34	SQL 文	○
35	SQL データ	○
36	ユーザ付加情報 1	○
37	ユーザ付加情報 2	○
38	ユーザ付加情報 3	○
39	関連製品付加情報 1	−
40	監査証跡表へのデータ登録時刻印	○
41	監査証跡表へのデータ登録時の通番	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- SDB データベース操作の要求単位に 1 レコードが記録されます。
- SDB データベース操作の延長で実行される SQL 文はありません。
- 要求を受け付けたフロントエンドサーバがあるユニットの監査証跡ファイルに出力されます。

注※1

SQL 通番は，SDB データベースを操作する API または DML の要求通番と読み替えてください。また，複数要求の場合は，SDB データベース操作の要求単位ごとに同じ通番となります。

注※2

オブジェクト種別には TBL が，オブジェクト名称にはレコード名が設定されます。

7.12.7　レコードの削除（ERASE）の場合

監査証跡のレコード項目（レコードの削除（ERASE）の場合）を次の表に示します。

表 7-31　監査証跡のレコード項目（レコードの削除（ERASE）の場合）

項番	監査証跡のレコード項目	イベントの終了
1	ユーザ識別子	○
2	イベント実行日	○
3	イベント実行時刻	○
4	イベント実行時刻（マイクロ秒）	○
5	イベントタイプ	SDA
6	イベントサブタイプ	ERS
7	イベント成否	○
8	使用した権限	△△△
9	UAP 名称	○
10	サービス名称	○
11	IP アドレス	○
12	プロセス ID	○
13	スレッド ID	○
14	ホスト名	○
15	ユニット識別子	○
16	サーバ名称	○
17	コネクト通番	○
18	SQL 通番※1	○
19	オブジェクトの所有者名	データベース名
20	オブジェクト名称※2	○
21	オブジェクト種別※2	TBL
22	付与，削除，変更した権限	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−
24	監査関連オペランドの定義値	−
25	監査証跡種別	E
26	SQL コードまたは終了コード	○
27	スワップ元監査証跡ファイル名称	−
28	スワップ先監査証跡ファイル名称	−
29	セキュリティ関連変更種別	−

項番	監査証跡のレコード項目	イベントの終了
30	セキュリティ関連変更前設定値	−
31	セキュリティ関連変更後設定値	−
32	監査証跡表オプション	−
33	アクセス件数	○
34	SQL 文	○
35	SQL データ	○
36	ユーザ付加情報 1	○
37	ユーザ付加情報 2	○
38	ユーザ付加情報 3	○
39	関連製品付加情報 1	−
40	監査証跡表へのデータ登録時刻印	○
41	監査証跡表へのデータ登録時の通番	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- SDB データベース操作の要求単位に 1 レコードが記録されます。
- SDB データベース操作の延長で実行される SQL 文はありません。
- 要求を受け付けたフロントエンドサーバがあるユニットの監査証跡ファイルに出力されます。

注※1

SQL 通番は，SDB データベースを操作する API または DML の要求通番と読み替えてください。また，複数要求の場合は，SDB データベース操作の要求単位ごとに同じ通番となります。

注※2

オブジェクト種別には TBL が，オブジェクト名称にはレコード名が設定されます。

7.12.8　レコードの一括削除の場合【4V DAM，4V SAM】

監査証跡のレコード項目（レコードの一括削除の場合）を次の表に示します。

表 7-32　監査証跡のレコード項目（レコードの一括削除の場合）

項番	監査証跡のレコード項目	イベントの終了
1	ユーザ識別子	○
2	イベント実行日	○
3	イベント実行時刻	○
4	イベント実行時刻（マイクロ秒）	○
5	イベントタイプ	SDA
6	イベントサブタイプ	CLA
7	イベント成否	○
8	使用した権限	△△△
9	UAP 名称	○
10	サービス名称	○
11	IP アドレス	○
12	プロセス ID	○
13	スレッド ID	○
14	ホスト名	○
15	ユニット識別子	○
16	サーバ名称	○
17	コネクト通番	○
18	SQL 通番※1	○
19	オブジェクトの所有者名	データベース名
20	オブジェクト名称※2	○
21	オブジェクト種別※2	SCH
22	付与，削除，変更した権限	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−
24	監査関連オペランドの定義値	−
25	監査証跡種別	E
26	SQL コードまたは終了コード	○
27	スワップ元監査証跡ファイル名称	−
28	スワップ先監査証跡ファイル名称	−
29	セキュリティ関連変更種別	−

項番	監査証跡のレコード項目	イベントの終了
30	セキュリティ関連変更前設定値	−
31	セキュリティ関連変更後設定値	−
32	監査証跡表オプション	−
33	アクセス件数	○
34	SQL 文	○
35	SQL データ	○
36	ユーザ付加情報 1	○
37	ユーザ付加情報 2	○
38	ユーザ付加情報 3	○
39	関連製品付加情報 1	−
40	監査証跡表へのデータ登録時刻印	○
41	監査証跡表へのデータ登録時の通番	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- SDB データベース操作の要求単位に 1 レコードが記録されます。
- SDB データベース操作の延長で実行される SQL 文はありません。
- 要求を受け付けたフロントエンドサーバがあるユニットの監査証跡ファイルに出力されます。

注※1

SQL 通番は，SDB データベースを操作する API の要求通番と読み替えてください。

注※2

オブジェクト種別には SCH が，オブジェクト名称にはデータベース名が設定されます。

7.12.9 複数レコードの検索（FETCHDB ALL）の場合【4V FMB】

監査証跡のレコード項目（複数レコードの検索（FETCHDB ALL）の場合）を次の表に示します。

表 7-33 監査証跡のレコード項目（複数レコードの検索（FETCHDB ALL）の場合）

項番	監査証跡のレコード項目	イベントの終了
1	ユーザ識別子	○

項番	監査証跡のレコード項目	イベントの終了
2	イベント実行日	○
3	イベント実行時刻	○
4	イベント実行時刻（マイクロ秒）	○
5	イベントタイプ	SDA
6	イベントサブタイプ	FTA
7	イベント成否	○
8	使用した権限	△△△
9	UAP 名称	○
10	サービス名称	○
11	IP アドレス	○
12	プロセス ID	○
13	スレッド ID	○
14	ホスト名	○
15	ユニット識別子	○
16	サーバ名称	○
17	コネクト通番	○
18	SQL 通番※1	○
19	オブジェクトの所有者名	データベース名
20	オブジェクト名称※2	○
21	オブジェクト種別※2	SCH
22	付与，削除，変更した権限	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−
24	監査関連オペランドの定義値	−
25	監査証跡種別	E
26	SQL コードまたは終了コード	○
27	スワップ元監査証跡ファイル名称	−
28	スワップ先監査証跡ファイル名称	−
29	セキュリティ関連変更種別	−
30	セキュリティ関連変更前設定値	−
31	セキュリティ関連変更後設定値	−

項番	監査証跡のレコード項目	イベントの終了
32	監査証跡表オプション	−
33	アクセス件数	○
34	SQL 文	○
35	SQL データ	○
36	ユーザ付加情報 1	○
37	ユーザ付加情報 2	○
38	ユーザ付加情報 3	○
39	関連製品付加情報 1	−
40	監査証跡表へのデータ登録時刻印	○
41	監査証跡表へのデータ登録時の通番	○

（凡例）

○：情報を取得します。

−：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- SDB データベース操作の要求単位に 1 レコードが記録されます。
- SDB データベース操作の延長で実行される SQL 文はありません。
- 要求を受け付けたフロントエンドサーバがあるユニットの監査証跡ファイルに出力されます。

注※1

SQL 通番は，SDB データベースを操作する API の要求通番と読み替えてください。

注※2

オブジェクト種別には SCH が，オブジェクト名称にはデータベース名が設定されます。

7.12.10　レコードの取得（GET）の場合【SD FMB】

監査証跡のレコード項目（レコードの取得（GET）の場合）を次の表に示します。

表 7-34　監査証跡のレコード項目（レコードの取得（GET）の場合）

項番	監査証跡のレコード項目	イベントの終了
1	ユーザ識別子	○
2	イベント実行日	○
3	イベント実行時刻	○

項番	監査証跡のレコード項目	イベントの終了
4	イベント実行時刻（マイクロ秒）	○
5	イベントタイプ	SDA
6	イベントサブタイプ	GET
7	イベント成否	○
8	使用した権限	△△△
9	UAP 名称	○
10	サービス名称	○
11	IP アドレス	○
12	プロセス ID	○
13	スレッド ID	○
14	ホスト名	○
15	ユニット識別子	○
16	サーバ名称	○
17	コネクト通番	○
18	SQL 通番※1	○
19	オブジェクトの所有者名	データベース名
20	オブジェクト名称※2	○
21	オブジェクト種別※2	TBL
22	付与，削除，変更した権限	−
23	権限を付与，削除，変更されたユーザ識別子，またはイベント対象のユーザ識別子	−
24	監査関連オペランドの定義値	−
25	監査証跡種別	E
26	SQL コードまたは終了コード	○
27	スワップ元監査証跡ファイル名称	−
28	スワップ先監査証跡ファイル名称	−
29	セキュリティ関連変更種別	−
30	セキュリティ関連変更前設定値	−
31	セキュリティ関連変更後設定値	−
32	監査証跡表オプション	−
33	アクセス件数	○

項番	監査証跡のレコード項目	イベントの終了
34	SQL 文	○
35	SQL データ	○
36	ユーザ付加情報 1	○
37	ユーザ付加情報 2	○
38	ユーザ付加情報 3	○
39	関連製品付加情報 1	－
40	監査証跡表へのデータ登録時刻印	○
41	監査証跡表へのデータ登録時の通番	○

（凡例）

○：情報を取得します。

－：情報を取得しません。または該当しません。

△：半角空白を意味しています。

注

- SDB データベース操作の要求単位に 1 レコードが記録されます。
- SDB データベース操作の延長で実行される SQL 文はありません。
- 要求を受け付けたフロントエンドサーバがあるユニットの監査証跡ファイルに出力されます。

注※1

SQL 通番は，SDB データベースを操作する DML の要求通番と読み替えてください。

注※2

オブジェクト種別には TBL が，オブジェクト名称にはレコード名が設定されます。

7.13　ユティリティ実行時の監査証跡の出力先ユニット

ユティリティ実行時の監査証跡の出力先ユニットについては，マニュアル「HiRDB システム運用ガイド」の「ユティリティ実行時の監査証跡の出力先ユニット」を参照してください。

7.14　バージョンアップ時の注意事項

バージョンアップ時の注意事項については，マニュアル「HiRDB システム運用ガイド」の「セキュリティ監査機能の運用」の「バージョンアップ時の注意事項」を参照してください。

7.15 注意事項

セキュリティ監査機能に関する注意事項を説明します。

なお，説明文中の SDB ユティリティとは，次のユティリティの総称を意味しています。

- HiRDB/SD 定義ユティリティ（pdsdbdef）
- HiRDB/SD データベース作成ユティリティ（pdsdblod）
- HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

注意事項

- SDB ユティリティの実行中に，pdaudbegin コマンド（監査証跡を取得開始するコマンド）を実行しても，実行中の SDB ユティリティに対しては有効になりません。
- SDB ユティリティの実行中に，定義系 SQL の CREATE AUDIT 文または DROP AUDIT 文を実行しても，実行中の SDB ユティリティに対しては有効になりません。
- SDB ユティリティの実行中に，pdaudend コマンド（監査証跡の取得をやめるコマンド）を実行した場合，SDB ユティリティでエラーメッセージが出力され，監査証跡は取得されません。
- SDB ユティリティ実行時の CONNECT 権限チェックで，CONNECT エラーが発生した場合，すべての監査証跡が取得されません。このため，CONNECT 関連セキュリティ機能の「連続認証失敗回数の制限」を使用することを推奨します。CONNECT 関連セキュリティ機能については，マニュアル「HiRDB システム運用ガイド」の「CONNECT 関連セキュリティ機能の運用」を参照してください。

8 ディザスタリカバリシステムの構築と運用【4V FMB，4V AFM】

この章では，ディザスタリカバリシステムの構築方法と運用方法について説明します。

8.1　ディザスタリカバリシステムの概要

ここでは，ディザスタリカバリシステムの概要，および前提製品について説明します。

ディザスタリカバリシステムは，4V FMB または 4V AFM の SDB データベースの場合に使用できます。

8.1.1　ディザスタリカバリシステムとは

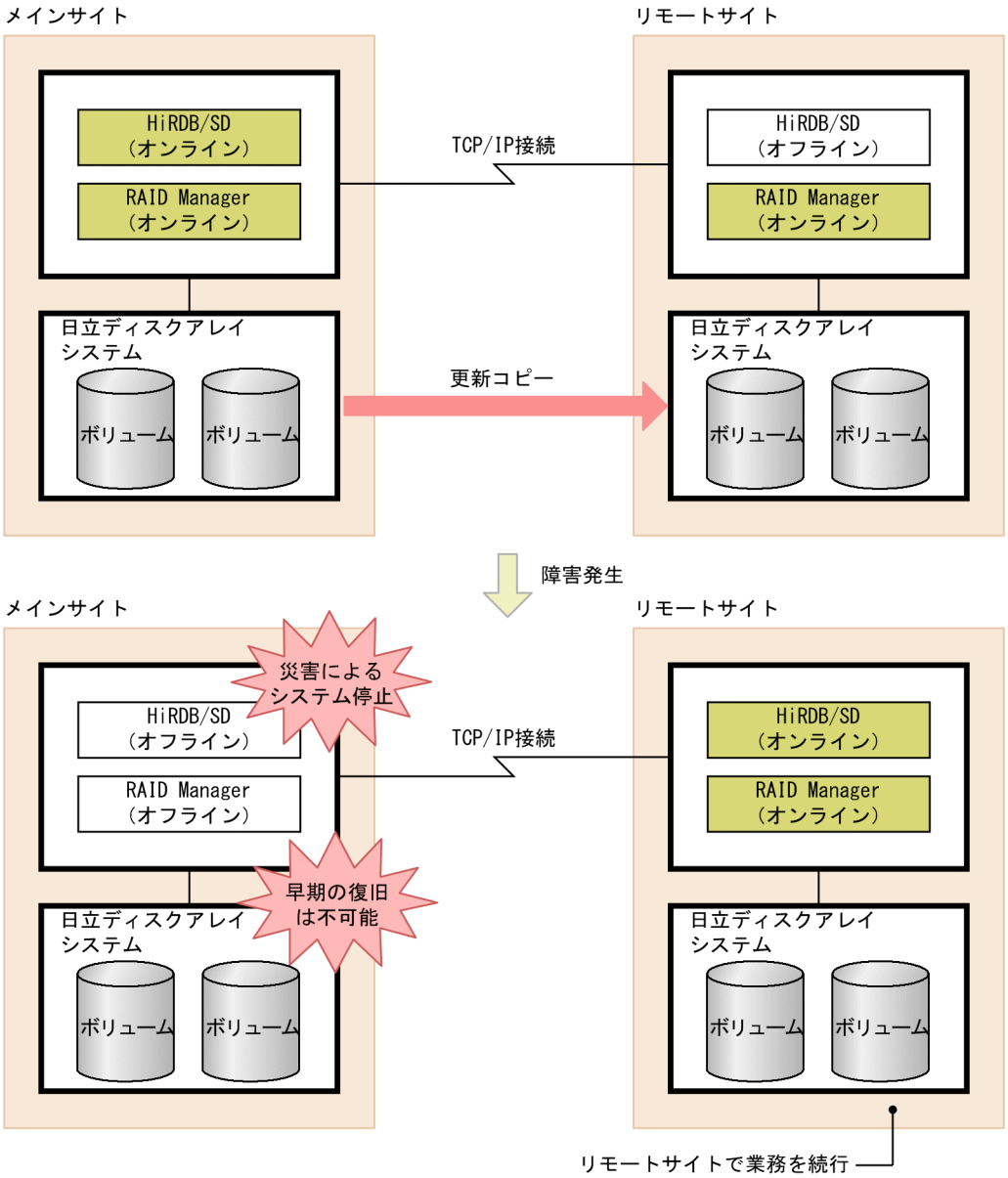
(1)　機能概要

通常使用しているシステムが地震，火災などの災害によって物理的に回復困難な状況になった場合，遠隔地に準備している予備のシステムを稼働して業務を続行できます。このシステムをディザスタリカバリシステム（リアルタイム SAN レプリケーション（RiSe））といいます。なお，通常使用しているシステムがあるサイトをメインサイトといい，遠隔地に準備している予備のシステムがあるサイトをリモートサイトといいます。

メインサイトおよびリモートサイトのデータは，日立ディスクアレイサブシステム上に配置し，メインサイトのデータに更新が発生した場合，日立ディスクアレイサブシステムの Universal Replicator または TrueCopy を使用してリモートサイトに反映（更新コピー）します。

ディザスタリカバリシステムの概要を次の図に示します。

図 8-1　ディザスタリカバリシステムの概要



[説明]

通常はメインサイトの HiRDB を使用して業務を行います。メインサイトのファイルに更新が発生した場合，その更新内容をリモートサイトに更新コピーします。更新コピーによってメインサイトとリモートサイトで同じデータを持ちます。

メインサイトで地震や火災などの大規模災害が発生し，メインサイトのシステムが早期に復旧できない場合，リモートサイトで HiRDB を再開始して業務を続行できます。

参考

- 更新コピーは，Universal Replicator または TrueCopy が行います。Universal Replicator および TrueCopy は，ホストを経由しないで，日立ディスクアレイサブシステム間で直接データをコピーします。

- RAID Manager とは，日立ディスクアレイサブシステムの付加プログラムプロダクトで，Universal Replicator および TrueCopy を制御，運用するコマンドなどを提供しています。

(2)　更新コピーの対象になるファイル

更新コピーの対象になるファイルを次に示します。これらのファイルに更新が発生すると，リモートサイトの同じファイルに更新情報がコピーされます。

- データベースファイル（RD エリアを構成する HiRDB ファイル）
- システムログファイル
- シンクポイントダンプファイル
- ステータスファイル

注意事項

SDB ディレクトリ情報ファイルは更新コピーの対象にはなりません。そのため，SDB データベースの定義を追加，変更，または削除した場合は，SDB ディレクトリ情報ファイルをリモートサイトに転送する必要があります。

(3)　非同期コピーの特徴

更新コピー処理は非同期コピーになります。非同期コピーの特徴を次に示します。

- 処理方式
  リモートサイトの更新処理の完了を待たないでメインサイトの更新処理を完了します。
- メインサイトとリモートサイトのデータの整合性
  データ欠損が発生することがあります。そのため，メインサイトとリモートサイトでデータが一致しないことがあります。
- 性能への影響※
  性能への影響はありません。
  注※　Universal Replicator および TrueCopy が保証する理論値に従った特徴です。

8.1.2　サポートしているリモートサイトへのデータ反映方式（全非同期方式）

メインサイトからリモートサイトにデータを反映するときの処理方式は，全非同期方式になります。

全非同期方式とは，リモートサイトへの更新コピー処理が非同期コピーで行われる方式のことです。非同期コピーの場合，リモートサイトの更新処理の完了を待たないでメインサイトの更新処理を完了するため，メインサイトのトランザクション性能に影響を与えません。

ただし，メインサイトのファイル（更新コピーの対象ファイル）の更新内容がリモートサイトに反映されないことがあります。そのため，メインサイトの HiRDB が災害などによって異常終了し，リモートサイトで HiRDB を再開始した場合，再開始したときの状態が異常終了直前の状態と異なることがあります。したがって，全非同期方式の場合，メインサイトから継続したサービスの開始を保証できません。

全非同期方式の概要を次の図に示します。また，リモートサイトに更新コピーするときの処理方式（全非同期方式の場合）を次の表に示します。

図 8-2　全非同期方式の概要

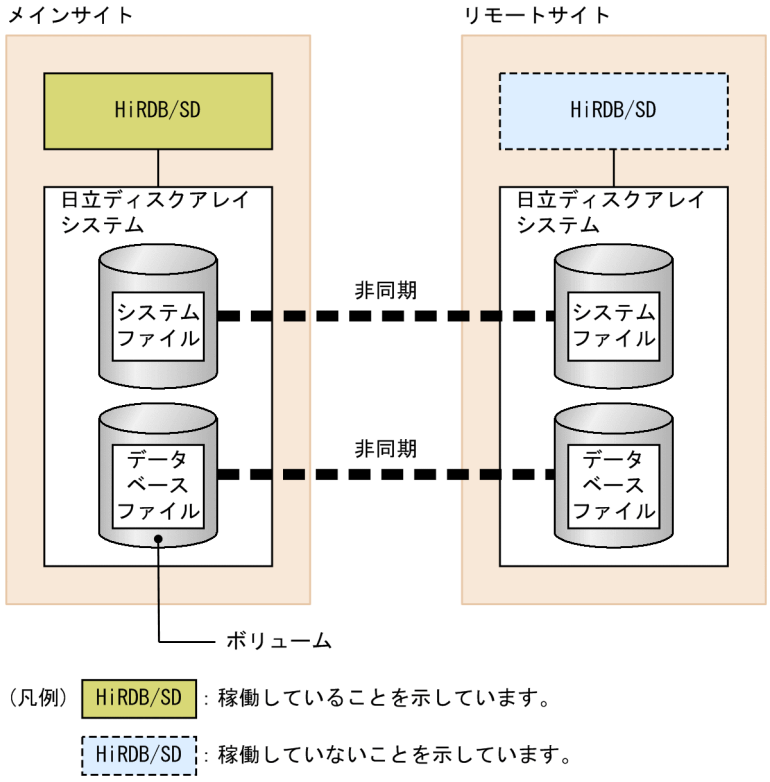


表 8-1　リモートサイトに更新コピーするときの処理方式（全非同期方式の場合）

リモートサイトにコピーされるファイル		更新コピーするときの処理方式
データベースファイル		非同期コピー
システムファイル	システムログファイル	
	シンクポイントダンプファイル	
	ステータスファイル	

全非同期方式の特徴については，マニュアル「HiRDB ディザスタリカバリシステム　構築・運用ガイド」の「各処理方式の特徴」を参照してください。

参考

マニュアル「HiRDB ディザスタリカバリシステム　構築・運用ガイド」には，全非同期方式以外の処理方式（全同期方式，ハイブリッド方式，およびログ同期方式）についても説明されていますが，これらの処理方式は使用できません。

8.1.3　前提製品

日立ディスクアレイサブシステムシリーズの製品が必要となります。前提製品を次の表に示します。なお，これらの前提製品はメインサイトとリモートサイトの両方に必要です。

表 8-2　前提製品

日立ディスクアレイサブシステムの機器名称	必要となる日立ディスクアレイサブシステム
Hitachi Unified Storage VM(HUSVM)	• Universal Replicator • TrueCopy • RAID Manager

8.2　ディザスタリカバリシステムの設計

ディザスタリカバリシステムを設計する際の考慮点については，マニュアル「HiRDB ディザスタリカバリシステム　構築・運用ガイド」の「第 2 編　全同期方式，全非同期方式，およびハイブリッド方式編」の「システム設計時の考慮点」を参照してください。

ただし，マニュアル「HiRDB ディザスタリカバリシステム　構築・運用ガイド」に記載されている内容とは仕様差があります。ここでは，その仕様差についてだけ説明します。

仕様差のある個所

マニュアル「HiRDB ディザスタリカバリシステム　構築・運用ガイド」の「HiRDB の環境設定時の考慮点」の「メインサイトとリモートサイトで一致させる項目」

HiRDB/SD を使用する場合の注意点（仕様差）

HiRDB/SD を使用する場合は，上記個所に記載されている「メインサイトとリモートサイトで一致させる項目」に加えて，次の点に注意してください。

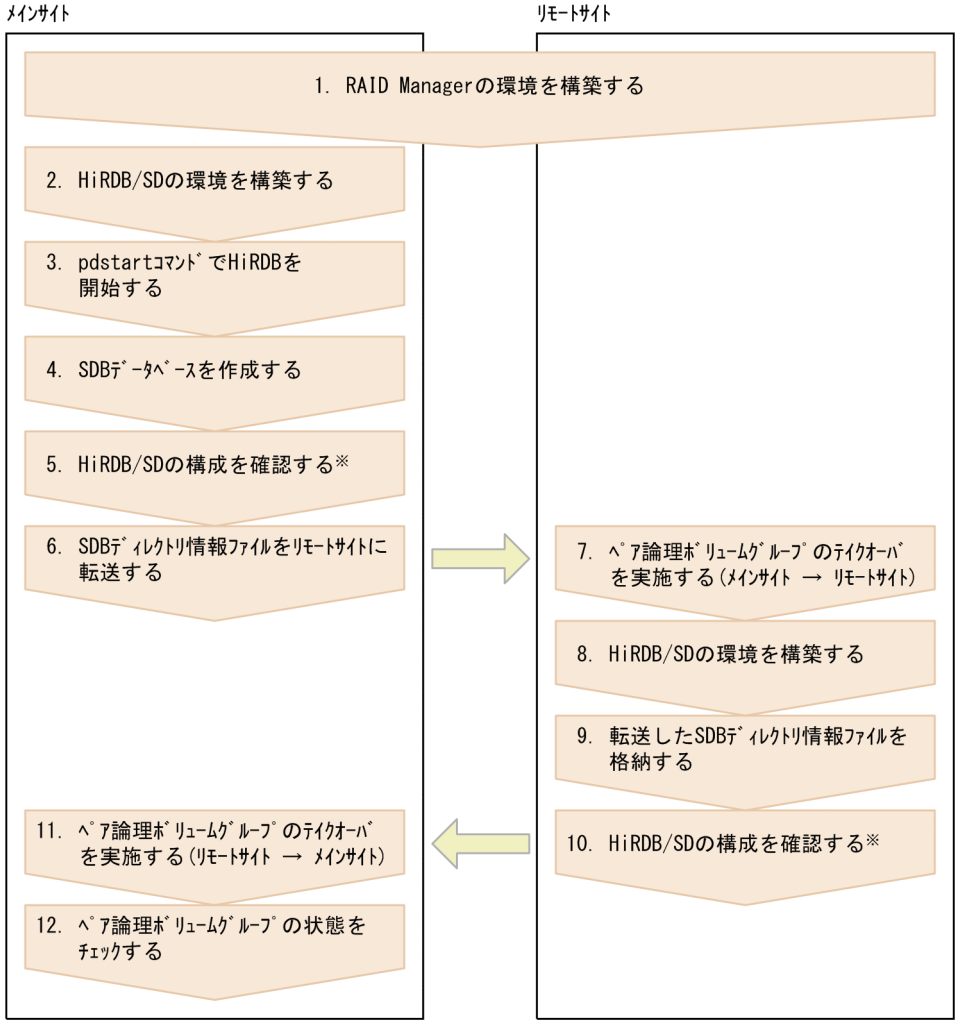
- メインサイトとリモートサイトが HiRDB/SD である必要があります。

8.3　ディザスタリカバリシステムの構築

8.3.1　ディザスタリカバリシステムの構築手順

ディザスタリカバリシステムの構築手順を次の図に示します。

図 8-3　ディザスタリカバリシステムの構築手順



注※

pdconfchk および pdrisechk コマンドでは，メインサイトとリモートサイトで同じ SDB ディレクトリ情報ファイルが格納されているかどうかはチェックできません。

ここでは，次の作業についてだけ説明します。

- 「4. SDB データベースを作成する」
- 「6. SDB ディレクトリ情報ファイルをリモートサイトに転送する」
- 「9. 転送した SDB ディレクトリ情報ファイルを格納する」

上記以外の作業については，マニュアル「HiRDB ディザスタリカバリシステム　構築・運用ガイド」の「第 2 編　全同期方式，全非同期方式，およびハイブリッド方式編」の「システムの構築」を参照してください。

(1) 「4. SDB データベースを作成する」の作業

作業手順については，「4.5　SDB データベースの作成」を参照してください。

(2) 「6. SDB ディレクトリ情報ファイルをリモートサイトに転送する」の作業

SDB ディレクトリ情報ファイルをメインサイトからリモートサイトの全サーバマシンに転送します。SDB ディレクトリ情報ファイルの転送は，OS の rcp コマンドなどを使用して，ユーザが手動で行う必要があります。ftp コマンドを使用する場合は，必ずバイナリモードで転送してください。

SDB ディレクトリ情報ファイルの転送後，SDB ディレクトリ情報ファイルが正しく転送されたかを確認してください。メインサイトの SDB ディレクトリ情報ファイルと，リモートサイトの SDB ディレクトリ情報ファイルの次に示す情報が同じであることを，OS の cksum コマンドで確認してください。

- CRC チェックサム
- バイト数

異なる場合は，SDB ディレクトリ情報ファイルを再転送してください。

(3) 「9. 転送した SDB ディレクトリ情報ファイルを格納する」の作業

リモートサイトの全ユニットのユニット制御定義の pd_structured_directory_path オペランドに指定したディレクトリ下に，メインサイトから転送した SDB ディレクトリ情報ファイルを格納します。

8.4　ディザスタリカバリシステムの運用

ディザスタリカバリシステムの運用方法については，マニュアル「HiRDB ディザスタリカバリシステム構築・運用ガイド」の「第 2 編　全同期方式，全非同期方式，およびハイブリッド方式編」の次の個所を参照してください。

- 「メインサイトの運用」
- 「リモートサイトへの切り替え」
- 「ペア論理ボリュームの構成変更」
- 「ほかの機能との関連」

なお，SDB データベースの定義追加，定義変更，または定義削除を実行して SDB ディレクトリ情報ファイルを更新した場合，更新後の SDB ディレクトリ情報ファイルをリモートサイトに転送する必要があります。

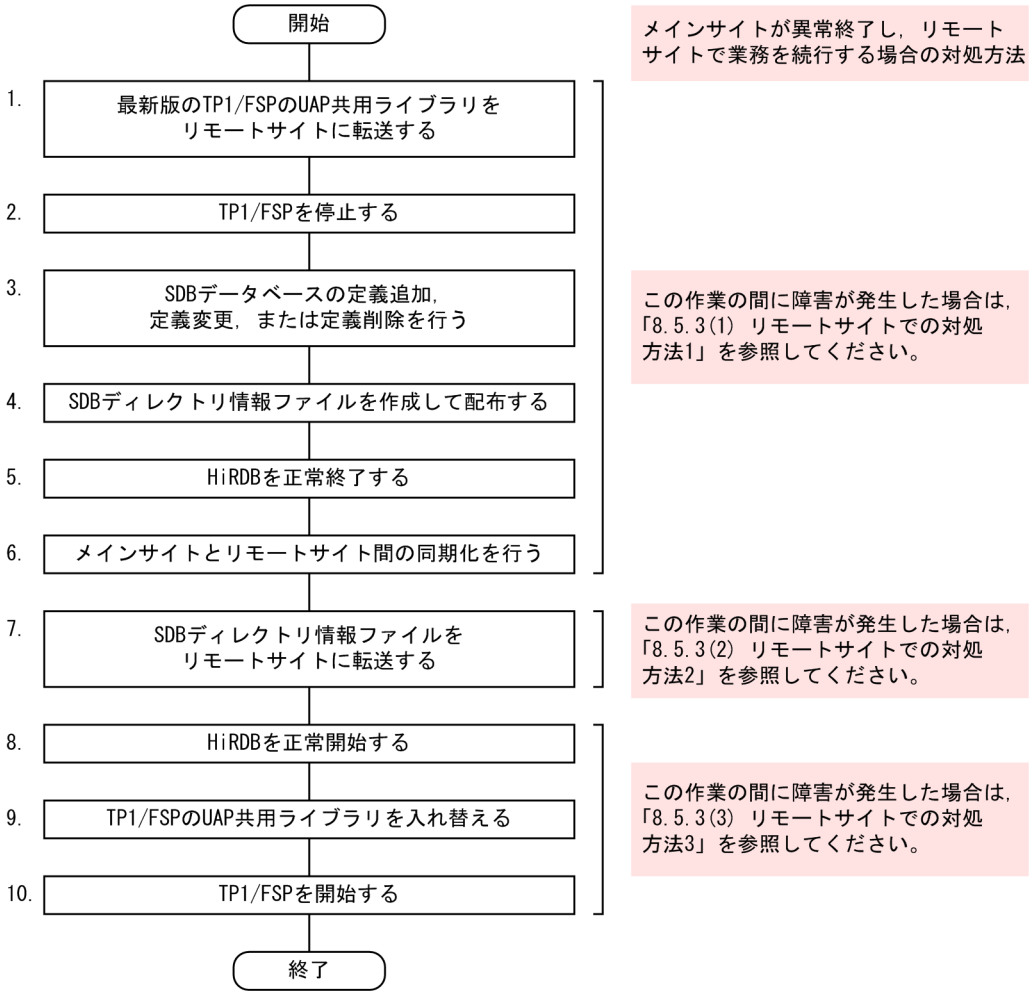
8.5　SDB データベースの定義追加，定義変更，または定義削除するときの運用（HiRDB の再起動を必要とする場合）

ここでは，ディザスタリカバリシステムを適用している場合の，HiRDB の再起動を必要とする SDB データベースの定義追加，定義変更，または定義削除の手順について説明します。

8.5.1　SDB データベースの定義追加，定義変更，または定義削除の手順（概要）

ディザスタリカバリシステムを適用している場合の，SDB データベースの定義追加，定義変更，または定義削除する手順（概要）を次の図に示します。

図 8-4　SDB データベースの定義追加，定義変更，または定義削除する手順（概要）



注

図中の項番は，「8.5.2　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）」の（　）の番号と対応しています。

8.5.2　SDB データベースの定義追加，定義変更，または定義削除の手順（各手順の詳細）

ディザスタリカバリシステムを適用している場合の，SDB データベースの定義追加，定義変更，または定義削除をする際の各手順の詳細について説明します。

(1)　最新版の TP1/FSP の UAP 共用ライブラリをリモートサイトに転送する

OS の rcp コマンドなどを使用して，最新版の TP1/FSP の UAP 共用ライブラリをリモートサイトに転送します。

(2)　TP1/FSP を停止する

TP1/FSP の eesvstop コマンドで TP1/FSP を停止します。この操作によってオンラインが停止状態になります。

(3)　SDB データベースの定義追加，定義変更，または定義削除を行う

pdsdbdef コマンドで，SDB データベースの定義追加，定義変更，または定義削除を行います。

(4)　SDB ディレクトリ情報ファイルを作成して配布する

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して全ユニットに配布します。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1　SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

参考

(3)および(4)の作業は，1 回の pdsdbdef コマンドで実行できます。

(5)　HiRDB を正常終了する

pdstop コマンドで HiRDB を正常終了します。必ず正常終了させてください。

(6)　メインサイトとリモートサイト間の同期化を行う

RD エリアが配置されているペア論理グループに対して，RAID Manager の pairsyncwait コマンド（-g aaaa_ALL -t 600 オプション指定）を実行し，メインサイトとリモートサイト間の同期化を行います。aaaa には，HiRDB 識別子を指定します。

pairsyncwait コマンドが正常終了し，実行結果が DONE となることを確認してください。

(7)　SDB ディレクトリ情報ファイルをリモートサイトに転送する

SDB ディレクトリ情報ファイルをリモートサイトに転送します。転送後，リモートサイトの全ユニットの次のディレクトリ下に SDB ディレクトリ情報ファイルを格納してください。

- pd_structured_directory_path オペランドに指定したディレクトリ

(8)　HiRDB を正常開始する

pdstart コマンドで HiRDB を正常開始します。SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報が常用常駐領域に常駐されます。

(9)　TP1/FSP の UAP 共用ライブラリを入れ替える

TP1/EE サービス定義の module_dir オペランドに指定した UAP 共用ライブラリ格納ディレクトリに，新しい UAP 共用ライブラリを格納します。または，module_dir オペランドの指定を，新しい UAP 共用ライブラリを格納したディレクトリに変更します。

(10)　TP1/FSP を開始する

TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

このとき，常用常駐領域中の SDB ディレクトリ情報を入力情報にして，TP1/FSP の SDB 定義情報領域に SDB 定義情報が格納されます。

この作業が完了すると，定義追加，定義変更，または定義削除した SDB データベースにアクセスできるようになります。

8.5.3　障害が発生したときの対処方法

メインサイトが異常終了し，リモートサイトで業務を続行する場合の対処方法について説明します。

障害が発生した時点によって対処方法が異なります。「図 8-4　SDB データベースの定義追加，定義変更，または定義削除する手順（概要）」を参照して，障害発生時点と選択する対処方法を確認してください。

(1)　リモートサイトでの対処方法 1

リモートサイトで次の対処をしてください。

手順

1. RAID Manager の pairvolchk コマンドで，ペア論理ボリュームグループのステータスを確認します。
2. RAID Manager の horctakeover コマンドで，ペア論理ボリュームグループのテイクオーバを実施します。

3. pdstart コマンドで HiRDB を開始します。
4. pdsdbdef コマンドで，SDB データベースの定義追加，定義変更，または定義削除を行います。
5. pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して全ユニットに配布します。
6. pdstop コマンドで HiRDB を終了します。
7. pdstart コマンドで HiRDB を開始します。
8. TP1/FSP の UAP 共用ライブラリを最新版に入れ替えます。
9. TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

(2) リモートサイトでの対処方法 2

リモートサイトで次の対処をしてください。

手順

1. RAID Manager の pairvolchk コマンドで，ペア論理ボリュームグループのステータスを確認します。
2. RAID Manager の horctakeover コマンドで，ペア論理ボリュームグループのテイクオーバを実施します。
3. pdstart コマンドで HiRDB を開始します。
4. pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して全ユニットに配布します。
5. pdstop コマンドで HiRDB を終了します。
6. pdstart コマンドで HiRDB を開始します。
7. TP1/FSP の UAP 共用ライブラリを最新版に入れ替えます。
8. TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

(3) リモートサイトでの対処方法 3

リモートサイトで次の対処をしてください。

手順

1. RAID Manager の pairvolchk コマンドで，ペア論理ボリュームグループのステータスを確認します。
2. RAID Manager の horctakeover コマンドで，ペア論理ボリュームグループのテイクオーバを実施します。
3. pdstart コマンドで HiRDB を開始します。
4. TP1/FSP の UAP 共用ライブラリを最新版に入れ替えます。
5. TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

8.6　SDB データベースの定義追加または定義変更するときの運用（HiRDB の再起動を必要としない場合）

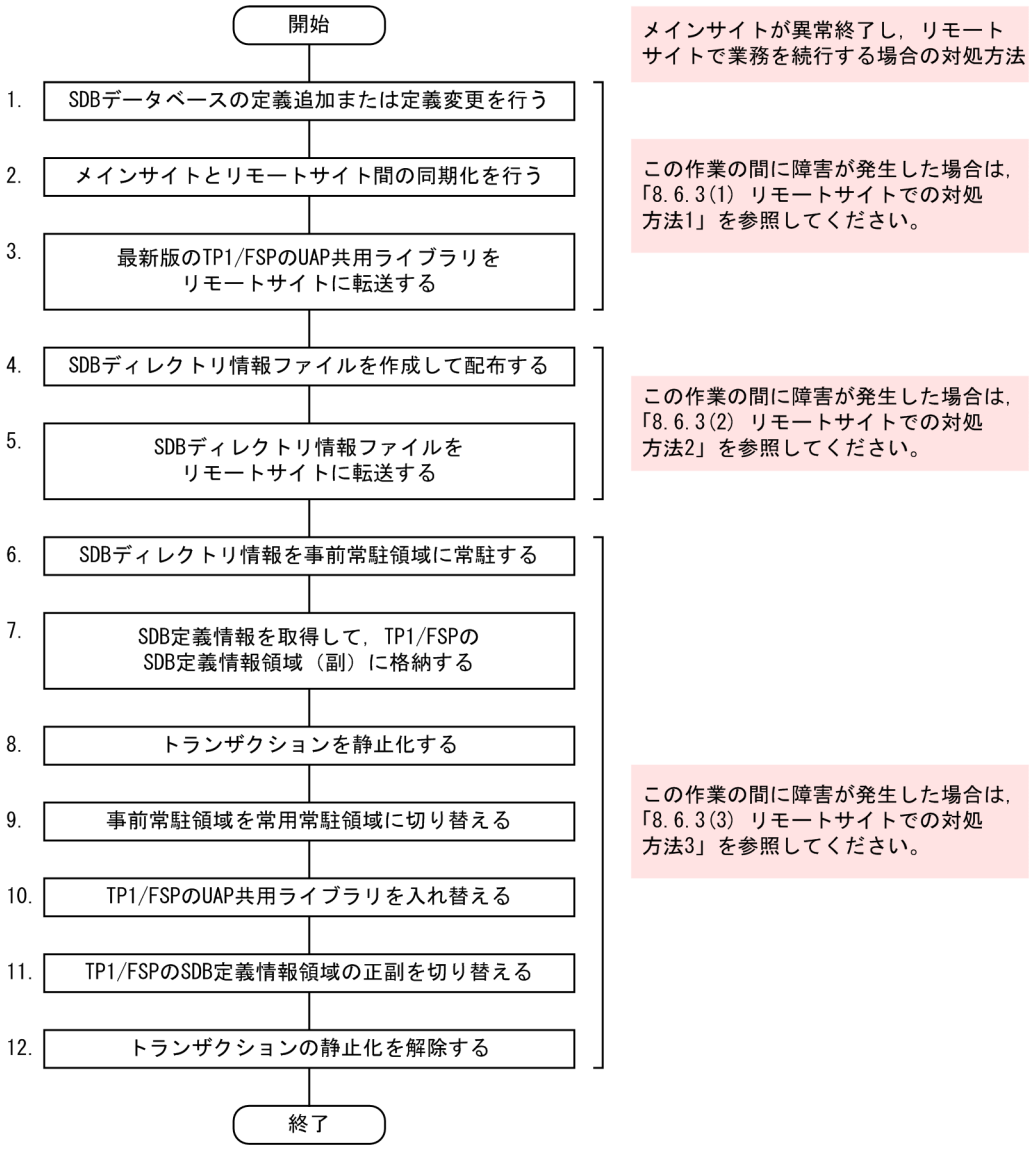
ここでは，ディザスタリカバリシステムを適用している場合の，HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更の手順について説明します。

前提条件および事前準備については，「5.7　SDB データベースの定義追加または定義変更（HiRDB の再起動を必要としない場合）」を参照してください。

8.6.1　SDB データベースの定義追加または定義変更の手順（概要）

ディザスタリカバリシステムを適用している場合の，SDB データベースの定義追加または定義変更する手順（概要）を次の図に示します。

図 8-5　SDB データベースの定義追加または定義変更する手順（概要）



注

図中の項番は，「8.6.2　SDB データベースの定義追加または定義変更の手順（各手順の詳細）」の（　）の番号と対応しています。

8.6.2　SDB データベースの定義追加または定義変更の手順（各手順の詳細）

ディザスタリカバリシステムを適用している場合の，SDB データベースの定義追加または定義変更をする際の各手順の詳細について説明します。

(1)　SDB データベースの定義追加または定義変更を行う

pdsdbdef コマンドで，SDB データベースの定義追加または定義変更を行います。

(2)　メインサイトとリモートサイト間の同期化を行う

RD エリアが配置されているペア論理グループに対して，RAID Manager の pairsyncwait コマンド（-g aaaa_ALL -t 600 オプション指定）を実行し，メインサイトとリモートサイト間の同期化を行います。aaaa には，HiRDB 識別子を指定します。

pairsyncwait コマンドが正常終了し，実行結果が DONE となることを確認してください。

(3)　最新版の TP1/FSP の UAP 共用ライブラリをリモートサイトに転送する

OS の rcp コマンドなどを使用して，最新版の TP1/FSP の UAP 共用ライブラリをリモートサイトに転送します。

(4)　SDB ディレクトリ情報ファイルを作成して配布する

pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して全ユニットに配布します。SDB ディレクトリ情報ファイルの配布方法の詳細については，「5.8.1　SDB ディレクトリ情報ファイルを作成して配布する場合」を参照してください。

(5)　SDB ディレクトリ情報ファイルをリモートサイトに転送する

SDB ディレクトリ情報ファイルをリモートサイトに転送します。転送後，リモートサイトの全ユニットの次のディレクトリ下に SDB ディレクトリ情報ファイルを格納してください。

- pd_structured_directory_path オペランドに指定したディレクトリ

(6)　SDB ディレクトリ情報を事前常駐領域に常駐する

pdsdbarc -e コマンドで，SDB ディレクトリ情報を事前常駐領域に常駐します。

pdsdbarc -e コマンドの実行時，HiRDB/SD は，事前常駐領域に常駐した SDB ディレクトリ情報の最終更新日時などをチェックします。チェックの結果，最終更新日時の不整合を検知した場合，不整合を検知した SDB ディレクトリ情報を無効状態にします。無効状態の SDB ディレクトリ情報は，(9)の操作で事前常駐領域を常用常駐領域に切り替える際，切り替えの対象外となります。切り替えの対象外となった SDB ディレクトリ情報を使用しているユニット（フロントエンドサーバまたはバックエンドサーバ）を経由する API はエラーになります。そのため，最終更新日時の不整合が検知された場合，出力されたメッセージに従って対処し，事前常駐領域の SDB ディレクトリ情報を正常な状態にしてください。

(7)　SDB 定義情報を取得して，TP1/FSP の SDB 定義情報領域（副）に格納する

TP1/FSP の eesdhchg -b コマンドで SDB 定義情報を取得し，SDB 定義情報領域（副）に格納します（事前常駐領域中の SDB ディレクトリ情報が入力情報になります）。

SDB 定義情報を取得する方法については，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「SDB 定義情報の事前取得」を参照してください。

(8)　トランザクションを静止化する

TP1/FSP の eesvctl -s コマンドで，トランザクションを静止化します。eesvctl -s コマンドが正常終了して，完全にトランザクションが静止化したことを確認してから次の手順に進んでください。

注意事項

完全にトランザクションが静止化していない状態で次の手順を行った場合，データベース破壊や UAP の異常終了などが発生するおそれがあります。そのため，eesvctl -s コマンドが正常終了するまで，eesvctl -s を実行し続けてください。

(9)　事前常駐領域を常用常駐領域に切り替える

pdsdbarc -w -q コマンドで，事前常駐領域を常用常駐領域に切り替えます。pdsdbarc -w -q コマンドはユニットごとに実行します。この操作は全ユニットで実行してください。

事前常駐領域中の SDB ディレクトリ情報が無効状態の場合，事前常駐領域を常用常駐領域に切り替えることはできません。切り替えられなかった SDB ディレクトリ情報を使用しているユニット（フロントエンドサーバまたはバックエンドサーバ）を経由する API はエラーになるため，無効状態の SDB ディレクトリ情報を次に示すどちらかの方法で正常な状態にしてください。

- 正しい SDB ディレクトリ情報ファイルが配布済みの場合
  事前常駐領域中の SDB ディレクトリ情報が無効状態のユニットに対して，「(6)　SDB ディレクトリ情報を事前常駐領域に常駐する」から作業を再実行してください。
- 正しい SDB ディレクトリ情報ファイルが未配布の場合

事前常駐領域中の SDB ディレクトリ情報が無効状態のユニットに対して，pdsdbdef コマンドで SDB ディレクトリ情報ファイルを配布し直してください。そのあと，「(6)　SDB ディレクトリ情報を事前常駐領域に常駐する」から作業を再実行してください。

SDB ディレクトリ情報ファイルの配布方法については，「5.8.2　SDB ディレクトリ情報ファイルの配布だけをする場合」を参照してください。

(10)　TP1/FSP の UAP 共用ライブラリを入れ替える

TP1/FSP の eechglib コマンドで，TP1/FSP の UAP 共用ライブラリを入れ替えます。

(11)　TP1/FSP の SDB 定義情報領域の正副を切り替える

TP1/FSP の eesdhchg -c コマンドで，SDB 定義情報領域の正副を切り替えます。この操作によって，定義追加または定義変更した SDB データベースに対して，UAP からアクセスできるようになります。

(12)　トランザクションの静止化を解除する

TP1/FSP の eesvctl -r コマンドで，トランザクションの静止化を解除します。

この作業が完了すると，定義追加または定義変更した SDB データベースにアクセスできるようになります。

8.6.3　障害が発生したときの対処方法

メインサイトが異常終了し，リモートサイトで業務を続行する場合の対処方法について説明します。

障害が発生した時点によって対処方法が異なります。「図 8-5　SDB データベースの定義追加または定義変更する手順（概要）」を参照して，障害発生時点と選択する対処方法を確認してください。

(1)　リモートサイトでの対処方法 1

リモートサイトで次の対処をしてください。

手順

1. RAID Manager の pairvolchk コマンドで，ペア論理ボリュームグループのステータスを確認します。
2. RAID Manager の horctakeover コマンドで，ペア論理ボリュームグループのテイクオーバを実施します。
3. pdstart コマンドで HiRDB を開始します。
4. TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

SDB ディレクトリ情報ファイルは，リモートサイトに転送済みのファイルを使用してください。

(2) リモートサイトでの対処方法 2

リモートサイトで次の対処をしてください。

手順

1. RAID Manager の pairvolchk コマンドで，ペア論理ボリュームグループのステータスを確認します。
2. RAID Manager の horctakeover コマンドで，ペア論理ボリュームグループのテイクオーバを実施します。
3. pdstart コマンドで HiRDB を開始します。
4. pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを作成して全ユニットに配布します。
5. pdstop コマンドで HiRDB を終了します。
6. pdstart コマンドで HiRDB を開始します。
7. TP1/FSP の UAP 共用ライブラリを最新版に入れ替えます。
8. TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

(3) リモートサイトでの対処方法 3

リモートサイトで次の対処をしてください。

手順

1. RAID Manager の pairvolchk コマンドで，ペア論理ボリュームグループのステータスを確認します。
2. RAID Manager の horctakeover コマンドで，ペア論理ボリュームグループのテイクオーバを実施します。
3. pdstart コマンドで HiRDB を開始します。
4. TP1/FSP の UAP 共用ライブラリを最新版に入れ替えます。
5. TP1/FSP の eesvstart コマンドで TP1/FSP を開始します。

8.7　障害発生時の対処方法

障害発生時の対処方法については，マニュアル「HiRDB ディザスタリカバリシステム　構築・運用ガイド」の「第 2 編　全同期方式，全非同期方式，およびハイブリッド方式編」の「障害発生時の運用」を参照してください。

ここでは，メインサイトとリモートサイトの SDB ディレクトリ情報ファイルの不一致による障害の対処方法だけを説明します。

この場合，メインサイトからリモートサイトに切り替えて，リモートサイトの HiRDB を開始したときに KFPB31010-E メッセージが出力されます。このとき，リモートサイトで次の対処をしてください。

手順

1. pdsdbdef コマンドで，SDB ディレクトリ情報ファイルを出力し，リモートサイトの全ユニットに配布します。
2. pdstop コマンドで，HiRDB を終了します。
3. pdstart コマンドで，HiRDB を開始します。
4. 1.で出力した SDB ディレクトリ情報に対応する TP1/FSP の UAP 共用ライブラリに入れ替えます。
5. TP1/FSP の eesvstart コマンドで，TP1/FSP を開始します。

9 システム定義

この章では，HiRDB のシステム定義について説明します。

9.1　システム共通定義

ここでは，システム共通定義のオペランドについて説明します。

ここに記載のないオペランドについては，マニュアル「HiRDB システム定義」を参照してください。

9.1.1　オペランドの形式と説明

(1)　オペランドの形式

```
 set pd_structured_shmpool_dicsize = SDBディレクトリ情報の常駐領域サイズ
〔set pd_structured_advance_resident = {use | nouse} 〕
```

(2)　オペランドの説明

●pd_structured_shmpool_dicsize =SDB ディレクトリ情報の常駐領域サイズ

～<符号なし整数>((1～2000000))（単位：キロバイト）

共用メモリ上に確保される SDB ディレクトリ情報の常駐領域の大きさを指定します。

このオペランドに指定した大きさの常用常駐領域が各ユニットに確保されます。高速系切り替え機能を使用している場合は，待機系ユニットにも常用常駐領域が確保されます。

また，pd_structured_advance_resident オペランドに use を指定した場合は，このオペランドに指定した大きさの事前常駐領域が，ユニットごとに確保されます。高速系切り替え機能を使用している場合は，待機系ユニットにも事前常駐領域が確保されます。

《指定値の目安》

- 指定値の目安については，「3.4.2(2)　システム共通定義 pd_structured_shmpool_dicsize オペランドに指定する値」を参照してください。
- pdsdbdef コマンドを実行して SDB ディレクトリ情報ファイルを作成したときに，SDB ディレクトリ情報長が KFPB61231-I メッセージに表示されます。表示された SDB ディレクトリ情報長を目安にこのオペランドの指定値を決めることもできます。

  （例）

  KFPB61231-I SDB directory information file assigned, file="/home/user01/pdsdb/pdsdbdir",

  num=48, size=577208, time stamp=20121206154201555(20121206151838222)

  size に SDB ディレクトリ情報長（単位：バイト）が表示されます。表示された SDB ディレクトリ情報長以上の値をこのオペランドに指定してください。
- HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更をする場合，定義追加または定義変更する分の SDB ディレクトリ情報長も加算して，このオペランドの指定値を決めてください。

《注意事項》

- このオペランドは必ず指定してください。省略した場合，HiRDB を開始できません。
- SDB ディレクトリ情報長が 2,000,000 キロバイトを超える場合は，このオペランドの指定値に関係なく，メモリに SDB ディレクトリ情報を常駐できないため，SDB データベースにはアクセスできません。
- SDB ディレクトリ情報長が 2,000,000 キロバイトを超えていないかどうかは，「3.4.2(2)(a) SDB ディレクトリ情報長の計算式」に記載されている計算式から算出して確認してください。または，pdsdbdef コマンドを実行して SDB ディレクトリ情報ファイルを作成したときに出力される KFPB61231-I メッセージの size に，SDB ディレクトリ情報の長さが表示されるため，そこで確認してください。
- SDB ディレクトリ情報の常駐領域サイズが不足している場合，HiRDB を開始することはできません（KFPB62900-E メッセージが出力されます）。この場合，このオペランドの指定値を大きくしてください。
- HiRDB はこのオペランドで指定されたメモリサイズに，HiRDB の開始に必要なメモリサイズを加算して共用メモリを確保します。このため，このオペランドの指定値が大き過ぎると，共用メモリが不足して，HiRDB を開始できないことがあります。

●pd_structured_advance_resident = {use | nouse}

SDB ディレクトリ情報を格納する事前常駐領域を確保するかどうかを指定します。HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更をする場合は，事前常駐領域が必要になるため use を指定してください。

use：

事前常駐領域を確保します。use を指定すると，常用常駐領域と事前常駐領域の 2 つの常駐領域が確保されます。

nouse：

事前常駐領域を確保しません。nouse を指定すると，常用常駐領域だけが確保されます。

《注意事項》

このオペランドに use を指定した場合，pd_structured_shmpool_dicsize オペランドで指定した大きさの常用常駐領域と事前常駐領域の 2 つの領域が共用メモリ上に確保されます。そのため，SDB ディレクトリ情報の常駐領域は 2 倍必要になります。

9.2　ユニット制御情報定義

ここでは，ユニット制御情報定義のオペランドについて説明します。

ここに記載のないオペランドについては，マニュアル「HiRDB システム定義」を参照してください。

9.2.1　オペランドの形式と説明

(1)　オペランドの形式

```
set pd_structured_directory_path = SDBディレクトリ情報ファイルの格納先ディレクトリ
```

(2)　オペランドの説明

●pd_structured_directory_path = SDB ディレクトリ情報ファイルの格納先ディレクトリ

～<パス名>

SDB ディレクトリ情報ファイルの格納先ディレクトリを絶対パスで指定します。

なお，1 つのサーバマシンに複数のユニットがある場合，ユニットごとに同じディレクトリを指定すると，1 つの SDB ディレクトリ情報ファイルを複数のユニット間で共用できます。

パス名は 140 文字以内とします。

《注意事項》

- 全ユニットのユニット制御情報定義に，このオペランドを必ず指定してください。指定しない場合，HiRDB を起動できません。
- 指定するディレクトリは，HiRDB を起動する前に作成し，HiRDB 管理者に対して参照権限を付与しておいてください。
- 共有ディスク上にディレクトリを作成した場合，実行系ユニットおよび待機系ユニット間で SDB ディレクトリ情報ファイルを共用することはできません。SDB ディレクトリ情報ファイルを共用すると，待機系ユニットの起動に失敗します。
- 実行系ユニットで SDB ディレクトリ情報ファイルへのアクセスエラーが検知された場合，KFPB62905-E および KFPB62906-E メッセージが出力されます。この場合，共用メモリ上に SDB ディレクトリ情報を常駐しないで，HiRDB の起動を完了します。ただし，このとき待機系ユニットは起動しません。
- このオペランドに指定されたパスが存在しない場合，または指定されたディレクトリ下に SDB ディレクトリ情報ファイルが存在しない場合，KFPB62910-W メッセージを出力し，共用メモリ上に SDB ディレクトリ情報を常駐しないで HiRDB の起動を完了します。ただし，このとき待機系ユニットは起動しません（KFPB62911-E メッセージを出力します）。

9.3　SDB 用 UAP 環境定義【SD FMB】

ここでは，SDB 用 UAP 環境定義のオペランドと文法規則について説明します。

SD FMB の SDB データベースにアクセスする場合に，SDB 用 UAP 環境定義の作成を検討してください。

9.3.1　オペランドの形式と説明

(1)　オペランドの形式

■set 形式

```
〔set lockrange = transaction｜cursorupdate〕
```

■コマンド形式

```
〔 { {subschema -s SDBデータベース名
              〔-t ルートレコードのレコード型名
              〔-e  {shared｜protected｜exclusive｜nonprotected}
               -a retrieve｜update〕 〕
              〔-p occupyroot｜shareroot〕
              〔-r レコード格納用RDエリア名
                   〔,レコード格納用RDエリア名〕…〕
 } } 〕
```

(2)　オペランドの説明

SDB 用 UAP 環境定義のオペランドについて説明します。

なお，サブページ分割をしている場合は，ページをサブページに読み替えてください。

(a)　set 形式

●lockrange = transaction ｜ cursorupdate

DML で SDB データベースを操作した結果，レコード位置指示子が指さなくなったレコード実現値を格納するページに対する排他の解除時期を指定します。

transaction：

トランザクション終了時に，ページに対する排他を解除します。

cursorupdate：

DML 終了時に，ページに対する排他を解除します。これを排他自動解除機能といいいます。排他自動解除機能については，「2.9.4　排他自動解除機能」を参照してください。

《利点》

このオペランドに cursorupdate を指定すると，ページの排他解除時期が早くなるため，ほかのトランザクションとの同時実行性能が向上します。

《適用基準》

通常は transaction（省略値）を指定してください。次のような場合に cursorupdate を指定してください。

- トランザクション内でデータの一貫性を保証する必要がなく，検索したレコード実現値を更新するほかのトランザクションとの同時実行性能を優先したい場合
- 更新を前提とする検索の結果，検索したレコード実現値に対する更新が不要となることがあり，そのレコード実現値を操作するほかのトランザクションとの同時実行性能を優先したい場合

《ほかのオペランドとの関連》

このオペランドの指定値を変更した場合，次に示すオペランドの指定値を見積もり直してください。

- サーバ共通定義またはバックエンドサーバ定義の pd_lck_pool_size オペランド

《見積もり式への影響》

このオペランドの指定値を変更した場合，「3.8.2(1)　レコードの検索（FETCH），または位置指示子の位置づけ（FIND）」の見積もり式を再計算してください。

(b)　コマンド形式

●subschema

```
subschema -s SDBデータベース名
         〔-t ルートレコードのレコード型名
         〔-e  {shared | protected | exclusive | nonprotected}
          -a retrieve | update〕〕
         〔-p occupyroot | shareroot〕
         〔-r レコード格納用RDエリア名〔, レコード格納用RDエリア名〕…〕
```

UAP を実行して SDB データベースにアクセスするときの動作オプションを指定します。

《オペランドの規則》

1 つの SDB 用 UAP 環境定義ファイルに，subschema オペランドを最大 1,600 個指定できます。

-s SDB データベース名

～<識別子>((1～30 バイト))

subschema オペランドで指定するオプションの対象とする SDB データベースの名称を指定します。

《オペランドの規則》

- SDB データベース名の記述規則については，「11.4.1(1)　名前の規則」を参照してください。
- subschema オペランドを複数個指定する場合，同じ SDB データベース名は指定できません。

-t ルートレコードのレコード型名

～<識別子>((1～30 バイト))

-s オプションに指定した SDB データベースのルートレコードのレコード型名を指定します。

《オペランドの規則》

- -e オプションおよび-a オプションを指定する場合に，-t オプションを指定してください。
- ルートレコードのレコード型名の記述規則については，「11.4.1(1)　名前の規則」を参照してください。

-e　{shared | protected | exclusive | nonprotected}

SD 排他モードを指定します。

shared：

ほかの UAP からの参照および更新を許します。

protected：

ほかの UAP からの更新を許しません。

exclusive：

ほかの UAP からの参照および更新を許しません。

nonprotected：

ほかの UAP からの参照および更新を許します（ページ（サブページ分割をする場合はサブページ）に対する排他制御を行いません）。

SD 排他モード（-e オプション）の指定と，アクセス目的（-a オプション）の指定によって，ページに対する排他制御を行うかどうかと，次に示す排他資源に掛かる排他の排他制御モードが決まります。

- ルートレコードのレコード型
- ルートレコードのレコード型（NOWAIT 検索時）
- レコード格納用 RD エリア

排他制御モードの詳細については，「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」を参照してください。

《オペランドの規則》

SD 排他モードに nonprotected を指定する場合は，アクセス目的に retrieve を指定してください。

《利点》

SD 排他モードとアクセス目的の組み合わせによって，UAP の目的に応じて排他の粒度と強さを制御できます。

また，SD 排他モードとアクセス目的の組み合わせが次の場合，ページに対する排他制御を行いません。

- SD 排他モードが protected で，アクセス目的が retrieve の場合
- SD 排他モードが exclusive で，アクセス目的が retrieve の場合
- SD 排他モードが exclusive で，アクセス目的が update の場合
- SD 排他モードが nonprotected で，アクセス目的が retrieve の場合

上記の組み合わせの場合，次の利点があります。

- 排他資源数を削減できます。
- 排他制御の処理時間を短縮できます。
- ページのアクセス順序によって発生するデッドロックを回避できます。

《適用基準》

SD 排他モードとアクセス目的の組み合わせ例を説明します。

オンライン業務の場合は，SD 排他モードとアクセス目的を次のように指定します。

- 参照業務の場合は，SD 排他モードに shared を，アクセス目的に retrieve を指定します。
- 更新業務の場合は，SD 排他モードに shared を，アクセス目的に update を指定します。

バッチ業務の場合は，SD 排他モードとアクセス目的を次のように指定します。

- 参照業務で，参照オンライン業務からのアクセスだけを許す場合は，SD 排他モードに protected を，アクセス目的に retrieve を指定します。
- 更新業務で，ほかの業務からのアクセスを許さない場合は，SD 排他モードに exclusive を，アクセス目的に update を指定します。

オンラインの参照業務の同時実行性を向上させたり，デッドロックを回避したりするために，無排他検索機能（SD 排他モードが nonprotected で，アクセス目的が retrieve）を選択することもできます。無排他検索機能の適用基準については，「2.9.8　SDB 用 UAP 環境定義の指定による無排他検索機能【SD FMB】」を参照してください。

《注意事項》

SD 排他モードに exclusive または protected を指定した場合，UAP の同時実行性が低下します。そのため，exclusive または protected を指定する場合は，同時に実行できなくなっても問題となる UAP がないことを確認してください。

《ほかのオペランドとの関連》

-e オプションと-a オプションの指定値の組み合わせを変更したことによって，ページに対する排他制御が変わった場合，次に示すオペランドの指定値を見積もり直してください。

- サーバ共通定義またはバックエンドサーバ定義の pd_lck_pool_size オペランド

《各見積もり式への影響》

-e オプションと-a オプションの指定値の組み合わせを変更したことによって，ページに対する排他制御が変わった場合，「3.8.2　SDB データベースを操作する API または DML の実行時の排他資源数」の見積もり式を再計算してください。

-a retrieve | update

アクセス目的を指定します。

retrieve：

レコードの参照だけを行います。次の DML を実行するとエラーになります。

- FETCH（FOR UPDATE オペランドの指定あり）

- FIND（FOR UPDATE オペランドの指定あり）
- STORE
- MODIFY
- ERASE

update：

レコードの参照および更新を行います。

SD 排他モード（-e オプション）の指定と，アクセス目的（-a オプション）の指定によって，ページに対する排他制御を行うかどうかと，次に示す排他資源に掛かる排他の排他制御モードが決まります。

- ルートレコードのレコード型
- ルートレコードのレコード型（NOWAIT 検索時）
- レコード格納用 RD エリア

排他制御モードの詳細については，「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」を参照してください。

なお，次の項目については，-e オプションの説明を参照してください。

- 利点
- 適用基準
- ほかのオペランドとの関連
- 各見積もり式への影響

-p occupyroot | shareroot

SDB データベースの階層数が 2 以上の SDB データベースに対して，ルートレコード検索，および子レコードのレコード型内検索時の，ルートレコード格納ページの排他制御方式を指定します。

occupyroot：

ルートレコード格納ページに，排他制御モード EX で排他制御を行います。

shareroot：

ルートレコード格納ページに，排他制御モード SR または PU で排他制御を行います（ルートレコード検索の場合，SDB データベースの階層数が 1 の SDB データベースのときと同じ排他制御モードで，排他制御を行います）。

排他制御モードの詳細については，「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」を参照してください。このオプションは，SD 排他モードとアクセス目的の組み合わせが shared update または protected update の場合に有効となります。

《利点》

デフォルト値では，複数の UAP から同一ファミリを検索すると，後続の UAP が排他待ちとなります。

-p オプションに shareroot を指定すると，後続の UAP が排他待ちすることなく，同一ファミリを検索できるようになります。

ただし，-p オプションに shareroot を指定しても，DML に FOR UPDATE を指定した UAP 同士を同時実行した場合は，複数の UAP から同一レコードを同時に更新することを抑止するため，後続の UAP が排他待ちをします。

《適用基準》

通常は-p オプションに occupyroot（省略値）を指定してください。

UAP の運用上，ルートレコード格納ページの排他待ちが問題となる場合は，-p オプションに shareroot を指定することを検討してください。

-p オプションに shareroot を指定すると，複数の UAP から同時に子レコードにアクセスできるようになります。

そのため，次に該当するような SDB データベースの操作をしない，またはその操作の際のデッドロックを許容できることを確認してください。

- 子レコードを対象に，レコードの更新（MODIFY），レコードの格納（STORE），またはレコードの削除（ERASE）を実行する。
- 複数の UAP 間で，ファミリ内の子レコードのアクセス順序を統一していない。

子レコードを対象に，レコードの更新（MODIFY），レコードの格納（STORE），またはレコードの削除（ERASE）を実行する場合は，レコードのアクセス順序を統一していても，デッドロックが発生することがあります。デッドロックの例については，「2.9.9(4)　ルートレコードを格納しているページと子レコードを格納しているページ間のデッドロックの例【4V FMB，SD FMB】」を参照してください。

最終的に，業務の際には，同時実行性能向上とデッドロック防止のどちらを優先するか判断した上で，-p オプションの指定値を決定してください。

《備考》

-p オプションに occupyroot を指定しているときに，ルートレコード格納ページに，排他制御モード EX で排他制御を行うのは，子レコード更新時のデッドロックを防止するためです。

次のどちらかの場合は，-p オプションの指定に関係なく，ルートレコード格納ページに，排他制御モード SR または PU で排他制御を行います。

- SDB データベースの階層数が 1 の SDB データベースの場合
- アクセス目的が retrieve の場合

-r レコード格納用 RD エリア名〔，レコード格納用 RD エリア名〕…

～＜識別子＞((1～30 バイト))

横分割している SDB データベースに対して，アクセスする RD エリアを限定したい場合，レコード格納用 RD エリア名を指定します。

subschema オペランド，または-r オプションを省略すると，SDB データベースを格納しているすべての RD エリアをアクセス対象として排他制御を行います。

横分割していない SDB データベースに-r オプションを指定した場合，エラーとなります。

《利点》

DML を実行する際，記述されたレコード格納用 RD エリアだけをアクセス対象として排他制御を行います。

バッチ業務（SD 排他モードに exclusvie を指定）の場合，-r オプションを省略すると，SDB データベースを格納しているすべての RD エリアに排他制御モード EX で排他制御を行います。このため，その SDB データベースをほかの業務からアクセスできません。

-r オプションを指定して，アクセスする RD エリアを限定すると，-r オプションに記述していないレコード格納用 RD エリアの排他制御を行わないので，ほかの業務からアクセスできます。

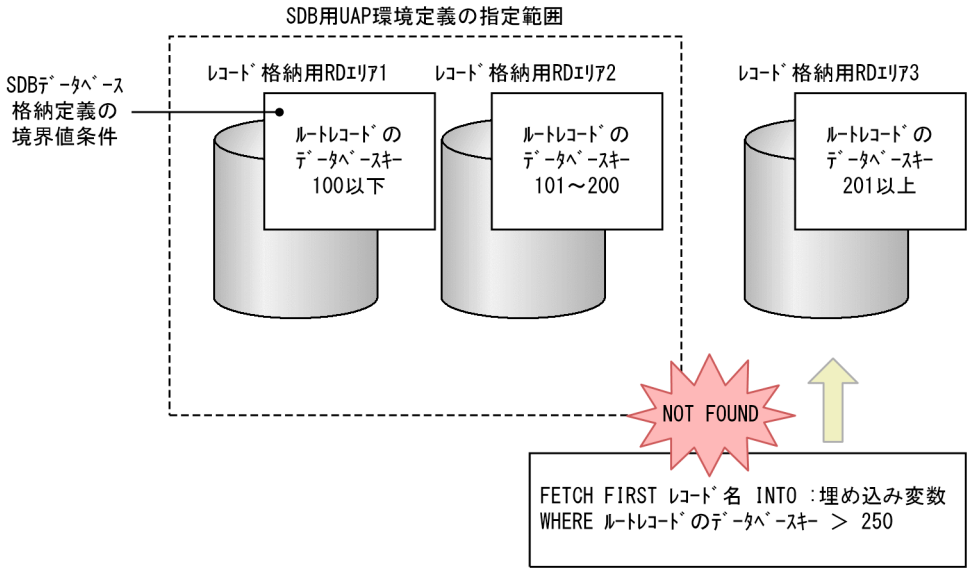
《オペランドの規則》

- レコード格納用 RD エリア名の記述規則については，「11.4.1(1)　名前の規則」を参照してください。
- レコード格納用 RD エリア名を複数記述する場合，重複したレコード格納用 RD エリア名は指定できません。
- 最大 1,024 個のレコード格納用 RD エリア名を指定できます。

《留意事項》

- -r オプションを指定した環境で，レコードの格納（STORE）を実行する場合，該当する RD エリアを指定していないと，エラーとなります。
- -r オプションを指定した環境で，レコードの検索（FETCH）または位置指示子の位置づけ（FIND）を実行する場合，キーの条件に該当する RD エリアを指定していないと，NOT FOUND となります。NOT FOUND の例については，次の図を参照してください。

図 9-1　NOT FOUND の例



[説明]

SDB データベース格納定義では，3 つの RD エリアに境界値分割しています。

- レコード格納用 RD エリア 1

  ルートレコードのデータベースキー 100 以下のレコードを格納

- レコード格納用 RD エリア 2
  ルートレコードのデータベースキー 101～200 のレコードを格納
- レコード格納用 RD エリア 3
  ルートレコードのデータベースキー 201 以上のレコードを格納

SDB 用 UAP 環境定義では，-r オプションに，レコード格納用 RD エリア 1 と 2 を指定しています。
レコードの検索（FETCH）のキーの条件に，「ルートレコードのデータベースキーが 250 より大きい」を指定して実行すると，該当する RD エリアがないので，NOT FOUND となります。

9.3.2 SDB 用 UAP 環境定義の指定例

■指定例

```
set lockrange = cursorupdate                          ...1
subschema -s SDB01 -t RECA -e shared -a retrieve      ...2
subschema -s SDB02 -t RECB -e exclusive -a update     ...3
subschema -s SDB03 -t RECC -e exclusive -a update ¥
          -r RDAREA1,RDAREA3                          ...4
```

[説明]

1. DML で SDB データベースを操作した結果，レコード位置指示子が指さなくなったレコード実現値を格納するページ（サブページ分割をした場合はサブページ）に対する排他の解除時期を指定します。
   DML の終了時に排他を解除するため，cursorupdate を指定します。
2. 参照業務で SDB データベースを操作する際，ほかの参照業務または更新業務からのアクセスを許す場合の動作オプションを指定します。
   -s：操作対象の SDB データベースの名称 SDB01 を指定します。
   -t：操作対象の SDB データベースのルートレコードのレコード型名 RECA を指定します。
   -e：ほかの参照業務または更新業務から，SDB データベース SDB01 へのアクセスを許すため，SD 排他モードに shared を指定します。
   -a：参照業務で SDB データベース SDB01 にアクセスするため，アクセス目的に retrieve を指定します。
3. 更新業務で SDB データベースを操作する際，ほかの業務からのアクセスを許さない場合の動作オプションを指定します。
   -s：操作対象の SDB データベースの名称 SDB02 を指定します。
   -t：操作対象の SDB データベースのルートレコードのレコード型名 RECB を指定します。
   -e：ほかの業務から SDB データベース SDB02 へのアクセスを許さないように，SD 排他モードに exclusive を指定します。
   -a：更新業務で SDB データベース SDB02 にアクセスするため，アクセス目的に update を指定します。
4. 更新業務で RD エリアを限定して SDB データベースを操作する際，ほかの業務からのアクセスを許さない場合の動作オプションを指定します。

-s：操作対象の SDB データベースの名称 SDB03 を指定します。

-t：操作対象の SDB データベースのルートレコードのレコード型名 RECC を指定します。

-e：ほかの業務から SDB データベース SDB03 へのアクセスを許さないように，SD 排他モードに exclusive を指定します。

-a：更新業務で SDB データベース SDB03 にアクセスするため，アクセス目的に update を指定します。

-r：RD エリアを RDAREA1 と RDAREA3 に限定するため，RDAREA1 と RDAREA3 を指定します。

9.3.3　SDB 用 UAP 環境定義の文法規則

SDB 用 UAP 環境定義の文法規則は，HiRDB システム定義の文法規則に従います。HiRDB システム定義の文法規則については，マニュアル「HiRDB システム定義」の「HiRDB システム定義を記述するときの文法規則」を参照してください。

ここでは，SDB 用 UAP 環境定義特有の文法規則について説明します。

(1)　1 オペランドの継続行数の最大値

1 オペランドの継続行数の最大値は 2,000 となります。1 オペランドの継続行数とは，継続記号の¥で継続する行数を意味しています。継続行数の数え方の例を次に示します。

（例）

```
subschema              ¥        ←継続行数1
    -s sdb01           ¥        ←継続行数2
    -t record01        ¥        ←継続行数3
    -e exclusive       ¥        ←継続行数4
    -a update                   ←継続行数5
```

上記の例の場合，1 オペランドの継続行数は 5 となります。

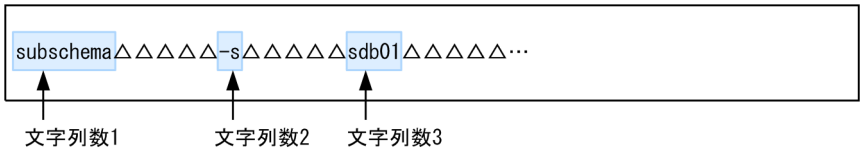
(2)　1 オペランドの文字列数の最大値

1 オペランドの文字列数の最大値は 2,000 となります。

- 1 オペランドの文字列数の数え方

  1 オペランドの文字列数とは，半角空白またはタブで区切られた文字列の数を意味しています。1 オペランドの文字列数の数え方の例を次に示します。

  （例）

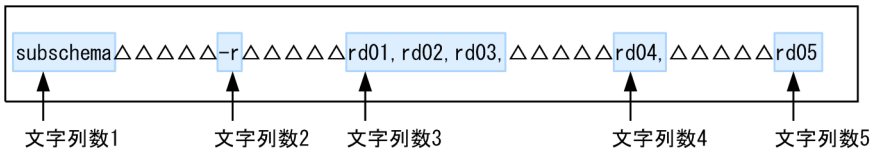


（凡例）△：半角空白またはタブ

上記の例の場合，1 オペランドの文字列数は 3 となります。

なお，コンマ（ , ）は，文字列の区切りとは見なされません。コンマを使用した場合の文字列数の数え方の例を次に示します。

（例）



（凡例）△：半角空白またはタブ

上記の例の場合，1 オペランドの文字列数は 5 となります。

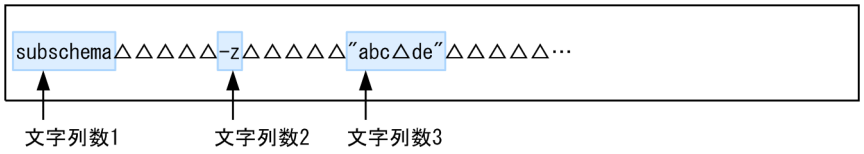
rd01，rd02，rd03 は，半角空白またはタブで区切られていないため，同じ文字列と見なされます。

rd04 と rd05 は，半角空白またはタブで区切られているため，異なる文字列と見なされます。

- **文字列の区切りと見なされないケース**

  半角空白またはタブを引用符（ " ）で囲んだ場合，半角空白とタブは文字列の区切りとは見なされません。半角空白またはタブを引用符で囲んだ場合の文字列数の数え方の例を次に示します。

  （例）



（凡例）△：半角空白またはタブ

上記の例の場合，1 オペランドの文字列数は 3 となります。

- **文字列と見なされないケース**

  行の末尾の継続記号である¥，およびコメント（#を含んだ行の末尾までの文字）は，文字列数に数えません。

(3) オプションの重複指定の禁止

SDB 用 UAP 環境定義のオプションを重複指定できません。エラーとなる例を次に示します。

（例）

```
subschema -s sdb01 -s sdb02 -t RECA -e shared -a retrieve
```

上記の例では，-s オプションが重複して指定されているため，エラーになります。

(4) 2 つのマイナス記号の使用禁止

SDB 用 UAP 環境定義では，"--"は指定できません。

参考

コマンド引数にマイナス記号を指定する場合，オプションの終わりに 2 つのマイナス記号"--"を記述すると，"--"以降をコマンド引数として解釈する製品があります。SDB 用 UAP 環境定義では該当しません。

-2 をオプションではなく，コマンド引数として指定する例（エラーとなる例）を次に示します。

（例）

```
subschema -s sdb01 -- -2
```

(5) 改行コード CRLF が指定できる

改行コードに，LF（X'0A'）だけでなく，CRLF（X'0D0A'）も指定できます。

参考

そのほかの HiRDB システム定義は，プラットフォームが Linux の場合，CRLF を指定できません。

9.4　指定値の見直しが必要なオペランド

指定できないオペランド，または指定値の見直しが必要なオペランドを次の表に示します。

それぞれのオペランドの詳細については，マニュアル「HiRDB システム定義」を参照してください。

表 9-1　指定値の見直しが必要なオペランド

項番	分類	定義名	オペランド名	指定可否および指定値の見直し	指定値の目安
1	最大同時実行数	SYS	pd_max_users	○	pd_max_users オペランドに，SDB データベースにアクセスするユーザ数を加算してください。 HiRDB/SD データベースアクセスユティリティ（pdsdbexe）を使用して SDB データベースにアクセスする場合も，1 ユーザとして pd_max_users オペランドに加算してください。HiRDB/SD データベースアクセスユティリティを複数同時に使用する場合は，それぞれ 1 ユーザとしてカウントする必要があります。 また，次に示すユティリティも，HiRDB に接続するため，pd_max_users オペランドにユーザ数を加算する必要があります。 ユティリティの最大同時実行数については，「付録 C　ユティリティの最大同時実行数およびコマンドの同時接続数」を参照してください。 • HiRDB/SD 定義ユティリティ（pdsdbdef） • HiRDB/SD データベース作成ユティリティ（pdsdblod） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
2		SYS	pd_max_access_tables	○	アクセスするレコード型数を指定値に加算してください。 HiRDB/SD 定義ユティリティ（pdsdbdef）を実行する場合は，計算式※で求めた値を加算してください。
3		SYS	pd_utl_exec_mode	○	次に示すユティリティは，このオペランドの対象になります。 • HiRDB/SD 定義ユティリティ（pdsdbdef）

項番	分類	定義名	オペランド名	指定可否 および 指定値の見直し	指定値の目安
					• HiRDB/SD データベース作成ユティリティ（pdsdblod） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog） ユティリティの最大同時実行数については，「付録 C　ユティリティの最大同時実行数およびコマンドの同時接続数」を参照してください。
4		SYS	pd_max_commit_write_reclaim_no	△	SDB データベースに対して，空きページ解放ユティリティ（pdreclaim）は使用できません。
5	HiRDBの処理方式	SYS	pd_space_level	△	SDB データベースでは，空白変換機能は使用しません。
6		SYS	pd_pageaccess_mode	○	SDB データベースに対しては，スナップショット方式によるページアクセスは適用されません。 更新可能なオンライン再編成の追い付き反映キー対応表には，スナップショット方式によるページアクセスが適用されます。
7		SYS	pd_dec_sign_normalize	○	SDB データベースでは，正規化された符号付きパック形式 10 進データと，正規化されていない符号付きパック形式 10 進データが混在した状態で，SDB データベースを操作すると，不当にデータベース破壊を検知したり，実行結果が不正になったりすることがあります。 そのため，新規に HiRDB/SD を導入する際には Y を指定し，以降は指定値を変更しないでください。
8	システム監視	SYS	pd_utl_exec_time	○	次に示すユティリティは，このオペランドの対象となります。 • HiRDB/SD データベース作成ユティリティ（pdsdblod） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
9		DS BES	pdwork	○	• バックエンドサーバ定義の場合 HiRDB/SD データベース作成ユティリティ（pdsdblod）のインデクス一括作成モードを使用する場合は指定する必要があります。

項番	分類	定義名	オペランド名	指定可否 および 指定値の見直し	指定値の目安
					• ディクショナリサーバ定義の場合 省略できます。ただし，省略する場合は，pd_work_buff_size オペランドの指定を省略（または 5MB 以上を指定）し，pd_work_buff_mode オペランドの指定を省略（または pool を指定）する必要があります。
10		SVR FES DS BES	pd_spd_syncpoint_skip_limit	○	次のコマンドは，シンクポイントダンプ有効化のスキップ回数監視機能の対象外です。 • pdsdbdef • pdsdblod • pdsdbrog
11	排他制御	SYS	pd_nowait_scan_option	△	指定値に関係なく，SDB データベースに対して，WITHOUT LOCK NOWAIT 指定の検索はできません。
12		SYS	pd_key_resource_type	△	SDB データベースに対しては，インデクスキー値で排他制御をしません。
13		SYS	pd_indexlock_mode	○	このオペランドを省略するか，または NONE を指定してください。
14		SYS	pd_lock_uncommited_delete_data	○	SDB データベースを操作する API または DML （FETCH，FIND，FETCHDB ALL，STORE）実行時，相手トランザクションが操作中のレコードを参照したときの動作を指定します。 WAIT： 次の動作を選択したい場合に指定します。 • レコードの検索（FETCH），位置指示子の位置づけ（FIND），または複数レコードの検索（FETCHDB ALL）実行時，相手トランザクションがレコードの削除またはレコードの更新を先に実行していた場合，相手トランザクションが決着するまで排他待ちをします。 トランザクション決着後のデータを基にした検索結果を取得したい場合に指定します。

項番	分類	定義名	オペランド名	指定可否 および 指定値の見直し	指定値の目安
					• レコードの格納（STORE）またはレコードの更新（MODIFY）実行時，相手トランザクションが同じインデクスキー値のレコードの格納または更新を先に実行していた場合，排他待ちをします。相手トランザクションがロールバックしても，該当レコードを必ず追加または更新したい場合に指定します。 NOWAIT（省略値）： 次の動作を選択したい場合に指定します。 • レコードの検索（FETCH），位置指示子の位置づけ（FIND），または複数レコードの検索（FETCHDB ALL）実行時，相手トランザクションがレコードの削除またはレコードの更新を先に実行していた場合，相手トランザクションの決着を待ちません。 削除されたデータを読み飛ばしたい場合に指定します。 削除されたデータとは，ERASE によって削除されたレコード，または MODIFY によって削除された更新前のインデクスキーを指します。 • レコードの格納（STORE）またはレコードの更新（MODIFY）実行時，相手トランザクションが同じインデクスキー値のレコードの格納または更新を先に実行していた場合，排他待ちをしません。即時，ユニークエラーとしたい場合に指定します。
15		SVR	pd_lck_until_disconnect_cnt	○	「3.8.4　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時の排他資源数」を参照して，使用する排他資源数を計算して，指定値に加算してください。
16		SVR DS BES	pd_lck_pool_size	○	「3.8　排他資源数の見積もり」を参照して，排他資源数を求めて，マニュアル「HiRDB システム定義」の pd_lck_pool_size の《指定値の目安》にある計算式に代入して指定値を求めてください。

項番	分類	定義名	オペランド名	指定可否および指定値の見直し	指定値の目安
17		SVR FES	pd_fes_lck_pool_size	△	SDB データベースでは，フロントエンドサーバで排他制御をしません。
18		SYS	pd_dbreuse_remaining_entries	△	SDB データベースに対しては，行識別子の再利用を抑止して，排他待ちの発生頻度を低くすることはできません。
19	RD エリア	SYS	pd_shared_rdarea_use	△	SDB データベースでは，共用 RD エリアは使用できません。
20	グローバルバッファ	SYS	pd_dbbuff_dev_sector_size	×	論理セクタ長 4,096 バイトのディスクは未サポートです。
21			pdbuffer	○	グローバルバッファ指定時の考慮点については，「3.3.3　グローバルバッファを割り当てる際の考慮点」を参照してください。
22	バッファ	SVR FES	pd_table_def_cache_size	○	デフォルト値で問題ないため，pd_table_def_cache_size オペランドの指定は省略してください。
23		SYS SVR FES DS BES	pd_sql_object_cache_size	○	このオペランドの指定値の目安および見積もり式については，マニュアル「HiRDB システム定義」の pd_sql_object_cache_size オペランドの説明を参照してください。 また，更新可能なオンライン再編成を実行する場合は，見積もった SQL オブジェクト用バッファ長に，次の計算式で求めた値を加算してください。 • 48 ＋（31×更新可能なオンライン再編成の対象 BES 数） 単位：キロバイト
24	メモリ DB	SYS	pd_max_resident_rdarea_no	△	SDB データベースに対して，メモリ DB は適用できません。
25	インデクスの予約数	SYS	pd_assurance_index_no	○	使用するインデクス数から指定値を決めてください。 使用するインデクス数については，現在定義されているインデクス数だけでなく，次回 HiRDB 開始から終了までの間に追加予定のインデクス数も考慮してください。
26	再編成時期予測機能	SYS	pd_rorg_predict	×	再編成時期予測機能は未サポートです。

項番	分類	定義名	オペランド名	指定可否 および 指定値の見直し	指定値の目安
27	系切り替え機能	UNT	pd_ha_agent	○	standbyunit を指定してください。
28	HiRDB Datareplicator	UNT	pd_rpl_hdepath	×	HiRDB Datareplicator は未サポートです。
29	リアルタイム SAN レプリケーション	SYS	pd_rise_pairvolume_combination	○	全非同期方式だけをサポートしているため，async だけを指定できます。
30		SYS	pd_rise_fence_level	×	全非同期方式以外は未サポートのため，指定しても無視されます。
31		SYS	pd_rise_disaster_mode	×	全非同期方式以外は未サポートのため，指定しても無視されます。
32	通信処理	SYS	pd_utl_buff_size	△	次に示すユティリティは，このオペランドの対象外となります。 • HiRDB/SD 定義ユティリティ（pdsdbdef） • HiRDB/SD データベース作成ユティリティ（pdsdblod） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
33		SYS	pd_utl_file_buff_size	○	ユティリティが使用するファイルへ入出力するデータの 1 回当たりのサイズを指定します。 次に示すユティリティは，このオペランドの対象となります。 • HiRDB/SD 定義ユティリティ（pdsdbdef） • HiRDB/SD データベース作成ユティリティ（pdsdblod） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
34	作業表	DS BES	pd_work_buff_mode	○	このオペランドを省略するか，または pool を指定してください。
35		DS BES	pd_work_buff_size	○	• バックエンドサーバ定義の場合 HiRDB/SD データベース作成ユティリティ（pdsdblod）のインデクス一括作成モードを使用する場合は，このオペランドの対象となります。 作業表用ファイルの容量を超えない範囲で，大きい値ほどユティリティの処理時間が短くなります。

項番	分類	定義名	オペランド名	指定可否および指定値の見直し	指定値の目安
					作業表用ファイルの容量については，「3.5.2　作業表用ファイルの容量の見積もり」を参照してください。 • ディクショナリサーバ定義の場合 HiRDB/SD 定義ユティリティ（pdsdbdef）を実行して，SDB ディレクトリ情報を削除する場合，このオペランドの指定を省略，または 5MB 以上を指定する必要があります。
36	ワークファイルの出力先ディレクトリの変更	UNT	pd_tmp_directory	○	ワークファイルの出力先ディレクトリ名を指定します。 次に示すユティリティは，このオペランドの対象となります。 • HiRDB/SD 定義ユティリティ（pdsdbdef） • HiRDB/SD データベース作成ユティリティ（pdsdblod） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
37	UAP の実行環境	UAP	pd_uap_wait	△	SDB データベースに対して，ローカルバッファは適用できません。
38		UAP	pdlbuffer	△	
39	プロセス	BES	pd_max_open_fds	○	• HiRDB/SD データベース作成ユティリティのインデクス一括作成モードを使用する場合 次の計算式から求めた値を指定してください。 MIN（7,320，320 ＋データロード対象のデータベースの全インデクス数×データロード対象の格納レコード用 RD エリア数） • HiRDB/SD データベース作成ユティリティのインデクス一括作成モードを使用しない場合 省略するか，320 を指定してください。
40	監視時間	SYS	pd_cmd_exec_time	○	次に示すユティリティは，このオペランドの対象となります。 • HiRDB/SD 定義ユティリティ（pdsdbdef） • HiRDB/SD データベース作成ユティリティ（pdsdblod）

項番	分類	定義名	オペランド名	指定可否 および 指定値の見直し	指定値の目安
					• HiRDB/SD データベース再編成ユティリティ（pdsdbrog） 上記ユティリティの実行時間が指定値よりも長時間となる場合は，指定値を見直してください。
41		SYS	pd_cmd_process_conwait time	○	次に示すユティリティは，このオペランドの対象となります。 • HiRDB/SD 定義ユティリティ（pdsdbdef） • HiRDB/SD データベース作成ユティリティ（pdsdblod） • HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

（凡例）

○：指定できます。ただし，指定値を見直してください。

△：該当する機能は SDB データベースでは使用できません。または指定しても SDB データベースには適用されません。

×：指定できません。指定した場合，動作は保証されません。

SYS：システム共通定義

UNT：ユニット制御情報定義

SVR：サーバ共通定義

FES：フロントエンドサーバ定義

DS：ディクショナリサーバ定義

BES：バックエンドサーバ定義

UAP：UAP 環境定義

注※

HiRDB/SD 定義ユティリティ（pdsdbdef）を実行する場合は，次に示す計算式で求めた値を pd_max_access_tables オペランドに加算してください。pd_max_access_tables オペランドの指定値が小さい場合，HiRDB/SD 定義ユティリティ（pdsdbdef）が異常終了するおそれがあります。

1. *DELETE DICTIONARY 文を実行する場合

   A ＋ B

   A：SDB データベース定義に指定しているレコード型数

   B：SDB データベース格納定義に指定しているインデクス数

2. *CHECK DICTIONARY 文を実行する場合

   *ENTRY DICTIONARY 文で追加した SDB ディクショナリ情報に対して，*CHECK DICTIONARY 文を実行する場合に，次の計算式の値を加算します。

   A ＋ B

   A：SDB データベース定義に指定しているレコード型数

B：SDB データベース格納定義に指定しているインデクス数

*ALTER DICTIONARY 文で変更した SDB ディクショナリ情報に対して，*CHECK DICTIONARY 文を実行する場合に，次の計算式の値を加算します。

A ＋ B

A：SDB データベース定義に追加したレコード型数

B：SDB データベース格納定義に追加したインデクス数

3. *ENTRY DIRECTORY 文を実行する場合

*CHECK DICTIONARY 文を実行していない場合にかぎり，次の計算式の値を加算します。

A ＋ B

A：SDB データベース定義に指定しているレコード型数

B：SDB データベース格納定義に指定しているインデクス数

4. *ALTER DIRECTORY 文を実行する場合

*CHECK DICTIONARY 文を実行していない場合にかぎり，次の計算式の値を加算します。

A ＋ B

A：SDB データベース定義に追加したレコード型数

B：SDB データベース格納定義に追加したインデクス数

なお，複数の SDB 定義文を実行する場合は，上記の各計算式を加算してください。例えば，*DELETE DICTIONARY 文と*CHECK DICTIONARY 文を同時に実行する場合は，上記の 1 と 2 の計算式の合計値を加算してください。

9.5　HiRDB の再開始時に指定値を変更できるオペランド

HiRDB の再開始時に指定値を変更できるオペランドと，変更できないオペランドがあります。オペランドの指定値の変更可否を次の表に示します。

次の表に記載のないオペランドについては，マニュアル「HiRDB システム定義」の「指定値を変更できるオペランド」を参照してください。

表 9-2　オペランドの指定値の変更可否（HiRDB の再開始時）

項番	オペランド名	強制終了，異常終了後の変更可否	計画停止後の変更可否
1	pd_structured_shmpool_dicsize	○※	○※
2	pd_structured_advance_resident	×	×
3	pd_structured_directory_path	○	○

（凡例）

○：指定値を変更できます。

×：指定値を変更できません。

注※

オペランドの指定値を小さくすると，HiRDB を再開始できない場合があります。

SDB 用 UAP 環境定義のオペランドについては，HiRDB の再開始時に指定値を変更できます。

9.6　pdconfchk コマンドでチェックできるオペランド

pdconfchk コマンドでチェックできるオペランドを次の表に示します。

次の表に記載のないオペランドについては，マニュアル「HiRDB システム定義」の「pdconfchk コマンドでチェックできるオペランド」を参照してください。

表 9-3　pdconfchk コマンドでチェックできるオペランド

項番	オペランド名	文法のチェック	ファイルのチェック	アクセス権限のチェック	重複指定のチェック	ホスト名のチェック	サーバマシン間のチェック
1	pd_structured_shmpool_dicsize	○	×	×	×	×	○
2	pd_structured_advance_resident	○	×	×	×	×	○
3	pd_structured_directory_path	○	×	×	×	×	○

（凡例）

○：チェック対象です。

×：チェック対象外です。

文法のチェック：

オペランドの文法が正しいかチェックします。

ファイルのチェック：

システムログファイル，シンクポイントダンプファイル，およびステータスファイルの有無をチェックします。pdconfchk コマンドで-n オプションを指定した場合はファイルのチェックをしません。

アクセス権限のチェック：

HiRDB 管理者にファイルのアクセス権があるかを確認します。

HiRDB 管理者に hosts ファイルのアクセス権があるかを確認します。

pdconfchk コマンドで-n オプションを指定した場合はアクセス権限のチェックをしません。

重複指定のチェック：

システムログファイル，シンクポイントダンプファイル，およびステータスファイルが重複していないかどうかをチェックします。

ホスト名のチェック：

ホスト名が hosts ファイルに記述されているかをチェックします。

サーバマシン間のチェック：

システムマネジャが定義されているサーバマシンを基準に，サーバマシン間の妥当性をチェックします。

注意事項

SDB 用 UAP 環境定義のオペランドは，pdconfchk コマンドのチェック対象外になります。

10 コマンド

この章では，運用コマンドおよびユティリティについて，マニュアル「HiRDB コマンドリファレンス」との差異を中心に説明します。

10.1　運用コマンドおよびユティリティの実行方法と実行可否

ここでは，運用コマンドおよびユティリティの実行方法と実行可否について説明します。

10.1.1　運用コマンドおよびユティリティの実行方法

運用コマンドおよびユティリティの実行方法については，マニュアル「HiRDB コマンドリファレンス」の「コマンドの入力方法」を参照してください。

運用コマンドおよびユティリティの記述形式については，マニュアル「HiRDB コマンドリファレンス」の「コマンドの記述形式」を参照してください。

10.1.2　運用コマンドおよびユティリティの一覧

HiRDB/SD の運用コマンドおよびユティリティの一覧を「表 10-1　HiRDB/SD の運用コマンドおよびユティリティの一覧」に示します。

なお，運用コマンドおよびユティリティの機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」で説明しているコマンドのうち，使用できないコマンドがあります。「表 10-1　HiRDB/SD の運用コマンドおよびユティリティの一覧」の「実行可否」欄が×のコマンドは実行できません。

また，運用コマンドおよびユティリティの機能および指定形式の詳細の説明が，マニュアル「HiRDB コマンドリファレンス」ではなく，このマニュアルに記載されている運用コマンドおよびユティリティがあります。該当する運用コマンドおよびユティリティについては，「表 10-1　HiRDB/SD の運用コマンドおよびユティリティの一覧」の「留意事項およびマニュアル参照先」欄にマニュアルの参照先を記載しています。

表 10-1　HiRDB/SD の運用コマンドおよびユティリティの一覧

分類	ユティリティ運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
システムの運用	pdsdbarc	SDB ディレクトリ情報の常駐化および最終更新日時のチェック	○	「10.9.1　pdsdbarc（SDB ディレクトリ情報の常駐化および最終更新日時のチェック）」を参照してください。
	pdadmvr	HiRDB バージョン情報の取得	○	なし
	pdcat	ファイルの内容表示	○	
	pdchgconf	システム構成変更コマンド	○	
	pdclibsync	C ライブラリファイルの操作	×	

分類	ユティリティ 運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
	pdconfchk	システム定義のチェック	○	
	pdcspool	トラブルシュート情報および作業用一時ファイルの削除	○	作業用一時ファイルは出力されません。
	pdinfoget	障害情報の取得および容量見積もり	○	なし
	pditvtrc	HiRDB の状態の定期取得	○	次に示すユティリティは対象外となります。 HiRDB/SD 定義ユティリティ（pdsdbdef） HiRDB/SD データベース作成ユティリティ（pdsdblod） HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
	pditvstop	HiRDB の状態の定期取得の停止	○	次に示すユティリティは対象外となります。 HiRDB/SD 定義ユティリティ（pdsdbdef） HiRDB/SD データベース作成ユティリティ（pdsdblod） HiRDB/SD データベース再編成ユティリティ（pdsdbrog）
	pdjarsync	JAR ファイルの操作	×	なし
	pdlodsv	インストールディレクトリの容量削減	×	
	pdlistls	リスト定義情報の表示	×	
	pdls	HiRDB システムの状態表示	○	次のオプションは指定できません。 • -d ris • -d rpl 系切り替え機能を使用していない場合は，次のオプションを使用できません。 • -d ha
	pdmemsv	メモリの削減	○	なし
	pdobjconv	64 ビットモードの HiRDB への SQL オブジェクトの移行	×	
	pdopsetup	付加 PP のセットアップ	○	「4.3.2(2)　付加 PP のインストール」を参照してください。

分類	ユティリティ 運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
	pdsetup	HiRDB システムの OS への登録・削除	○	指定できる文字コードは，4V FMB または 4V AFM の SDB データベースの場合は，シフト JIS 漢字コードまたは Unicode（UTF-8）のどちらかになります。 SD FMB の SDB データベースの場合は，シフト JIS 漢字コードになります。
	pdsvhostname	サーバのホスト名表示	○	なし
	pdvrup	HiRDB バージョンアップ	○	
HiRDB ファイルシステム	pdfbkup	HiRDB ファイルシステム領域のバックアップ	○	なし
	pdfls	HiRDB ファイルシステムの内容表示	○	
	pdfmkfs	HiRDB ファイルシステム領域の初期設定	○	
	pdfrm	HiRDB ファイルの削除	○	
	pdfrstr	HiRDB ファイルシステム領域のリストア	○	
	pdfstatfs	HiRDB ファイルシステム領域の状態表示	○	
	pdffsck	HiRDB ファイルシステム領域の整合性の検証および修復	○	
ログ関係のファイル	pdlogadpf	ログ関係のファイルの割り当て	○	なし
	pdlogatul	自動ログアンロード機能の制御	○	
	pdlogchg	ログ関係のファイルのステータス変更	○	
	pdlogcls	ログ関係のファイルのクローズ	○	
	pdloginit	ログ関係のファイルの初期設定	○	
	pdlogls	ログ関係のファイルの情報表示	○	
	pdlogopen	ログ関係のファイルのオープン	○	
	pdlogrm	ログ関係のファイルの削除	○	

分類	ユティリティ運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
	pdlogswap	ログ関係のファイルのスワップ	○	
	pdlogsync	シンクポイントダンプの取得	○	
	pdlogucat	アンロードログファイルの情報表示	○	
	pdlogunld	ログ関係のファイルのアンロード	○	
ステータスファイル	pdstscls	ステータスファイルのクローズ	○	なし
	pdstsinit	ステータスファイルの初期設定	○	
	pdstsopen	ステータスファイルのオープン	○	
	pdstsrm	ステータスファイルの削除	○	
	pdstsswap	ステータスファイルのスワップ	○	
HiRDB の開始・終了	pdstart	HiRDB システム・ユニット・サーバの開始	○	なし
	pdstop	HiRDB システム・ユニット・サーバの終了	○	
統計ログ	pdstbegin	統計情報の出力開始	○	なし
	pdstend	統計情報の出力停止	○	
	pdstjswap	統計ログファイルの切り替え	○	
	pdstjsync	統計ログファイルへの統計ログバッファの反映	○	
RD エリア	pdclose	RD エリアのクローズ	○	なし
	pddbls	RD エリアの状態表示	○	「10.9.3　pddbls（RD エリアの状態表示）」を参照してください。
	pdhold	RD エリアの閉塞	○	なし
	pdopen	RD エリアのオープン	○	なし
	pdrels	RD エリアの閉塞解除	○	
	pddbfrz	ユーザ LOB 用 RD エリアの満杯 HiRDB ファイルの更新凍結	×	

分類	ユティリティ運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
	pdrdconstck	インナレプリカグループ内のRDエリアの整合性チェック	×	
	pdrdrefls	関連RDエリアの表示	○	-tオプションには，処理対象とするレコード型名を指定します。
グローバルバッファ	pdbufls	グローバルバッファ情報の表示	○	なし
	pdbufmod	グローバルバッファの動的変更	○	
トランザクションの制御	pdcmt	トランザクションのコミット	○	なし
	pdfgt	トランザクションの強制終了	○	
	pdrbk	トランザクションのロールバック	○	
	pdtrndec	未決着トランザクションの強制自動決着	○	
プロセスの制御	pdcancel	UAP，ユティリティ処理の強制終了	○	なし
	pdchprc	サーバプロセスの起動本数変更	○	
	pdpfresh	サーバプロセスのリフレッシュ	○	
	pdrpause	プロセスサーバプロセスの再起動	○	
修正版HiRDBの入れ替え	pdprgcopy	修正版HiRDBのコピー	○	なし
	pdprgrenew	修正版HiRDBの入れ替え	○	HiRDBのバージョンアップまたはバージョンダウンとなるような入れ替えを行った場合，動作を保証できません。
HiRDB Datareplicator連携	pdrplstart	HiRDB Datareplicator連携の開始	×	なし
	pdrplstop	HiRDB Datareplicator連携の終了	×	
インナレプリカ機能	pddbchg	レプリカRDエリアのレプリカステータスの切り替え	○	なし
更新可能なオンライン再編成	pdorbegin	オンライン再編成のデータベース静止化	○	-cおよび-eオプションは指定できません。
	pdorcheck	オンライン再編成の適用条件チェック	○	なし

分類	ユティリティ 運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
	pdorchg	オンライン再編成のカレントRDエリアの切り替え	○	
	pdorcreate	オンライン再編成の追い付き反映環境の作成	○	
	pdorend	オンライン再編成の追い付き反映	○	-pオプションは指定できません
	pdsdborcrt	追い付き反映キー対応表の操作	○	なし
セキュリティ監査	pdaudbegin	監査証跡の取得開始	○	なし
	pdaudend	監査証跡の取得停止	○	
	pdaudrm	閉塞中の監査証跡ファイルの削除	○	
	pdaudswap	現用の監査証跡ファイルのスワップ	○	
	pdaudatld	監査証跡表の自動データロード	○	
	pdaudput	JP1/NETM/Audit用監査ログ出力ユティリティ	○	
CONNECT関連セキュリティ機能	pdacunlck	連続認証失敗アカウントロック状態の解除	○	なし
トランザクションキューイング機能	pdtrnqing	トランザクションキューイング機能の開始，解除	○	なし
リアルタイムSANレプリケーション	pdrisechk	リアルタイムSANレプリケーションの構成確認	○	-l，-n，-rオプションは指定できません。
	pdrisedbto	リアルタイムSANレプリケーションのデータベース引き継ぎ	×	ログ同期方式は未サポートのため，指定できません。
	pdriseset	リアルタイムSANレプリケーションのサイト状態の設定	×	
インメモリデータ処理	pdmemdb	インメモリデータ処理に関する操作	×	なし
SQLトレースの取得	pdclttrc	SQLトレースの動的取得	○	なし

分類	ユティリティ 運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
SQL オブジェクトの情報表示	pdobils	SQL オブジェクト用バッファの統計情報表示	×	なし
SQL の翻訳	pdcbl	COBOL プリプロセサ	×	なし
	pdcpp	C プリプロセサ	×	
	pdocb	OOCOBOL プリプロセサ	×	
	pdocc	C++プリプロセサ	×	
SDB データベースの生成・操作	pdinit	データベース初期設定ユティリティ	○	「10.2　データベース初期設定ユティリティ（pdinit）」を参照してください。
	pdsdbdef	HiRDB/SD 定義ユティリティ	○	「11.　HiRDB/SD 定義ユティリティ（pdsdbdef）」を参照してください。
	pddef	データベース定義ユティリティ	○	次に示す定義系 SQL が実行できます。 • CREATE SCHEMA • DROP SCHEMA • GRANT • REVOKE • CREATE CONNECTION SECURITY • DROP CONNECTION SECURITY • CREATE AUDIT • DROP AUDIT
	pdsdblod	HiRDB/SD データベース作成ユティリティ	○	「12.　HiRDB/SD データベース作成ユティリティ（pdsdblod）」を参照してください。
	pdload	データベース作成ユティリティ	○	SDB データベースに対するデータロードは，HiRDB/SD データベース作成ユティリティ（pdsdblod）で実行します。 監査証跡表に監査証跡のデータを登録する際に，データベース作成ユティリティ（pdload）を実行します。
	pdparaload	パラレルローディング	×	なし
	pdsdbexe	HiRDB/SD データベースアクセスユティリティ	○	「14.　HiRDB/SD データベースアクセスユティリティ（pdsdbexe）」を参照してください。

分類	ユティリティ 運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
	pdsql	会話型 SQL 実行ユティリティ	○	次に示す SQL が実行できます。 • CREATE SCHEMA • DROP SCHEMA • GRANT • REVOKE • CREATE CONNECTION SECURITY • DROP CONNECTION SECURITY • CONNECT • DISCONNECT • COMMIT • ROLLBACK
	pddefrev	定義系 SQL の生成	×	なし
データベースの運用	pdsdbrog	HiRDB/SD データベース再編成ユティリティ	○	「13.　HiRDB/SD データベース再編成ユティリティ（pdsdbrog）」を参照してください。
	pdrorg	データベース再編成ユティリティ	○	ディクショナリ表の再編成，および監査証跡表のアンロードの際に実行します。 SDB データベースの再編成は，HiRDB/SD データベース作成ユティリティ（pdsdblod）および HiRDB/SD データベース再編成ユティリティ（pdsdbrog）で実行します。
	pdmod	データベース構成変更ユティリティ	○	データベース構成変更ユティリティ使用時の制限については，「10.3 データベース構成変更ユティリティ（pdmod）」を参照してください。
	pdexp	ディクショナリ搬出入ユティリティ	×	なし
	pdrbal	リバランスユティリティ	×	
	pdreclaim	空きページ解放ユティリティ	×	
	pdpgbfon	グローバルバッファ常駐化ユティリティ	○	「10.5　グローバルバッファ常駐化ユティリティ（pdpgbfon）」を参照してください。
	pdconstck	整合性チェックユティリティ	×	なし
チューニング	pdstedit	統計解析ユティリティ	○	なし

分類	ユティリティ 運用コマンド	内容	実行可否	留意事項およびマニュアル参照先
	pddbst	データベース状態解析ユティリティ	○	SDB データベースを格納する RD エリアに対して，RD エリア単位の状態解析の論理的解析および物理的解析だけが実行できます。 データベース状態解析ユティリティ使用時の制限については，「10.7 データベース状態解析ユティリティ(pddbst)」を参照してください。
	pdgetcst	最適化情報収集ユティリティ	×	なし
	pdvwopt	アクセスパス表示ユティリティ	×	
データベースの障害対策	pdcopy	データベース複写ユティリティ	○	なし
	pdbkupls	バックアップファイルの情報表示	○	「10.9.2　pdbkupls（バックアップファイルの情報表示)」を参照してください。
	pdrstr	データベース回復ユティリティ	○	なし
プラグイン関連	pdplgrgst	プラグインの登録	×	なし
	pdplgset	プラグインのセットアップ	×	
	pdreginit	レジストリ機能初期設定ユティリティ	×	

(凡例)

○：HiRDB/SD で実行できるコマンドです。

×：HiRDB/SD で実行できないコマンドです。実行した場合の動作は保証しません。

なし：留意事項およびマニュアルの参照先はありません。

10.1.3　OS やほかのプログラムが提供しているコマンド名と HiRDB/SD のコマンド名が同じ場合

OS やほかのプログラムが提供しているコマンド名と，HiRDB/SD のコマンド名（運用コマンド名またはユティリティ名）が同じ場合，HiRDB/SD のコマンドが実行できないことがあります。このような場合，次に示すどちらかの方法で対処してください。

- 環境変数の設定を HiRDB/SD のコマンド優先にする
- 絶対パスを指定して HiRDB/SD のコマンドを実行する

上記の 2 つの対処が実施できない場合，HiRDB/SD のコマンド名を変更する（別名で実行する）シェルスクリプトを作成してください。次に示すシェルスクリプトを作成すると，HiRDB/SD のコマンド名を変更できます。オプションもすべて同様に指定できます。

Bourne シェルの場合

```
#!/bin/sh -f

#set HiRDB/SD command here
HiRDBSD_COMMAND=XX....XX   ←ここに変更したいHiRDB/SDのコマンド名を記述します。

${PDDIR}/bin/${HiRDBSD_COMMAND}  $*
exit $?
```

C シェルの場合

```
#!/bin/csh -f

#set HiRDB/SD command here
set HiRDBSD_COMMAND=XX....XX  ←ここに変更したいHiRDB/SDのコマンド名を記述します。

${PDDIR}/bin/${HiRDBSD_COMMAND}  $argv
exit $status
```

[説明]

- 作成したシェルスクリプトのファイル名は，HiRDB/SD のコマンド名とは異なる名称にしてください。ファイル名が変更したコマンド名になります。
- HiRDB 運用ディレクトリ下の bin または lib の 2 つのディレクトリは，アンインストール時にディレクトリごと削除されるおそれがあるため，この 2 つのディレクトリ下にシェルスクリプトファイルを作成しないでください。
- シェルスクリプトファイルを格納したディレクトリを環境変数 PATH または path に設定してください。

10.2　データベース初期設定ユティリティ（pdinit）

データベース初期設定ユティリティは，HiRDB のデータベースとして使用するために，ファイルの物理構成を設定するユティリティです。

データベース初期設定ユティリティ（pdinit）の機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

10.2.1　create rdarea 文

create rdarea 文のオペランドについて説明します。

項番	オペランド
1	create 〔shared〕
2	rdarea RD エリア名
3	for {masterdirectory | datadirectory | datadictionary 〔of routines〕 | user used by {PUBLIC | 認可識別子〔，認可識別子〕…} | LOB used by {HiRDB（ディクショナリ表の名称） | PUBLIC | 認可識別子〔，認可識別子〕…} | list}
4	〔server name サーバ名〕
5	〔data model {structured | relational}〕
6	〔open attribute {INITIAL | DEFER | SCHEDULE}〕
7	〔page ページ長 characters〕
8	〔storage control segment セグメントサイズ pages〕
9	〔max entries 最大リスト登録数〕
10	〔extension {use 増分セグメント数 segments | nouse}〕
11	file name "HiRDB ファイルシステム領域名/HiRDB ファイル名"
12	initial HiRDB ファイル内セグメント数 segments
−	〔file name "HiRDB ファイルシステム領域名/HiRDB ファイル名"
	initial HiRDB ファイル内セグメント数 segments〕…
	;

注

項番 11，12 は，必ず create rdarea 文の最後に指定してください。

ここでは，項番 5 の data model オペランドについてだけ説明します。そのほかのオペランドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

●data model　{structured ｜ relational}

RD エリアに格納するデータベースの形式を指定します。

structured：

RD エリアに SDB データベースを格納する場合に指定します。

relational：

RD エリアにリレーショナル DB を格納する場合に指定します。次の RD エリアを定義する場合はこのオペランドを省略するか，または relational を指定してください。

- システム用 RD エリア
- 追い付き反映キー対応表を格納する RD エリア
- 追い付き反映キー対応表のインデクスを格納する RD エリア
- 追い付き状態管理表を格納する RD エリア
- 監査証跡表を格納する RD エリア

> **参考**
>
> システム用 RD エリアのデータベースの形式は，リレーショナル DB に分類されます。

《注意事項》

- for オペランドには公用 RD エリアの指定である user used by PUBLIC を指定してください。ただし，監査証跡表を格納する RD エリアについては，マニュアル「HiRDB システム運用ガイド」の「監査人の登録，監査証跡表を格納する RD エリアの作成，および監査証跡表の作成」の「監査証跡表を格納する RD エリアの作成」の「RD エリアの作成」を参照して，格納できる RD エリアの種類を確認してから，for オペランドを指定するようにしてください。
- RD エリア作成後に格納するデータベースの形式を変更する場合は，データベース構成変更ユティリティ（pdmod）で RD エリアを削除してから，RD エリアを追加してください。

10.2.2　HiRDB ファイルのセグメント数の指定

データベース初期設定ユティリティ（pdinit）では，create rdarea 文の initial オペランドに，HiRDB ファイルのセグメント数を指定します。

データディクショナリ用 RD エリアを構成する HiRDB ファイルの総セグメント数は次の表に示す値より大きい値を指定してください。

表 10–2　データディクショナリ用 RD エリアの最小セグメント数

RD エリアの種別	最小セグメント数
データディクショナリ用 RD エリア	207

データディクショナリ用 RD エリアが複数ある場合，次に示す表の順に，セグメント数の多い RD エリアから 1RD エリアに 1 ディクショナリ表ずつ格納されます。ディクショナリ表のインデクスは，対応するディクショナリ表と同じ RD エリアに格納されます。したがって，データディクショナリ用 RD エリアが複数ある場合は，格納されるディクショナリ表およびそのディクショナリ表のインデクスの数が最小セグメント数となります。

表 10–3　ディクショナリ表に定義されているインデクス数

格納順序	ディクショナリ表の名称	インデクス数
1	SQL_PHYSICAL_FILES	2
2	SQL_RDAREAS	2
3	SQL_TABLES	2
4	SQL_COLUMNS	3
5	SQL_INDEXES	3
6	SQL_USERS	1
7	SQL_RDAREA_PRIVILEGES	2
8	SQL_TABLE_PRIVILEGES	2
9	SQL_DIV_TABLE	3
10	システムで使用する表	2
11	SQL_INDEX_COLINF	2
12	SQL_TABLE_STATISTICS	1
13	SQL_COLUMN_STATISTICS	1
14	SQL_INDEX_STATISTICS	2
15	SQL_VIEW_TABLE_USAGE	3
16	SQL_VIEWS	2
17	システムで使用する表	2
18	SQL_DIV_INDEX	2
19	SQL_DIV_COLUMN	2
20	SQL_REFERENTIAL_CONSTRAINTS	3
21	SQL_ALIASES	1
22	SQL_DATATYPES	3

格納順序	ディクショナリ表の名称	インデクス数
23	SQL_DATATYPE_DESCRIPTORS	2
24	SQL_TABLE_RESOURCES	3
25	SQL_PLUGINS	3
26	SQL_PLUGIN_ROUTINES	4
27	SQL_PLUGIN_ROUTINE_PARAM	3
28	SQL_INDEX_TYPES	2
29	SQL_INDEX_RESOURCES	2
30	SQL_INDEX_DATATYPE	2
31	SQL_INDEX_FUNCTION	2
32	SQL_TYPE_RESOURCES	3
33	SQL_INDEX_TYPE_FUNCTION	1
34	SQL_EXCEPT	3
35	システムで使用する表	0
36	SQL_FOREIGN_SERVERS	0
37	SQL_USER_MAPPINGS	1
38	SQL_IOS_GENERATIONS	0
39	SQL_TRIGGERS	4
40	システムで使用する表	2
41	SQL_TRIGGER_COLUMNS	2
42	SQL_TRIGGER_DEF_SOURCE	2
43	SQL_TRIGGER_USAGE	4
44	SQL_PARTKEY	1
45	SQL_PARTKEY_DIVISION	1
46	SQL_AUDITS	3
47	システムで使用する表	2
48	SQL_KEYCOLUMN_USAGE	2
49	SQL_TABLE_CONSTRAINTS	2
50	SQL_CHECKS	2
51	SQL_CHECK_COLUMNS	2
52	SQL_DIV_TYPE	1
53	SQL_SYSPARAMS	1

格納順序	ディクショナリ表の名称	インデクス数
54	システムで使用する表	1
55	SQL_INDEX_XMLINF	2
56	SQL_SEQUENCES	2
57	SQL_SDB_DB	3
58	SQL_SDB_DB_VIEW	4
59	SQL_SDB_STORAGE_DB	5
60	SQL_SDB_NAME_CODE	4
61	SQL_SDB_DIR	3
62	SQL_ROUTINES	4
63	SQL_ROUTINE_RESOURCES	4
64	SQL_ROUTINE_PARAM	3

10.3　データベース構成変更ユティリティ（pdmod）

データベース構成変更ユティリティ（pdmod）は，RD エリアの追加，拡張，または再初期化など，データベースの物理構成を変更するためのユティリティです。

データベース構成変更ユティリティ（pdmod）の機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

10.3.1　RD エリアの追加

create rdarea 文のオペランドについて説明します。

項番	オペランド
1	create 〔shared〕
2	rdarea RD エリア名
3	〔globalbuffer グローバルバッファ名〕
4	for {datadictionary of {routines | dbmanagement} | user used by {PUBLIC | 認可識別子〔，認可識別子〕…} | LOB used by {HiRDB（ディクショナリ表の名称） | PUBLIC | 認可識別子〔，認可識別子〕…} | list}
5	〔server name サーバ名〕
6	〔data model {structured | relational}〕
7	〔open attribute {INITIAL | DEFER | SCHEDULE}〕
8	〔page ページ長 characters〕
9	〔storage control segment セグメントサイズ pages〕
10	〔max entries 最大リスト登録数〕
11	〔extension {use 増分セグメント数 segments | nouse}〕
12	file name "HiRDB ファイルシステム領域名/HiRDB ファイル名"
13	initial HiRDB ファイル内セグメント数 segments
−	〔file name "HiRDB ファイルシステム領域名/HiRDB ファイル名"
	initial HiRDB ファイル内セグメント数 segments〕…
	;

注

項番 12，13 は，必ず create rdarea 文の最後に指定してください。

ここでは，項番 6 の data model オペランドについてだけ説明します。そのほかのオペランドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

●data model　{structured ｜ relational}

RD エリアに格納するデータベースの形式を指定します。

structured：

RD エリアに SDB データベースを格納する場合に指定します。

relational：

RD エリアにリレーショナル DB を格納する場合に指定します。次の RD エリアを定義する場合はこのオペランドを省略するか，または relational を指定してください。

- 追い付き反映キー対応表を格納する RD エリア
- 追い付き反映キー対応表のインデクスを格納する RD エリア
- 追い付き状態管理表を格納する RD エリア
- 監査証跡表を格納する RD エリア

> **参考**
>
> システム用 RD エリアのデータベースの形式は，リレーショナル DB に分類されます。

《注意事項》

- for オペランドには公用 RD エリアの指定である user used by PUBLIC を指定してください。ただし，監査証跡表を格納する RD エリアについては，マニュアル「HiRDB システム運用ガイド」の「監査人の登録，監査証跡表を格納する RD エリアの作成，および監査証跡表の作成」の「監査証跡表を格納する RD エリアの作成」の「RD エリアの作成」を参照して，格納できる RD エリアの種類を確認してから，for オペランドを指定するようにしてください。
- RD エリア作成後に格納するデータベースの形式を変更する場合は，データベース構成変更ユティリティ（pdmod）で RD エリアを削除してから，RD エリアを追加してください。
- globalbuffer オペランドを指定して，追加した RD エリアにグローバルバッファを割り当ててください。globalbuffer オペランドでグローバルバッファを割り当てない場合，HiRDB をいったん終了して，システム定義の pdbuffer オペランドでグローバルバッファを割り当てないと，追加した RD エリアを使用できません。

10.3.2　RD エリアの再初期化

(1)　注意事項

- SDB データベース格納定義の格納レコード用 RD エリアに指定した RD エリアに対して，RD エリアの再初期化でページ長を変更する場合は，変更前のページ長以上の値を指定してください。SDB データベース定義に，変更後のページ長に収まらないレコード長のレコード型が存在する場合，該当するレコードの格納時にエラーとなります。
- SDB データベース格納定義で SUBPAGE NUMBER 句を指定したレコード型を格納する RD エリアに対しては，RD エリアの再初期化でページ長を変更しないでください。ページ長を変更した場合，該当するレコード型のレコードの格納時や検索時などにエラーとなります。
- SDB データベースを格納する RD エリアに対しては，without lock table オペランドを指定した再初期化を行わないでください。without lock table オペランドを指定した場合，エラーとなります。

(2)　RD エリアの再初期化後のレコードおよびインデクスの状態

- **レコードを格納している RD エリアを再初期化した場合**

  再初期化した RD エリア内の全レコードが削除されます。また，レコード型に定義されているインデクスのデータも同時に削除されます。

  SDB データベースを横分割している場合は，再初期化対象の RD エリアに格納されているレコードだけが削除されます。このとき，削除されたレコードに対応するインデクスのデータも同時に削除されます。

  > **参考**
  >
  > RD エリアを再初期化した場合，レコード型およびインデクスの定義情報は削除されずに引き継がれます。

- **インデクスを格納している RD エリアを再初期化した場合**

  再初期化した RD エリア内のインデクスの全データが削除されます。ただし，レコードは削除されません。

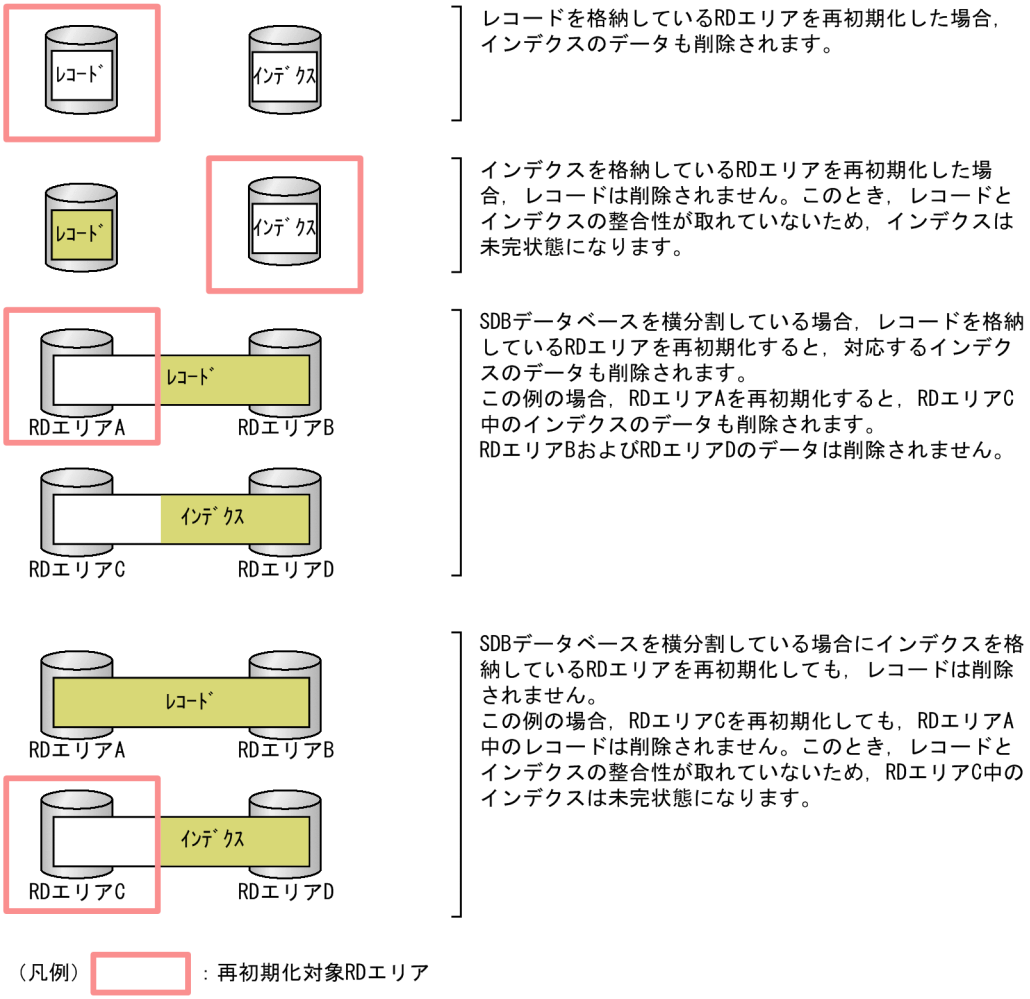
  SDB データベースを横分割している場合は，再初期化対象の RD エリアに格納されているインデクスのデータだけが削除されます。レコードは削除されません。

  このように，インデクスを格納している RD エリアを再初期化すると，レコードとインデクスの整合性が取れなくなります。この状態をインデクスの未完状態といいます。未完状態のインデクスは使用できないため，基本的にはインデクスを格納している RD エリアだけを再初期化しないようにしてください。

  インデクスを格納している RD エリアだけを再初期化してしまった場合，インデクスを格納している RD エリアを閉塞解除かつオープン状態にして，レコードを格納している RD エリアを pdmod コマンドで再初期化するか，または pdsdblod コマンドでインデクスを再作成してください。

RD エリアの再初期化後のレコードおよびインデクスの状態を次の図に示します。

図 10-1　RD エリアの再初期化後のレコードおよびインデクスの状態



(3)　実行結果リストの出力

RD エリアの再初期化を実行した場合，実行結果リストが出力されます。実行結果リストに出力される情報については，「HiRDB コマンドリファレンス」の「RD エリアの再初期化」の「実行結果リストの出力」を参照してください。

なお，「HiRDB コマンドリファレンス」の「実行結果リストの出力」で説明されている次の出力情報の表識別子には，レコード型名が表示されます。

4. 再初期化した RD エリア内の表識別子

認可識別子．表識別子

注

上記の「4.」は，「HiRDB コマンドリファレンス」の「実行結果リストの出力」の説明で使用している項番です。

10.3.3　RD エリアの削除

RD エリア内にレコードまたはインデクスがある場合（レコードまたはインデクスの定義がある場合），その RD エリアは削除できません。

10.3.4　RD エリアの属性変更

ユーザ用 RD エリアの名称は変更できません。

10.3.5　RD エリアの移動

ユーザ用 RD エリアの移動はできません。

10.4　データベース再編成ユティリティ（pdrorg）

データベース再編成ユティリティ（pdrorg）はデータベースのメンテナンスをするためのユティリティです。

データベース再編成ユティリティ（pdrorg）では，ディクショナリ表の再編成だけが実行できます。SDBデータベースを再編成する場合は，HiRDB/SD データベース再編成ユティリティ（pdsdbrog）を使用してください。

データベース再編成ユティリティ（pdrorg）の機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

10.4.1　ディクショナリ表の再編成

すべてのディクショナリ表を再編成するだけでなく，特定のディクショナリ表だけを指定して再編成することもできます。また，HiRDB/SD が使用するディクショナリ表も指定できます。

処理対象となる表の名称を-t オプションで指定します。ディクショナリ表の場合，次の形式で指定してください。詳細については，マニュアル「HiRDB コマンドリファレンス」の「データベース再編成ユティリティ（pdrorg）」の-t オプションの説明を参照してください。

```
-t　{表識別子〔，表識別子〕…}
```

すべてのディクショナリ表を再編成したい場合は，-t オプションを省略してください。特定のディクショナリ表を再編成する場合は，次の表に示す表識別子を-t オプションに指定します。なお，すべてのディクショナリ表を再編成する場合，次の表に示す順でアンロード・リロードされます。

表 10–4　-t オプションに指定するディクショナリ表の表識別子

項番	表識別子	備考
1	SQL_PHYSICAL_FILES	なし
2	SQL_RDAREAS	
3	SQL_TABLES	
4	SQL_COLUMNS	
5	SQL_INDEXES	
6	SQL_USERS	
7	SQL_RDAREA_PRIVILEGES	
8	SQL_TABLE_PRIVILEGES	
9	SQL_DIV_TABLE	

項番	表識別子	備考
10	SQL_DIV_TABLE_REGULARIZE	システムで使用する表
11	SQL_INDEX_COLINF	なし
12	SQL_TABLE_STATISTICS	
13	SQL_COLUMN_STATISTICS	
14	SQL_INDEX_STATISTICS	
15	SQL_VIEW_TABLE_USAGE	
16	SQL_VIEWS	
17	SQL_VIEW_DEF	システムで使用する表
18	SQL_DIV_INDEX	なし
19	SQL_DIV_COLUMN	
20	SQL_REFERENTIAL_CONSTRAINTS	
21	SQL_ALIASES	
22	SQL_ROUTINES	
23	SQL_ROUTINE_RESOURCES	
24	SQL_ROUTINE_PARAMS	
25	SQL_DATATYPES	
26	SQL_DATATYPE_DESCRIPTORS	
27	SQL_TABLE_RESOURCES	
28	SQL_PLUGINS	
29	SQL_PLUGIN_ROUTINES	
30	SQL_PLUGIN_ROUTINE_PARAMS	
31	SQL_REGISTRY_CONTEXT	システムで使用する表
32	SQL_REGISTRY_KEY	
33	SQL_INDEX_TYPES	なし
34	SQL_INDEX_RESOURCES	
35	SQL_INDEX_DATATYPE	
36	SQL_INDEX_FUNCTION	
37	SQL_TYPE_RESOURCES	
38	SQL_INDEX_TYPE_FUNCTION	
39	SQL_EXCEPT	
40	SQL_FOREIGN_SERVER	

項番	表識別子	備考
41	SQL_USER_MAPPINGS	
42	SQL_USAGE_PRIVILEGES	システムで使用する表
43	SQL_IOS_GENERATIONS	
44	SQL_TRIGGERS	なし
45	SQL_TRIGGER_ACTCOND	システムで使用する表
46	SQL_TRIGGER_COLUMNS	なし
47	SQL_TRIGGER_DEF_SOURCE	
48	SQL_TRIGGER_USAGE	
49	SQL_PARTKEY	
50	SQL_PARTKEY_DIVISION	
51	SQL_AUDITS	
52	SQL_AUDIT_REGULARIZE	システムで使用する表
53	SQL_KEYCOLUMN_USAGE	なし
54	SQL_TABLE_CONSTRAINTS	
55	SQL_CHECKS	
56	SQL_CHECK_COLUMNS	
57	SQL_DIV_TYPE	
58	SQL_SYSPARAMS	
59	SQL_DB_MANAGEMENT	システムで使用する表
60	SQL_DB_STATE_ANALYZED	
61	SQL_PUBLICVIEW_SAME_USERS	
62	SQL_INDEX_XMLINF	なし
63	SQL_SEQUENCES	
64	SQL_SDB_DB	
65	SQL_SDB_DB_VIEW	システムで使用する表
66	SQL_SDB_STORAGE_DB	なし
67	SQL_SDB_NAME_CODE	システムで使用する表
68	SQL_SDB_DIR	なし

10.5 グローバルバッファ常駐化ユティリティ（pdpgbfon）

グローバルバッファ常駐化ユティリティ（pdpgbfon）は HiRDB 開始直後や，オンライン業務開始前などに，SDB データベースのデータ（レコードやインデクス）をグローバルバッファにあらかじめ読み込んでおくためのユティリティです。

グローバルバッファ常駐化ユティリティ（pdpgbfon）の機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

10.5.1 オプションの説明

- -t {〔認可識別子.〕表識別子｜〔認可識別子.〕all}

  表識別子には，ルートレコードまたは仮想ルートレコードのレコード型名を指定します。常駐化の対象となる資源を次に示します。

  - -k オプションに table を指定した場合（データページを常駐化する場合）

    指定したルートレコードまたは仮想ルートレコードが定義されている SDB データベースの全レコード（ルートレコードとその子レコード，または仮想ルートレコードとその子レコード）が常駐化の対象になります。

  - -k オプションに index を指定した場合（インデクスページを常駐化する場合）

    指定したルートレコードまたは仮想ルートレコードが定義されている SDB データベースの全インデクス（シーケンシャルインデクスおよび二次インデクス）が常駐化の対象になります。

  all を指定した場合，その認可識別子（スキーマ）が所有している SDB データベースのレコード，またはインデクスのどちらかを常駐化します。-k オプションの指定によって，レコードまたはインデクスのどちらを常駐化するかが決まります。

  なお，all を指定した場合，idxname 文は指定できません。

- -b

  BINARY 型は未サポートのため，-b オプションは指定できません。指定しても無視されます。

10.5.2 ユティリティを実行できるユーザ

グローバルバッファ常駐化ユティリティ（pdpgbfon）を実行する場合は，DBA 権限が必要になります。そのため，このユティリティを実行する場合は，環境変数 PDUSER に DBA 権限を持ったユーザの認可識別子とパスワードを設定しておく必要があります。

10.6　統計解析ユティリティ（pdstedit）

統計解析ユティリティ（pdstedit）は，HiRDB の稼働情報など，統計情報を編集・表示するためのユティリティです。

統計解析ユティリティ（pdstedit）の機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

10.6.1　対象となる統計情報

次に示す統計情報が対象となります。

- システムの稼働に関する統計情報
- グローバルバッファプールに関する統計情報
- インデクスに関する統計情報
- データベース操作に関する HiRDB ファイルの統計情報
- デファードライト処理に関する統計情報
- UAP に関する統計情報
- データベースの入出力に関する統計情報

10.6.2　UAP に関する統計情報（標準出力）

UAP に関する統計情報に出力される情報を次に示します。

```
pdstedit 09-60(Object Option) ***** UAP INFORMATION *****
INPUT          :/tmp/stjdata
OUTPUT RANGE   :**/**/** **:**:** - **/**/** **:**:**
--------------------------------------------------------------------------------
HOST = test
--------------------------------------------------------------------------------
EDIT TIME 2015/11/29 16:00:00 - 2015/11/29 17:00:00
UAP NAME
  SERVICE NAME
                                EXEC  NORM  ERROR TTIME TAVG  CACHE LOCK
-----------------------------   ----- ----- ----- ----- ----- ----- -----
                                LMAX  LAVG  CTIME CTAVG SVEXT SVEXA
                                ----- ----- ----- ----- -----
   *SQL OBJECT*                 REQ   RAVG  HITS  HAVG  CRT   CAVG  SMAX
                                ----- ----- ----- ----- ----- ----- -----
   *LN1* COMT  ROLB  FROW  DROW  IROW  UROW  SET   OPEN  FETC  CLOS  DESC
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   *LN2* SEL   INS   UPD   DEL   LOCK  CRTT  DRPT  ALTT  CRTI  DRPI  CMTT
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
```

```
   *LN3* CMTC  CRTS  DRPS  GRTR  GRTS  GRTA  GRTC  GRTD  RVKR  RVKS  RVKA
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   *LN4* RVKC  RVKD  CRTV  DRPV  PRGT  CRTP  DRPP  ALTP  CALL  DESI  MISC
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   *STORED ROUTINE OBJECT*       REQ   RAVG  HITS  HAVG
                                 ----- ----- ----- -----
   *DB ACCESS*                   MAXIO MINIO MWFN  MWFEC MWFVL
                                 ----- ----- ----- ----- -----
                                 MWHTS MBSL1 MBSL2 MBSL3
                                 ----- ----- ----- -----
   *LN1* IOTIM DIDRC DIDUC DIDHC DIDHR DIDRD DIDWT LBRFC LBUPC LBRHC LBUHC
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   *LN2* LRFHR LUPHR LBRDC LBWTC BFSHC BRDWC BWTWC BLKWC WFRDC WFWTC WBFOC
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   *LN3* SCHSK SCHCH LDIRC LDIUC LDIHC LDIRD LDIWT LBFSHC ARREQ ARWC  ARWT
         ----- ----- ----- ----- ----- ----- ----- -----  ----- ----- -----
   *LN4* ARSTA HJMAX HJAVG TUSEG COMPC COMPT CMPMT EXTEC EXTET EXTMT
         ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   *THREAD LOCK*
   *LN1* LCKTC LCKWC LCKWT LCKTM LCKWQ LCKQM JBFTC JBFWC JBFWT JBFTM JBFWQ
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
   *LN2* JBFQM BUFTC BUFWC BUFWT BUFTM BUFWQ BUFQM
         ----- ----- ----- ----- ----- ----- -----
   *SDB ACCESS*
         [  1] [  2] [  3] [  4] [  5] [  6] [  7] [  8] [  9] [010] [ 11]
   *LN1* DMLC  DMLS  DMLE  DMLFT DMLER DMLST DMLMO DMLDE DMLIF VRTFT VRTST
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
         [ 12] [ 13] [ 14] [ 15] [ 16] [ 17] [ 18] [ 19] [020] [ 21] [ 22]
   *LN2* VRTMO ONYS  ONYE  SDBFT SDBER SDBST SDBMO SDFTR SDERR SDSTR SDMOR
         ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
         [ 23] [ 24] [ 25] [ 26]
   *LN3* SDTFC DMLGT SDBGT SDGTR
         ----- ----- ----- -----
Unknown
  ******************************
                                    68    68     0 40.6M  597k     0     0
                                     0     0     0     0    22     0
   *SQL OBJECT*                      0     0     0     0     0     0     0
   *LN1*     3     0     0     0     0     0     0     0     0     0     0 [ 27]
      **     0     0     0     0     0     0     0     0     0     0     0 [ 28]
   *LN2*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN4*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *STORED ROUTINE OBJECT*           0     0     0     0
   *DB ACCESS*                       0     0     0     0     0
                                     0     0     0     0
   *LN1*     0     3     0     2    67     1     0     0     0     0     0
      **     0     0     0     0           0     0     0     0     0     0
   *LN2*     0     0     0     0     1     0     0     0     0     0     0
      **                 0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0     0     0     0     0     0     0     0
      **                 0     0     0     0     0     0     0     0     0
   *LN4*     0     0     0     0     0     0     0     0     0     0
      **     0
   *THREAD LOCK*
```

```
   *LN1* 1.08k     0     0     0     0     0    12     0     0     0     0
      **                 0           0                       0           0
   *LN2*     0 46.5k     0     0     0     0     0
      **                       0           0
   *SDB ACCESS*
   *LN1*     9     1     0     1     0     0     0     0     0     1     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN2*     0     1     0     2     0     0     0     2     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0
      **     0     0     0     0
pdsdblod
  ******************************
                                  1     1     0     9     9 9.38k     0
                                  0     0     0     0     7     7
   *SQL OBJECT*                   5     5     1     1     4     4 2.23k
   *LN1*     4     0     5     0     0     0     1     1     1     1     0
      **     4     0     5     0     0     0     1     1     1     1     0
   *LN2*     4     0     0     0     0     0     0     0     0     0     0
      **     4     0     0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN4*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *STORED ROUTINE OBJECT*        0     0     0     0
   *DB ACCESS*                    0     0     0     0     0
                                  0     0     0     0
   *LN1*     0   170     0   143    84    27     0     0     0     0     0
      **     0   170     0   143          27     0     0     0     0     0
   *LN2*     0     0     0     0     0     0     0     0     0     0     0
      **                 0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0     0     0     0     0     0     0     0
      **                 0     0     0     0     0     0     0     0     0
   *LN4*     0     0     0     0     0     0     0     0     0     0
      **     0
   *THREAD LOCK*
   *LN1* 1.08k     0     0     0     0     0    12     0     0     0     0
      **                 0           0                       0           0
   *LN2*     0 46.5k     0     0     0     0     0
      **                       0           0
   *SDB ACCESS*
   *LN1*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN2*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0
      **     0     0     0     0
*TOTAL* [ 29]                     69    69     0 40.6M  588k   139     0
                                  0     0     0     0    29     0
   *SQL OBJECT*                   5     0     1     0     4     0 2.23k
   *LN1*     7     0     5     0     0     0     1     1     1     1     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN2*     4     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN4*     0     0     0     0     0     0     0     0     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
```

```
   *STORED ROUTINE OBJECT*          0     0     0     0
   *DB ACCESS*                      0     0     0     0     0
                                    0     0     0     0
   *LN1*     0   173     0   145    84    28     0     0     0     0     0
      **     0     3     0     2           0     0     0     0     0     0
   *LN2*     0     0     0     0     1     0     0     0     0     0     0
      **                 0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0     0     0     0     0     0     0     0
      **                 0     0     0     0     0     0     0     0     0
   *LN4*     0     0     0     0     0     0     0     0     0     0
      **     0
   *THREAD LOCK*
   *LN1* 2.16k     0     0     0     0     0    24     0     0     0     0
      **                 0           0                       0           0
   *LN2*     0 93.0k     0     0     0     0     0
      **                       0           0
   *SDB ACCESS*
   *LN1*     9     1     0     1     0     0     0     0     0     1     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN2*     0     1     0     2     0     0     0     2     0     0     0
      **     0     0     0     0     0     0     0     0     0     0     0
   *LN3*     0     0     0     0
      **     0     0     0     0
--------------------------------------------------------------------------------
FILE KIND   LOG KIND   FIRST                   LAST                          NUM
STJ         uap        2015/11/29 16:43:26   2015/11/29 16:45:07            23
--------------------------------------------------------------------------------

NO FILE KIND:LOG FILE NAME
  LOG KIND             FIRST                   LAST                          NUM
 1 STJ64    :/tmp/stjdata/pdstj2
  sys                  2015/11/29 16:44:26   2015/11/29 16:44:26             4
  uap                  2015/11/29 16:43:26   2015/11/29 16:45:07            23
  sql                  2015/11/29 16:43:26   2015/11/29 16:45:07            79
  sop                  2015/11/29 16:43:43   2015/11/29 16:43:43             1
  dop                  2015/11/29 16:43:26   2015/11/29 16:45:06            47
  pcd                  2015/11/29 16:43:26   2015/11/29 16:45:07           172
  obj                  ****/**/** **:**:**   ****/**/** **:**:**             0
  sqh                  2015/11/29 16:43:26   2015/11/29 16:45:07            35
  buf                  2015/11/29 16:43:26   2015/11/29 16:45:15           422
  fil                  2015/11/29 16:43:26   2015/11/29 16:45:15           422
  dfw                  2015/11/29 16:43:26   2015/11/29 16:45:15             3
  idx                  2015/11/29 16:43:26   2015/11/29 16:45:15             3
  fsv                  ****/**/** **:**:**   ****/**/** **:**:**             0
  hba                  ****/**/** **:**:**   ****/**/** **:**:**             0
  dio                  ****/**/** **:**:**   ****/**/** **:**:**             0
```

注

［1］～［29］は，SDB データベースへのアクセス時にだけ出力される情報です。SDB データベースにアクセスしていないときは，「****」が出力されます。

［説明］

［1］～［29］以外の情報については，マニュアル「HiRDB コマンドリファレンス」の「UAP に関する統計情報」の説明を参照してください。

1. SDB データベースを操作する API または DML の実行を要求した回数

2. API（個別開始）の実行を要求した回数，または DML の実行要求に対してシステムが個別開始の実行を要求した回数
3. API（個別終了）の実行を要求した回数
4. API または DML（FETCH，FIND，または FETCHDB ALL）の実行を要求した回数
5. API または DML（ERASE）の実行を要求した回数
6. API または DML（STORE）の実行を要求した回数
7. API または DML（MODIFY）の実行を要求した回数
8. API（一括削除）の実行を要求した回数
9. API（容量情報取得）の実行を要求した回数
10. API（FETCH，STORE，一括削除，または容量情報取得）の実行要求に対して，仮想ルートレコードに FETCH を実行した回数※
11. API（STORE）の実行要求に対して，仮想ルートレコードに STORE を実行した回数※
12. API（STORE，または一括削除）の実行要求に対して，仮想ルートレコードに MODIFY を実行した回数※
13. API（個別開始）の実行要求に対して，実際に個別開始を実行した回数，または DML の実行要求に対するシステムによる個別開始の実行要求に対して，実際に個別開始を実行した回数
14. API（個別終了）の実行要求に対して，実際に個別終了を実行した回数
15. API または DML（FETCH，FIND，または FETCHDB ALL）の実行要求に対して，実際に FETCH，または FETCHDB ALL を実行した回数※
16. API または DML（ERASE）の実行要求に対して，実際に ERASE を実行した回数※
17. API または DML（STORE）の実行要求に対して，実際に STORE を実行した回数※
18. API または DML（MODIFY）の実行要求に対して，実際に MODIFY を実行した回数※
19. API または DML（FETCH，FIND，または FETCHDB ALL）の実行要求に対して，実際に FETCH，または FETCHDB ALL を実行して取得したレコード件数※
20. API または DML（ERASE）の実行要求に対して，実際に ERASE を実行して削除したレコード件数※

    削除対象のレコードの子レコードも削除された場合，そのレコードも数に含まれます。
21. API または DML（STORE）の実行要求に対して，実際に STORE を実行して追加したレコード件数※
22. API または DML（MODIFY）の実行要求に対して，実際に MODIFY を実行して更新したレコード件数※
23. SDB データベースを操作する API の実行要求時，HiRDB/SD が内部的に実行する終端検知用の（レコードの検索時に最終のレコードかどうかを判定するための），FETCH または FETCHDB ALL を実行した回数

24. DML（GET）の実行を要求した回数

25. DML（GET）の実行要求に対して，実際に GET を実行した回数

26. DML（GET）の実行要求に対して，実際に GET を実行して取得したレコード件数

27. 1 回の UAP，またはトランザクション実行ごとの，各項目の合計値

28. 1 回の UAP，またはトランザクション実行ごとの，各項目の平均値

29. 全体の各項目の合計値

注※

API または DML の実行要求に対して，FETCH，FETCHDB ALL，STORE，MODIFY，または ERASE が複数回実行された場合，実行された回数分，実行回数およびレコード件数がカウントされます。複数回実行する途中でエラーが発生した場合は，エラーが発生するまでの実行回数，レコード件数となります。

10.6.3 UAP に関する統計情報（DAT 形式ファイル）

UAP に関する統計情報を DAT 形式ファイルに出力する場合の出力情報を次の表に示します。

SDB データベースにアクセスしていないときは，「**********」（-e オプションに er1 を指定した場合は「*」）が出力されます

なお，これ以外の出力情報については，マニュアル「HiRDB コマンドリファレンス」の「DAT 形式ファイルのレコード形式（UAP に関する統計情報）」を参照してください。

表 10-5 DAT 形式ファイルのレコード形式（UAP に関する統計情報）

項番	フィールド名（タイトルバー）	属性	最大長
1	SDB データベースを操作する API または DML の実行を要求した回数（DML COUNT）	数値	10
2	API（個別開始）の実行を要求した回数，または DML の要求に対してシステムが個別開始の実行を要求した回数（DML START COUNT）		
3	API（個別終了）の実行を要求した回数（DML END COUNT）		
4	API または DML（FETCH，FIND，または FETCHDB ALL）の実行を要求した回数（DML FETCH COUNT）		
5	API または DML（ERASE）の実行を要求した回数（DML ERASE COUNT）		
6	API または DML（STORE）の実行を要求した回数（DML STORE COUNT）		
7	API または DML（MODIFY）の実行を要求した回数（DML MODIFY COUNT）		
8	API（一括削除）の実行を要求した回数（DML PURGE COUNT）		
9	API（容量情報取得）実行要求をした回数（DML INFO COUNT）		
10	API（FETCH，STORE，一括削除，または容量情報取得）の実行要求に対して，仮想ルートレコードに FETCH を実行した回数（VROOT FETCH COUNT）		

項番	フィールド名（タイトルバー）	属性	最大長
11	API（STORE）の実行要求に対して，仮想ルートレコードに STORE を実行した回数（VROOT STORE COUNT）		
12	API（STORE または一括削除）の実行要求に対して，仮想ルートレコードに MODIFY を実行した回数（VROOT MODIFY COUNT）		
13	API（個別開始）の実行要求に対して，実際に個別開始を実行した回数，または DML の実行要求に対するシステムによる個別開始の実行要求に対して，実際に個別開始を実行した回数（ONLY START COUNT）		
14	API（個別終了）の実行要求に対して，実際に個別終了を実行した回数（ONLY END COUNT）		
15	API または DML（FETCH，FIND，または FETCHDB ALL）の実行要求に対して，実際に FETCH，または FETCHDB ALL を実行した回数（FETCH COUNT）		
16	API または DML（ERASE）の実行要求に対して，実際に ERASE を実行した回数（ERASE COUNT）		
17	API または DML（STORE）の実行要求に対して，実際に STORE を実行した回数（STORE COUNT）		
18	API または DML（MODIFY）の実行要求に対して，実際に MODIFY を実行した回数（MODIFY COUNT）		
19	API または DML（FETCH，FIND，または FETCHDB ALL）の実行要求に対して，実際に FETCH，または FETCHDB ALL を実行して取得したレコード件数（FETCH RECORD）		
20	API または DML（ERASE）の実行要求に対して，実際に ERASE を実行して削除したレコード件数（ERASE RECORD）		
21	API または DML（STORE）の実行要求に対して，実際に STORE を実行して追加したレコード件数（STORE RECORD）		
22	API または DML（MODIFY）の実行要求に対して，実際に MODIFY を実行して更新したレコード件数（MODIFY RECORD）		
23	API の実行要求に対して，HiRDB/SD が内部的に実行した終端検知用の（レコードの検索時に最終のレコードかどうかを判定するための），FETCH または FETCHDB ALL を実行した回数（TERMINATION FETCH COUNT）		
24	DML（GET）の実行を要求した回数（DML GET COUNT）		
25	DML（GET）の実行要求に対して，実際に GET を実行した回数（GET COUNT）		
26	DML（GET）の実行要求に対して，実際に GET を実行して取得したレコード件数（GET RECORD）		

10.7　データベース状態解析ユティリティ（pddbst）

データベース状態解析ユティリティ（pddbst）は，データベースの状態解析を実行するためのユティリティです。

データベース状態解析ユティリティ（pddbst）の機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

10.7.1　実行できる機能

SDB データベースに対しては，データベース状態解析ユティリティ（pddbst）の次に示す機能だけが実行できます。

- RD エリア単位の状態解析（論理的解析）
- RD エリア単位の状態解析（物理的解析）

10.7.2　コマンドの形式およびオプションの説明

(1)　コマンドの形式

データベース状態解析ユティリティ（pddbst）の指定形式を次に示します。

RD エリア単位の状態解析

```
pddbst -r  {RDエリア名〔, RDエリア名〕…|ALL}
      〔-k  {logi  〔-d〕  〔-z〕  〔 {-a  〔-h〕 |-A 〔 {r|i} 〕} 〕
           |phys  〔-f〕  〔-a  〔-h〕〕} 〕
      〔-u 認可識別子  〔-p パスワード〕〕
      〔-q 世代番号〕  〔-b〕
      〔-X サーバ間通信の応答監視時間〕
      〔-v 制御文ファイル名〕
```

データベース状態解析ユティリティ（pddbst）の指定形式およびオプションの詳細については，マニュアル「HiRDB コマンドリファレンス」の「データベース状態解析ユティリティ（pddbst）」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差があるオプションがあります。ここでは，その仕様差があるオプションについてだけ説明します。

●-A〔{r ｜ i}〕

SDB データベースを格納している RD エリアの論理的解析の解析結果を，DAT 形式で標準出力に表示する場合に指定します。

r：

ルートレコード型を格納している RD エリアの論理的解析の解析結果を表示する場合に指定します。

インデクスを格納している RD エリアに対してこのオプションを指定した場合，その RD エリアは解析対象にはなりません（処理をスキップします）。

i：

SDB データベースのインデクスを格納している RD エリアの論理的解析の解析結果を表示する場合に指定します。

レコード型を格納している RD エリアに対してこのオプションを指定した場合，その RD エリアは解析対象にはなりません（処理をスキップします）。

-A オプションだけを指定した場合（r オプションおよび i オプションの両方を省略した場合）は，ルートレコード型を格納している RD エリアの解析結果と，インデクスを格納している RD エリアの解析結果の両方を表示します。

(2) 注意事項

- SDB データベースを格納している RD エリアに対しては，-a オプションおよび-h オプションを指定することはできません。-a オプションおよび-h オプションは，リレーショナル DB を格納している RD エリアを解析するときに指定します。
- -A オプションと-a オプションを同時に指定することはできません。
- -a オプションが指定されていない場合，-h オプションの指定は無視されます。

10.7.3 RD エリア単位の状態解析（論理的解析）

RD エリア内の SDB データベースに対する，全セグメントおよび全ページの格納状態を解析します。ただし，複数の RD エリアに分割格納された SDB データベースについては，指定した RD エリアに格納された部分だけを解析します。

なお，ここではレコードが格納されている RD エリアに対して論理的解析を実行したときに出力される情報について説明します。

RD エリアにインデクスが格納されているときに出力される情報については，マニュアル「HiRDB コマンドリファレンス」の「RD エリア単位の状態解析（論理的解析）」を参照してください。その際，インデクスに関する情報の RowID Count に表示される「インデクス管理行数」については，次の留意事項があります。

- シーケンシャルインデクスの場合，仮想ルートレコードに対しては，1 レコード実現値当たり最大 2 つのキー値を持ちます。そのため，インデクス管理行数とレコード実現値数が 1 対 1 とはなりません。

(1) 使用目的

RD エリア内のレコードおよびインデクスの乱れの検知，容量の状態を確認できます。

(2) 解析結果

RD エリア単位の状態解析（論理的解析）の出力例を次の図に示します。

図 10-2 RD エリア単位の状態解析（論理的解析）の出力例（サブページ分割機能を適用していない場合）

```
pddbst 09-00(Object Option) ** RD Area Logical Analysis **  2015/03/06 18:32:28
RD Area Name    : RDAREA52
 Server         : testbesm
 Total Segment :         100   Segment Size :          10 Pages
 Unused Segment:          82   Page Size    :        4096 Bytes
 Original RD Area Name : RDAREA52
 Generation Number :  0   Replica RD Area Count :  0
 History1 Hold Status :       Hold Code :     0  Hold Time :
 History2 Hold Status :       Hold Code :     0  Hold Time :
--------------------------------------------------------------------------------
 SDB database kind: FMB
 Root Record Name : FMB_ROOT_RCD
 Auth Id          : PT1
 SubPage Number   :  1
 <Common information>
         Used(Full)        Used(      Full)         Sum
 Segment 100%(  0%)          18(         0)          18
 Page    100%( 50%)         180(        89)         180
 <Normal Page information>
         Used(Full(Occupied))       Used(      Full(  Occupied))         Sum
 Page    100%( 47%(    100%))        166(        77(         77))         166
 Collect Prearranged Page :          0
 Used Page Ratio       Page(Ratio)       Occupied Page
            0% :          0(   0%)                   0
          1- 10% :        145(  88%)                  77
         11- 20% :         10(   6%)                   0
         21- 30% :          8(   5%)                   0
         31- 40% :          2(   2%)                   0
         41- 50% :          1(   1%)                   0
         51- 60% :          0(   0%)                   0
         61- 70% :          0(   0%)                   0
         71- 80% :          0(   0%)                   0
         81- 90% :          0(   0%)                   0
         91-100% :          0(   0%)                   0
 Total                    166                         77
 Stored Family   :          2
 Stored Children:                   200
 Page Change     :                  164
 Collection Degree
      Page       :  18%
      Segment    :  92%
 Virtual Root Record(disuse):                     0
 Logical Cleared Record     :                     0
 Del RowID    :                    0
```

「(a)基本情報」を参照

「(b)Common information」を参照

「(c)Normal Page information」を参照

（続く）

（続き）

```
<Reserved Page information>
        Used(Full)        Used(       Full)         Sum
Page     100%( 86%)          14(         12)          14
NonContinuous Reserved Pages          :           0
Used Page Ratio       Page(Ratio)
           0% :           0(   0%)
        1- 10% :           0(   0%)
       11- 20% :           0(   0%)
       21- 30% :           0(   0%)
       31- 40% :           2(  15%)
       41- 50% :           0(   0%)
       51- 60% :           0(   0%)
       61- 70% :           0(   0%)
       71- 80% :           0(   0%)
       81- 90% :           0(   0%)
      91-100% :          12(  86%)
Total                    14
Stored Family  :           2
Stored Children:                    140
Page Change    :                     12
Collection Degree
     Page      :  92%
     Segment   : 100%
Del RowID   :                         0
```

「(d)Reserved Page information」を参照

（凡例）：pddbstコマンドの-dオプションを指定した場合に出力される情報です。

：pddbstコマンドの-dオプションおよび-zオプションを指定した場合に出力される情報です。

図 10-3　RD エリア単位の状態解析（論理的解析）の出力例（サブページ分割機能を適用している場合）

```
pddbst 09-00(Object Option) ** RD Area Logical Analysis **  2015/03/06 18:50:22
RD Area Name    : RDAREA52
 Server         : testbesm
 Total Segment :         100   Segment Size :          10 Pages
 Unused Segment:          97   Page Size    :        4096 Bytes
 Original RD Area Name : RDAREA52
 Generation Number :  0   Replica RD Area Count :  0
 History1 Hold Status :      Hold Code :     0  Hold Time :
 History2 Hold Status :      Hold Code :     0  Hold Time :
--------------------------------------------------------------------------------
SDB database kind: FMB
Root Record Name : FMB_ROOT_RCD
Auth Id          : PT1
SubPage Number   : 16
SubPage Size     :   252 Bytes
<Common information>
        Used(Full)       Used(      Full)        Sum
Segment 100%( 67%)          3(         2)          3
Page     77%( 74%)         23(        22)         30
<Normal SubPage information>
         Used(Full(Occupied))        Used(       Full(   Occupied))         Sum
SubPage   60%( 29%(     94%))         166(         79(          74))         280
Collect Prearranged Page :          0
Used SubPage Ratio        SubPage(Ratio)        Occupied SubPage
               0% :              0(   0%)                       0
             1- 10% :              0(   0%)                       0
            11- 20% :              0(   0%)                       0
            21- 30% :            140(  85%)                      74
            31- 40% :              0(   0%)                       0
            41- 50% :              0(   0%)                       0
            51- 60% :              0(   0%)                       0
            61- 70% :             16(  10%)                       0
            71- 80% :              0(   0%)                       0
            81- 90% :              0(   0%)                       0
           91-100% :              10(   6%)                       0
   Total                         166                            74
```

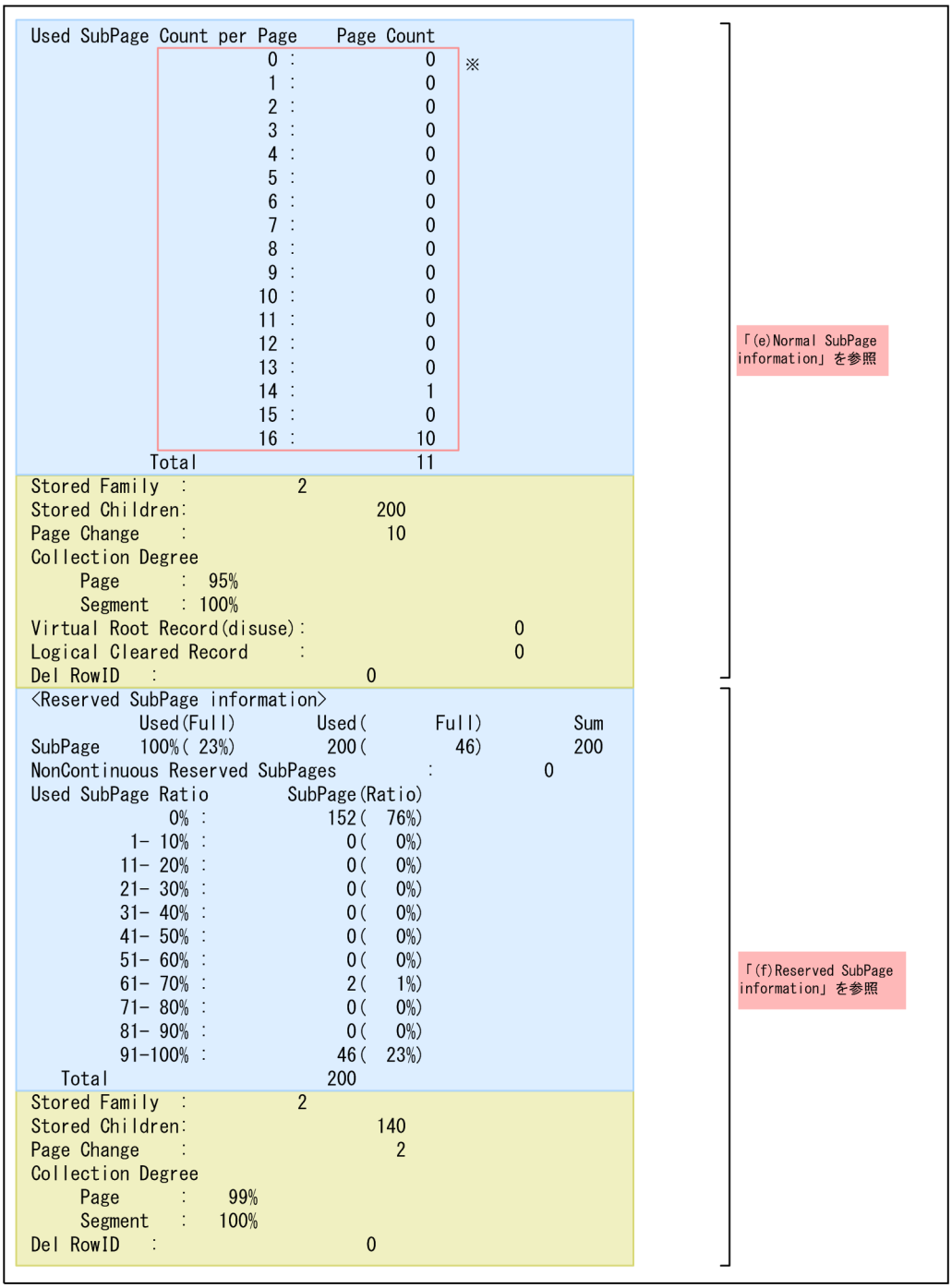
「(a)基本情報」を参照

「(b)Common information」を参照

「(e)Normal SubPage information」を参照

（続く）

（続き）



（凡例） ：pddbstコマンドの-dオプションを指定した場合に出力される情報です。

：pddbstコマンドの-dオプションおよび-zオプションを指定した場合に出力される情報です。

注※　当該ルートレコード型に指定されたサブページ分割数まで表示されます。

各出力項目の説明を次に示します。

なお，単位が%の項目については，小数点以下切り上げとなります。

(a) 基本情報

出力項目のヘッダ名		出力項目	説明
RD Area Name		RD エリア名	解析対象の RD エリア名です。
Server		サーバ識別子	RD エリアを管理する BES のサーバ識別子です。
Total Segment		RD エリアの総セグメント数	解析対象の RD エリアの総セグメント数です（使用中セグメント数＋未使用セグメント数）。
Segment Size		RD エリアのセグメント内ページ数	解析対象の RD エリアのセグメントサイズです。1 セグメントのサイズをセグメント内のページ数で表示します。
Unused Segment		未割り当てセグメント数	RD エリア内でルートレコード型に割り当てられていないセグメントの数です。
Page Size		ページサイズ	RD エリアのページサイズです。
Original RD Area Name※		オリジナル RD エリア名	RD エリアのオリジナル RD エリア名です。
Generation Number※		世代番号	RD エリアの世代番号です。
Replica RD Area Count		レプリカ RD エリア数	レプリカ RD エリアの数です。
History1	Hold Status	閉塞種別	RD エリアの閉塞履歴 1 の閉塞種別です。
	Hold Code	閉塞要因コード	RD エリアの閉塞履歴 1 の閉塞要因コードです。閉塞要因コードについては，マニュアル「HiRDB コマンドリファレンス」の「データベース状態解析ユティリティ(pddbst)」の「RD エリア単位の状態解析（論理的解析)」を参照してください。
	Hold Time	閉塞時刻	RD エリアの閉塞履歴 1 の閉塞時刻です。
History2	Hold Status	閉塞種別	RD エリアの閉塞履歴 2 の閉塞種別です。
	Hold Code	閉塞要因コード	RD エリアの閉塞履歴 2 の閉塞要因コードです。閉塞要因コードについては，マニュアル「HiRDB コマンドリファレンス」の「データベース状態解析ユティリティ(pddbst)」の「RD エリア単位の状態解析（論理的解析)」を参照してください。
	Hold Time	閉塞時刻	RD エリアの閉塞履歴 2 の閉塞時刻です。
SDB database kind		SDB データベース種別	SDB データベースの種別です。FMB，DAM，MAM，TAM，または SAM が表示されます。 4V FMB または SD FMB の SDB データベースの場合は，FMB が出力されます。
Root Record Name		ルートレコードのレコード型名	ルートレコードのレコード型名です。
Auth Id		認可識別子	ルートレコード型の所有者の認可識別子です。
SubPage Number		サブページ分割数	サブページの分割数です。 サブページ分割を適用していない場合は，サブページ分割数に 1 が表示されます。

出力項目のヘッダ名	出力項目	説明
SubPage Size	サブページサイズ	サブページのサイズです。 ルートレコード型にサブページ分割が適用されている場合に表示されます。

注※

この項目（インナレプリカ情報）は，HiRDB Staticizer Option を使用している場合に表示されます。また，指定した世代にレプリカ RD エリアが存在しない場合も，解析結果は出力されません。このとき，その旨のメッセージが出力されます。

(b) Common information

各ルートレコード型のセグメントに関する情報（未割り当てセグメントを含まない），および各ルートレコード型のページに関する情報が表示されます。

出力項目のヘッダ名		出力項目	説明
Segment	Used(%)	使用セグメント率	↑使用セグメント数÷セグメント数の合計↑の値が表示されます。
	Full(%)	満杯セグメント比率	↑満杯セグメント数÷総セグメント数↑の値が表示されます。
	Used	使用セグメント数	使用ページが含まれているセグメントの数です。
	Full	満杯セグメント数	セグメント内ページがすべて満杯ページであるセグメントの数です。
	Sum	セグメント数の合計	割り当てられているセグメント数の合計です。
Page※	Used(%)	使用ページ率	↑使用ページ数÷総ページ数↑の値が表示されます。
	Full(%)	満杯ページ率	↑満杯ページ数÷総ページ数↑の値が表示されます。
	Used	使用ページ数	次のどれかのページの合計数が表示されます。 • データが格納されているページ • 事前割り当てページ • 事前割り当てサブページを含むページ
	Full	満杯ページ数	**■サブページ分割が適用されていない場合** 次のどちらかのページ数が表示されます。 • 割り当てられているセグメント内で，これ以上データを格納できない使用ページ数 • 割り当てられているセグメント内で，占有状態の使用ページ数 **■サブページ分割が適用されている場合** 割り当てられているセグメント内で，すべてのサブページが使用サブページである使用ページ数です。
	Sum	総ページ数	全セグメントの総ページ数です。

注※

通常の（サブ）ページと事前割り当て（サブ）ページの区別をしません。

(c) Normal Page information

通常のページに関する情報が表示されます。Normal Page information は，次に示す条件をすべて満たす場合に表示されます。

- 解析対象のルートレコード型にサブページ分割が適用されていない場合
- pddbst コマンドの-d オプションを指定した場合

出力項目のヘッダ名		出力項目	説明
Page	Used(%)	通常のページの使用ページ率	↑通常のページの使用ページ数÷通常のページ数の合計↑の値が表示されます。
	Full(%)	通常のページの満杯ページ率	↑通常のページの満杯ページ数÷通常のページ数の合計↑の値が表示されます。
	Occupied(%)	通常のページの占有ページ率	↑通常のページの占有ページ数÷通常のページの満杯ページ数↑の値が表示されます。
	Used	通常のページの使用ページ数	データが入っている通常のページ数です。
	Full	通常のページの満杯ページ数	次のどちらかの通常のページの合計数が表示されます。 • これ以上レコードを格納できない通常のページ • 占有状態の通常のページ
	Occupied	通常のページの占有ページ数	ページ切り替えオプションに O を指定したレコードが格納されているページの数です。
	Sum	通常のページ数の合計	全セグメントの通常のページ数の合計です。割り当てられているセグメント中の通常のページ数の合計です。未使用ページは，通常のページと見なされます。
Collect Prearranged Page		再編成による解放可能なページ数	データが格納されていない通常のページの，使用ページ数です。 SDB データベースの再編成によって，解放可能な通常のページ数が表示されます。データ格納済みの通常のページで，使用ページ数にカウントされているが，実際にはページ内の全レコードが削除されていて，使用率が0%になっている通常のページ数が表示されます。 事前割り当てページは，データが格納されていない状態であっても，そのあとの運用で指定ファミリに割り当てられることが前提のため，解放可能なページ数にカウントされません。
Used Page Ratio	Page	通常のページの使用比率別ページ数	使用ページの使用比率ごと（10%単位）の，通常のページ数です。
	Ratio	通常のページ数の合計に対する比率	通常のページ数の合計に対する比率です。 小数点以下切り上げで比率を求めるため，すべての比率を合計したときに 100%を超えることがあります。 格納対象レコードが空き領域に収まらない場合，そのページは満杯ページとして設定されるため，満杯ページであっても，使用率が 91－100%にならないことがあります。

出力項目のヘッダ名		出力項目	説明
	Occupied Page	使用ページの比率ごとの，占有ページ数	使用ページの比率ごとの，占有ページ数です。
	Total	合計値	次の値が表示されます。 • 通常のページ数の合計値 • 占有ページ数の合計値
Stored Family※		通常のページに格納されているルートレコードの数	事前ページ割り当て機能が適用されていないファミリの数です。
Stored Children※		通常のページに格納されている非ルートレコードの数	通常のページに格納されている非ルートレコードの数です。
Page Change※		通常のページの，ページのまたがり回数	PRIOR ポインタが別ページのレコードを指す，通常のページの非ルートレコードの数です。
Collection Degree※	Page	通常のページの，ページ内レコード集約度	↑PRIOR ポインタが同一ページのレコードを指す，通常のページの非ルートレコードの数÷通常のページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一ページのレコードを指すレコードの比率を表示します。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 通常のページの使用ページ数が 1 以上の場合 • 通常のページの非ルートレコード数が 0 の場合
	Segment	通常のページの，セグメント内レコード集約度	↑PRIOR ポインタが同一セグメントのレコードを指す，通常のページの非ルートレコードの数÷通常のページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一セグメントのレコードを指す，レコードの比率を表示します。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 通常のページの使用ページ数が 1 以上の場合 • 通常のページの非ルートレコード数が 0 の場合
Virtual Root Record(disuse)※		通常のページの，子レコードを 1 件も管理していない仮想ルートレコードの数	通常のページに格納されている仮想ルートレコードのうち，子レコードを 1 件も管理していない仮想ルートレコードの数です。
Logical Cleared Record※		通常のページの，論理的に削除状態になっている非ルートレコードの数	通常のページに格納されている非ルートレコードのうち，一括削除機能によって論理的に削除状態になっている非ルートレコードの数です。
Del RowID※		通常のページの，削除状態の管理行数	通常のページのレコード管理情報のうち，レコードの削除（ERASE）によって削除状態になっているレコード管理情報数です。

注※

pddbst コマンドの-d オプションおよび-z オプションを指定した場合に表示される情報です。

(d) Reserved Page information

事前割り当てページに関する情報が表示されます。Reserved Page information は，次に示す条件をすべて満たす場合に表示されます。

- 解析対象のルートレコード型にサブページ分割が適用されていない場合
- pddbst コマンドの-d オプションを指定した場合

出力項目のヘッダ名		出力項目	説明
Page	Used(%)	事前割り当てページの使用ページ率	↑事前割り当てページの使用ページ数÷事前割り当てページ数の合計↑の値が表示されます。
	Full(%)	事前割り当てページの満杯ページ率	↑事前割り当てページの満杯ページ数÷事前割り当てページ数の合計↑の値が表示されます。
	Used	事前割り当てページの使用ページ数	事前割り当てページ数です。事前割り当てページは，レコードが格納されていなくても使用ページとして扱われます。
	Full	事前割り当てページの満杯ページ数	これ以上レコードを格納できない事前割り当てページ数です。
	Sum	事前割り当てページ数の合計	事前割り当てページ数の合計です。
	Ratio	事前割り当てページの，使用比率別ページ数	事前割り当てページ内の使用比率ごとに，次の項目が表示されます。 • ページ数 • 事前割り当てページ数に対する比率
NonContinuous Reserved Pages		非連続に割り当てられている事前割り当てページ数	該当ファミリに割り当てられている事前割り当てページの間に，ほかのデータページが割り込んでいる回数です。
Used Page Ratio	Page	事前割り当てページの使用比率別ページ数	使用ページの使用比率ごと（10%単位）の，事前割り当てページ数です。
	Ratio	事前割り当てページの合計に対する比率	事前割り当てページ数の合計に対する比率です。 小数点以下切り上げで比率を求めるため，すべての比率を合計したときに 100%を超えることがあります。 格納対象レコードが空き領域に収まらない場合，そのページは満杯ページとして設定されるため，満杯ページであっても，使用率が 91－100%にならないことがあります。
	Total	合計値	事前割り当てページ数の合計値です。
Stored Family※		事前ページ割り当て機能が適用されているファミリの数	事前ページ割り当て機能が適用されているファミリの数です。事前割り当てページに格納されているルートレコードの数です。
Stored Children※		事前割り当てページに格納されている非ルートレコードの数	事前割り当てページに格納されている非ルートレコードの数です。

出力項目のヘッダ名		出力項目	説明
Page Change※		事前割り当てページの，ページのまたがり回数	PRIOR ポインタが別ページのレコードを指す，事前割り当てページの非ルートレコードの数です。
Collection Degree※	Page	事前割り当てページの，ページ内レコード集約度	↑PRIOR ポインタが同一ページのレコードを指す，事前割り当てページの非ルートレコードの数÷事前割り当てページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一ページのレコードを指す，レコードの比率を表示します。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 事前割り当てページの使用ページ数が 1 以上の場合 • 事前割り当てページの非ルートレコード数が 0 の場合
	Segment	事前割り当てページの，セグメント内レコード集約度	↑PRIOR ポインタが同一セグメントのレコードを指す，事前割り当てページの非ルートレコードの数÷事前割り当てページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一セグメントのレコードを指す，レコードの比率を表示します。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 事前割り当てページの使用ページ数が 1 以上の場合 • 事前割り当てページの非ルートレコード数が 0 の場合
Del RowID※		事前割り当てページの，削除状態の管理行数	事前割り当てページのレコード管理情報のうち，レコードの削除（ERASE）によって削除状態になっているレコード管理情報数です。

注※

pddbst コマンドの-d オプションおよび-z オプションを指定した場合に表示される情報です。

(e) Normal SubPage information

通常のサブページに関する情報が表示されます。Normal SubPage information は，次に示す条件をすべて満たす場合に表示されます。

- 解析対象のルートレコード型にサブページ分割が適用されている場合
- pddbst コマンドの-d オプションを指定した場合

出力項目のヘッダ名		出力項目	説明
SubPage	Used(%)	通常のサブページの使用サブページ率	↑通常のサブページの使用サブページ数÷通常のサブページ数の合計↑の値が表示されます。
	Full(%)	通常のサブページの満杯サブページ率	↑通常のサブページの満杯サブページ数÷通常のサブページ数の合計↑の値が表示されます。
	Occupied(%)	通常のサブページの占有サブページ比率	↑通常のサブページの占有サブページ数÷通常のサブページの満杯サブページ数↑の値が表示されます。

出力項目のヘッダ名		出力項目	説明
	Used	通常のサブページの使用サブページ数	レコードが格納されている通常のサブページの数です。
	Full	通常のサブページの満杯サブページ数	次のどちらかのサブページの数が表示されます。 • これ以上レコードを格納できない通常のサブページ数 • 占有状態の通常のサブページ数
	Occupied	通常のサブページの占有サブページ数	ページ切り替えオプションに O を指定したレコードが格納されている通常のサブページの数です。
	Sum	通常のサブページ数の合計	通常のサブページ数の合計です。 割り当てられているセグメント中の，通常のサブページ数の合計です。未使用サブページは，通常のサブページと見なされます。
Collect Prearranged Page		再編成による解放可能なページ数	すべてのサブページが未使用サブページである使用ページ数です。 解放可能な通常のページ数が表示されます。データ格納済みの通常のページで，使用ページ数にカウントされているが，実際にはページ内の全レコードが削除されていて，使用率が 0%になっている通常のページ数が表示されます。 事前割り当てサブページは，データが格納されていない状態であっても，使用サブページとして扱われます。
Used SubPage Ratio	SubPage	通常のサブページの使用比率別サブページ数	使用サブページの使用比率ごと（10%単位）の，通常のサブページ数です。
	Ratio	通常のサブページ数の合計に対する比率	通常のサブページ数の合計に対する比率です。 小数点以下切り上げで比率を求めるため，すべての比率を合計したときに 100%を超えることがあります。 格納対象レコードが空き領域に収まらない場合，そのサブページは満杯サブページとして設定されるため，満杯サブページであっても，使用率が 91－100%にならないことがあります。
	Occupied SubPage	使用サブページの比率ごとの，占有ページ数	使用サブページの比率ごとの，占有ページ数です。
	Total	合計値	次の値が表示されます。 • 通常のサブページ数の合計値 • 占有ページ数の合計値
Used SubPage Count per Page	Page Count	使用サブページ数ごとのページ数	各ページの使用サブページ数ごとに，使用ページ数が表示されます。 事前割り当てサブページがあるページについては，次の条件を満たすページを解析対象にします。 • 同一ページ内に，事前割り当てサブページと通常のサブページの，使用サブページが混在しているページ 解析対象のルートレコード型に定義されているサブページ分割数まで表示されます。

出力項目のヘッダ名		出力項目	説明
	Total	合計値	上記の項目の合計値です。
Stored Family※		通常のサブページに格納されているルートレコードの数	事前ページ割り当て機能が適用されていないファミリの数です。
Stored Children※		通常のサブページに格納されている非ルートレコードの数	通常のサブページに格納されている非ルートレコードの数です。
Page Change※		通常のサブページの，ページのまたがり回数	PRIOR ポインタが別ページのレコードを指す，通常のサブページの非ルートレコードの数です。 サブページに格納されている場合でも，ページ単位に解析を行います。
Collection Degree※	Page	通常のサブページの，ページ内レコード集約度	↑PRIOR ポインタが同一ページのレコードを指す，通常のサブページの非ルートレコードの数÷通常のサブページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一ページのレコードを指す，レコードの比率を表示します。 サブページに格納されている場合でも，ページ単位に解析を行います。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 通常のサブページの使用サブページ数が 1 以上の場合 • 通常のサブページの非ルートレコード数が 0 の場合
	Segment	通常のサブページの，セグメント内レコード集約度	↑PRIOR ポインタが同一セグメントのレコードを指す，通常のサブページの非ルートレコードの数÷通常のサブページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一セグメントのレコードを指す，レコードの比率を表示します。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 通常のサブページの使用サブページ数が 1 以上の場合 • 通常のサブページの非ルートレコード数が 0 の場合
Virtual Root Record(disuse)※		通常のサブページの，子レコードを 1 件も管理していない仮想ルートレコードの数	通常のサブページに格納されている仮想ルートレコードのうち，子レコードを 1 件も管理していない仮想ルートレコードの数です。
Logical Cleared Record※		通常のサブページの，論理的に削除状態になっている非ルートレコードの数	通常のサブページに格納されている非ルートレコードのうち，一括削除機能によって論理的に削除状態になっている非ルートレコードの数です。
Del RowID※		通常のサブページの，削除状態の管理行数	通常のサブページのレコード管理情報のうち，レコードの削除（ERASE）によって削除状態になっているレコード管理情報数です。

注※

pddbst コマンドの-d オプションおよび-z オプションを指定した場合に表示される情報です。

(f) Reserved SubPage information

事前割り当てサブページに関する情報が表示されます。Reserved SubPage information は，次に示す条件をすべて満たす場合に表示されます。

- 解析対象のルートレコード型にサブページ分割が適用されている場合
- pddbst コマンドの-d オプションを指定した場合

出力項目のヘッダ名		出力項目	説明
SubPage	Used(%)	事前割り当てサブページの使用サブページ率	↑事前割り当てサブページの使用サブページ数÷事前割り当てサブページ数の合計↑の値が表示されます。
	Full(%)	事前割り当てサブページの満杯サブページ率	↑事前割り当てサブページの満杯サブページ数÷事前割り当てサブページ数の合計↑の値が表示されます。
	Used	事前割り当てサブページの使用サブページ数	割り当てサブページ数です。事前割り当てサブページは，レコードが格納されていなくても使用サブページとして扱われます。
	Full	事前割り当てサブページの満杯サブページ数	これ以上レコードを格納できない事前割り当てサブページ数です。
	Sum	事前割り当てサブページ数の合計	事前割り当てサブページ数の合計です。割り当てられているセグメント中の，事前割り当てサブページの合計です。
NonContinuous Reserved SubPages		非連続に割り当てられている事前割り当てサブページ数	該当ファミリに割り当てられている事前割り当てサブページの間に，ほかのデータページが割り込んでいる回数です。
Used SubPage Ratio	SubPage	事前割り当てサブページの使用比率別サブページ数	使用サブページの使用比率ごと（10%単位）の事前割り当てサブページ数です。
	Ratio	事前割り当てサブページの合計に対する比率	事前割り当てサブページ数の合計に対する比率です。 小数点以下切り上げで比率を求めるため，すべての比率を合計したときに 100%を超えることがあります。 格納対象レコードが空き領域に収まらない場合，そのサブページは満杯サブページとして設定されるため，満杯サブページであっても，使用率が 91－100%にならないことがあります。
	Total	合計値	事前割り当てサブページ数の合計値です。
Stored Family※		事前ページ割り当て機能が適用されているファミリの数	事前ページ割り当て機能が適用されているファミリの数です。事前割り当てサブページに格納されているルートレコードの数です。
Stored Children※		事前割り当てサブページに格納されている非ルートレコードの数	事前割り当てサブページに格納されている非ルートレコードの数です。
Page Change※		事前割り当てサブページの，ページのまたがり回数	PRIOR ポインタが別ページのレコードを指す，事前割り当てサブページの非ルートレコードの数です。

出力項目のヘッダ名		出力項目	説明
			サブページに格納されている場合でも，ページ単位に解析を行います。
Collection Degree※	Page	事前割り当てサブページの，ページ内レコード集約度	↑PRIOR ポインタが同一ページのレコードを指す，事前割り当てサブページの非ルートレコードの数÷事前割り当てサブページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一ページのレコードを指す，レコードの比率を表示します。 サブページに格納されている場合でも，ページ単位に解析を行います。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 事前割り当てサブページの使用サブページ数が 1 以上の場合 • 事前割り当てサブページの非ルートレコード数が 0 の場合
	Segment	事前割り当てサブページの，セグメント内レコード集約度	↑PRIOR ポインタが同一セグメントのレコードを指す，事前割り当てサブページの非ルートレコードの数÷事前割り当てサブページの非ルートレコード数↑の値が表示されます。 PRIOR ポインタが同一セグメントのレコードを指す，レコードの比率を表示します。 なお，次の条件をすべて満たす場合は，100%と見なされます。 • 事前割り当てサブページの使用サブページ数が 1 以上の場合 • 事前割り当てサブページの非ルートレコード数が 0 の場合
Del RowID※		事前割り当てサブページの，削除状態の管理行数	事前割り当てサブページのレコード管理情報のうち，レコードの削除（ERASE）によって削除状態になっているレコード管理情報数です。

注※

pddbst コマンドの-d オプションおよび-z オプションを指定した場合に表示される情報です。

(3) DAT 形式で解析結果を出力する場合

RD エリア単位の状態解析（論理的解析）の DAT 形式での出力規則を次に示します。

- DAT 形式で出力する場合の出力規則については，マニュアル「HiRDB コマンドリファレンス」の「コマンド実行結果を DAT 形式で出力する場合の規則」を参照してください。
- RD エリアごとに，その RD エリア内のルートレコード型につき 1 行表示します。
- RD エリアの情報は，RD エリア名の昇順に表示します。
- 単位が%の項目については，小数点以下切り上げとなります。

RD エリア単位の状態解析（論理的解析）の DAT 形式出力内容を次の表に示します。

表 10-6　RD エリア単位の状態解析（論理的解析）の DAT 形式出力内容

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
1	VERSION	HiRDB のバージョンです。	文字列(vv-rr)	5	◎
2	INF_GET_TIME	情報取得が完了した年月日時刻です。出力対象の情報をすべて取得完了した日時です。DAT 形式で出力される全行で同じ日時が出力されます。	文字列 (yyyy/mm/dd hh:mm:ss)	19	◎
3	RDAREA_NAME	解析対象の RD エリア名です。	文字列	30	◎
4	SERVER	RD エリアがある BES 名です。	文字列	8	◎
5	TOTAL_SEGMENT	RD エリア内の総セグメント数です。	数値	10	◎
6	UNUSED_SEGMENT	RD エリア内の未割り当てセグメント数です。	数値	10	◎
7	SEGMENT_SIZE	セグメント内ページ数です。	数値	10	◎
8	PAGE_SIZE	ページサイズです。	数値	10	◎
9	ORIGINAL_RDAREA_NAME	オリジナル RD エリア名です。	文字列	30	◎※2
10	GENERATION_NUMBER	RD エリアの世代番号です。	数値	2	◎※2
11	REPLICA_RDAREA_COUNT	レプリカ RD エリア数です。	数値	2	◎※2
12	HOLD_STATUS(HISTORY1)	RD エリアの閉塞種別です。閉塞解除する 1 つ前の閉塞種別（閉塞中の場合は現在の閉塞種別）になります。 • "CMD"：HiRDB の障害検知によるコマンド閉塞 • "FLT"：障害閉塞	文字列	3	◎
13	HOLD_CODE(HISTORY1)	RD エリアの閉塞要因コードです。閉塞解除する 1 つ前の閉塞要因コード（閉塞中の場合は現在の閉塞要因コード）になります。閉塞要因コードについては，マニュアル「HiRDB コマンドリファレンス」の「データベース状態解析ユティリティ（pddbst)」の「RD エリア単位の状態解析（論理的解析)」を参照してください。	数値	5	◎

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
14	HOLD_TIME(HISTORY1)	RD エリアの閉塞時刻です。閉塞解除する 1 つ前の閉塞時刻（閉塞中の場合は現在の閉塞時刻）になります。	文字列 (yyyy/mm/dd hh:mm:ss)	19	◎
15	HOLD_STATUS(HISTORY2)	RD エリアの閉塞種別です。History1 の 1 つ前の閉塞種別になります。 • "CMD"：HiRDB の障害検知によるコマンド閉塞 • "FLT"：障害閉塞	文字列	3	◎
16	HOLD_CODE(HISTORY2)	RD エリアの閉塞要因コードです。History1 の 1 つ前の閉塞要因コードです。閉塞要因コードについては，マニュアル「HiRDB コマンドリファレンス」の「データベース状態解析ユティリティ(pddbst)」の「RD エリア単位の状態解析（論理的解析)」を参照してください。	数値	5	◎
17	HOLD_TIME(HISTORY2)	RD エリアの閉塞時刻です。History1 の 1 つ前の閉塞時刻です。	文字列 (yyyy/mm/dd hh:mm:ss)	19	◎
18	INF_KIND	表示情報の種別です。 • "RCD"：RD エリアの解析（ルートレコード型の解析情報）	文字列	3	◎
19	SDB_DATABASE_KIND	SDB データベースの種別です。	文字列	3	◎
20	ROOT_RECORD_NAME	ルートレコードのレコード型名です。	文字列	30	◎
21	AUTH_ID	ルートレコード型の所有者の認可識別子です。	文字列	8	◎
22	SUBPAGE_NUMBER	サブページ分割数です。	数値	2	◎
23	SUBPAGE_SIZE	サブページサイズです。	数値	5	◎※3
24	USED_SEGMENT	使用セグメント数です。	数値	10	◎
25	USED_SEGMENT_RATIO	使用セグメント率です。(単位：%)	数値	3	◎
26	FULL_SEGMENT	満杯セグメント数です。	数値	10	◎
27	FULL_SEGMENT_RATIO	満杯セグメント率です。(単位：%)	数値	3	◎

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
28	SUM_SEGMENT	割り当てられているセグメント数の合計です。	数値	10	◎
29	USED_PAGE	使用ページ数です。	数値	10	◎
30	USED_PAGE_RATIO	使用ページ率です。（単位：%）	数値	3	◎
31	FULL_PAGE	満杯ページ数です。	数値	10	◎
32	FULL_PAGE_RATIO	満杯ページ率です。（単位：%）	数値	3	◎
33	SUM_PAGE	割り当てられているセグメント中のページ数の合計です。	数値	10	◎
34	USED_NORMAL_PAGE	通常のページの使用ページ数です。	数値	10	○
35	USED_NORMAL_PAGE_RATIO	通常のページの使用ページ率です。（単位：%）	数値	3	○
36	FULL_NORMAL_PAGE	通常のページの満杯ページ数です。	数値	10	○
37	FULL_NORMAL_PAGE_RATIO	通常のページの満杯ページ率です。（単位：%）	数値	3	○
38	OCCUPIED_NORMAL_PAGE	通常のページの，占有状態のページ数です。	数値	10	○
39	OCCUPIED_NORMAL_PAGE_RATIO	通常のページの，占有状態のページ率です。（単位：%）	数値	3	○
40	SUM_NORMAL_PAGE	割り当てられているセグメント中の通常のページ数の合計です。未使用ページは，通常のページと見なされます。	数値	10	○
41	USED_NORMAL_SUBPAGE	通常のサブページの使用サブページ数です。	数値	11	○
42	USED_NORMAL_SUBPAGE_RATIO	通常のサブページの使用サブページ率です。（単位：%）	数値	3	○
43	FULL_NORMAL_SUBPAGE	通常のサブページの満杯サブページ数です。	数値	11	○
44	FULL_NORMAL_SUBPAGE_RATIO	通常のサブページの満杯サブページ率です。（単位：%）	数値	3	○
45	OCCUPIED_NORMAL_SUBPAGE	通常のサブページの占有状態のサブページ数です。	数値	11	○
46	OCCUPIED_NORMAL_SUBPAGE_RATIO	通常のサブページの占有状態のサブページ率です。（単位：%）	数値	3	○
47	SUM_NORMAL_SUBPAGE	割り当てられているセグメント中の，通常のサブページ数の合計で	数値	11	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
		す。未使用サブページは，通常のサブページと見なされます。			
48	COLLECT_PREARRANGED_PAGE	再編成による解放可能な通常のページ数です。 データ格納済みの通常のページで，使用ページ数にカウントされているが，実際にはページ内の全レコードが削除されていて，使用率が 0%になっている通常のページ数が表示されます。	数値	10	○
49	STORED_NORMAL_PAGE_FAMILY	事前ページ割り当て機能が適用されていないファミリの数です（サブページ分割機能を適用していない場合）。	数値	10	△
50	STORED_NORMAL_SUBPAGE_FAMILY	事前ページ割り当て機能が適用されていないファミリの数です（サブページ分割機能を適用している場合）。	数値	11	△
51	NORMAL_PAGE_CHILDREN	通常のページに格納されている非ルートレコードの数です。	数値	仮数 16，指数 5	△
52	NORMAL_SUBPAGE_CHILDREN	通常のサブページに格納されている非ルートレコードの数です。	数値	仮数 16，指数 5	△
53	NORMAL_PAGE_PAGECHANGE	通常のページに格納されている，PRIOR ポインタが別ページのレコードを指す非ルートレコードの数です。	数値	仮数 16，指数 5	△
54	NORMAL_SUBPAGE_PAGECHANGE	通常のサブページに格納されている，PRIOR ポインタが別ページのレコードを指す非ルートレコードの数です。	数値	仮数 16，指数 5	△
55	NORMAL_PAGE_SAME_PAGE_RECORD_RATIO	通常のページに格納されている，PRIOR ポインタが同一ページのレコードを指すレコードの比率です。	数値	3	△
56	NORMAL_SUBPAGE_SAME_PAGE_RECORD_RATIO	通常のサブページに格納されている，PRIOR ポインタが同一ページのレコードを指すレコードの比率です。 サブページに格納されている場合でも，ページ単位に解析を行います。	数値	3	△

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
57	NORMAL_PAGE_SAME_SEGMENT_RECORD_RATIO	通常のページに格納されている，PRIOR ポインタが同一セグメントのレコードを指すレコードの比率です。	数値	3	△
58	NORMAL_SUBPAGE_SAME_SEGMENT_RECORD_RATIO	通常のサブページに格納されている，PRIOR ポインタが同一セグメントのレコードを指すレコードの比率です。	数値	3	△
59	NORMAL_PAGE_DISUSE_VIRTUAL_ROOT_RECORD	通常のページに格納されている，子レコードを 1 件も管理していない仮想ルートレコードの数です。	数値	10	△
60	NORMAL_SUBPAGE_DISUSE_VIRTUAL_ROOT_RECORD	通常のサブページに格納されている，子レコードを 1 件も管理していない仮想ルートレコードの数です。	数値	11	△
61	NORMAL_PAGE_LOGICAL_CLEARED_RECORD	通常のページに格納されている，一括削除機能によって論理的に削除状態になっているレコードの数です。	数値	仮数 16，指数 5	△
62	NORMAL_SUBPAGE_LOGICAL_CLEARED_RECORD	通常のサブページに格納されている，一括削除機能によって論理的に削除状態になっているレコードの数です。	数値	仮数 16，指数 5	△
63	RATIO0_NORMAL_PAGE	使用率が 0%の通常のページ数です。	数値	10	○
64	RATIO0_NORMAL_PAGE_RATIO	使用率が 0%の通常のページの割合です。（単位：%） 小数点以下切り上げで比率を求めるため，RATIO0_NORMAL_PAGE_RATIO～RATIO10_NORMAL_PAGE_RATIO の比率を合計したときに 100%を超えることがあります。	数値	3	○
65	RATIO0_OCCUPIED_PAGE	使用率が 0%の通常のページのうち，占有状態のページ数です。	数値	10	○
66	RATIO1_NORMAL_PAGE	使用率が 1～10%の通常のページ数です。	数値	10	○
67	RATIO1_NORMAL_PAGE_RATIO	使用率が 1～10%の通常のページの割合です。（単位：%）	数値	3	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
68	RATIO1_OCCUPIED_PAGE	使用率が 1～10%の通常のページのうち，占有状態のページ数です。	数値	10	○
69	RATIO2_NORMAL_PAGE	使用率が 11～20%の通常のページ数です。	数値	10	○
70	RATIO2_NORMAL_PAGE_RATIO	使用率が 11～20%の通常のページの割合です。（単位：%）	数値	3	○
71	RATIO2_OCCUPIED_PAGE	使用率が 11～20%の通常のページのうち，占有状態のページ数です。	数値	10	○
72	RATIO3_NORMAL_PAGE	使用率が 21～30%の通常のページ数です。	数値	10	○
73	RATIO3_NORMAL_PAGE_RATIO	使用率が 21～30%の通常のページの割合です。（単位：%）	数値	3	○
74	RATIO3_OCCUPIED_PAGE	使用率が 21～30%の通常のページのうち，占有状態のページ数です。	数値	10	○
75	RATIO4_NORMAL_PAGE	使用率が 31～40%の通常のページ数です。	数値	10	○
76	RATIO4_NORMAL_PAGE_RATIO	使用率が 31～40%の通常のページの割合です。（単位：%）	数値	3	○
77	RATIO4_OCCUPIED_PAGE	使用率が 31～40%の通常のページのうち，占有状態のページ数です。	数値	10	○
78	RATIO5_NORMAL_PAGE	使用率が 41～50%の通常のページ数です。	数値	10	○
79	RATIO5_NORMAL_PAGE_RATIO	使用率が 41～50%の通常のページの割合です。（単位：%）	数値	3	○
80	RATIO5_OCCUPIED_PAGE	使用率が 41～50%の通常のページのうち，占有状態のページ数です。	数値	10	○
81	RATIO6_NORMAL_PAGE	使用率が 51～60%の通常のページ数です。	数値	10	○
82	RATIO6_NORMAL_PAGE_RATIO	使用率が 51～60%の通常のページの割合です。（単位：%）	数値	3	○
83	RATIO6_OCCUPIED_PAGE	使用率が 51～60%の通常のページのうち，占有状態のページ数です。	数値	10	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
84	RATIO7_NORMAL_PAGE	使用率が61～70%の通常のページ数です。	数値	10	○
85	RATIO7_NORMAL_PAGE_RATIO	使用率が61～70%の通常のページの割合です。（単位：%）	数値	3	○
86	RATIO7_OCCUPIED_PAGE	使用率が61～70%の通常のページのうち，占有状態のページ数です。	数値	10	○
87	RATIO8_NORMAL_PAGE	使用率が71～80%の通常のページ数です。	数値	10	○
88	RATIO8_NORMAL_PAGE_RATIO	使用率が71～80%の通常のページの割合です。（単位：%）	数値	3	○
89	RATIO8_OCCUPIED_PAGE	使用率が71～80%の通常のページのうち，占有状態のページ数です。	数値	10	○
90	RATIO9_NORMAL_PAGE	使用率が81～90%の通常のページ数です。	数値	10	○
91	RATIO9_NORMAL_PAGE_RATIO	使用率が81～90%の通常のページの割合です。（単位：%）	数値	3	○
92	RATIO9_OCCUPIED_PAGE	使用率が81～90%の通常のページのうち，占有状態のページ数です。	数値	10	○
93	RATIO10_NORMAL_PAGE	使用率が91～100%の通常のページ数です。 格納対象レコードが空き領域に収まらない場合，そのページは満杯ページとして設定されるため，満杯ページであっても，使用率が91～100%にならないことがあります。	数値	10	○
94	RATIO10_NORMAL_PAGE_RATIO	使用率が91～100%の通常のページの割合です。（単位：%）	数値	3	○
95	RATIO10_OCCUPIED_PAGE	使用率が91～100%の通常のページのうち，占有状態のページ数です。	数値	10	○
96	TOTAL_NORMAL_PAGE	通常のページの，使用率別の使用ページ数の合計です。	数値	10	○
97	TOTAL_OCCUPIED_PAGE	通常のページの，使用率別の占有状態のページ数の合計です。	数値	10	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
98	RATIO0_NORMAL_SUBPAGE	使用率が 0%の通常のサブページ数です。	数値	11	○
99	RATIO0_NORMAL_SUBPAGE_RATIO	使用率が 0%の通常のサブページの割合です。（単位：%） 小数点以下切り上げで比率を求めるため，RATIO0_NORMAL_SUBPAGE_RATIO～RATIO10_NORMAL_SUBPAGE_RATIO の比率を合計したときに 100%を超えることがあります。	数値	3	○
100	RATIO0_OCCUPIED_SUBPAGE	使用率が 0%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
101	RATIO1_NORMAL_SUBPAGE	使用率が 1～10%の通常のサブページ数です。	数値	11	○
102	RATIO1_NORMAL_SUBPAGE_RATIO	使用率が 1～10%の通常のサブページの割合です。（単位：%）	数値	3	○
103	RATIO1_OCCUPIED_SUBPAGE	使用率が 1～10%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
104	RATIO2_NORMAL_SUBPAGE	使用率が 11～20%の通常のサブページ数です。	数値	11	○
105	RATIO2_NORMAL_SUBPAGE_RATIO	使用率が 11～20%の通常のサブページの割合です。（単位：%）	数値	3	○
106	RATIO2_OCCUPIED_SUBPAGE	使用率が 11～20%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
107	RATIO3_NORMAL_SUBPAGE	使用率が 21～30%の通常のサブページ数です。	数値	11	○
108	RATIO3_NORMAL_SUBPAGE_RATIO	使用率が 21～30%の通常のサブページの割合です。（単位：%）	数値	3	○
109	RATIO3_OCCUPIED_SUBPAGE	使用率が 21～30%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
110	RATIO4_NORMAL_SUBPAGE	使用率が 31～40%の通常のサブページ数です。	数値	11	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
111	RATIO4_NORMAL_SUBPAGE_RATIO	使用率 21～30%の通常のサブページの割合です。（単位：%）	数値	3	○
112	RATIO4_OCCUPIED_SUBPAGE	使用率が 31～40%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
113	RATIO5_NORMAL_SUBPAGE	使用率が 41～50%の通常のサブページ数です。	数値	11	○
114	RATIO5_NORMAL_SUBPAGE_RATIO	使用率が 41～50%の通常のサブページの割合です。（単位：%）	数値	3	○
115	RATIO5_OCCUPIED_SUBPAGE	使用率が 41～50%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
116	RATIO6_NORMAL_SUBPAGE	使用率が 51～60%の通常のサブページ数です。	数値	11	○
117	RATIO6_NORMAL_SUBPAGE_RATIO	使用率が 51～60%の通常のサブページの割合です。（単位：%）	数値	3	○
118	RATIO6_OCCUPIED_SUBPAGE	使用率が 51～60%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
119	RATIO7_NORMAL_SUBPAGE	使用率が 61～70%の通常のサブページ数です。	数値	11	○
120	RATIO7_NORMAL_SUBPAGE_RATIO	使用率が 61～70%の通常のサブページの割合です。（単位：%）	数値	3	○
121	RATIO7_OCCUPIED_SUBPAGE	使用率が 61～70%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
122	RATIO8_NORMAL_SUBPAGE	使用率が 71～80%の通常のサブページ数です。	数値	11	○
123	RATIO8_NORMAL_SUBPAGE_RATIO	使用率が 71～80%の通常のサブページの割合です。（単位：%）	数値	3	○
124	RATIO8_OCCUPIED_SUBPAGE	使用率が 71～80%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
125	RATIO9_NORMAL_SUBPAGE	使用率が 81～90%の通常のサブページ数です。	数値	11	○
126	RATIO9_NORMAL_SUBPAGE_RATIO	使用率が 81～90%の通常のサブページの割合です。（単位：%）	数値	3	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
127	RATIO9_OCCUPIED_SUBPAGE	使用率が 81～90%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
128	RATIO10_NORMAL_SUBPAGE	使用率が 91～100%の通常のサブページ数です。 格納対象レコードが空き領域に収まらない場合，そのサブページは満杯サブページとして設定されるため，満杯サブページであっても，使用率が 91～100%にならないことがあります。	数値	11	○
129	RATIO10_NORMAL_SUBPAGE_RATIO	使用率が 91～100%の通常のサブページの割合です。（単位：%）	数値	3	○
130	RATIO10_OCCUPIED_SUBPAGE	使用率が 91～100%の通常のサブページのうち，占有状態のサブページ数です。	数値	11	○
131	TOTAL_NORMAL_SUBPAGE	通常のサブページの，使用率別の使用サブページ数の合計です。	数値	11	○
132	TOTAL_OCCUPIED_SUBPAGE	通常のサブページの，使用率別の占有状態のサブページ数の合計です。	数値	11	○
133	NORMAL_PAGE_DEL_ROWID	通常のページの，レコードの削除状態の管理行数です。	数値	仮数 16， 指数 5	△
134	NORMAL_SUBPAGE_DEL_ROWID	通常のサブページの，レコードの削除状態の管理行数です。	数値	仮数 16， 指数 5	△
135	USE0_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページを使用していない通常のページ数です。※4	数値	10	○
136	USE1_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，1 サブページを使用している通常のページ数です。※4	数値	10	○
137	USE2_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，2 サブページを使用している通常のページ数です。※4	数値	10	○
138	USE3_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，3 サブページを使用している通常のページ数です。※4	数値	10	○
139	USE4_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，4 サブページを使用している通常のページ数です。※4	数値	10	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
140	USE5_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，5サブページを使用している通常のページ数です。※4	数値	10	○
141	USE6_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，6サブページを使用している通常のページ数です。※4	数値	10	○
142	USE7_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，7サブページを使用している通常のページ数です。※4	数値	10	○
143	USE8_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，8サブページを使用している通常のページ数です。※4	数値	10	○
144	USE9_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，9サブページを使用している通常のページ数です。※4	数値	10	○
145	USE10_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，10サブページを使用している通常のページ数です。※4	数値	10	○
146	USE11_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，11サブページを使用している通常のページ数です。※4	数値	10	○
147	USE12_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，12サブページを使用している通常のページ数です。※4	数値	10	○
148	USE13_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，13サブページを使用している通常のページ数です。※4	数値	10	○
149	USE14_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，14サブページを使用している通常のページ数です。※4	数値	10	○
150	USE15_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，15サブページを使用している通常のページ数です。※4	数値	10	○
151	USE16_SUBPAGE_PER_NORMAL_PAGE	ページ内のサブページのうち，16サブページを使用している通常のページ数です。※4	数値	10	○
152	TOTAL_NORMAL_PAGE_HAVING_SUBPAGE	サブページを持つ通常のページの，使用ページの合計です。	数値	10	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
153	USED_RESERVED_PAGE	事前割り当てページの使用ページ数です。	数値	10	○
154	USED_RESERVED_PAGE_RATIO	事前割り当てページの使用ページ率です。(単位：%)	数値	3	○
155	FULL_RESERVED_PAGE	事前割り当てページの満杯ページ数です。	数値	10	○
156	FULL_RESERVED_PAGE_RATIO	事前割り当てページの満杯ページ率です。(単位：%)	数値	3	○
157	USED_RESERVED_SUBPAGE	事前割り当てサブページの使用サブページ数です。	数値	11	○
158	USED_RESERVED_SUBPAGE_RATIO	事前割り当てサブページの使用サブページ率です。(単位：%)	数値	3	○
159	FULL_RESERVED_SUBPAGE	事前割り当てサブページの満杯サブページ数です。	数値	11	○
160	FULL_RESERVED_SUBPAGE_RATIO	事前割り当てサブページの満杯サブページ率です。(単位：%)	数値	3	○
161	SUM_RESERVED_PAGE	割り当てられているセグメント中の，事前割り当てページの合計です。	数値	10	○
162	SUM_RESERVED_SUBPAGE	割り当てられているセグメント中の，事前割り当てサブページの合計です。	数値	11	○
163	RESERVED_PAGE_FAMILY	事前ページ割り当て機能が適用されているファミリの数です（サブページ分割機能を適用していない場合)。	数値	10	△
164	RESERVED_SUBPAGE_FAMILY	事前ページ割り当て機能が適用されているファミリの数です（サブページ分割機能を適用している場合)。	数値	11	△
165	RESERVED_PAGE_CHILDREN	事前割り当てページに格納されている非ルートレコードの数です。	数値	仮数 16, 指数 5	△
166	RESERVED_SUBPAGE_CHILDREN	事前割り当てサブページに格納されている非ルートレコードの数です。	数値	仮数 16, 指数 5	△
167	NONCONTINUOUS_RESERVED_PAGES	非連続に割り当てられている事前割り当てページ数です。	数値	10	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
168	NONCONTINUOUS_RESERVED_SUBPAGES	非連続に割り当てられている事前割り当てサブページ数です。	数値	11	○
169	RESERVED_PAGE_PAGECHANGE	事前割り当てページに格納されている非ルートレコードのうち，PRIOR ポインタが別ページのレコードを指す非ルートレコードの数です。	数値	仮数 16，指数 5	△
170	RESERVED_SUBPAGE_PAGECHANGE	事前割り当てサブページに格納されているレコードのうち，PRIOR ポインタが別ページのレコードを指すレコードの数です。 サブページに格納されている場合でも，ページ単位に解析を行います。	数値	仮数 16，指数 5	△
171	RESERVED_PAGE_SAME_PAGE_RECORD_RATIO	事前割り当てページに格納されているレコードのうち，PRIOR ポインタが同一ページのレコードを指すレコードの比率です。	数値	3	△
172	RESERVED_SUBPAGE_SAME_PAGE_RECORD_RATIO	事前割り当てサブページに格納されているレコードのうち，PRIOR ポインタが同一ページのレコードを指すレコードの比率です。 サブページに格納されている場合でも，ページ単位に解析を行います。	数値	3	△
173	RESERVED_PAGE_SAME_SEGMENT_RECORD_RATIO	事前割り当てページに格納されているレコードのうち，PRIOR ポインタが同一セグメントのレコードを指すレコードの比率です。	数値	3	△
174	RESERVED_SUBPAGE_SAME_SEGMENT_RECORD_RATIO	事前割り当てサブページに格納されているレコードのうち，PRIOR ポインタが同一セグメントのレコードを指すレコードの比率です。	数値	3	△
175	RATIO0_RESERVED_PAGE	事前割り当てページで，使用率が 0%のページ数です。	数値	10	○
176	RATIO0_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が 0%のページの割合です。（単位：%） 小数点以下切り上げで比率を求めるため，RATIO0_RESERVED_PAGE_RATIO～	数値	3	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
		RATIO10_RESERVED_PAGE_RATIO の比率を合計したときに100%を超えることがあります。			
177	RATIO1_RESERVED_PAGE	事前割り当てページで，使用率が1～10%のページ数です。	数値	10	○
178	RATIO1_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が1～10%のページの割合です。（単位：%）	数値	3	○
179	RATIO2_RESERVED_PAGE	事前割り当てページで，使用率が11～20%のページ数です。	数値	10	○
180	RATIO2_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が11～20%のページの割合です。（単位：%）	数値	3	○
181	RATIO3_RESERVED_PAGE	事前割り当てページで，使用率が21～30%のページ数です。	数値	10	○
182	RATIO3_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が21～30%のページの割合です。（単位：%）	数値	3	○
183	RATIO4_RESERVED_PAGE	事前割り当てページで，使用率が31～40%のページ数です。	数値	10	○
184	RATIO4_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が31～40%のページの割合です。（単位：%）	数値	3	○
185	RATIO5_RESERVED_PAGE	事前割り当てページで，使用率が41～50%のページ数です。	数値	10	○
186	RATIO5_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が41～50%のページの割合です。（単位：%）	数値	3	○
187	RATIO6_RESERVED_PAGE	事前割り当てページで，使用率が51～60%のページ数です。	数値	10	○
188	RATIO6_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が51～60%のページの割合です。（単位：%）	数値	3	○
189	RATIO7_RESERVED_PAGE	事前割り当てページで，使用率が61～70%のページ数です。	数値	10	○
190	RATIO7_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が61～70%のページの割合です。（単位：%）	数値	3	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
191	RATIO8_RESERVED_PAGE	事前割り当てページで，使用率が71～80%のページ数です。	数値	10	○
192	RATIO8_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が71～80%のページの割合です。（単位：%）	数値	3	○
193	RATIO9_RESERVED_PAGE	事前割り当てページで，使用率が81～90%のページ数です。	数値	10	○
194	RATIO9_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が81～90%のページの割合です。（単位：%）	数値	3	○
195	RATIO10_RESERVED_PAGE	事前割り当てページで，使用率が91～100%のページ数です。	数値	10	○
196	RATIO10_RESERVED_PAGE_RATIO	事前割り当てページで，使用率が91～100%のページの割合です。（単位：%） 格納対象レコードが空き領域に収まらない場合，そのページは満杯ページとして設定されるため，満杯ページであっても，使用率が91～100%にならないことがあります。	数値	3	○
197	TOTAL_RESERVED_PAGE	事前割り当てページの，使用率別の使用ページ数の合計です。	数値	10	○
198	RATIO0_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が0%のサブページ数です。	数値	11	○
199	RATIO0_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が0%のサブページの割合です。（単位：%） 小数点以下切り上げで比率を求めるため，RATIO0_RESERVED_SUBPAGE_RATIO～RATIO10_RESERVED_SUBPAGE_RATIOの比率を合計したときに100%を超えることがあります。	数値	3	○
200	RATIO1_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が1～10%のサブページ数です。	数値	11	○
201	RATIO1_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が1～10%のサブページの割合です。（単位：%）	数値	3	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
202	RATIO2_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 11～20%のサブページ数です。	数値	11	○
203	RATIO2_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 11～20%のサブページの割合です。（単位：%）	数値	3	○
204	RATIO3_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 21～30%のサブページ数です。	数値	11	○
205	RATIO3_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 21～30%のサブページの割合です。（単位：%）	数値	3	○
206	RATIO4_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 31～40%のサブページ数です。	数値	11	○
207	RATIO4_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 31～40%のサブページの割合です。（単位：%）	数値	3	○
208	RATIO5_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 41～50%のサブページ数です。	数値	11	○
209	RATIO5_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 41～50%のサブページの割合です。（単位：%）	数値	3	○
210	RATIO6_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 51～60%のサブページ数です。	数値	11	○
211	RATIO6_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 51～60%のサブページの割合です。（単位：%）	数値	3	○
212	RATIO7_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 61～70%のサブページ数です。	数値	11	○
213	RATIO7_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 61～70%のサブページの割合です。（単位：%）	数値	3	○
214	RATIO8_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 71～80%のサブページ数です。	数値	11	○

項番	タイトルバー	内容	出力形式	最大長（単位：バイト）※1	出力条件
215	RATIO8_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 71～80%のサブページの割合です。（単位：%）	数値	3	○
216	RATIO9_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 81～90%のサブページ数です。	数値	11	○
217	RATIO9_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 81～90%のサブページの割合です。（単位：%）	数値	3	○
218	RATIO10_RESERVED_SUBPAGE	事前割り当てサブページで，使用率が 91～100%のサブページ数です。 格納対象レコードが空き領域に収まらない場合，そのサブページは満杯サブページとして設定されるため，満杯サブページであっても，使用率が 91～100%にならないことがあります。	数値	11	○
219	RATIO10_RESERVED_SUBPAGE_RATIO	事前割り当てサブページで，使用率が 91～100%のサブページの割合です。（単位：%）	数値	3	○
220	TOTAL_RESERVED_SUBPAGE	事前割り当てサブページの，使用率別の使用サブページ数の合計です。	数値	11	○
221	RESERVED_PAGE_DEL_ROWID	事前割り当てページの，レコードの削除状態の管理行数です。	数値	仮数 16， 指数 5	△
222	RESERVED_SUBPAGE_DEL_ROWID	事前割り当てサブページの，レコードの削除状態の管理行数です。	数値	仮数 16， 指数 5	△

（凡例）

◎：pddbst コマンドの-d オプションの指定に関係なく出力されます。

○：pddbst コマンドの-d オプションを指定した場合に出力されます。

△：pddbst コマンドの-d オプションおよび-z オプションを指定した場合に出力されます。

注

- 出力データがない場合は空要素となります。
- 単位が%の項目については，小数点以下切り上げとなります。

注※1

出力形式が文字列の場合，" "で囲んで表示します。最大長の値には，" "を含みません。

注※2

HiRDB Staticizer Option がインストールされていない場合，空要素が出力されます。

注※3

サブページ分割を適用していない場合は，0 が出力されます。

注※4

- 事前割り当てサブページがあるページについては，同一ページ内に事前割り当てサブページと通常のサブページの，使用サブページが混在しているページを解析対象にします。
- ルートレコード型に定義されているサブページ分割数以降は，0 が表示されます。

(4) 解析結果の分析

(a) ページの割り当て状態による分析

解析結果に出力された総未使用ページ比率が，SDB データベースの定義時に指定したセグメント内の空きページ比率とかい離している場合は，SDB データベースのデータが次のどちらかの状態であると考えられます。

- レコードを繰り返し格納したため，未使用ページが少なくなっている。
- データの削除を繰り返したため，空きページが多くなっている。

このような場合は，必要に応じてルートレコード型が格納されている RD エリアを拡張したあとに，SDB データベースを再編成することで，ページの割り当て状態の乱れを修正できます。

(b) 詳細情報による分析

RD エリア単位の状態解析（論理的解析）の解析結果から，次の表のように分析できます。

表 10–7 RD エリア単位（論理的解析）の解析結果による分析

項番	出力項目	分析内容
1	通常のページの使用比率別ページ数（Normal Page information の Used Page Ratio）	• 使用率が低いページが多い場合 使用率が低いページから，占有状態のページ数を除外し，この値が PCTFREE オペランドの指定値とかい離しているかどうかを確認してください。かい離している場合は，データの削除や更新によって，一度割り当てられた領域が未解放のまま，空き領域として残っています。再使用できない状態になっているため，SDB データベースの再編成が必要です。
2	通常のサブページの使用比率別サブページ数（Normal SubPage information の Used SubPage Ratio）	• 使用率が低いサブページが多い場合 使用率が低いサブページから，占有状態のサブページ数を除外し，この値が PCTFREE オペランドの指定値とかい離しているかどうかを確認してください。かい離している場合は，データの削除や更新によって，一度割り当てられた領域が未解放のまま，空き領域として残っています。再使用できない状態になっているため，SDB データベースの再編成が必要です。

項番	出力項目	分析内容
3	事前割り当てページの使用比率別ページ数（Reserved Page information の Used Page Ratio）	• 使用率 0%の使用ページ数が少ない場合 使用率 0%のページとは，レコードが格納されていないページのことです。このページ数が少ない場合は，事前割り当てページとしてレコードを格納できる残りのページ数が少なくなっています。割り当て済みの事前割り当てページにレコードを格納できなくなると，追加で事前割り当てページを割り当てることができます。ただし，これが多発すると，非連続のページにファミリ内のレコードが分散して配置されるため，出力項目「非連続に割り当てられた事前割り当て（サブ）ページ数」の分析内容に示す対策が必要です。
4	事前割り当てサブページの使用比率別サブページ数（Reserved SubPage information の Used SubPage Ratio）	• 使用率 0%の使用サブページ数が少ない場合 使用率 0%のサブページとは，レコードが格納されていないサブページのことです。このサブページ数が少ない場合は，事前割り当てサブページとしてレコードを格納できる残りのサブページ数が少なくなっています。割り当て済みの事前割り当てサブページにレコードを格納できなくなると，追加で事前割り当てサブページを割り当てることができます。ただし，これが多発すると，非連続のページにファミリ内のレコードが分散して配置されるため，出力項目「非連続に割り当てられた事前割り当て（サブ）ページ数」の分析内容に示す対策が必要です。
5	使用サブページ数ごとのページ数（Normal SubPage information の Used SubPage Count per Page）	• 使用サブページが少ないページが多い場合 使用ページ数に対して未使用サブページが多く存在しているため，それが性能低下の原因となっています。SDB データベースの再編成が必要です。
6	再編成による解放可能なページ数（Normal Page information または Normal SubPage information の Collect Prearranged Page）	• 再編成による解放可能なページ数が多い場合 データの削除によって，一度割り当てられた領域が未解放のまま空きページとして残っています。それが性能低下の原因となっています。再編成による解放可能ページ数の，通常のページ数の合計に対する比率が 30%以上の場合は，SDB データベースの再編成が必要です。
7	削除状態の管理行数（各 information の Del RowID）	• 削除済みレコードの管理情報数が多い場合 データの削除によって，レコードの管理情報が未使用状態のままページに残っています。それが性能低下の原因となっています。同一（サブ）ページ種別での削除済みレコードの管理情報の，総レコードの管理行数に対する比率が，30%以上の場合は，SDB データベースの再編成が必要です。 ページ種別： 通常のページ，通常のサブページ，事前割り当てページ，および事前割り当てサブページを表す種別のことです。 管理行数： ファミリ格納件数，非ルートレコード格納件数，および削除済みレコードの管理情報数の合計です。
8	レコード集約度（各 information の Collection Degree）	見積もり値と比較した結果，次のどちらかの状態になっている場合は，SDB データベースの再編成が必要です。見積もり値の見積もり

項番	出力項目	分析内容
		方法については，「(5)　レコードの集約度の見積もり方法」を参照してください。 • ページまたがり回数が多い，またはページ内のレコード集約度が低い場合 親子近傍配置に乱れが発生しているため，レコード間ポインタをたどる際に，ページまたがりが多く発生しています。それが性能低下の原因となっています。SDB データベースの再編成を実行して，レコードを配置し直す必要があります。 • セグメント内のレコード集約度が低い場合 親子近傍配置に乱れが発生したため，レコード間ポインタをたどる際に，セグメントまたがりが多く発生しています。それが性能低下の原因となっています。SDB データベースの再編成を実行して，レコードを配置し直す必要があります。
9	非連続に割り当てられている事前割り当て（サブ）ページ数（Reserved Page information の NonContinuous Reserved Pages または Reserved SubPage information の NonContinuous Reserved SubPages）	• 非連続に割り当てられた事前割り当て（サブ）ページ数が多い場合 データベースの初期作成後の運用によって，割り当てられた事前割り当て（サブ）ページが非連続になっています。それが性能低下の原因となっています。SDB データベースの再編成を実行して，事前割り当て（サブ）ページが連続領域に割り当てられるようにしてください。
10	子レコードを 1 件も管理していない仮想ルートレコードの数（Normal Page information または Normal SubPage information の Virtual Root Record(disuse)）	• 子レコードを 1 件も持たない仮想ルートレコードが多い場合 レコードの削除（ERASE）または一括削除によって，子レコードがすべて削除された仮想ルートレコードによる不要な領域が増加しています。SDB データベースの再編成を実行して，その仮想ルートレコードを解放する必要があります。ただし，その後，レコードを格納する際に，この領域が再利用されることがあるため，今後の運用予定を考慮し，SDB データベースを再編成するかどうかを判断する必要があります。
11	論理的に削除状態になっている非ルートレコードの数（Normal Page information または Normal SubPage information の Logical Cleared Record）	• 論理的に削除状態になっている非ルートレコードの件数が多い場合 一括削除によって論理的に削除状態になっている非ルートレコードによる領域が増加しています。SDB データベースの再編成を実行して，その領域を解放する必要があります。ただし，その後，レコードを格納する際に，この領域が再利用されることがあるため，今後の運用予定を考慮し，SDB データベースを再編成するかどうかを判断する必要があります。

レコード格納用 RD エリアの状態解析結果と対策を次の表に示します。

表 10–8　レコード格納用 RD エリアの状態解析結果と対策

RD エリア内の状態			対策
ページ	空きページ	多い	SDB データベースを定義したときのセグメント内の空きページ比率が大き過ぎます。または，レコードの削除によって，空きページができています。SDB データベースの再編成が必要です。
		少ない	RD エリア内に空きセグメントが少ない場合は，RD エリアの拡張が必要です。

RD エリア内の状態			対策
			サブページ分割機能適用時，または事前ページ割り当て機能適用時には，pddbst コマンドの実行時に-d オプションを指定して，詳細情報を解析することを推奨します。

(5)　レコードの集約度の見積もり方法

レコードの集約度について，データベース状態解析ユティリティ（pddbst）による解析結果（実態値）と，見積もった値を比較した結果，両方の値にかい離がある場合，基本的には SDB データベースの再編成の効果が見込めます。そのため，解析結果の値と比較するために，事前に見積もりの値を算出しておく必要があります。

なお，事前割り当て（サブ）ページと通常（サブ）ページでは，再編成契機を判断する基準が異なるため，各項目をそれぞれ見積もってください。レコードの集約度の見積もり方法を以降で説明します。

(a)　使用中ページ数，使用中サブページ数

- 事前割り当てページまたは事前割り当てサブページ

  「3.5.5(3)　事前割り当てページ数または事前割り当てサブページ数の算出【4V FMB】」を参照して，事前ページ割り当て機能を適用する全ファミリ分の事前割り当てページ数または事前割り当てサブページ数を求めてください。

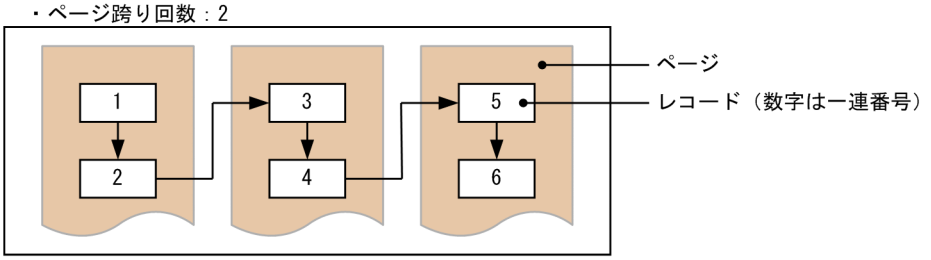
- 通常のページまたは通常のサブページ

  「3.5.5(4)　平均的なファミリが使用するページ数またはサブページ数の算出」を参照して，事前ページ割り当て機能を適用しない全ファミリ分のページ数またはサブページ数を求めてください。

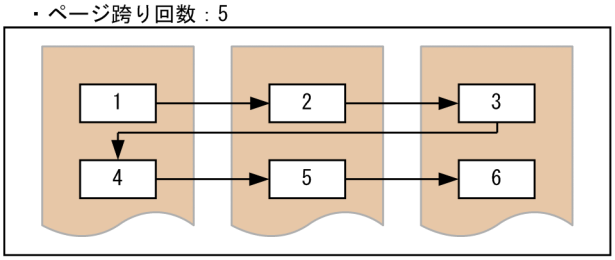
(b)　ページの跨り回数

レコード間ポインタを辿って検索を行う際，なるべくページ替わりが発生しない方がアクセス性能が良くなります。したがって，1 ファミリに着目した場合，最も理想的なページ跨り回数は，「当該ファミリによる使用ページ数－1」になります。

■理想的な状況
・使用ページ数：3
・ページ跨り回数：2



■不要なページ跨りの発生状況
・使用ページ数：3
・ページ跨り回数：5



（凡例）⟶：レコード間ポインタ

ただし，見積もり時には，単純に「使用中ページ数－1」とするのではなく，次の理由によって，ファミリ間のページ替わり回数分を考慮する必要があります。

- 1RD エリアに対して複数のファミリを格納することができる。
- ページおよびサブページを異なるファミリで共有しない。
- 異なるファミリ間はポインタによって接続されない。

■サブページ分割を適用していない場合

- **事前割り当てページ**

  ページ跨り回数の見積もり＝

  （レコードが格納されている事前割り当てページ数－1）

  －（事前割り当てページを使用しているファミリ数－1）※

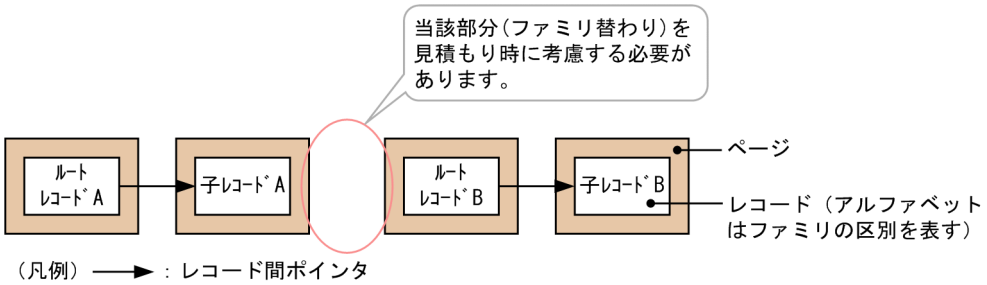
- **通常ページ**

  ページ跨り回数の見積もり＝

  （通常ページの使用ページ数－1）

  －（事前割り当てページを使用していないファミリ数－1）※

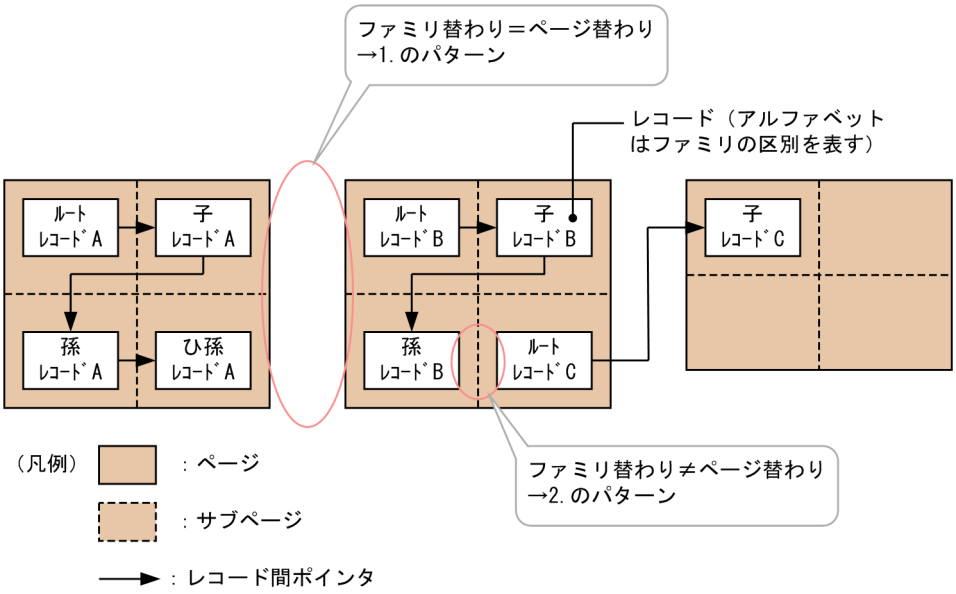
注※　ファミリ間のページ替わり回数です。



■サブページ分割を適用している場合

サブページ分割の適用時には，異なるファミリを同一ページ（異なるサブページ）に格納することがあるため，ファミリ替わりのタイミングとして次の２つのパターンがあります。

1. ページ内の全サブページを使い切った上でファミリが切り替わる。
2. ページの途中でファミリが切り替わる。



サブページ分割を適用していないときのように，ファミリの切り替わりとページの切り替わりが必ずしも一致しないため，見積もり式を共通化することはできません。そのため，サブページ分割の適用時には，平均ファミリサイズ（平均的なファミリが使用するページまたはサブページ数）を基に，ページ跨り発生比率を求めた上で，当該比率と使用中ページ数からページ跨り回数を見積もってください。

なお，平均ファミリサイズの単位は，サブページ分割を適用していないときはページとなり，サブページ分割を適用しているときはサブページとなります。

平均ファミリサイズ（平均的なファミリが使用するページまたはサブページ数）の見積もり方法については，「3.5.5(4)　平均的なファミリが使用するページ数またはサブページ数の算出」を参照してください。

- **事前割り当てサブページ**

  ページ跨り回数の見積もり＝

  （レコードが格納されている事前割り当てサブページ数÷サブページ分割数）

  ×ページ跨り発生比率

ページ跨り発生比率※＝（A－1）÷A

A ＝↑B÷サブページ分割数↑

B ＝最小公倍数（事前割り当てサブページを使用するファミリの平均ファミリサイズ，サブページ分割数）

注※　ページ跨り（ファミリ替わり≠ページ替わり）が，Aページごとに「A－1」回発生します。

- **通常サブページ**

ページ跨り回数の見積もり＝

（通常サブページの使用サブページ数÷サブページ分割数）

×ページ跨り発生比率

ページ跨り発生比率※＝（C－1）÷C

C ＝↑D÷サブページ分割数↑

D ＝最小公倍数（事前割り当てサブページを使用しないファミリの平均ファミリサイズ，サブページ分割数）

注※　ページ跨り（ファミリ替わり≠ページ替わり）が，Cページごとに「C－1」回発生します。

計算例を次に示します。

（例）

- データベース状態解析ユティリティによる解析結果

  使用中ページ数：100

- 事前準備による見積もり結果

  サブページ分割数：7

  平均ファミリサイズ：3サブページ

- ページ跨り回数の見積もり

  α ＝↑最小公倍数（平均ファミリサイズ，サブページ分割数）÷サブページ分割数↑

  ＝↑21÷7↑

  ＝3

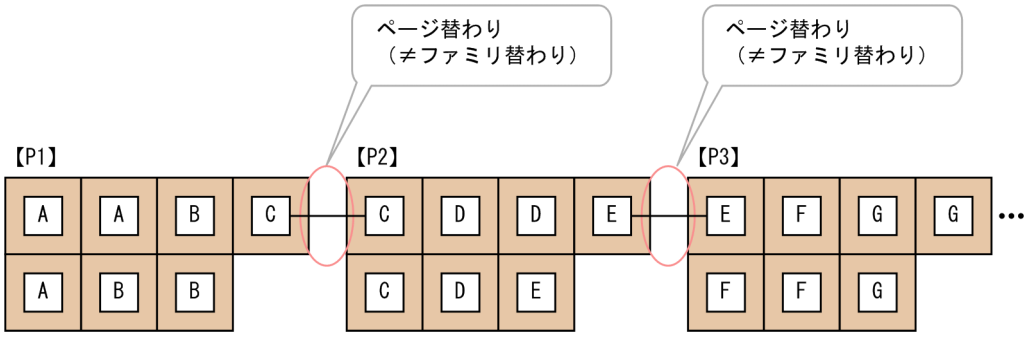
  ページ跨り発生比率＝（α－1）÷α

  ＝（3－1）÷3

  ＝2÷3

したがって，平均ファミリサイズのファミリを順次格納していくと，3ページ当たり，ページ跨りが2回発生します。

ページ跨り回数の見積もり＝（100×2）÷3＝66



なお，次のケースでは，実際のページ跨り回数との間で誤差が発生することがあります。

- SDB データベースの実際の格納状況で，各ファミリのサイズに幅がある場合
- 各ファミリに割り当てた事前割り当てサブページ数に幅がある場合

(c)　ページ内のレコード集約度

PRIOR ポインタが同一ページのレコードを指しているレコードの比率を表す「ページ内のレコード集約度」は，「(b)　ページの跨り回数」の見積もり結果を利用することによって見積もることができます。

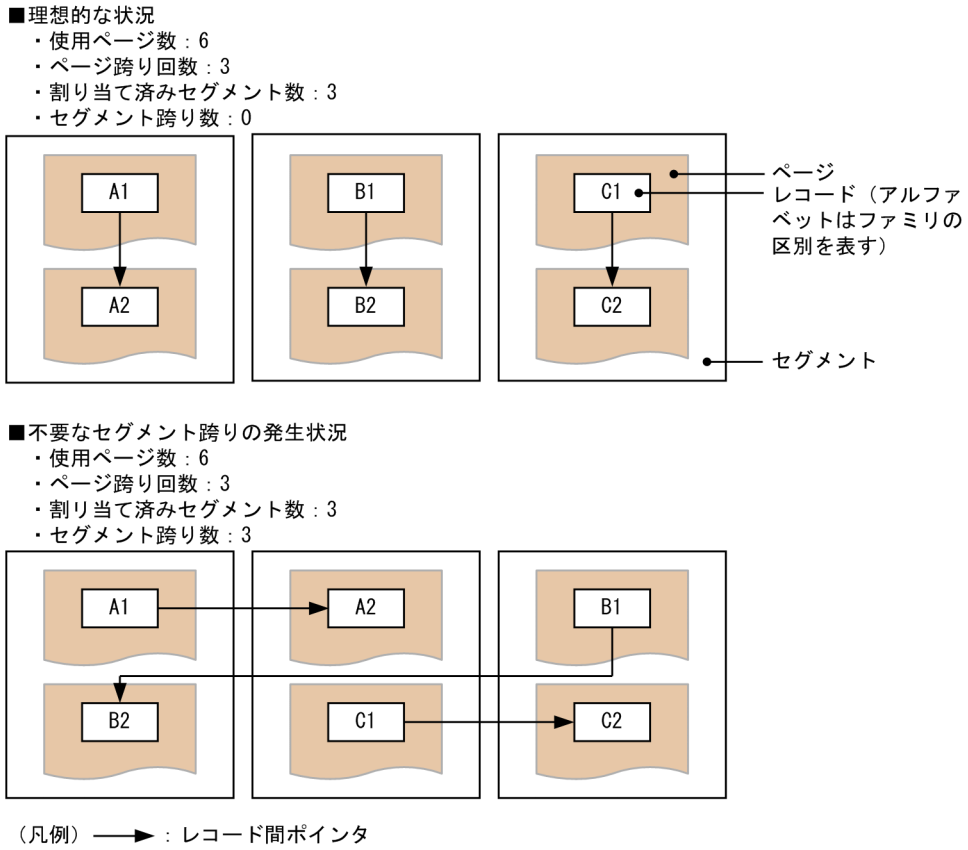
なお，この見積もり式では，すべてのレコードを対象としてレコード集約度を見積もるため，ページ切り替えオプションに'O'や'C'を指定したレコードが多い場合，SDB データベースを設計したときのとおりに格納されたとしても，見積もり結果の値が低くなることがあります。

ページ内のレコード集約度の見積もり＝
　　PRIOR ポインタが同一ページのレコードを指しているレコード数
　　÷格納レコードの合計数
PRIOR ポインタが同一ページのレコードを指しているレコード数
　　＝格納レコードの合計数－ページの跨り回数

(d)　セグメント内のレコード集約度

レコード間ポインタを辿って検索を行う際，なるべくセグメント替わりが発生しない方がアクセス性能が良くなります。そのため，SDB データベースの設計時にセグメントサイズを「平均的なファミリの使用ページ数」と同じにすることを推奨しています。

したがって，最も理想的なセグメント内のレコード集約度は 100％になります。



ただし，前述のとおり，セグメントサイズは平均なファミリの使用ページ数を基に設計するため，各ファミリの使用ページ数に幅がある場合，セグメント内のレコード集約度が必ずしも 100%にならないことがあります。

10.7.4　RD エリア単位の状態解析（物理的解析）

RD エリア単位の状態解析（物理的解析）では，RD エリア内の全セグメントおよび全ページの格納状態を解析します。

次の出力項目以外については，マニュアル「HiRDB コマンドリファレンス」の「RD エリア単位の状態解析（物理的解析)」を参照してください。

- FULL_PAGE（満杯ページ数）
  - サブページ分割機能が適用されていない場合
    次の条件を満たすページ数の合計が表示されます。
    ・これ以上データを格納することができない使用ページ数
    ・占有状態の使用ページ数
  - サブページ分割が適用されている場合
    すべてのサブページが使用サブページである使用ページ数が表示されます。

10.8 JP1/NETM/Audit 用監査ログ出力ユティリティ（pdaudput）

JP1/NETM/Audit 用監査ログ出力ユティリティ（pdaudput）は，HiRDB の監査証跡表のデータを，JP1/NETM/Audit で読み込める形式でファイルに出力するユティリティです。JP1/NETM/Audit 用監査ログ出力ユティリティが出力する監査証跡表のデータを JP1/NETM/Audit 用監査ログといいます。また，出力されたファイルを，JP1/NETM/Audit 用監査ログ出力ファイルといいます。

JP1/NETM/Audit 用監査ログ出力ユティリティ（pdaudput）の機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差があります。ここでは，その仕様差についてだけ説明します。

10.8.1 JP1/NETM/Audit 用監査ログ出力ファイルの内容

JP1/NETM/Audit 用監査ログ出力ファイルの項目のうち，構造型 DB 機能の監査対象イベントの種別を次の表に示します。

表 10–9 構造型 DB 機能が出力するイベントタイプおよびイベントサブタイプと，pdaudput が出力する監査対象イベントの種別についての対応

構造型 DB 機能が出力するイベントタイプおよびイベントサブタイプ				pdaudput が出力する値と説明	
監査イベント	イベントタイプの値	イベントサブタイプの値	イベント	EventCategory に出力する値	説明
SDB データベース操作イベント	SDA	FTC	• レコードの検索（FETCH） • 位置指示子の位置づけ（FIND）	ContentAccess	重要情報アクセス
		STR	レコードの格納（STORE）	ContentAccess	重要情報アクセス
		MOD	レコードの変更（MODIFY）	ContentAccess	重要情報アクセス
		ERS	レコードの削除（ERASE）	ContentAccess	重要情報アクセス
		CLA	レコードの一括削除	ContentAccess	重要情報アクセス
		FTA	複数レコードの検索（FETCHDB ALL）	ContentAccess	重要情報アクセス
		GET	レコードの取得（GET）	ContentAccess	重要情報アクセス

構造型 DB 機能が出力するイベントタイプおよびイベントサブタイプ				pdaudput が出力する値と説明	
監査イベント	イベントタイプの値	イベントサブタイプの値	イベント	EventCategory に出力する値	説明
SDB ユティリティ操作イベント	SDU	DEF	HiRDB/SD データベース定義ユティリティ（pdsdbdef）	ContentAccess	重要情報アクセス
		LOD	HiRDB/SD データベース作成ユティリティ（pdsdblod）	ContentAccess	重要情報アクセス
		ROG	HiRDB/SD データベース再編成ユティリティ（pdsdbrog）	ContentAccess	重要情報アクセス

JP1/NETM/Audit 用監査ログ出力ファイルの項目のうち，構造型 DB 機能のサブジェクト識別情報を次の表に示します。

表 10–10　構造型 DB 機能のイベントおよびイベントタイプとサブジェクト識別情報の対応

イベントの種類			サブジェクト識別情報
監査イベント	イベントタイプの値	イベントサブタイプの値	
SDB データベース操作イベント	SDA	FTC	AccountID
		STR	AccountID
		MOD	AccountID
		ERS	AccountID
		CLA	AccountID
		FTA	AccountID
		GET	AccountID
SDB ユティリティ操作イベント	SDU	DEF	AccountID
		LOD	AccountID
		ROG	AccountID

10.9 運用コマンド

ここでは，次の運用コマンドについて説明します。

- pdsdbarc（SDB ディレクトリ情報の常駐化，および最終更新日時のチェックをするコマンド）
- マニュアル「HiRDB コマンドリファレンス」に記載されている内容と差異があるコマンド

10.9.1 pdsdbarc（SDB ディレクトリ情報の常駐化および最終更新日時のチェック）

(1) 機能

pdsdbarc コマンドには次に示す機能があります。

- SDB ディレクトリ情報を事前常駐領域に常駐する。
- SDB ディレクトリ情報の最終更新日時のチェックを行う。
- SDB ディレクトリ情報に関する情報を表示する。
- 事前常駐領域を常用常駐領域に切り替える。

(2) 実行者

HiRDB 管理者が実行できます。

(3) 形式

■SDB ディレクトリ情報の事前常駐領域への常駐，最終更新日時のチェック，または情報表示の場合

```
pdsdbarc {-e | -c | -a} 〔-u ユニット識別子〔，ユニット識別子〕…〕
```

■事前常駐領域を常用常駐領域に切り替える場合

```
pdsdbarc -w -q
```

(4) オプション

- -e

  SDB ディレクトリ情報ファイルから SDB ディレクトリ情報を読み込み，事前常駐領域に常駐します。

  HiRDB/SD は，事前常駐領域中の SDB ディレクトリ情報の最終更新日時と，ディクショナリ表中の SDB ディレクトリ情報の最終更新日時をチェックします。チェックの結果，ディクショナリ表中の

SDB ディレクトリ情報の最終更新日時の方が古い場合，事前常駐領域中の SDB ディレクトリ情報を無効状態にします。この場合，出力されたメッセージの対処に従ってください。

- -c

  常用常駐領域中の SDB ディレクトリ情報の最終更新日時のチェックを行います。

  HiRDB/SD は，常用常駐領域中の SDB ディレクトリ情報の最終更新日時と，ディクショナリ表中の SDB ディレクトリ情報の最終更新日時をチェックします。チェックの結果，ディクショナリ表中の SDB ディレクトリ情報の最終更新日時の方が古い場合，常用常駐領域中の SDB ディレクトリ情報を無効状態にします。この場合，出力されたメッセージに従って対処してください。SDB ディレクトリ情報が無効状態になったユニットでは SDB データベースへのアクセスができなくなります。

  注意事項

  チェックの結果によっては，常用常駐領域中の SDB ディレクトリ情報が無効になることがあります。そのため，業務実行中に pdsdbarc -c コマンドを実行しないようにしてください。pdsdbarc -c コマンドは，HiRDB の開始直後など，業務実行中でないときに実行するようにしてください。

- -a

  ディクショナリ表，SDB ディレクトリ情報ファイル，事前常駐領域，および常用常駐領域の SDB ディレクトリ情報に関する情報を表示します。

- -u ユニット識別子〔，ユニット識別子〕…

  pdsdbarc コマンドの処理対象とするユニットのユニット識別子を指定します。-u オプションを省略した場合，全ユニットが pdsdbarc コマンドの処理対象になります。

  なお，複数のユニット識別子を指定する場合は，コンマの前後に空白を入力しないでください。

- -w -q

  事前常駐領域を常用常駐領域に切り替え，常用常駐領域を事前常駐領域に切り替えます。

  pdsdbarc -w -q コマンドは，事前常駐領域に SDB ディレクトリ情報が正しく常駐されているユニットに対してだけ実行できます。事前常駐領域の SDB ディレクトリ情報が無効状態の場合は，その事前常駐領域を常用常駐領域に切り替えることはできません。

  注意事項

  pdsdbarc -w -q コマンドを実行する前に次に示すことを確認してください。

  - トランザクションが静止化している。トランザクションの静止化方法については，「5.7.3(8) トランザクションを静止化する」を参照してください。
  - 実行中のコマンドおよびユティリティがない。
  - ロールバック中のトランザクションがない。

  常駐領域の切り替え後，事前常駐領域中の SDB ディレクトリ情報は無効状態になります。そのため，pdsdbarc -w -q コマンドの実行後，再度 pdsdbarc -w -q コマンドを実行することはできません。

(5) 規則

- pdsdbarc コマンドは，HiRDB が稼働中のときに実行できます。
- pdsdbarc コマンド（-e，-c，および-a 指定）を実行する場合は，次に示すクライアント環境定義を指定しておく必要があります。
  - PDUSER：CONNECT 権限を持ったユーザの認可識別子とパスワードを指定します。
    このとき，OS ログインユーザの簡易認証機能は使用できません。
- pdsdbarc コマンド（-e，-c，および-a 指定）は，システムマネジャがあるサーバマシンで実行してください。
- pdsdbarc -w -q コマンドは，コマンドの実行対象とするユニットがある各サーバマシンで実行してください。
- pdsdbarc コマンドのリターンコードが 0 の場合は，pdsdbarc コマンドは正常終了しています。0 以外の場合は pdsdbarc コマンドが異常終了しています。出力されたメッセージに従って対処してください。

(6) 留意事項

- pdsdbarc コマンドは複数同時に実行できません。あとから実行しようとした pdsdbarc コマンドはエラーになります。
- pdsdbarc コマンド（-e または-w -q オプション指定）を実行する場合は，pd_structured_advance_resident オペランドに use を指定している必要があります。use を指定していない場合，pdsdbarc コマンドは実行できません。
- 高速系切り替え機能を使用している場合，実行系ユニットだけが pdsdbarc コマンドの処理対象になります。待機系ユニットは処理対象になりません。
- -u オプションの指定を省略するか，または-u オプションに複数のユニットを指定した場合，1 つのユニットに対するコマンド処理が失敗しても処理を続行し，次のユニットに対してコマンド処理を行います。ただし，通信障害が発生した場合は，コマンド処理を行いません。
- pdsdbarc コマンド（-e，-c，および-a 指定）を実行する場合は，次に示すクライアント環境定義の指定は無視されます。
  - PDSQLTRACE
  - PDVWOPTMODE
  - PDCLTAPNAME
  - PDAUTORECONNECT

(7) リターンコード

pdsdbarc コマンドのリターンコードを次に示します。

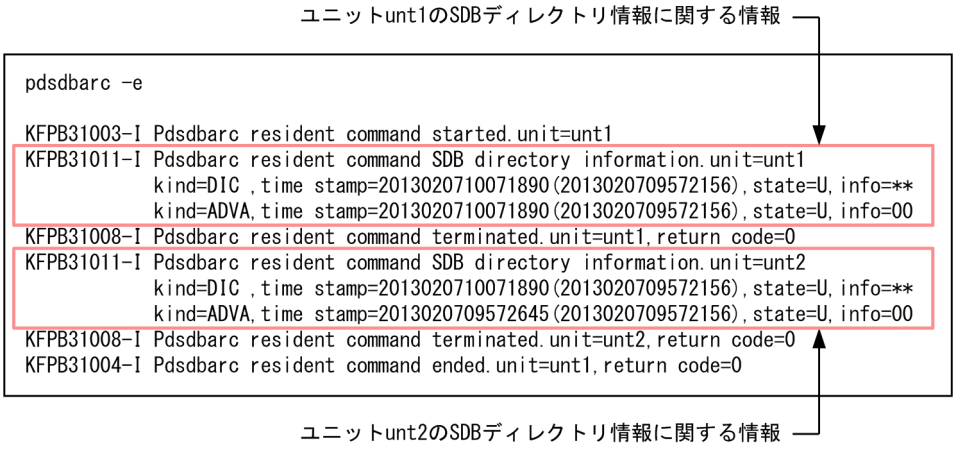
- 0：pdsdbarc コマンドが正常終了しました。

- 4：pdsdbarc コマンドが異常終了しました。ただし，一部のユニットに対しては処理が完了しています。または，警告メッセージが出力されています。
- 8：pdsdbarc コマンドが異常終了しました。

(8)　出力形式

pdsdbarc コマンドの実行結果（標準出力への出力形式）について説明します。

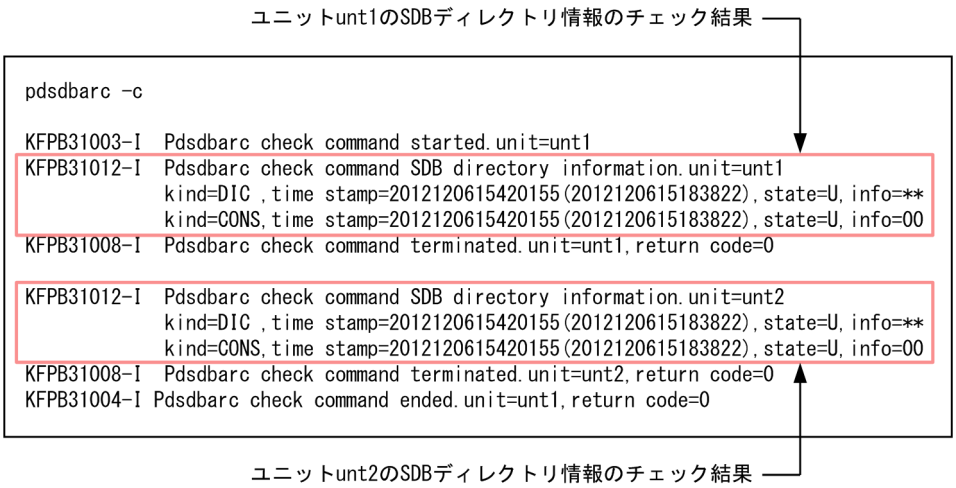
(a)　pdsdbarc -e コマンドの実行時（SDB ディレクトリ情報を事前常駐領域に常駐する場合）



KFPB31011-I メッセージの出力内容の各項目については，「(9)　pdsdbarc コマンドの実行結果の出力内容」を参照してください。

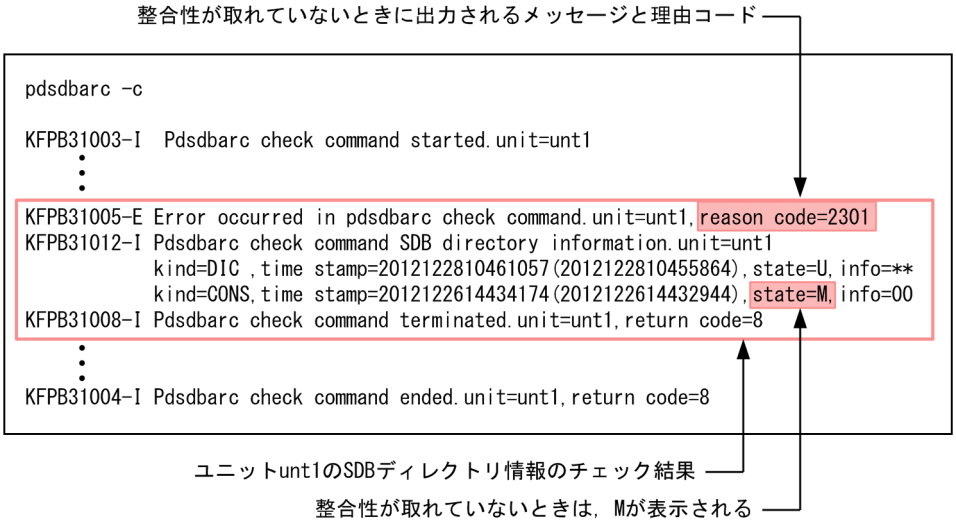
(b)　pdsdbarc -c コマンドの実行時（SDB ディレクトリ情報の最終更新日時をチェックする場合）

■チェックの結果，整合性が取れている場合の例

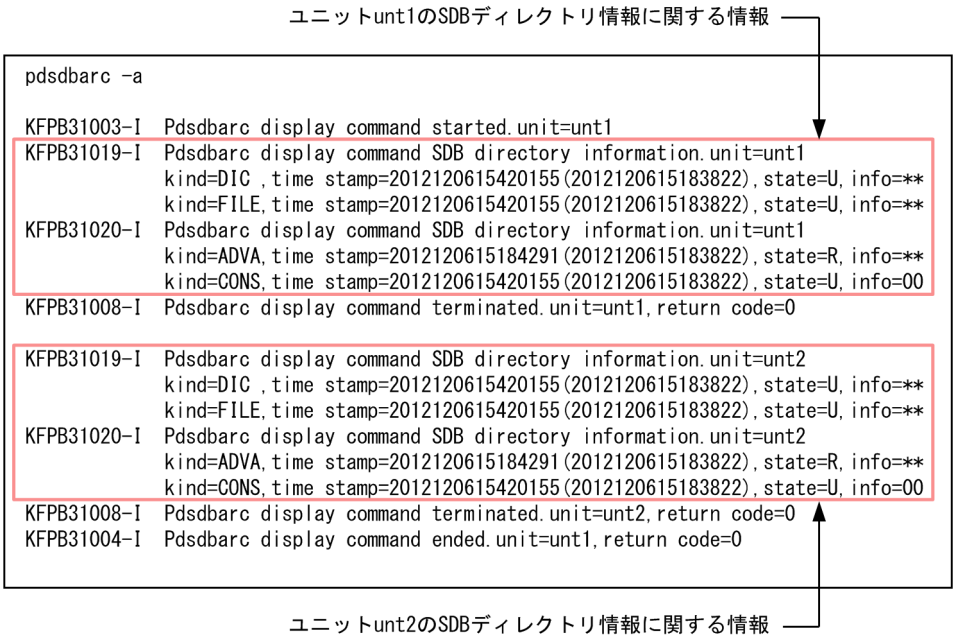


KFPB31012-I メッセージの出力内容の各項目については，「(9)　pdsdbarc コマンドの実行結果の出力内容」を参照してください。

■チェックの結果，整合性が取れていない場合の例



(c)　pdsdbarc -a コマンドの実行時（SDB ディレクトリ情報に関する情報を表示する場合）



KFPB31019-I および KFPB31020-I メッセージの出力内容の各項目については，「(9)　pdsdbarc コマンドの実行結果の出力内容」を参照してください。

(d) pdsdbarc -w -q コマンドの実行時（事前常駐領域を常用常駐領域に切り替える場合）

```
pdsdbarc -q -w

KFPB31014-I Pdsdbarc switch command terminated.unit=unt1,return code=0
```

ユニット unt1 の事前常駐領域を常用常駐領域に切り替えた例です。

(9) pdsdbarc コマンドの実行結果の出力内容

pdsdbarc コマンドの実行結果の出力内容を次の表に示します。

表 10–11 pdsdbarc コマンドの実行結果の出力内容

出力情報	説明
unit	ユニット識別子
kind	SDB ディレクトリ情報の格納場所 • DIC：ディクショナリ表 • FILE：SDB ディレクトリ情報ファイル • ADVA：事前常駐領域 • CONS：常用常駐領域
time stamp	SDB ディレクトリ情報の最終更新日時 （ ）内は SDB 定義文の最終更新日時
state	SDB ディレクトリ情報の状態 • U：有効な状態です。 • E：常駐化に失敗している状態です。 • I：SDB ディレクトリ情報の常駐処理が行われていない状態です。 • M：SDB ディレクトリ情報の整合性が取れていない状態です。 • N：SDB ディレクトリ情報が存在しない状態です。 • R：無効な状態です。事前常駐領域を常用常駐領域に切り替えたため，切り替え前に使用していた SDB ディレクトリ情報を無効にしました。 • S：事前常駐領域を使用しない設定の場合に表示されます。pd_structured_advance_resident オペランドに nouse を指定するか，またはこのオペランドを省略している場合に表示されます。 • T：処理が競合したため，事前常駐処理が失敗した状態です。 • W：常用常駐領域と事前常駐領域の切り替え処理で内部矛盾が発生した状態です。 • *：日時情報の取得に失敗した状態です。

(10) ディクショナリ表中の最終更新日時と SDB ディレクトリ情報ファイル中の最終更新日時が異なる場合の SDB データベースへのアクセス制限

SDB データベースへのアクセスが実行される際，HiRDB/SD はディクショナリ表中の次に示す最終更新日時と，SDB ディレクトリ情報ファイル中の次に示す最終更新日時が一致しているかどうかを確認しています。

- SDB 定義文の最終更新日時
- SDB ディレクトリ情報の最終更新日時

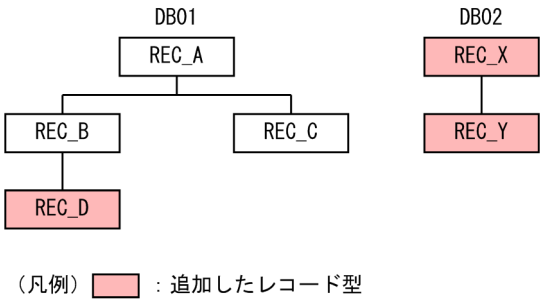
ディクショナリ表中の上記 2 つの最終更新日時と，SDB ディレクトリ情報ファイル中の上記 2 つの最終更新日時が一致している場合は，SDB データベースにアクセスできます。一方，ディクショナリ表中の上記 2 つの最終更新日時と，SDB ディレクトリ情報ファイル中の上記 2 つの最終更新日時が両方とも一致していない場合は，SDB データベースにアクセスできません。

なお，次に示す場合は，一部のレコード型にアクセスできません。

SDB 定義文の最終更新日時は一致しているが，SDB ディレクトリ情報の最終更新日時が一致していない場合

> **参考**
>
> HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更をしてから，事前常駐領域を常用常駐領域に切り替えるまでの間は，上記の状態になります。

（例 1）レコード型の追加の場合



上記のようにレコード型を追加した場合，アクセスできるレコード型は次のようになります。

アクセス対象		pd_structured_advance_resident オペランドの指定値	アクセス可否
SDB データベース	レコード型		
DB01	REC_A	use	○
	REC_B		○
	REC_C		○
	REC_D		×※1
DB02	REC_X		×※1

アクセス対象		pd_structured_advance_resident オペランドの指定値	アクセス可否
SDB データベース	レコード型		
	REC_Y		×※1
DB01	REC_A	nouse または省略	×※2
	REC_B		×※2
	REC_C		×※2
	REC_D		×※2
DB02	REC_X		×※2
	REC_Y		×※2

（凡例）

○：アクセスできます。

×：アクセスできません。

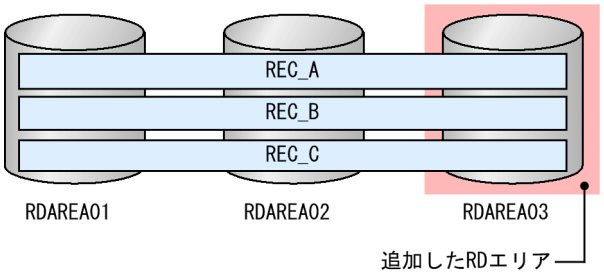
注※1

新しい SDB ディレクトリ情報ファイルを全ユニットに配布して，事前常駐領域を常用常駐領域に切り替えたあとにアクセスできるようになります。

注※2

新しい SDB ディレクトリ情報ファイルを全ユニットに配布して，HiRDB を再起動したあとにアクセスできるようになります。

（例 2）RD エリアの追加の場合



上記のように格納 RD エリアを追加した場合，アクセスできる RD エリアのレコード型は次のようになります。

アクセス対象		pd_structured_advance_resident オペランドの指定値	アクセス可否
格納 RD エリア	レコード型		
RDAREA01	REC_A	use	○
	REC_B		○
	REC_C		○
RDAREA02	REC_A		○
	REC_B		○

アクセス対象		pd_structured_advance_resident オペランドの指定値	アクセス可否
格納 RD エリア	レコード型		
	REC_C		○
RDAREA03	REC_A		×※1
	REC_B		×※1
	REC_C		×※1
RDAREA01	REC_A	nouse または省略	×※2
	REC_B		×※2
	REC_C		×※2
RDAREA02	REC_A		×※2
	REC_B		×※2
	REC_C		×※2
RDAREA03	REC_A		×※2
	REC_B		×※2
	REC_C		×※2

（凡例）

○：アクセスできます。

×：アクセスできません。

注※1

新しい SDB ディレクトリ情報ファイルを全ユニットに配布して，事前常駐領域を常用常駐領域に切り替えたあとにアクセスできるようになります。

注※2

新しい SDB ディレクトリ情報ファイルを全ユニットに配布して，HiRDB を再起動したあとにアクセスできるようになります。

10.9.2 pdbkupls（バックアップファイルの情報表示）

(1) 機能

pdcopy コマンドで取得したバックアップの情報（RD エリアの一覧，バックアップ取得日時など）を表示します。

pdbkupls コマンドの機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

なお，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

(2) 実行者

HiRDB 管理者が実行できます。

(3) 出力形式

pdbkupls コマンドの実行結果に出力される RD エリアの種別（<RDAREA kind>）に，次に示す情報も出力されます。

USER_STRUCTURED：SDB データベースを格納しているユーザ用 RD エリア

10.9.3 pddbls（RD エリアの状態表示）

(1) 機能

RD エリアの状態を表示します。

pddbls コマンドの機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」を参照してください。

なお，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差がある部分があります。ここでは，その仕様差についてだけ説明します。

(2) 実行者

HiRDB 管理者が実行できます。

(3) 形式

```
pddbls  { {-r RDエリア名〔, RDエリア名〕…|-r ALL
          |-s サーバ名〔, サーバ名〕…}
        〔 {-l|-a} 〕  〔 {-b|-o} 〕  〔-m〕  〔-x 〔-y〕〔-M〕〔-D〕〔-S〕〕
          | {-q 世代番号|-C}  〔- {l|a} 〕  〔-m〕  〔-x 〔-y〕〕〔-M〕〔-D〕〔-S〕}
```

(4) オプション

-S オプション以外の説明については，マニュアル「HiRDB コマンドリファレンス」の「pddbls（RD エリアの状態表示)」を参照してください。

- -S

  RD エリアに格納されているデータベースの形式を表示します。

(5) 出力形式

-S オプション指定時かつ RD エリア種別が USER（ユーザ用 RD エリア）の場合，RD エリアに格納されているデータベースの形式（[DATAMODEL]）として，次の情報が出力されます。

- STRUCTURED：SDB データベースを格納しているユーザ用 RD エリアを意味しています。
- RELATIONAL：次の表またはインデクスを格納しているユーザ用 RD エリアを意味しています。
  - 追い付き反映キー対応表を格納している RD エリア
  - 追い付き反映キー対応表のインデクスを格納している RD エリア
  - 追い付き状態管理表を格納している RD エリア
  - 監査証跡表を格納している RD エリア

10.9.4 pdrdrefls（関連 RD エリアの表示）

(1) 機能

関連する RD エリアの情報を表示します。pdrdrefls コマンドの機能および指定形式の詳細については，マニュアル「HiRDB コマンドリファレンス」の「pdrdrefls（関連する RD エリアの情報の表示)」を参照してください。ただし，マニュアル「HiRDB コマンドリファレンス」に記載されている内容とは，仕様差があります。ここでは，その仕様差についてだけ説明します。

(2) オプション

- -r ALL

  すべてのユーザ用 RD エリアを処理対象とする場合に指定します。

  同時に-c オプションを指定した場合，c オプションの指定は，リレーショナル DB を格納する RD エリアの情報表示では有効となりますが，SDB データベースを格納する RD エリアでは無視されます。

- -t 〔認可識別子.〕レコード型名

  ～<識別子>((認可識別子：1～8 バイト，レコード型名：1～30 バイト))

  SDB データベースの情報を表示する場合，-t オプションには表識別子ではなく 4V FMB の SDB データベースのレコード型名を指定します。

  このオプションを指定した場合，指定したレコード型を格納している全 RD エリアと，その RD エリア内に定義されている資源について解析を行い，関連 RD エリア名を表示します。

  同時に-c オプションを指定した場合はエラーとなります。

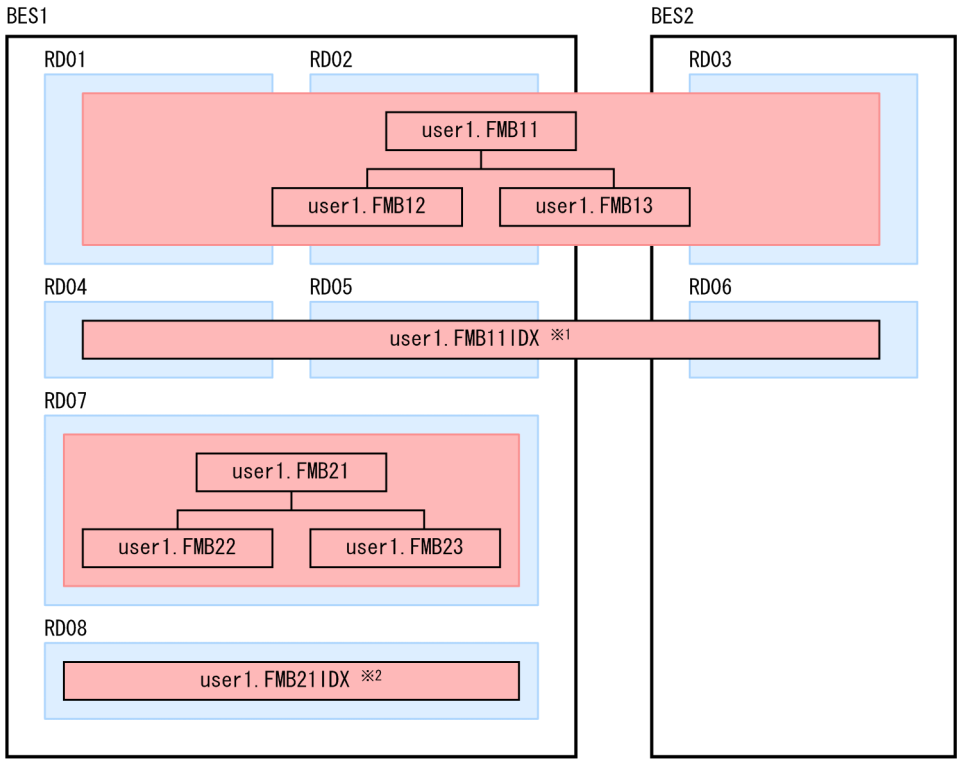
(3) 出力形式

ここでは，差異がある部分についてだけ説明します。

- 資源種別（3 バイト）：
  資源の種別です。
  TBL：レコード型
  IDX：インデクス
- 資源名（最大 43 バイト）：
  該当する RD エリアに格納されている資源の名称です。資源名は引用符で囲んで表示されます。
  資源種別が TBL の場合：認可識別子.レコード型名
  資源種別が IDX の場合：認可識別子.インデクス識別子
- 関連資源名（最大 45 バイト）：
  資源種別が IDX の資源の関連資源名です。関連資源名は（認可識別子.レコード型名）の形式で表示され，引用符で囲まれます。
- 制約による関連資源名（最大 47 バイト）：
  出力されません。

(4)　コマンドの実行結果の例

pdrdrefls コマンドを実行して，関連 RD エリアを表示する際の実行結果の例を示します。SDB データベースの構成は次のとおりとします。



（凡例）
RD01～RD08：RD エリア名

注※1　user1.FMB11 に定義したインデクス

注※2　user1.FMB21 に定義したインデクス

項番	オプションの指定				実行結果
	-r	-t	-l	-a	

1	RD01	−	−	○	SERVER NAME : BES1
					  RDAREA NAME : ”RD01”
					  AREA TYPE : USER
					  RESOURCE : TBL : ”user1”.”FMB11”
					  ”user1”.”FMB12”
					  ”user1”.”FMB13”
					  RDAREA NAME : ”RD04”
					  AREA TYPE : USER
					  RESOURCE : IDX : ”user1”.”FMB11IDX”
					  (”user1”.”FMB11”)

2		−	−	−	RDAREA NAME : ”RD01” RDAREA NAME : ”RD04”
3		−	○	○	BES1 ”RD01” USER TBL ”user1”.”FMB11” BES1 ”RD01” USER TBL ”user1”.”FMB12” BES1 ”RD01” USER TBL ”user1”.”FMB13” BES1 ”RD04” USER IDX ”user1”.”FMB11IDX” ”user1”.”FMB11”
4		−	○	−	”RD01” ”RD04”

5	RD02	−	−	○	SERVER NAME : BES1
					  RDAREA NAME : ”RD02”
					  AREA TYPE : USER
					  RESOURCE : TBL : ”user1”.”FMB11”
					  ”user1”.”FMB12”
					  ”user1”.”FMB13”
					  RDAREA NAME : ”RD05”
					  AREA TYPE : USER
					  RESOURCE : IDX : ”user1”.”FMB11IDX”
					  (”user1”.”FMB11”)

6	RD03	−	−	○	SERVER NAME : BES2
					  RDAREA NAME : ”RD03”
					  AREA TYPE : USER
					  RESOURCE : TBL : ”user1”.”FMB11”
					  ”user1”.”FMB12”
					  ”user1”.”FMB13”
					  RDAREA NAME : ”RD06”
					  AREA TYPE : USER
					  RESOURCE : IDX : ”user1”.”FMB11IDX”

項番	オプションの指定				実行結果
	-r	-t	-l	-a	
					("user1"."FMB11")
7	RD07	−	−	○	SERVER NAME : BES1 RDAREA NAME : "RD07" AREA TYPE : USER RESOURCE : TBL : "user1"."FMB21" "user1"."FMB22" "user1"."FMB23" RDAREA NAME : "RD08" AREA TYPE : USER RESOURCE : IDX : "user1"."FMB21IDX" ("user1"."FMB21")
8	−	FMB11	−	○	SERVER NAME : BES1 RDAREA NAME : "RD01" AREA TYPE : USER RESOURCE : TBL : "user1"."FMB11" "user1"."FMB12" "user1"."FMB13" RDAREA NAME : "RD02" AREA TYPE : USER RESOURCE : TBL : "user1"."FMB11" "user1"."FMB12" "user1"."FMB13" RDAREA NAME : "RD04" AREA TYPE : USER RESOURCE : IDX : "user1"."FMB11IDX" ("user1"."FMB11") RDAREA NAME : "RD05" AREA TYPE : USER RESOURCE : IDX : "user1"."FMB11IDX" ("user1"."FMB11") SERVER NAME : BES2 RDAREA NAME : "RD03" AREA TYPE : USER RESOURCE : TBL : "user1"."FMB11" "user1"."FMB12" "user1"."FMB13" RDAREA NAME : "RD06" AREA TYPE : USER RESOURCE : IDX : "user1"."FMB11IDX" ("user1"."FMB11")

項番	オプションの指定				実行結果
	-r	-t	-l	-a	
9			−	−	RDAREA NAME : ”RD01” RDAREA NAME : ”RD02” RDAREA NAME : ”RD04” RDAREA NAME : ”RD05” RDAREA NAME : ”RD03” RDAREA NAME : ”RD06”
10			○	○	BES1 ”RD01” USER TBL ”user1”.”FMB11” BES1 ”RD01” USER TBL ”user1”.”FMB12” BES1 ”RD01” USER TBL ”user1”.”FMB13” BES1 ”RD02” USER TBL ”user1”.”FMB11” BES1 ”RD02” USER TBL ”user1”.”FMB12” BES1 ”RD02” USER TBL ”user1”.”FMB13” BES1 ”RD04” USER IDX ”user1”.”FMB11IDX” ”user1”.”FMB11” BES1 ”RD05” USER IDX ”user1”.”FMB11IDX” ”user1”.”FMB11” BES2 ”RD03” USER TBL ”user1”.”FMB11” BES2 ”RD03” USER TBL ”user1”.”FMB12” BES2 ”RD03” USER TBL ”user1”.”FMB13” BES2 ”RD06” USER IDX ”user1”.”FMB11IDX” ”user1”.”FMB11”
11			○	−	”RD01” ”RD02” ”RD04” ”RD05” ”RD03” ”RD06”

（凡例）

○：指定あり

−：指定なし

(5) 注意事項

HiRDB の再起動を必要としない SDB データベースの定義変更でレコード型を追加し，その二次インデクスを格納するために追加した RD エリアや，分割格納条件の変更機能で追加した RD エリアは，ディクショナリ表中の SDB ディクショナリ情報が更新されると，関連 RD エリアとして表示されます（「5.7.5(2) SDB ディクショナリ情報を変更する」の操作が終わると，関連 RD エリアとして表示されます)。ただし，この時点では，追加した RD エリアは関連 RD エリアとして表示されますが，更新可能なオンライン再編成の対象にはできません。HiRDB の再起動を必要としない SDB データベースの定義変更の操作が完了するまで，更新可能なオンライン再編成の対象にはできません。

10.10　運用コマンド（更新可能なオンライン再編成で使用するコマンド）【4V FMB】

ここでは，更新可能なオンライン再編成で使用する運用コマンドについて説明します。

10.10.1　pdsdborcrt（追い付き反映キー対応表の操作）

(1)　機能

pdsdborcrt コマンドには次の機能があります。

- 追い付き反映キー対応表の作成
  追い付き反映キー対応表を作成します。また，追い付き反映キー対応表のインデクスを作成します。
- 追い付き反映キー対応表の削除
  追い付き反映キー対応表を削除します。また，追い付き反映キー対応表のインデクスも削除します。
- 追い付き反映キー対応表の初期化
  追い付き反映キー対応表の全行を削除します。

pdsdborcrt コマンドの使い方については，「5.13.3　更新可能なオンライン再編成の準備作業」，および「5.23　障害が発生したときの対処方法（更新可能なオンライン再編成の実行時）【4V FMB】」を参照してください。

(2)　実行者

HiRDB 管理者が実行できます。

(3)　形式

■追い付き反映キー対応表を作成する場合

```
pdsdborcrt -k crt -r RDエリア名〔-i RDエリア名〕
```

■追い付き反映キー対応表を削除する場合

```
pdsdborcrt -k del -s サーバ名
```

■追い付き反映キー対応表を初期化する場合

```
pdsdborcrt -k init -s サーバ名
```

(4) オプション

- -k

  追い付き反映キー対応表に対して実行する操作を指定します。

  crt：追い付き反映キー対応表を作成します。

  del：追い付き反映キー対応表を削除します。

  init：追い付き反映キー対応表を初期化します（追い付き反映キー対応表の全行を削除します）。

- -r RD エリア名

  ～<識別子>((1～30 バイト))

  追い付き反映キー対応表を格納する RD エリアの名称を指定します。追い付き反映キー対応表を格納する RD エリアは，BES ごとに作成する必要があります。

  なお，レプリカ RD エリアが定義されている RD エリアは指定できません。

- -i RD エリア名

  ～<識別子>((1～30 バイト))

  追い付き反映キー対応表のインデクスを格納する RD エリアの名称を指定します。追い付き反映キー対応表のインデクスを格納する RD エリアは，BES ごとに作成する必要があります。

  -i オプションの指定を省略した場合，-r オプションに指定した RD エリアにインデクスも格納されます。

  なお，レプリカ RD エリアが定義されている RD エリアは指定できません。

- -s サーバ名

  ～<識別子>((1～8 バイト))

  削除または初期化する追い付き反映キー対応表がある BES の名称を指定します。ここで指定した BES にある追い付き反映キー対応表が削除または初期化されます。

(5) 規則

- pdsdborcrt コマンドは，HiRDB が稼働中のときに実行できます。
- pdsdborcrt コマンドは，システムマネジャがあるサーバマシンで実行してください。
- -r および-i オプションには，リレーショナル DB を格納する指定をした RD エリアを指定してください。pdinit または pdmod コマンドで RD エリアを作成する際，create rdarea 文の data model オペランドに relational を指定した RD エリアを指定します。
- -r および-i オプションには，更新可能なオンライン再編成の対象 RD エリアがある BES の RD エリアを指定してください。ただし，更新可能なオンライン再編成の対象 RD エリアを指定することはできません。
- 追い付き反映キー対応表，およびそのインデクスは，1 つの BES に 1 つだけ作成できます。そのため，追い付き反映キー対応表がすでに存在する BES の RD エリアを，-r および-i オプションに指定した場合，pdsdborcrt コマンドがエラーになります。
- RD エリア名に英小文字または空白が含まれている場合は，RD エリア名を引用符（"）で囲んでください。囲まないと，英小文字は英大文字と見なされます。

（例）-r "RDarea01"

また，sh（Bourne シェル），csh（C シェル），または ksh（K シェル）の場合は，さらにアポストロフィ（’）で囲む必要があります。

（例）-r '"RDarea01"'

(6) 注意事項

- pdsdborcrt コマンドはバックグラウンドで実行できません。
- pdsdborcrt コマンドを複数同時に実行できません。
- pdsdborcrt コマンドと，ほかのコマンドおよびユティリティを同時に実行できません。
- SDB データベースがサーバ間横分割されている場合，BES ごとに pdsdborcrt コマンドを実行し，各 BES に追い付き反映キー対応表を作成してください。
- 追い付き反映キー対応表を初期化または削除する場合は，追い付き反映処理がすべて終了したあとに実行してください。追い付き反映処理が中断状態の RD エリアがある場合は，追い付き反映処理を再開始し，すべての追い付き反映処理が終了したあとに実行してください。追い付き反映処理が終了したかどうかは，更新可能なオンライン再編成の対象 RD エリアの状態を pddbls コマンドで確認するとわかります。オンライン再編成閉塞の RD エリアが 1 つもない場合は，追い付き反映処理が終了しています。
- 更新可能なオンライン再編成の実行中に，追い付き反映キー対応表を初期化または削除する必要が生じた場合は，pdsdborcrt コマンドの実行後に警告メッセージと確認メッセージが出力されます。次の流れで応答してください。

  1. 警告メッセージ（KFPT82005-W）が出力されます。KFPT82005-W メッセージに出力された BES 名を確認してください。
  2. 1.で確認した BES 内に，オンライン再編成閉塞の RD エリアがあるかどうかを pddbls コマンドで確認してください。
  3. 確認メッセージ（KFPT82006-Q）が出力されます。追い付き反映キー対応表を初期化または削除する場合は y を，初期化または削除をやめる場合は n を応答してください。

  注意事項

  次のどちらかの条件を満たす場合は，y を応答してください。

  - BES 内にオンライン再編成閉塞の RD エリアがない場合
  - pdsdblod コマンドによるデータロードで障害が発生し，追い付き反映キー対応表の初期化または削除が必要な場合

  上記の条件を満たさない場合は，必ず n を応答してください。上記の条件を満たさない場合に y を応答すると，SDB データベースの内容が不整合な状態になるおそれがあります。

(7) リターンコード

pdsdborcrt コマンドのリターンコードを次に示します。

- 0：pdsdborcrt コマンドが正常終了しました。
- 4：pdsdborcrt コマンドが正常終了しました。ただし，警告メッセージが出力されています。
- 8：pdsdborcrt コマンドが異常終了しました。

10.10.2 pdorbegin（オンライン再編成のデータベース静止化）

(1) 機能

指定したオリジナル RD エリアと，レプリカグループ内の指定した世代のレプリカ RD エリアをオンライン再編成閉塞にします。

レコード型名を指定した場合，そのレコード型の関連 RD エリアを処理対象とし，各オリジナル RD エリアと，レプリカグループ内の指定した世代のレプリカ RD エリアをオンライン再編成閉塞にします。関連 RD エリアについては，pdrdrefls コマンドで確認できます。

pdorbegin コマンドを実行してから pdorchg コマンドを実行するまでの間は，指定したオリジナル RD エリアがあるレプリカグループ内の RD エリアにアクセスするトランザクションは排他待ち状態になります。

オンライン再編成閉塞を解除する場合は-u オプションを指定します。

pdorbegin コマンドの使い方については，「5.13.4 更新可能なオンライン再編成の運用手順」を参照してください。

(2) 実行者

HiRDB 管理者が実行できます。

(3) 形式

```
pdorbegin {-r オリジナルRDエリア名〔, オリジナルRDエリア名〕…
          |-t 〔認可識別子.〕レコード型名 〔-s サーバ名〔, サーバ名〕…〕}
          〔-q 世代番号〕〔-w 排他待ち時間〕〔-u〕〔-W 実行監視時間〕
          〔-I {put | noput}〕
```

(4) オプション

-t 以外のオプションについては，マニュアル「HiRDB コマンドリファレンス」の「pdorbegin（オンライン再編成のデータベース静止化）」を参照してください。

形式にあるオプション以外のオプションを指定した場合，コマンドがエラーになるか，またはオプションの指定が無視されます。

- -t 〔認可識別子.〕レコード型名
  ～＜識別子＞((認可識別子：1～8 バイト，レコード型名：1～30 バイト))
  更新可能なオンライン再編成を実行するレコード型の名称を指定します。ここで指定したレコード型の関連 RD エリアのオリジナル RD エリアと，レプリカグループ内の指定した世代のレプリカ RD エリアをオンライン再編成閉塞にします。
  指定時の注意事項を次に示します。
  - 認可識別子およびレコード型名を引用符（"）で囲んだ場合は，英字の大文字と小文字が区別されますが，引用符（"）で囲まない場合はすべて大文字として扱われます。なお，sh（Bourne シェル），csh（C シェル），および ksh（K シェル）の場合は，さらにアポストロフィ（’）で囲む必要があります。
  - 認可識別子を省略した場合，pdorbegin コマンド実行時の環境変数 PDUSER の認可識別子が仮定されます。環境変数 PDUSER を設定していない場合，ログインウィンドウのユーザ名が仮定されます。
  - -t オプションを指定する場合，そのレコード型と，そのレコード型に関連するすべての RD エリアの総数が，4,096 個以下である必要があります。

このコマンドの規則および留意事項については，マニュアル「HiRDB コマンドリファレンス」の「pdorbegin（オンライン再編成のデータベース静止化）」を参照してください。

(5) 注意事項

pdorbegin コマンドを実行する前に，更新可能なオンライン再編成が実行できる状態かどうかを，次のコマンドで確認する必要があります。

- pdorcheck
- pdsdbarc -a

確認方法については，「5.13.3 更新可能なオンライン再編成の準備作業」を参照してください。

10.10.3 pdorcheck（オンライン再編成の適用条件チェック）

(1) 機能

更新可能なオンライン再編成が実行できる状態かどうかを確認します。

pdorcheck コマンドの使い方については，「5.13.3 更新可能なオンライン再編成の準備作業」を参照してください。

(2) 実行者

HiRDB 管理者が実行できます。

(3) 形式

```
pdorcheck {-rオリジナルRDエリア名〔，オリジナルRDエリア名〕… | -r ALL}
```

(4) オプション

- -r オリジナル RD エリア名〔，オリジナル RD エリア名〕…
  ～<識別子>((1～30 バイト))
  適用条件のチェックを実施する（更新可能なオンライン再編成の対象とする）オリジナル RD エリアの名称を指定します。

  注意事項

  更新可能なオンライン再編成の対象となるオリジナル RD エリアを漏れなく指定してください。RD エリアの指定が漏れた場合，適用条件のチェックが正しく実行されません。

- -r ALL
  すべてのオリジナル RD エリアに対して適用条件のチェックをする場合に指定します。

このコマンドの規則および留意事項については，マニュアル「HiRDB コマンドリファレンス」の「pdorcheck（オンライン再編成の適用条件チェック）」を参照してください。

(5) チェックの内容

pdorcheck コマンドを実行すると，次の表に示す項目がチェックされます。条件を満たしていない項目については，メッセージが出力されます。次の表に示す項目を 1 つでも満たさない場合，更新可能なオンライン再編成は実行できません。

表 10-12 pdorcheck コマンドで実施するチェック内容

項番	チェック項目	チェック内容	チェック内容を満たさない場合の対処
1	SDB データベース種別	指定した RD エリアに格納されている SDB データベースが，4V FMB の SDB データベースであることをチェックします。4V FMB の SDB データベース以外の SDB データベースが，RD エリアに格納されている場合，更新可能なオンライン再編成は実行できません。 また，インデクスを格納している RD エリアを指定した場合，更新可能なオンライン再編成の対象外のインデクスが，その RD エリア	更新可能なオンライン再編成以外の方法で，SDB データベースを再編成してください。

項番	チェック項目	チェック内容	チェック内容を満たさない場合の対処
		に格納されていると，更新可能なオンライン再編成は実行できません。	
2	RD エリアの種別	指定した RD エリアが，SDB データベースを格納する RD エリアであるかをチェックします。リレーショナル DB を格納する RD エリアが含まれている場合，更新可能なオンライン再編成は実行できません。	更新可能なオンライン再編成の対象 RD エリアから，リレーショナル DB を格納する RD エリアを除いてください。
3	追い付き反映キー対応表	指定した RD エリアがあるサーバに，追い付き反映キー対応表があることをチェックします。同じサーバにない場合，更新可能なオンライン再編成は実行できません。	更新可能なオンライン再編成の対象 RD エリアがあるサーバに，追い付き反映キー対応表を作成してください。 pddbst コマンドの表単位の状態解析で，次のことを確認できます。 • 追い付き反映キー対応表を格納している RD エリアの構成 • 追い付き反映キー対応表が作成されているかどうか 確認方法の詳細については，「5.13.6　追い付き反映キー対応表を格納している RD エリアを確認したい場合」を参照してください。
4		追い付き反映キー対応表が，作成直後であるか，または初期化済みであるかをチェックします。これ以外の場合は，更新可能なオンライン再編成は実行できません。	pdsdborcrt コマンドを実行して，追い付き反映キー対応表を作成するか，または追い付き反映キー対応表を初期化してください。 なお，前回の更新可能なオンライン再編成で使用した追い付き反映キー対応表を再利用する場合は，追い付き反映キー対応表を初期化する必要があります。
5	SDB ディレクトリ情報	指定した RD エリアの SDB データベース情報が，SDB ディレクトリ情報中にあることをチェックします。指定した RD エリアの SDB データベース情報が，SDB ディレクトリ情報中にない場合，更新可能なオンライン再編成は実行できません。	pdorcheck コマンドに指定した RD エリア名が正しいか確認してください。 また，HiRDB の再起動を必要としない SDB データベースの定義変更中に追加した RD エリアを，SDB データベースの定義変更が完了する前に指定していないか確認してください。詳細については，「5.13.2(6)　HiRDB の再起動を必要としない SDB データベースの定義追加，定義変更との関連」を参照してください。

10.10.4　pdorend（オンライン再編成の追い付き反映）

(1)　機能

更新可能なオンライン再編成の追い付き反映処理を実行します。レプリカ RD エリアの更新内容をオリジナル RD エリアに反映します。

pdorend コマンドの使い方については，「5.13.4　更新可能なオンライン再編成の運用手順」を参照してください。

(2)　実行者

HiRDB 管理者が実行できます。

(3)　形式

```
pdorend 〔-s サーバ名〔，サーバ名〕…〕
       {〔-n 最大リトライ回数〕
        〔-w 追い付き反映処理の最大待ち時間〕
        〔-t トランザクション処理の最大待ち時間〕
        〔-m pdorend反映プロセスの多重度〕
        〔-z〕
        〔-f 追い付き反映制御ファイル名〕
        〔-I {put｜noput}〕
       |〔-u〕}
```

(4)　オプション

ここでは，-m および-f オプションについてだけ説明します。そのほかのオプションについては，マニュアル「HiRDB コマンドリファレンス」の「pdorend（オンライン再編成の追い付き反映）」を参照してください。

形式にあるオプション以外のオプションを指定した場合，コマンドがエラーになるか，またはオプションの指定が無視されます。

- -m pdorend 反映プロセスの多重度

  ～<符号なし整数>((1～8))《2》

  レプリカ RD エリアに対する更新処理をオリジナル RD エリアに反映する際の，pdorend 反映プロセスの BES ごとの多重度を指定します。

  <指定値の目安>

  -m オプションの指定値の目安を次に示します。

  ```
  MIN {8 , MAX (↑SDBDefCnt÷8↑, ↑SDBKeyCnt÷SDBDefCnt↑) }
  ```

  SDBDefCnt：更新可能なオンライン再編成の対象 BES に存在する SDB データベースの数

  SDBKeyCnt：更新可能なオンライン再編成の実行時（pdorchg コマンド実行時～pdorend コマンド完了時の間）に，1 つの SDB データベースに対して更新するレコードの DBKEY 値の数（異なる DBKEY 値の数）

  <留意事項>

  pdorend コマンドを実行すると，次の計算式の数の pdorend 反映プロセスが起動されます。

  ```
  pdorend反映プロセス起動数＝M×S　（単位：プロセス数）
  ```

M：-m オプションの指定値

S：-s オプションの指定数（-s オプションを省略した場合，オンライン再編成閉塞の RD エリアがある全 BES 数）

なお，HiRDB/SD 全体で保証される pdorend 反映プロセス数は，あらかじめシステム定義の pd_max_reflect_process_count オペランドで指定しておく必要があります。

- **-f 追い付き反映制御ファイル名**

  ～<パス名>((255 文字以内))

  追い付き反映処理の制御文を記述したファイル（追い付き反映制御ファイル）を，絶対パス名で指定します。-f オプションを省略した場合，追い付き反映制御ファイルに記述できる制御文の省略時仮定値が仮定されます。なお，追い付き反映制御ファイル中にコメントは記述できません。

  追い付き反映制御ファイルに記述できる制御文を次に示します。

制御文	内容	指定できる数
reflection 文	追い付き反映処理の動作内容を指定します。指定形式については，「(5)　reflection 文の指定形式」を参照してください。	1
report 文	追い付き反映処理の実行結果を出力するファイルを指定します。指定形式については，「(6)　report 文の指定形式」を参照してください。	1

  追い付き反映制御ファイルの指定例を次に示します。

```
reflection dml_error=skip
report dml_skip_info=/home/user01/dmlskip_outputfile01
```

  ＜指定規則＞

  - reflection 文と report 文は両方とも必ず指定してください。
  - 各文は 1 行で記述してください。
  - 指定値を記述する際，＝の前後に空白を入れないでください。
  - 同じオペランドを複数指定しないでください。

(5)　reflection 文の指定形式

```
reflection 〔dml_error= {stop | skip} 〕
```

- dml_error= {stop | skip}

  追い付き反映処理で，HiRDB/SD が内部的に発行する，SDB データベースを操作する API の実行時にエラーが発生した場合の，HiRDB/SD の処理を指定します。

  stop：

  追い付き反映処理中に，HiRDB/SD が内部的に発行する API の実行時にエラーが発生した場合，追い付き反映処理をエラー終了します。このとき，追い付き反映処理に対してロールバックが実行され，SDB データベースは前回のコミット時点の状態になります。追い付き反映処理では，一定の API 処理件数ごとにコミットが実行されています。

skip：

追い付き反映処理中に，HiRDB/SD が内部的に発行する API で特定のエラーが発生した場合，その API を無視して追い付き反映処理を続行します。 スキップ対象の API の情報は，report 文の dml_skip_info オペランドで指定したファイルに出力されます。

注意事項

skip を指定した場合に，HiRDB/SD が内部的に発行する API でエラーが発生すると，更新可能なオンライン再編成が完了したあとのオリジナル RD エリアとレプリカ RD エリアの整合性が保証できなくなります。そのため，skip を指定する場合は十分に運用を検討してから指定するようにしてください。

skip を指定する場合のケースについては，「5.13.10　追い付き反映処理時に HiRDB/SD が内部的に発行する API のエラーをスキップしたい場合」を参照してください。

スキップ対象となった API を特定するために，スキップ情報中にカレント RD エリア（レプリカ RD エリア）のデータベースキー情報が出力されます。データベースキー情報からスキップしたレコードを特定し，SDB データベースへの反映が必要な場合は，SDB データベースを操作する API などを実行して SDB データベースに反映してください。

skip を指定した場合に，HiRDB/SD が内部的に発行する API でエラーが発生したときの HiRDB/SD の処理を次の表に示します。

表 10–13　HiRDB/SD が内部的に発行する API でエラーが発生したときの HiRDB/SD の処理

API のエラーが発生したレコード	API 種別	SQLCODE	HiRDB/SD の処理
ルートレコード	FETCH	100	対象レコードがないため，HiRDB/SD が内部的に発行する API がエラーになりますが，この API を無視して追い付き反映処理を続行します。
		上記以外	追い付き反映処理をエラー終了します。
上記以外			

注意事項

reflection 文には，dml_error オペランドだけが指定できます。マニュアル「HiRDB コマンドリファレンス」の「pdorend（オンライン再編成の追い付き反映）」の reflection 文で説明しているオペランドは指定できません。

(6)　report 文の指定形式

```
report 〔dml_skip_info=DMLスキップ情報出力ファイル名〕
```

- dml_skip_info=DML スキップ情報出力ファイル名
  ～<パス名>((235 文字以内))

  スキップ対象となった API（追い付き反映処理中に HiRDB/SD が内部的に発行する API）の情報を出力するファイルを絶対パス名で指定します。このオペランドは，reflection 文の dml_error オペランドに skip を指定した場合に指定してください。

  なお，実際に出力されるファイルの名称は，次のようになります。

  （例）dml_skip_info=/home/user01/dmlskip_outputfile01 と指定した場合

  ファイル名：dmlskip_outputfile01_bes1_25182

  bes1：スキップ対象となった API を実行しようとした BES 名

  25182：スキップ対象となった API を実行しようとしたプロセスのプロセス ID

  ＜規則＞

  - DML スキップ情報出力ファイルを出力するディレクトリは，すべてのサーバマシンで同じにしてください。出力するディレクトリがない場合は，pdorend コマンドがエラーになります。
  - DML スキップ情報の出力時にエラーが発生した場合，処理を中断してロールバックします（pdorend コマンドがエラーになります）。
  - 同じ名称の DML スキップ情報出力ファイルが存在する場合，存在するファイルにスキップ対象となった API の情報を追加出力します。
  - このオペランドを省略した場合，DML スキップ情報出力ファイルの出力先は，次の優先順位で決定されます（1 の優先順位がいちばん高くなります）。

    1．ユニット制御情報定義の pd_tmp_directory オペランドに指定したディレクトリ

    2．環境変数 TMPDIR※に指定したディレクトリ

    3．/tmp ディレクトリ

    ファイル名は，pdskipdml_BES 名_プロセス ID となります。

  注※

  init プロセス（OS のプロセス）から起動された HiRDB（pdprcd）に設定される TMPDIR 環境変数のことです。

注意事項

report 文には，dml_skip_info オペランドだけが指定できます。マニュアル「HiRDB コマンドリファレンス」の「pdorend（オンライン再編成の追い付き反映）」の report 文で説明しているオペランドは指定できません。

(7) DML スキップ情報出力ファイルの出力形式

DML スキップ情報出力ファイルの出力形式の例を次に示します。

```
2013/12/27 09:51:26  SVR(bes1) PID(5704)     ...1
**** SKIP DML INFORMATION ****               ...2
```

```
SCHEMA NAME : "FMB1"                          ...3
RECORD NAME : "RECB"                          ...4
REFLECTION KIND : STORE                       ...5
**** ERROR DML INFORMATION ****              ...6
RECORD NAME : "RECA"                          ...7
DML KIND : FETCH                              ...8
SQLCODE : 100                                 ...9
SQLWARN :                                     ...10
**** CURRENT DBKEY INFORMATION ****          ...11
      Address  Loc  +0+1+2+3 +4+5+6+7 +8+9+a+b +c+d+e+f 0123456789abcdef     ...12
      59bc3170 0000 30345f30 30303032 03000000          04_00002....
```

[説明]

API（追い付き反映処理中に HiRDB/SD が内部的に発行する API）のスキップが発生した数の分だけ，次の情報が表示されます。

1. API のスキップが発生した日時，BES 名，プロセス ID が表示されます。
2. スキップ対象となった API の情報を示すヘッダが表示されます。
3. SDB データベース名が表示されます。
4. 追い付き反映対象のレコード型名が表示されます。
5. 追い付き反映処理時に HiRDB/SD が内部的に発行した API の種別が表示されます。

   STORE：レコードの格納（STORE）の追い付き反映

   MODIFY：レコードの更新（MODIFY）の追い付き反映

   ERASE：レコードの削除（ERASE）の追い付き反映

   ROLLBACK STORE：レコードの格納（STORE）のロールバックに対する追い付き反映

   ROLLBACK MODIFY：レコードの更新（MODIFY）のロールバックに対する追い付き反映

   ROLLBACK ERASE(REC)：レコードの削除（ERASE）のロールバックに対する追い付き反映（レコードの回復）

   ROLLBACK ERASE(USER)：レコードの削除（ERASE）のロールバックに対する追い付き反映（USER ポインタの回復）
6. API のエラー情報を示すヘッダが表示されます。
7. API のエラーが発生したレコード型名が表示されます。

   追い付き反映対象とするレコードの位置づけ用の API などでエラーが発生した場合は，追い付き反映対象のレコードより上位階層のレコード型名が出力されます。
8. エラーが発生した API の種別が表示されます。

   FETCH：レコードの検索（追い付き反映を行うレコードに対する位置づけ用の API を意味します）

   MODIFY：レコードの更新

   STORE：レコードの格納

   ERASE：レコードの削除
9. API のエラーの SQLCODE が表示されます。
10. 警告情報が表示されます。

警告情報を次の形式で出力します。

”SQLWARNn”

n：0～F

警告情報が複数ある場合は，警告情報を空白で区切って出力します。

警告情報がない場合は，”SQLWARN ： ”が出力されます。

11. カレント RD エリア（レプリカ RD エリア）側のデータベースキーの情報を示すヘッダが表示されます。

12. 追い付き反映対象のレコードに対するデータベースキーの内容が出力されます。ダンプ形式（16 進数表示＋文字列表示）で出力されます。

(8) 注意事項

- pdorend コマンドの結果は，pdls -d org コマンド，またはリターンコードで確認できます。リターンコードを次に示します。

  0：正常終了

  4：警告終了※1

  8：異常終了

  12：異常終了（コマンドがエラーメッセージを表示できない事象が発生）※2

  注※1

  次のどちらかの場合にリターンコードが 4 になります。

  ・一部の BES だけで処理が成功した場合

  ・スキップ対象となった API（追い付き反映処理中に HiRDB/SD が内部的に発行する API）がある場合

  注※2

  ディクショナリサーバがあるサーバマシンの syslogfile に出力されたエラーメッセージを参照し，エラーの要因を取り除いたあとに，pdorend コマンドを再実行してください。syslogfile にエラーメッセージが出力されていない場合は，保守員に連絡してください。

- このコマンドの規則およびそのほかの注意事項については，マニュアル「HiRDB コマンドリファレンス」の「pdorend（オンライン再編成の追い付き反映)」を参照してください。

11 HiRDB/SD 定義ユティリティ（pdsdbdef）

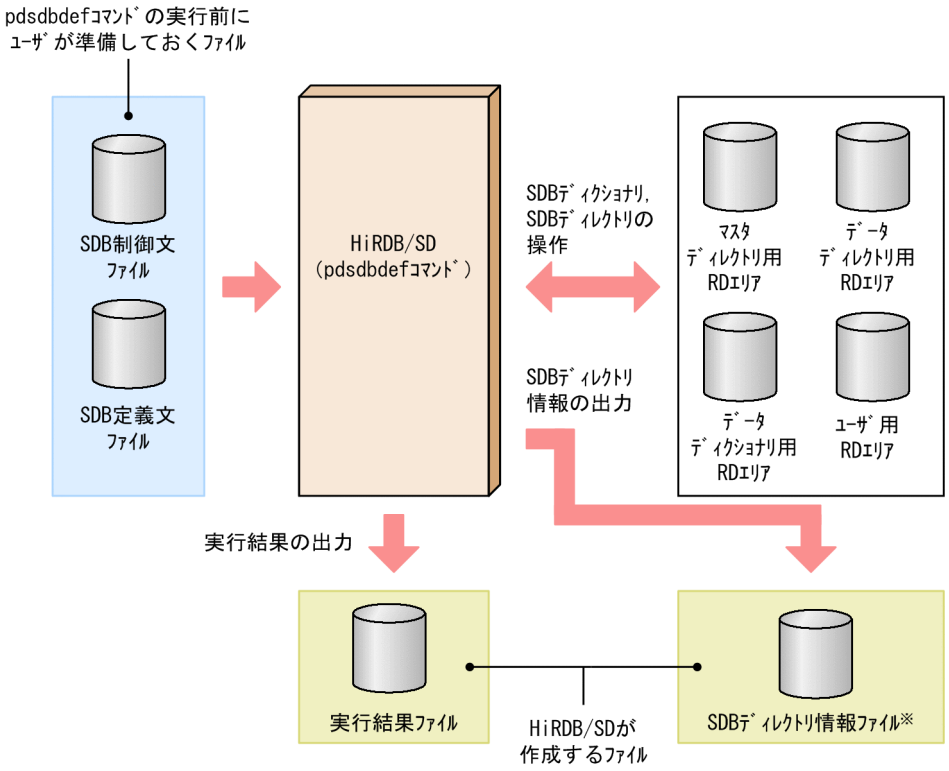
この章では，SDB データベース定義をする HiRDB/SD 定義ユティリティの機能と使い方について説明します。

なお，この章では，HiRDB/SD 定義ユティリティを pdsdbdef コマンドと表記しています。

11.1　機能

pdsdbdef コマンドを使用して SDB データベースを定義します。pdsdbdef コマンドの概要を次の図に示します。

図 11-1　pdsdbdef コマンドの概要



注※

SDB ディレクトリ情報のファイル出力後，作成した SDB ディレクトリ情報ファイルを全ユニットに配布できます。

pdsdbdef コマンドの制御情報（SDB 制御文）を SDB 制御文ファイルに記述します。また，pdsdbdef コマンドで操作する SDB データベースの定義情報（SDB 定義文）を SDB 定義文ファイルに記述します。

pdsdbdef コマンドで操作する SDB データベースの定義情報には，SDB ディクショナリ情報と SDB ディレクトリ情報があります。

SDB ディクショナリ情報と SDB ディレクトリ情報の内容を次の表に示します。

表 11-1　SDB ディクショナリ情報と SDB ディレクトリ情報の内容

定義情報	内容
SDB ディクショナリ情報	ユーザが定義した SDB データベースの定義情報を，ユーザが記述した形式に近い状態で保持したものです。
SDB ディレクトリ情報	SDB データベースの SDB ディクショナリ情報を，HiRDB/SD が解釈しやすい形式に変換したものです。

pdsdbdef コマンドの機能を次の表に示します。

表 11-2　pdsdbdef コマンドの機能

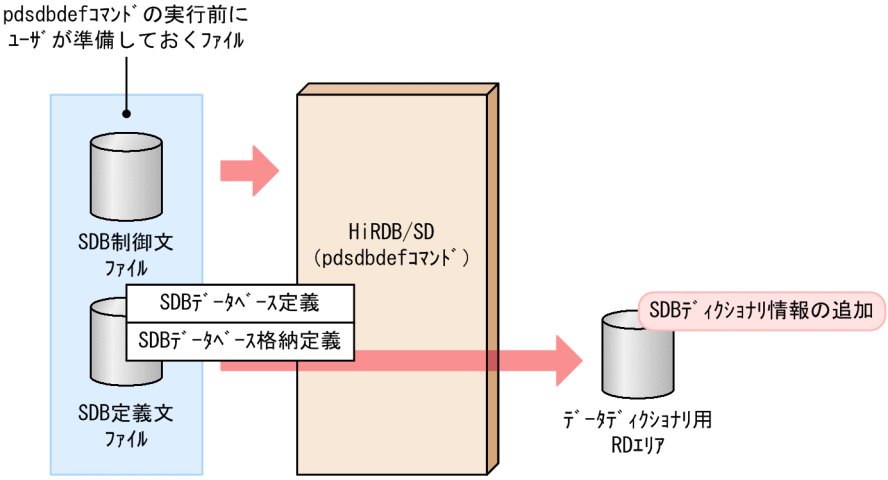
種別	機能	内容
ディクショナリ操作機能	SDB ディクショナリ情報の追加	*ENTRY DICTIONARY 文に指定した SDB データベース定義および SDB データベース格納定義から SDB ディクショナリ情報を作成し，ディクショナリ表に追加します。
	SDB ディクショナリ情報の変更	*ALTER DICTIONARY 文に指定した SDB データベース定義および SDB データベース格納定義から SDB ディクショナリ情報を作成し，ディクショナリ表に登録されている SDB ディクショナリ情報を変更します。
	SDB ディクショナリ情報の削除	*DELETE DICTIONARY 文に指定した SDB データベースの SDB ディクショナリ情報をディクショナリ表から削除します。
	データベース定義のチェック	*CHECK DICTIONARY 文に指定した SDB データベースの SDB データベース定義および SDB データベース格納定義の整合性をチェックします。
ディレクトリ操作機能	SDB ディレクトリ情報の追加	*ENTRY DIRECTORY 文に指定した SDB データベースの SDB ディレクトリ情報を作成し，ディクショナリ表に追加します。
	SDB ディレクトリ情報の変更	*ALTER DIRECTORY 文に指定した SDB データベースの SDB ディレクトリ情報を作成し，ディクショナリ表に登録されている SDB ディレクトリ情報を変更します。
	SDB ディレクトリ情報の削除	*DELETE DIRECTORY 文に指定した SDB データベースの SDB ディレクトリ情報をディクショナリ表から削除します。
	SDB ディレクトリ情報のファイル出力および配布	ディクショナリ表に登録されている SDB ディレクトリ情報を SDB ディレクトリ情報ファイルに出力し，全ユニットに配布します。

11.1.1　ディクショナリ操作機能

(1)　SDB ディクショナリ情報の追加

*ENTRY DICTIONARY 文に指定した SDB データベース定義および SDB データベース格納定義から SDB ディクショナリ情報を作成し，ディクショナリ表に追加します。

図 11-2　SDB ディクショナリ情報の追加



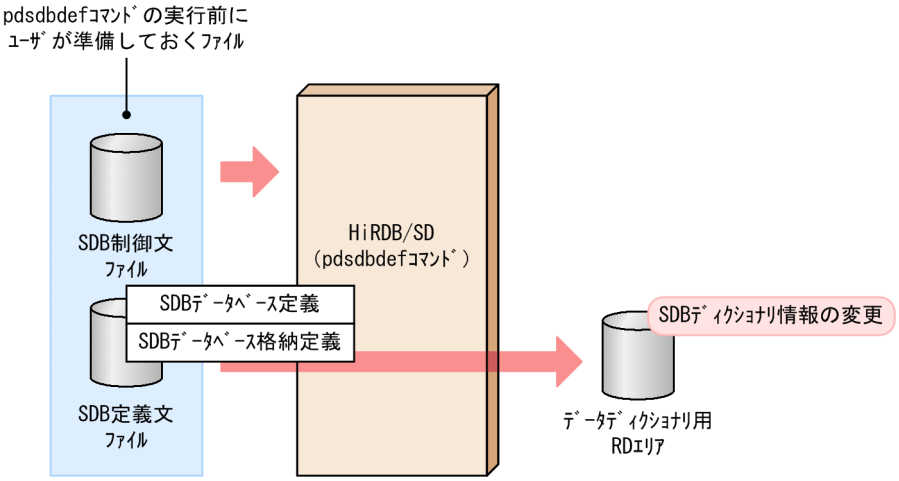
SDB データベース定義，SDB データベース格納定義は，SDB 定義文ファイルに記述します。SDB データベース定義および SDB データベース格納定義については，次の個所を参照してください。

- 4V FMB の SDB データベースを定義する場合
  「11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】」
- 4V AFM の SDB データベースを定義する場合
  「11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】」
- SD FMB の SDB データベースを定義する場合
  「11.7　SDB データベース定義，SDB データベース格納定義【SD FMB】」

(2)　SDB ディクショナリ情報の変更【4V FMB，4V AFM】

*ALTER DICTIONARY 文に指定した SDB データベース定義および SDB データベース格納定義から SDB ディクショナリ情報を作成し，ディクショナリ表に登録されている SDB ディクショナリ情報を変更します。

図 11-3　SDB ディクショナリ情報の変更



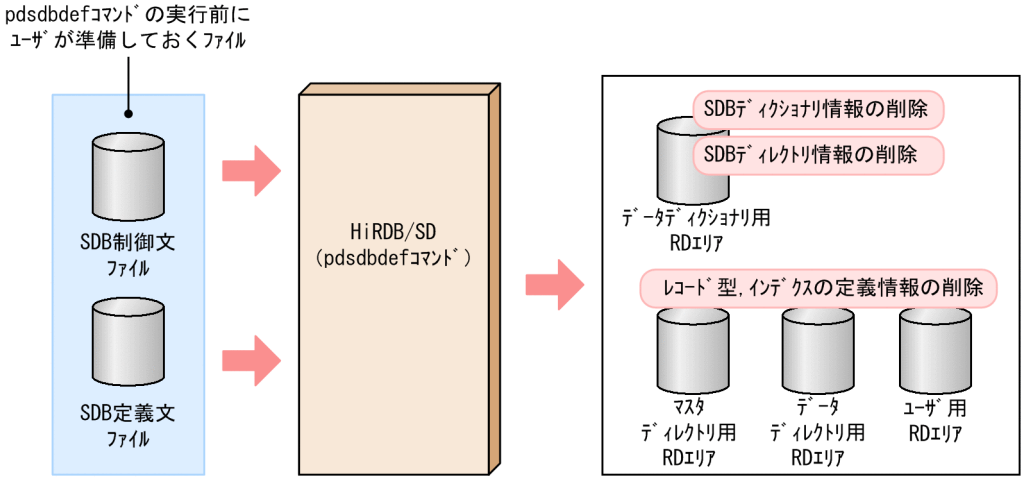
SDB データベース定義，SDB データベース格納定義は，SDB 定義文ファイルに記述します。SDB データベース定義および SDB データベース格納定義については，次の個所を参照してください。

- 4V FMB の SDB データベースの定義を変更する場合
  「11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】」
- 4V AFM の SDB データベースの定義を変更する場合
  「11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】」

(3)　SDB ディクショナリ情報の削除

*DELETE DICTIONARY 文に指定した SDB データベースの SDB ディクショナリ情報（SDB データベース定義および SDB データベース格納定義）をディクショナリ表から削除します。また，SDB ディレクトリ情報をディクショナリ表から削除します。

図 11-4　SDB ディクショナリ情報の削除



同時に，次に示す RD エリアに登録されている SDB データベースのレコード型およびインデクスの定義情報を削除します。

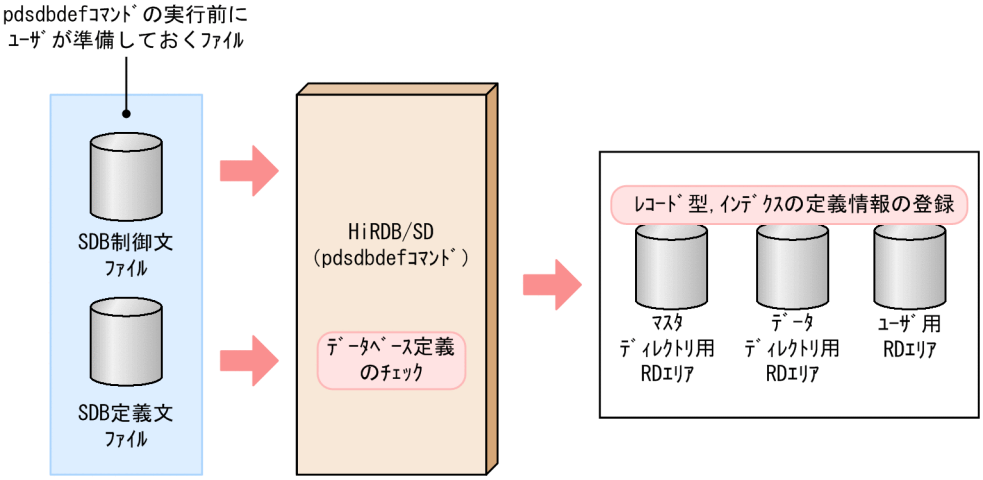
- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- ユーザ用 RD エリア

(4)　データベース定義のチェック

*CHECK DICTIONARY 文に指定した SDB データベースの SDB データベース定義および SDB データベース格納定義の整合性をチェックします。

SDB データベース格納定義は，RD エリアの定義，および SDB データベース定義と関連しています。そのため，SDB データベース定義，SDB データベース格納定義，および RD エリアの定義の間で指定内容に矛盾がないかどうかをチェックします。

図 11-5　データベース定義のチェック



チェックの結果，指定内容に矛盾がない場合は，次に示す RD エリアに SDB データベースのレコード型およびインデクスの定義情報を登録します。

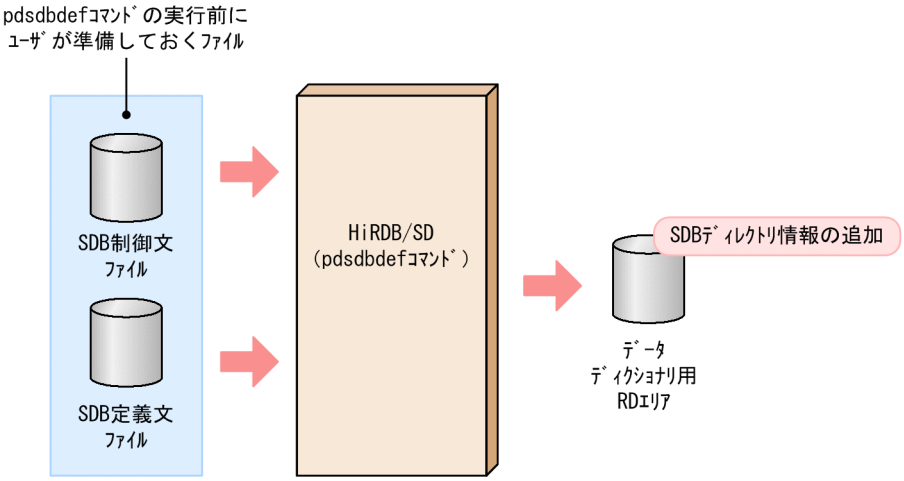
- マスタディレクトリ用 RD エリア
- データディレクトリ用 RD エリア
- ユーザ用 RD エリア

11.1.2　ディレクトリ操作機能

(1)　SDB ディレクトリ情報の追加

*ENTRY DIRECTORY 文に指定した SDB データベースの SDB ディレクトリ情報を SDB ディクショナリ情報から作成し，ディクショナリ表に追加します。

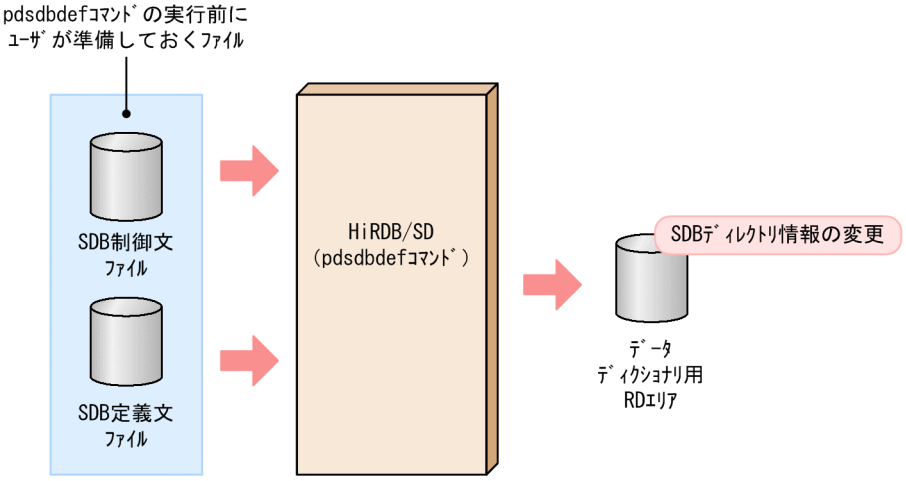
図 11-6　SDB ディレクトリ情報の追加



(2)　SDB ディレクトリ情報の変更【4V FMB，4V AFM】

*ALTER DIRECTORY 文に指定した SDB データベースの SDB ディレクトリ情報を SDB ディクショナリ情報から作成し，ディクショナリ表に登録されている SDB ディレクトリ情報を変更します。

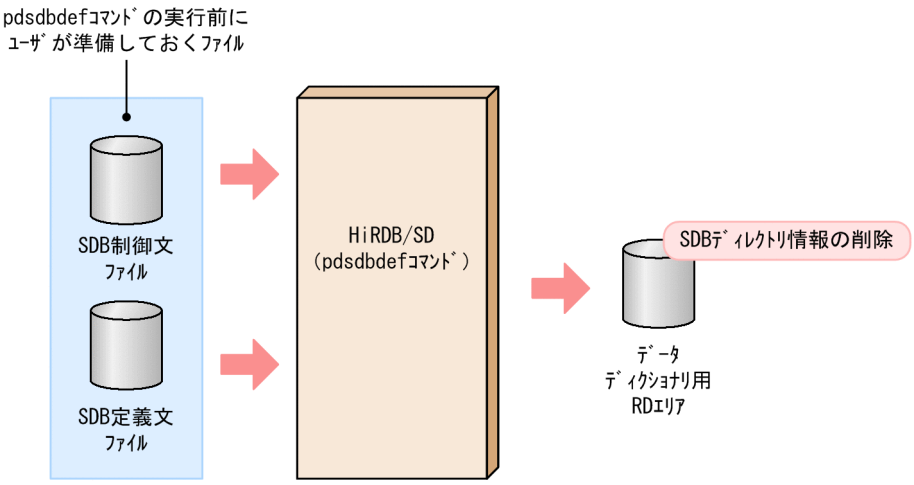
図 11-7　SDB ディレクトリ情報の変更



(3)　SDB ディレクトリ情報の削除

*DELETE DIRECTORY 文に指定した SDB データベースの SDB ディレクトリ情報をディクショナリ表から削除します。

図 11-8　SDB ディレクトリ情報の削除



(4)　SDB ディレクトリ情報のファイル出力および配布

ディクショナリ表に登録されている SDB ディレクトリ情報を SDB ディレクトリ情報ファイルに出力します。SDB ディレクトリ情報ファイルの出力先は，SDB 制御文の dirinf 文で指定します。SDB ディレクトリ情報ファイルのファイル名は，pdsdbdir 固定となります。

また，SDB ディレクトリ情報ファイルの作成と同時に，SDB ディレクトリ情報ファイルを全ユニットに配布します。配布先は，ユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下になります。

高速系切り替え機能を使用している場合は，待機系ユニットも配布対象になります。

注意事項

ディザスタリカバリシステムのリモートサイトは配布対象外です。

11.2　コマンドの形式

pdsdbdef コマンドの形式について説明します。

```
pdsdbdef△〔-i〕△SDB制御文ファイル名
```

注

△は，空白またはタブを区切り文字として，1 つ以上指定できることを示します。

-i：

次に示す SDB 定義文を指定する場合は，-i オプションを指定してください。-i オプションを指定しないと，pdsdbdef コマンドの実行がエラーになります。

- *DELETE DICTIONARY 文（SDB ディクショナリ情報の削除）
- *DELETE DIRECTORY 文（SDB ディレクトリ情報の削除）
- *ALTER DICTIONARY 文（格納 RD エリアを変更する SDB ディクショナリ情報の変更）
- *ALTER DIRECTORY 文（格納 RD エリアを変更する SDB ディレクトリ情報の変更）

注意事項

-i オプションを指定して pdsdbdef コマンドを実行した場合，HiRDB の再起動が必要になります。

SDB ディクショナリ情報の変更および SDB ディレクトリ情報の変更で，-i オプションの指定が必要となるのは，格納 RD エリアの分割，削除，または統合を行う場合です。格納 RD エリアの追加や，最下位のレコード型とその親子集合の追加をする場合は，-i オプションを指定する必要はありません（HiRDB の再起動の必要はありません）。

なお，-i オプションは，SDB 制御文ファイル名の後ろに指定することもできます。

SDB 制御文ファイル名：

～<パス名>

pdsdbdef コマンドの制御情報を記述したファイル（SDB 制御文ファイル）の名称を絶対パスまたは相対パスで指定します。

SDB 制御文ファイルに記述できる制御文（SDB 制御文）については，「11.3　SDB 制御文」を参照してください。

11.3 SDB 制御文

SDB 制御文には，pdsdbdef コマンドの制御情報を指定します。

SDB 制御文ファイルに記述できる制御文（SDB 制御文）を次の表に示します。

表 11–3 SDB 制御文ファイルに記述できる制御文

SDB 制御文	内容	指定できる制御文の最大数
source 文※1	SDB 定義文ファイルの情報を指定します。	1
environment 文※1，※2	pdsdbdef コマンドの動作環境に関する指定をします。	1
dirinf 文※3	SDB ディレクトリ情報ファイルを作成し，ユニットに配布する場合に指定します。	1

注※1

SDB ディレクトリ情報ファイルの作成だけを実行する場合は，source 文および environment 文は省略してください。

注※2

source 文の指定がある場合に有効となります。

注※3

source 文と同時に指定した場合，すべてのディクショナリ操作の処理が正常終了した場合に有効となります。

11.3.1 SDB 制御文の記述規則

SDB 制御文の記述規則を次に示します。

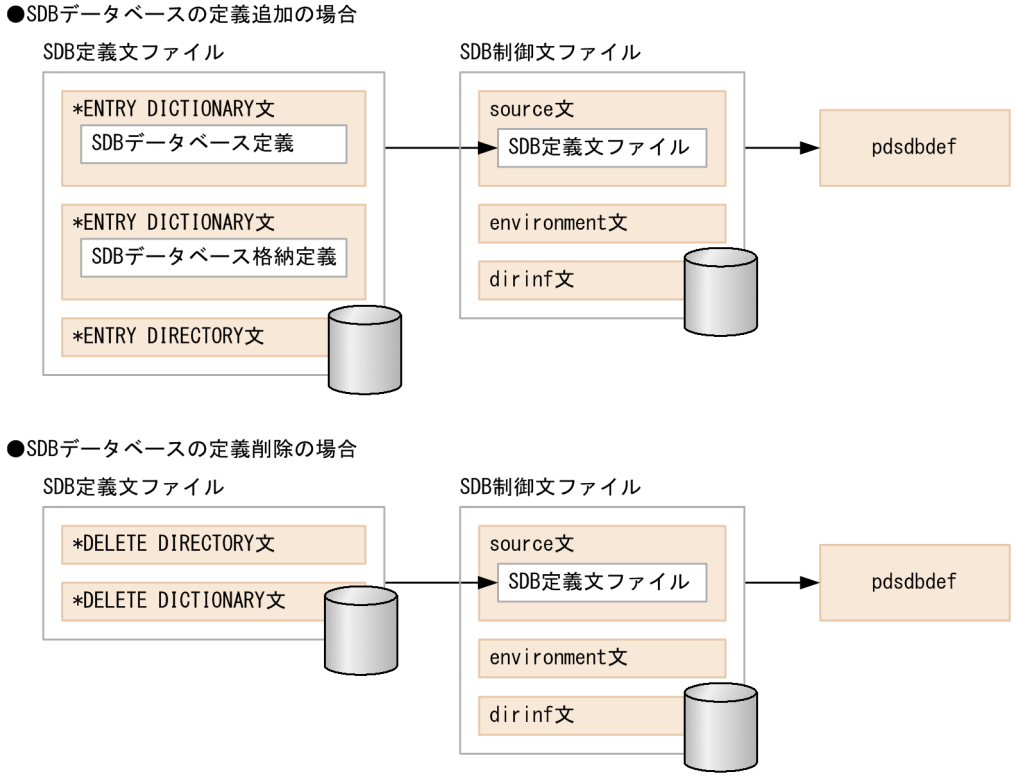
- パス名は引用符（"）で囲まないでください。
- パス名にスペースおよびタブを含むファイルおよびディレクトリは指定できません。
- 改行には，NL（X'0a'）または CR ＋ NL（X'0d'＋ X'0a'）を使用してください。
- 1 行の長さは 1,023 バイト以内（改行は含まない）で記述してください。
- SDB 制御文の指定順序は任意です。
- SDB 制御文ファイル中にコメントは記述できません。

11.3.2　pdsdbdef コマンドに指定する定義ファイルの関係

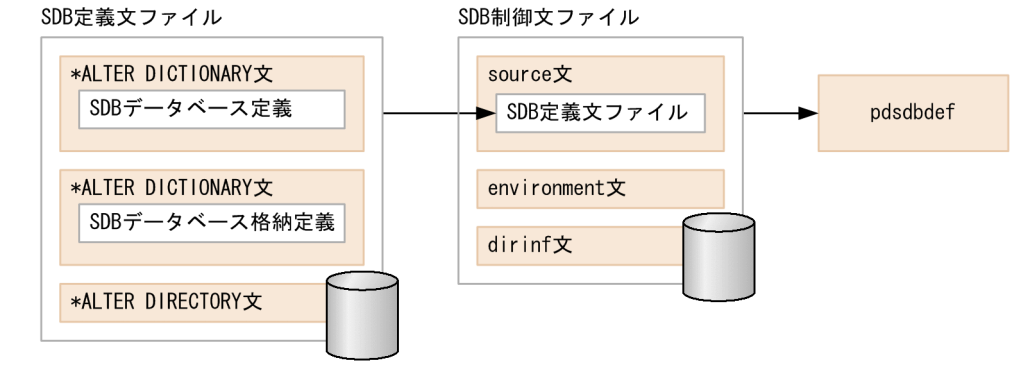
pdsdbdef コマンドでは，コマンドを制御する情報を SDB 制御文ファイルに記述し，さらに SDB 制御文ファイル中の source 文で SDB データベースの定義情報を記述したファイル（SDB 定義文ファイル）を指定します。

pdsdbdef コマンドに指定する定義ファイルの関係を次の図に示します。

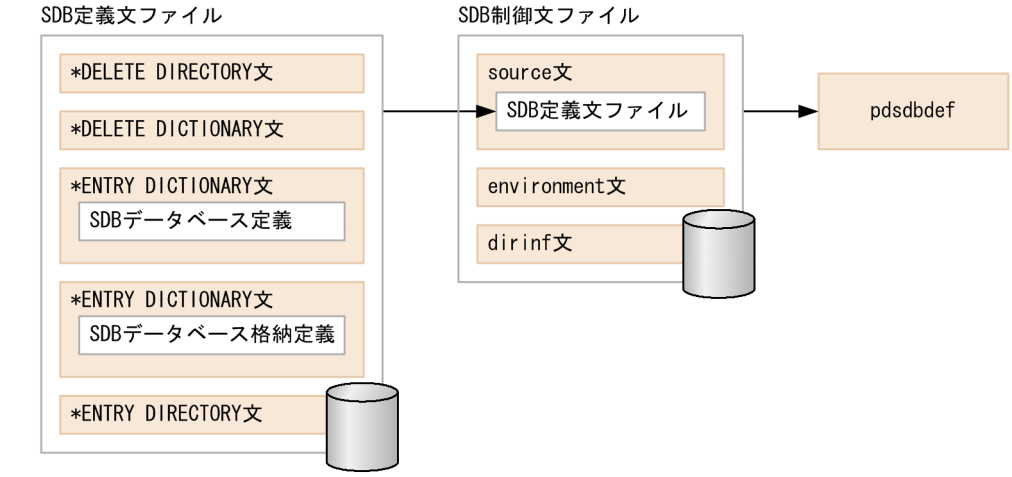
図 11-9　pdsdbdef コマンドに指定する定義ファイルの関係



●SDBデータベースの定義変更の場合



●SDBデータベースの定義変更(定義削除後の定義追加)の場合



注

- *ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文を別々に実行する場合は，*ENTRY DICTIONARY 文と一緒に*CHECK DICTIONARY 文を指定してください。
- *ALTER DICTIONARY 文と*ALTER DIRECTORY 文を別々に実行する場合は，*ALTER DICTIONARY 文と一緒に*CHECK DICTIONARY 文を指定してください。

11.3.3 source 文（SDB 定義文ファイルの指定）

source 文には SDB 定義文ファイルの情報を指定します。

(1) 適用条件

SDB 定義文を指定する場合は，source 文を必ず指定します。

SDB ディレクトリ情報ファイルの作成または配布だけをする場合（dirinf 文だけを指定する場合）は，source 文を省略できます。

(2) 形式

```
source△SDB定義文ファイル名
```

△は，空白またはタブを区切り文字として，1 つ以上指定できることを示しています。

(3) 説明

●SDB 定義文ファイル名

～<パス名>

SDB 定義文ファイルの名称を絶対パス名で指定します。

SDB 定義文ファイルに，SDB 定義文を記述します。SDB 定義文については「11.4 SDB 定義文」を参照してください。

11.3.4 environment 文（コマンド実行時の動作環境の指定）

environment 文には，pdsdbdef コマンド実行時に必要な動作環境を指定します。

(1) 適用条件

次のどちらかの場合に，environment 文を指定します。

- SDB 定義文ファイルの実行結果の出力ファイルを指定する場合

  システム定義の pd_tmp_directory オペランド，および環境変数 TMPDIR を指定していない場合，作業用一時ファイル格納ディレクトリ（$PDDIR/tmp ディレクトリ）の容量不足を発生させないために，environment 文の msglog オペランドをできるだけ指定してください。

- SDB 定義文ファイルの実行時間を監視する場合※

  注※ dirinf 文の処理は監視対象ではありません。

(2) 形式

```
environment△〔msglog=実行結果ファイル名〕△〔exectime=実行監視時間〕
```

△は，空白またはタブを区切り文字として，1 つ以上指定できることを示しています。

なお，msglog，exectime と=の間に空白やタブなどの区切り文字を指定することもできます。

(3) 説明

●msglog=実行結果ファイル名

～<パス名>

SDB 定義文ファイルの実行結果を出力する，実行結果ファイルの名称を絶対パス名で指定します。実行結果の出力形式については，「11.10　実行結果ファイルの出力形式」を参照してください。

- source 文に指定した SDB 定義文ファイルと同じファイル名は指定できません。
- SDB ディレクトリ情報ファイルの格納先ディレクトリにあるファイルと同じファイル名を指定すると，SDB ディレクトリ情報ファイルが上書きされることがあるので，指定しないでください。
- 指定したファイル名が存在しない場合，指定したファイル名でファイルが作成されます。ただし，ディレクトリが存在しない場合はエラーとなります。

注意事項

指定したファイル名が存在する場合，そのファイルは上書きされます。pdsdbdef コマンドが正常終了しなかった場合は，原因を調査するため，必ず実行結果ファイルを退避してください。

なお，このオペランドを省略すると，次に示す形式でファイルが作成されます。ただし，絶対パス名が 1,024 バイト以上となる場合はエラーとなります。

```
/ディレクトリ名/SDBDEF-xxxxxxxxx
```

ディレクトリ名：

pdsdbdef コマンド実行時，SDB 制御文ファイルでファイル出力先を指定しない場合，次の表に示すディレクトリにファイルを出力します。

表 11-4　pdsdbdef コマンドがファイルを出力するディレクトリ

指定したディレクトリのアクセス可否	msglog の指定			
	あり	なし		
		システム定義の pd_tmp_directory オペランドの指定		
		あり	なし	
			環境変数 TMPDIR の指定	
			あり	なし
アクセス可	msglog に指定したファイル	pd_tmp_directory に指定したディレクトリ	TMPDIR に指定したディレクトリ※2	$PDDIR/tmp ディレクトリ
アクセス不可	エラー	$PDDIR/tmp ディレクトリ※1		エラー

注※1

KFPS04603-W が出力されて，$PDDIR/tmp ディレクトリが仮定されます。処理は続行されます（ただし，KFPS04643-I は出力されません）。

注※2

環境変数 TMPDIR に指定したディレクトリ＋/SDBDEF-xxxxxxxxx の絶対パス名が 1,024 バイト以上の場合はエラーとなります。

SDBDEF-：

実行結果ファイルのプリフィクス

xxxxxxxxx：

ファイル作成時間とプロセス ID を文字列に変換した値

8 桁の 16 進数表記の現在日時＋ 10 進数表記のプロセス ID（最大 10 桁）です。

●exectime=実行監視時間

～＜符号なし整数＞((0～3,600))≪0≫

pdsdbdef コマンドの実行時間を監視する場合に，その監視時間を分単位で指定します。

0 を指定した場合は，実行時間は監視しません。

指定した監視時間内に pdsdbdef コマンドの処理が終了しなかった場合，pdsdbdef コマンドは強制終了します。その際，タイムアウト発生と強制終了のメッセージが出力され，次に示す原因調査用の障害情報が取得されます。

取得される障害情報

- 取得される障害情報の内容
  pdls コマンド（-d lck 指定）の表示結果
  pdls コマンド（-d rpc -a 指定）の表示結果
- 取得される障害情報の格納先
  ディレクトリ：$PDDIR/tmp
  作成ファイル名：pdsdbdefYYYYMMDDHHMMSS プロセス ID.txt

このオペランドの値は，システム定義の pd_cmd_exec_time オペランドの指定値よりも優先されます。

exectime オペランドとシステム定義の pd_cmd_exec_time オペランドとの関係を次の表に示します。

表 11-5　exectime オペランドとシステム定義の pd_cmd_exec_time オペランドの関係

exectime オペランドの指定値	システム定義の pd_cmd_exec_time オペランドの指定値	
	省略または 0	指定値 A
省略	－	指定値 A で監視
0	－	－
指定値 B	指定値 B で監視	指定値 B で監視

（凡例）

－：実行時間を監視しません。

pd_cmd_exec_time オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

- 実行時間監視（exectime オペランド）の適用基準
  監視時間を設定しておくことで，pdsdbdef コマンド実行中に何らかの障害（通信障害（一時的な障害を含む）やディスク障害など）が発生して pdsdbdef コマンドが無応答状態になった場合，pdsdbdef コマンドが異常終了します。このため，早期に障害を検知でき，回復処理に着手できます。

- 実行時間監視（exectime オペランド）の指定値の目安

  SDB ディクショナリ情報や SDB ディレクトリ情報の処理時間を基準に余裕を持った値を指定してください。例えば，7，8 分程度で終了する pdsdbdef コマンドの実行時間を監視する場合は，exectime=10 と指定しないで，exectime=20 と指定するなどしてください。

11.3.5　dirinf 文（SDB ディレクトリ情報ファイルの作成および配布）

dirinf 文には，SDB ディレクトリ情報ファイルの作成および配布についての情報を指定します。

ポイント

- 作成した SDB ディレクトリ情報ファイルは，全ユニットに配布する必要があります。
- 配布先は，ユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下となります。
- SDB ディレクトリ情報ファイルの作成と配布を 1 回の pdsdbdef コマンドで実行できます。また，SDB ディレクトリ情報ファイルの作成だけを実行することもできますし，配布だけを実行することもできます。

全ユニットに配布した SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報が，HiRDB の開始時に共用メモリ上の常用常駐領域に格納されます。これによって，SDB データベースにアクセスできるようになります。また，HiRDB の稼働中に，pdsdbarc コマンドで SDB ディレクトリ情報を更新することができます。pdsdbarc -e コマンドで SDB ディレクトリ情報を事前常駐領域に格納してから，pdsdbarc -w -q コマンドで事前常駐領域を常用常駐領域に切り替えることで SDB ディレクトリ情報を更新します。

(1)　適用条件

次の場合に dirinf 文を指定します。

- SDB ディレクトリ情報ファイルを作成する場合
- SDB ディレクトリ情報ファイルを各ユニットに配布する場合

なお，上記の 2 つの操作を同時に実行することもできます。

(2)　形式

```
dirinf△SDBディレクトリ情報ファイルの格納ディレクトリ
       [generate= {yes | no} ]
       [transfer= {yes | no}
         [checkmode= {match | range} ]
         [unit=ユニット名 [,ユニット名] …]
```

△は，空白またはタブを区切り文字として，1 つ以上指定する必要があることを意味しています。

(3) 説明

●SDB ディレクトリ情報ファイルの格納ディレクトリ

～<パス名>

SDB ディレクトリ情報ファイルの格納ディレクトリを指定します。ファイル名を含まない絶対パスで指定してください。

SDB ディレクトリ情報ファイルを作成する場合は，ここで指定したディレクトリ下に SDB ディレクトリ情報ファイルが出力されます。

SDB ディレクトリ情報ファイルを配布する場合は，ここで指定したディレクトリ下の SDB ディレクトリ情報ファイルが配布されます。

≪留意事項≫

- SDB ディレクトリ情報ファイルのファイル名は，「pdsdbdir」固定です。ファイル名は変更しないでください。
- 相対パスで指定した場合や，指定したパスが存在しない場合は，pdsdbdef コマンドの実行がエラーになります。
- SDB ディレクトリ情報ファイルの容量によっては，SDB ディレクトリ情報ファイルの格納ディレクトリを圧迫するおそれがあります。そのため，HiRDB 運用ディレクトリ以外のディレクトリを指定することを推奨します。

●generate= {yes | no}

SDB ディレクトリ情報ファイルを作成するかどうかを指定します。

yes：SDB ディレクトリ情報ファイルを作成します。

no：SDB ディレクトリ情報ファイルを作成しません。

yes を指定するか，または指定を省略した場合，SDB ディレクトリ情報ファイルの格納ディレクトリ下に SDB ディレクトリ情報ファイルが出力されます。

≪留意事項≫

- SDB ディレクトリ情報ファイルの格納ディレクトリ下に SDB ディレクトリ情報ファイルが存在する場合，SDB ディレクトリ情報ファイルは上書き更新されます。
- *ENTRY DIRECTORY 文で 1 つも SDB ディレクトリ情報を登録していない場合に，SDB ディレクトリ情報ファイルを作成しても，管理情報だけが格納された SDB ディレクトリ情報ファイルが作成されるため，SDB ディレクトリ情報ファイルを作成しても意味がありません（その SDB ディレクトリ情報ファイルは使用できません）。
- SDB ディレクトリ情報ファイルの格納ディレクトリ下に SDB ディレクトリ情報ファイルが存在するときでも，SDB ディレクトリ情報ファイルの作成中にエラーが発生した場合は，SDB ディレクトリ情報ファイルは削除されます。

●transfer= {yes | no}

SDB ディレクトリ情報ファイルを配布するかどうかを指定します。

yes：SDB ディレクトリ情報ファイルを配布します。

no：SDB ディレクトリ情報ファイルを配布しません。

yes を指定した場合，SDB ディレクトリ情報ファイルの格納ディレクトリ下の SDB ディレクトリ情報ファイルを，全ユニットに配布します。特定のユニットに対してだけ SDB ディレクトリ情報ファイルを配布したい場合は，unit オペランドを指定してください。

なお，高速系切り替え機能を使用している場合は，待機系ユニットも配布対象になります。

配布先は，ユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ下となります。

注意事項

ディザスタリカバリシステムのリモートサイトは配布対象外です。リモートサイトに SDB ディレクトリ情報ファイルを配布する方法については，「8.3.1(2) 「6. SDB ディレクトリ情報ファイルをリモートサイトに転送する」の作業」を参照してください。

●checkmode= {match | range}

SDB ディレクトリ情報ファイルを配布する場合（transfer = yes を指定した場合）にこのオペランドを指定します。

配布する SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報の最終更新日時のチェックレベルを指定します。SDB ディレクトリ情報ファイルを配布する前に，次の 1.と 2.の最終更新日時を比較します。

1. 配布する SDB ディレクトリ情報ファイル中の SDB ディレクトリ情報の最終更新日時
2. ディクショナリ表中の SDB ディレクトリ情報の最終更新日時

match：

1.と 2.の最終更新日時を比較し，最終更新日時が同じ場合に限り，SDB ディレクトリ情報ファイルを配布します。

range：

1.と 2.の最終更新日時を比較し，最終更新日時が同じ場合に SDB ディレクトリ情報ファイルを配布します。

また，1.の最終更新日時が古い場合でも，SDB ディレクトリ情報ファイルを配布します。

≪指定値の目安≫

- 通常は，match を指定してください。
- 障害発生時，一時的に旧 SDB ディレクトリ情報ファイルを使用するために配布する場合に range を指定してください。

≪留意事項≫

- システム共通定義の pd_structured_advance_resident オペランドに nouse を指定するか，または pd_structured_advance_resident オペランドを省略した場合，checkmode オペランドに range を指定することはできません。

- 配布する SDB ディレクトリ情報ファイル中の SDB 定義文の最終更新日時と，ディクショナリ表中の SDB 定義文の最終更新日時が異なる場合は，1.と 2.の最終更新日時が同じであっても，SDB ディレクトリ情報ファイルを配布しません。

●unit=ユニット名［,ユニット名］…

特定のユニットに対してだけ SDB ディレクトリ情報ファイルを配布したい場合に，このオペランドを指定します。このオペランドを省略した場合は，全ユニットに SDB ディレクトリ情報ファイルが配布されます。

例えば，一部のユニットに対して SDB ディレクトリ情報ファイルが配布されなかった場合に，そのユニットに対して SDB ディレクトリ情報ファイルを再配布するときにこのオペランドを指定してください。

≪留意事項≫

- このオペランドは，SDB ディレクトリ情報ファイルの配布だけをする場合に限り指定できます。このオペランドを指定する場合は，次のオペランドが指定されている必要があります。

  ```
  generate = no
  ```

  ```
  transfer = yes
  ```

- ユニット名は，最大 64 個指定できます。

(4) 指定例

dirinf 文の指定例を次に示します。

例 1

SDB ディレクトリ情報ファイルを作成して，全ユニットに配布します。

```
dirinf /usr/sdb_dir
       generate=yes
       transfer=yes
       checkmode=match
```

例 2

作成済みの SDB ディレクトリ情報ファイルを全ユニットに配布します。

```
dirinf /usr/sdb_dir
       generate=no
       transfer=yes
       checkmode=match
```

例 3

SDB ディレクトリ情報ファイルを特定のユニット（UNT3）に配布します（SDB ディレクトリ情報ファイルの作成はしません）。

```
dirinf /usr/sdb_dir
       generate=no
       transfer=yes
       checkmode=match
       unit=UNT3
```

(5) 記述規則

- SDB 制御文の 1 行の行長を超えない範囲内であれば，複数のオペランドを 1 行にまとめて記述できます。この場合，各オペランドの間に，空白またはタブを区切り文字として 1 つ以上指定する必要があります。
- 各オペランドは，形式で説明している順序のとおりに指定してください。

(6) 注意事項

- dirinf 文の実行時にエラーが発生した場合（SDB ディレクトリ情報ファイルの作成および配布処理でエラーが発生した場合），SDB 定義文の処理はロールバックされないため，SDB データベースの定義は追加，変更，削除されています。そのため，SDB ディレクトリ情報ファイルの作成および配布をするために，再度 pdsdbdef コマンドを実行する場合は，SDB 制御文を変更する必要があります。
- SDB ディレクトリ情報ファイルを配布する際，pd_structured_direcotry_path オペランドに指定したディレクトリ下，またはワークディレクトリ下に次の形式のディレクトリまたはファイルがある場合，そのディレクトリまたはファイルを削除対象とします。そのため，pd_structured_direcotry_path オペランドに指定したディレクトリ下，またはワークディレクトリ下には，ユーザが作成したファイルを保管しないことを推奨します。
  - "SDBDEF-AS_プロセス ID_プロセス開始時刻"の形式をしたディレクトリ名
  - "SDBDEF-AD_ユニット名"の形式をしたディレクトリ名
  - "SDBDEF-run"のファイル名
  - "SDBDEF-skip"のファイル名
  - "SDBDEF-result ＋"で始まるファイル名
  - "SDBDEF-tmp_"で始まるファイル名
  - "pdsdbdir_bk_"で始まるファイル名

  なお，ワークディレクトリとは，次のディレクトリを意味しています。
  - ユニット制御情報定義の pd_tmp_directory オペランドに指定したディレクトリ
  - 環境変数 TMPDIR に指定したディレクトリ
  - /tmp
- pdsdbdef コマンドで transfer = yes を指定する場合は，ユティリティを実行するマシン自体を対象にリモートシェルの実行を許可する設定が必要です。設定例については，「4.3.3(4)　リモートシェル実行環境の設定」を参照してください。

11.4 SDB 定義文

SDB 定義文には，pdsdbdef コマンドで操作する SDB データベースの定義情報を指定します。

SDB 定義文ファイルに記述できる制御文（SDB 定義文）を次の表に示します。

表 11–6 SDB 定義文ファイルに記述できる制御文

SDB 定義文	内容
*ENTRY DICTIONARY 文	SDB ディクショナリ情報を追加するときに指定します。
*ALTER DICTIONARY 文	SDB ディクショナリ情報を変更するときに指定します。
*DELETE DICTIONARY 文	SDB ディクショナリ情報を削除するときに指定します。
*CHECK DICTIONARY 文	データベース定義をチェックするときに指定します。
*ENTRY DIRECTORY 文	SDB ディレクトリ情報を追加するときに指定します。
*ALTER DIRECTORY 文	SDB ディレクトリ情報を変更するときに指定します。
*DELETE DIRECTORY 文	SDB ディレクトリ情報を削除するときに指定します。

11.4.1 SDB 定義文の記述規則

SDB 定義文の記述規則を次の表に示します。

表 11–7 SDB 定義文の記述規則

項目	記述規則
SDB 定義文の構成	SDB 定義文は，SDB 定義文の種類や定義句を表すキーワードとユーザ指定値から成ります。 1 つの SDB 定義文内のキーワードとキーワードの間，およびキーワードとユーザ指定値の間は，区切り文字で区切ります。
1 行の長さ	改行コードを含まずに，最大 600 バイトまで指定できます。
改行コード	改行コードには次の文字を指定します。 • NL（X'0a'） • CR + NL（X'0d'+ X'0a'）
分離記号	分離記号は，キーワードとキーワードを区切る文字です。また，キーワード自体としても扱われます。分離記号には次の文字を指定できます。 • ,（X'2c'） • .（X'2e'） • (（X'28'） •)（X'29'） • ;（X'3b'） • *（X'2a'） • /（X'2f'）

項目	記述規則
	• =（X'3d'） • <（X'3c'） • <= • >（X'3e'） • >= • <>
区切り文字	区切り文字には次の文字を指定できます。 • 空白（X'20'） • タブ（X'09'） • NL（X'0a'） • CR（X'0d'） • 全角空白 • 注釈（コメント）
区切り文字を挿入してはいけない位置	区切り文字を挿入してはいけない位置を次に示します。 • キーワードの中 • 引用符（"）で囲まれていない名前の中 • 名前を囲む初めの引用符（"）の後ろ • 名前を囲む終わりの引用符（"）の前 • 数値の中 • 16 進数を表す「X'…'」の「X」と初めの「'」の間
コメントの記述	• 1 行の長さの範囲で，かつ区切り文字が指定できる位置であれば，どこにでもコメントを記述できます。 • コメントは，スラントとアスタリスク（/*），アスタリスクとスラント（*/）で囲んで記述してください。 • 1 つのコメントを複数行にわたって記述できます。ただし，スラントとアスタリスクを 2 行に分けて記述することはできません。 • コメントは入れ子にできません。
継続行	1 つの SDB 定義文を，複数行にわたって記述できます。 キーワードやユーザ指定値自体を複数行に分けて記述することはできません。
SDB 定義文の指定制限	同一行に複数の SDB 定義文は記述できません。 1 つの SDB 定義文内での，定義句を表すキーワードの指定順序は各 SDB 定義文の規則に従います。
キーワードに関する記述規則	ユーザ指定値には，SDB 定義文で使用しているキーワードと同じものを指定しないでください。 指定した場合，SDB 定義文で文法エラーが発生すると，不適切なメッセージが出力されるおそれがあります。
符号なし整数の記述規則	符号なし整数の先頭に 0（ゼロ）を指定する場合，ユーザ指定値の有効桁数（各定義句で指定できる上限値の桁数）の範囲であれば指定できます。

(1) 名前の規則

SDB 定義文で記述する名前の種類と，名前に使用できる文字と長さ（バイト数）の規則を次の表に示します。

表 11–8 SDB 定義文で記述する名前の種類と規則

名前の種類	最大文字数	半角文字							全角文字
		英大文字 ※3	英小文字	数字	カナ文字	下線（_）	ハイフン（-）	空白	
識別子	30 バイト	○	○※1	○	×	○	○※2	○ ※2，※4	×
RD エリア名	30 バイト	○	○※1	○	×	○	○※2	○※2	×

（凡例）

○：使用できます。

×：使用できません。

注

名前の先頭の文字は半角英大英字（A～Z，¥，@，#），または半角英小文字（a～z）にしてください。

注※1

引用符（"）で囲まれている場合，半角英大文字と半角英小文字は区別されます。引用符（"）で囲まれていない半角英小文字は，半角英大文字として扱われます。

注※2

名前中にハイフンまたは空白がある場合，名前を引用符（"）で囲んで指定してください。

注※3

ここでいう英大文字とは，A～Z，¥，@，#を意味しています。

注※4

識別子の最後尾に空白は指定できません。

(2) 予約語

次に示す予約語は，SDB データベース名，SDB データベース格納名，レコード型名，構成要素名，親子集合型名，インデクス名，RD エリア名の名前として定義できません。

- ALL
- HiRDB
- MASTER
- PUBLIC
- SQL および sql で始まる名称
- SYSTEM
- SYSTEM_FLAG_FIELD

- SYSTEM_RECTYPE_NO

11.4.2 *ENTRY DICTIONARY 文（SDB ディクショナリ情報の追加）

SDB データベース定義および SDB データベース格納定義から SDB ディクショナリ情報を作成し，ディクショナリ表に追加します。

(1) 形式

```
*ENTRY  DICTIONARY
    SDBデータベース定義またはSDBデータベース格納定義
```

(2) 説明

*ENTRY DICTIONARY：

SDB ディクショナリ情報を追加（新規に登録）するときに指定します。

*ENTRY DICTIONARY のあとに SDB データベース定義，SDB データベース格納定義を指定します。

SDB データベース定義：

SDB データベース定義を指定します。SDB データベース定義の指定形式については，次の個所を参照してください。

- 4V FMB の SDB データベースを定義する場合
  「11.5.1 SDB データベース定義【4V FMB】」
- 4V AFM の SDB データベースを定義する場合
  「11.6.1 SDB データベース定義【4V AFM】」
- SD FMB の SDB データベースを定義する場合
  「11.7.1 SDB データベース定義【SD FMB】」

SDB データベース格納定義：

SDB データベース格納定義を指定します。SDB データベース格納定義の指定形式については，次の個所を参照してください。

- 4V FMB の SDB データベースを定義する場合
  「11.5.2 SDB データベース格納定義【4V FMB】」
- 4V AFM の SDB データベースを定義する場合
  「11.6.2 SDB データベース格納定義【4V AFM】」
- SD FMB の SDB データベースを定義する場合
  「11.7.2 SDB データベース格納定義【SD FMB】」

(3) 留意事項

- SDB データベース定義と SDB データベース格納定義は，それぞれ 10,000 個まで登録できます。
- 追加する SDB ディクショナリ情報のサイズが 2,048,000,000 バイトを超える場合，*ENTRY DICTIONARY 文の実行がエラーになります。SDB データベース定義情報長，および SDB データベース格納定義情報長の算出式については，次の表を参照してください。
  - 「表 3-26　計算式 2（SDB データベース定義情報長）」
  - 「表 3-27　計算式 3（SDB データベースビュー定義情報長）」
  - 「表 3-28　計算式 4（SDB データベース格納定義情報長）」
- SDB データベース名と SDB データベース格納名は 1 対 1 で対応しています。そのため，登録済みの SDB データベース格納定義で指定されている SDB データベース名は，別の SDB データベース格納定義では指定できません。
- 同一名称の SDB ディクショナリ情報（SDB データベース定義または SDB データベース格納定義）は登録できません。

11.4.3 *ALTER DICTIONARY 文（SDB ディクショナリ情報の変更）【4V FMB，4V AFM】

SDB データベース定義および SDB データベース格納定義から SDB ディクショナリ情報を作成し，ディクショナリ表に登録されている SDB ディクショナリ情報を変更します。

SD FMB の SDB データベースの SDB ディクショナリ情報を*ALTER DICTIONARY 文で変更することはできません。

(1) 形式

```
*ALTER DICTIONARY
    SDBデータベース定義またはSDBデータベース格納定義
```

(2) 説明

*ALTER DICTIONARY：

SDB ディクショナリ情報を変更するときに指定します。*ALTER DICTIONARY のあとに SDB データベース定義，SDB データベース格納定義を指定します。

SDB データベース定義：

変更後の SDB データベース定義を指定します。

SDB データベース定義を変更する場合は，変更個所だけを指定するのではなく，変更後の SDB データベース定義をすべて指定してください。SDB データベース定義の指定形式，および変更できるオペランドについては，次の個所を参照してください。

- 4V FMB の SDB データベースの定義を変更する場合
  「11.5.1　SDB データベース定義【4V FMB】」
- 4V AFM の SDB データベースの定義を変更する場合
  「11.6.1　SDB データベース定義【4V AFM】」

SDB データベース格納定義：

変更後の SDB データベース格納定義を指定します。

SDB データベース格納定義を変更する場合は，変更個所だけを指定するのではなく，変更後の SDB データベース格納定義をすべて指定してください。SDB データベース格納定義の指定形式，および変更できるオペランドについては，次の個所を参照してください。

- 4V FMB の SDB データベースの定義を変更する場合
  「11.5.2　SDB データベース格納定義【4V FMB】」
- 4V AFM の SDB データベースの定義を変更する場合
  「11.6.2　SDB データベース格納定義【4V AFM】」

(3)　留意事項

- *ALTER DICTIONARY 文および*ALTER DIRECTORY 文による SDB データベースの定義変更では，最下位のレコード型とその親子集合の追加，および横分割格納した RD エリアの変更（追加，削除，分割，統合）ができます。SDB データベース定義および SDB データベース格納定義で変更できるオペランドについては，次の個所を参照してください。
  - 4V FMB の SDB データベースの定義を変更する場合
    「11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】」
  - 4V AFM の SDB データベースの定義を変更する場合合
    「11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】」
- *ALTER DICTIONARY 文および*ALTER DIRECTORY 文による RD エリアの削除，分割，または統合を行う場合，pdsdbdef コマンドの実行時に-i オプションを指定する必要があります。-i オプションを指定しないと，pdsdbdef コマンドがエラーとなります。
- 変更後の SDB ディクショナリ情報のサイズが 2,048,000,000 バイトを超える場合，*ALTER DICTIONARY 文の実行がエラーになります。
- 追加する子レコード型の親レコード型で，子レコード用の管理領域の予備領域が不足している場合，子レコード型を追加できないため，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文によるデータベース定義のチェック時にエラーとなります。
- *ALTER DICTIONARY 文による RD エリアの分割または統合を行った場合，*CHECK DICTIONARY 文または*ALTER DIRECTORY 文の延長で，変更対象の RD エリアに格納されているデータが削除されることがあります。詳細については，「表 11-23　RD エリアの定義変更時のデータの削除有無（格納条件指定の横分割の場合）」または「表 11-24　RD エリアの定義変更時のデータの削除有無（境界値指定の横分割の場合）」を参照してください。

また，このデータ削除を伴う操作に限り，例外的に pdsdbdef コマンドの途中に同期点処理が行われることがあります。そのため，pdsdbdef コマンドの実行中にエラーが発生した場合，SDB 定義文ファイル中に指定した SDB 定義文は，完了した処理と無効となった処理に分かれます。したがって，pdsdbdef コマンドを再実行する場合は，SDB 定義文の変更が必要となります。詳細については，「11.4.10　エラー発生時の再実行について」を参照してください。

11.4.4　*DELETE DICTIONARY 文（SDB ディクショナリ情報の削除）

SDB ディクショナリ情報をディクショナリ表から削除します。

(1)　形式

■SDB データベース定義の SDB ディクショナリ情報を削除する場合

```
*DELETE DICTIONARY
    SCHEMA SDBデータベース名
```

■SDB データベース格納定義の SDB ディクショナリ情報を削除する場合

```
*DELETE DICTIONARY
    STORAGE SCHEMA SDBデータベース格納名 FOR SDBデータベース名
```

(2)　説明

*DELETE DICTIONARY：

SDB ディクショナリ情報を削除するときに指定します。

注意事項

SDB ディクショナリ情報および SDB ディレクトリ情報を削除する際は，必ず*DELETE DIRECTORY 文を先に実行してください。

SCHEMA SDB データベース名：

SDB ディクショナリ情報を削除する SDB データベースの SDB データベース名を指定します。

STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名：

SDB ディクショナリ情報を削除する SDB データベースの SDB データベース格納名および SDB データベース名を指定します。

(3)　留意事項

- *DELETE DICTIONARY 文を実行する場合，pdsdbdef コマンドの実行時，-i オプションの指定が必須になります。-i オプションの指定がないと pdsdbdef コマンドの実行時にエラーとなります。

- *DELETE DICTIONARY 文を実行すると，SDB データベースのレコードおよびインデクスも削除されます。

11.4.5 *CHECK DICTIONARY 文（データベース定義のチェック）

SDB データベースの SDB データベース定義および SDB データベース格納定義の整合性をチェックします。定義の指定内容に誤りがないか，または関連する情報と矛盾がないかがチェックされます。

*CHECK DICTIONARY 文は次に示す場合に指定します。

- *ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文を別々に実行する場合に，*ENTRY DICTIONARY 文と一緒に*CHECK DICTIONARY 文を指定します。
- *ALTER DICTIONARY 文と*ALTER DIRECTORY 文を別々に実行する場合に，*ALTER DICTIONARY 文と一緒に*CHECK DICTIONARY 文を指定します。

*ENTRY DICTIONARY 文（または*ALTER DICTIONARY 文）と*CHECK DICTIONARY 文を一緒に指定した場合，データベース定義の不整合が検知されると，SDB ディレクトリ情報はディクショナリ表に追加されません。

なお，横分割格納した RD エリアの変更を行う場合は，*CHECK DICTIONARY 文の実行時に，変更対象となる RD エリアに格納されたデータが削除されます。

(1) 形式

```
*CHECK DICTIONARY
    STORAGE SCHEMA SDBデータベース格納名 FOR SDBデータベース名
```

(2) 説明

*CHECK DICTIONARY：

SDB データベースの SDB データベース定義および SDB データベース格納定義の整合性をチェックするときに指定します。

STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名

整合性をチェックする SDB データベースの SDB データベース格納名および SDB データベース名を指定します。

11.4.6 *ENTRY DIRECTORY 文（SDB ディレクトリ情報の追加）

SDB データベースの SDB ディレクトリ情報を作成し，ディクショナリ表に追加します。

(1) 形式

```
*ENTRY DIRECTORY
    DBSCHM SDBデータベース名
```

(2) 説明

*ENTRY DIRECTORY：

SDB ディレクトリ情報を追加するときに指定します。

DBSCHM SDB データベース名：

SDB ディレクトリ情報を追加する SDB データベースの SDB データベース名を指定します。

(3) 留意事項

*ENTRY DIRECTORY 文の実行時，データベース定義（SDB データベース定義および SDB データベース格納定義）の整合性チェックが実行されます。定義の指定内容に誤りがないか，または関連する情報と矛盾がないかがチェックされます（*CHECK DICTIONARY 文と同じ処理が実行されます）。その結果，データベース定義の指定内容に不整合が検知された場合，*ENTRY DIRECTORY 文の実行がエラーになります。

ただし，一度でも*CHECK DICTIONARY 文を実行したデータベース定義に対しては，*ENTRY DIRECTORY 文の実行時にデータベース定義のチェックは実行されません。

11.4.7 *ALTER DIRECTORY 文（SDB ディレクトリ情報の変更）【4V FMB，4V AFM】

SDB データベースの SDB ディレクトリ情報を作成し，ディクショナリ表に登録されている SDB ディレクトリ情報を変更します。

SD FMB の SDB データベースの SDB ディレクトリ情報を*ALTER DIRECTORY 文で変更することはできません。

(1) 形式

```
*ALTER DIRECTORY
    DBSCHM SDBデータベース名
```

(2) 説明

*ALTER DIRECTORY：

SDB ディレクトリ情報を変更するときに指定します。

DBSCHM SDB データベース名：

SDB ディレクトリ情報を変更する SDB データベースの SDB データベース名を指定します。

(3) 留意事項

- *ALTER DIRECTORY 文の実行時，データベース定義（SDB データベース定義および SDB データベース格納定義）の整合性チェックが実行されます。定義の指定内容に誤りがないか，または関連する情報と矛盾がないかがチェックされます（*CHECK DICTIONARY 文と同じ処理が実行されます）。その結果，データベース定義の指定内容に不整合が検知された場合，*ALTER DIRECTORY 文の実行がエラーになります。

  ただし，一度でも*CHECK DICTIONARY 文を実行したデータベース定義に対しては，*ALTER DIRECTORY 文の実行時にデータベース定義のチェックは実行されません。
- *ALTER DICTIONARY 文および*ALTER DIRECTORY 文による RD エリアの削除，分割，または統合を行う場合，pdsdbdef コマンドの実行時に-i オプションを指定する必要があります。-i オプションを指定しないと，pdsdbdef コマンドがエラーとなります。

11.4.8 *DELETE DIRECTORY 文（SDB ディレクトリ情報の削除）

SDB データベースの SDB ディレクトリ情報をディクショナリ表から削除します。

(1) 形式

```
*DELETE DIRECTORY
    DBSCHM SDBデータベース名
```

(2) 説明

*DELETE DIRECTORY：

SDB ディレクトリ情報を削除するときに指定します。

DBSCHM SDB データベース名：

SDB ディレクトリ情報を削除する SDB データベースの SDB データベース名を指定します。

(3) 留意事項

*DELETE DIRECTORY 文を実行する場合，pdsdbdef コマンドの実行時に-i オプションの指定が必須になります。-i オプションの指定がないと pdsdbdef コマンドの実行時にエラーとなります。

11.4.9　SDB 定義文の同期点処理と推奨する SDB 定義文の指定方法

SDB 定義文の同期点処理は，pdsdbdef コマンド単位に実行されます。したがって，SDB 定義文ファイル中に指定したすべての SDB 定義文の処理が正常終了した場合にかぎり，pdsdbdef コマンドが正常終了します。SDB 定義文ファイル中に指定した SDB 定義文のうち，1 つでもエラーが発生した場合，すべての SDB 定義文の処理が無効になります。

推奨する SDB 定義文の指定方法を次に示します。

- HiRDB の再起動を必要としない SDB データベースの定義追加をする場合
  1. *ENTRY DICTIONARY 文と*CHECK DICTIONARY 文を指定して，pdsdbdef コマンドを実行します。※1
  2. *ENTRY DIRECTORY 文を指定して，pdsdbdef コマンドを実行します。
- HiRDB の再起動を必要としない SDB データベースの定義変更をする場合
  1. *ALTER DICTIONARY 文と*CHECK DICTIONARY 文を指定して，pdsdbdef コマンドを実行します。※1
  2. *ALTER DIRECTORY 文を指定して，pdsdbdef コマンドを実行します。
- HiRDB の再起動を必要とする SDB データベースの定義追加をする場合

  *ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文を指定して，pdsdbdef コマンドを実行します。※2
- HiRDB の再起動を必要とする SDB データベースの定義変更をする場合

  *ALTER DICTIONARY 文と*ALTER DIRECTORY 文を指定して，pdsdbdef コマンドを実行します。※2

注※1

*ENTRY DICTIONARY 文（または*ALTER DICTIONARY 文）と*CHECK DICTIONARY 文を別々に実行した場合，*CHECK DICTIONARY 文の実行時にデータベース定義の不整合が検知されると，不整合が検知された SDB ディクショナリ情報を*DELETE DICTIONARY 文で削除する必要が生じます。しかし，*DELETE DICTIONARY 文を実行した場合，HiRDB を再起動する必要があるため，*ENTRY DICTIONARY 文と*CHECK DICTIONARY 文は一緒に指定するようにしてください。

注※2

*ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文（または*ALTER DICTIONARY 文と*ALTER DIRECTORY 文）を別々に実行した場合，*ENTRY DIRECTORY 文（または*ALTER DIRECTORY 文）の実行時にデータベース定義の不整合が検知されると，不整合が検知された SDB ディクショナリ情報を*DELETE DICTIONARY 文で削除する必要が生じます。そのため，*ENTRY DICTIONARY 文と*ENTRY DIRECTORY 文（または*ALTER DICTIONARY 文と*ALTER DIRECTORY 文）は，一緒に指定するようにしてください。

11.4.10　エラー発生時の再実行について

同期点処理は，基本的には pdsdbdef コマンド単位に実行されます。したがって，エラーの発生時は，エラー原因を取り除いたあとに，SDB 定義文ファイルの内容を変更しないで，そのまま pdsdbdef コマンドを再実行できます。

ただし，RD エリアの分割や統合はデータ部にも影響がある操作のため，pdsdbdef コマンドの途中で同期点が取得されることがあります。このとき，KFPB61244-I メッセージが出力されて，同期点が取得されたことが通知されます。データ削除と同期点の取得については，「表 11-23　RD エリアの定義変更時のデータの削除有無（格納条件指定の横分割の場合）」または「表 11-24　RD エリアの定義変更時のデータの削除有無（境界値指定の横分割の場合）」を参照してください。これら 2 つの表の※1 のケースが該当します。

このメッセージの出力後にエラーが発生した場合は，pdsdbdef コマンドの再実行前に SDB 定義文ファイルの内容を修正する必要があります。修正方法を次の表に示します。

表 11–9　KFPB61244-I メッセージの出力後にエラーが発生した場合の SDB 定義文の修正方法

KFPB61244-I メッセージの出力	pdsdbdef コマンドの再実行前の SDB 定義文の修正方法
なし	SDB 定義文を修正する必要はありません。
あり	KFPB61244-I メッセージが出力された SDB 定義文の前に指定した SDB 定義文を削除してください。KFPB61244-I メッセージが出力された SDB 定義文は削除しないでください。

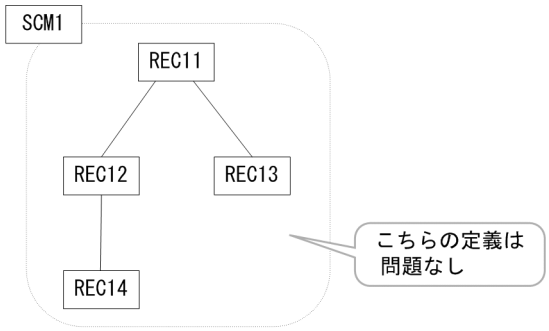
SDB 定義文の修正例を次の図に示します。

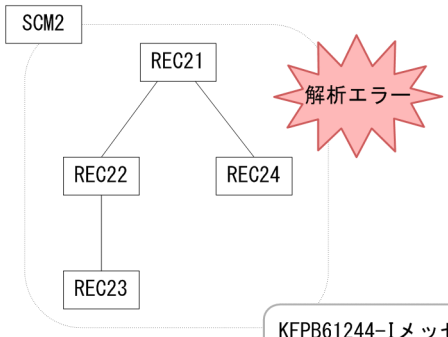
図 11-10　SDB 定義文の修正例

●＜1回目の定義実行時＞

＜定義文ファイル＞

```
*ALTER  DICTIONARY
    STORAGE  SCHEMA  SCM1  OF  SCM1
      …

*ALTER  DIRECTORY
    DBSCHM  SCM1

*ENTRY  DICTIONARY
    SCHEMA  SCM2
        REC21  …
        REC22  …
        REC23  …
        REC24  …
      …
    END  SCHEMA  SCM2

*ENTRY  DICTIONARY
    STORAGE  SCHEMA  SCM2  OF  SCM2
      …

*CHECK  DICTIONARY
    STORAGE  SCHEMA  SCM2  OF  SCM2
```





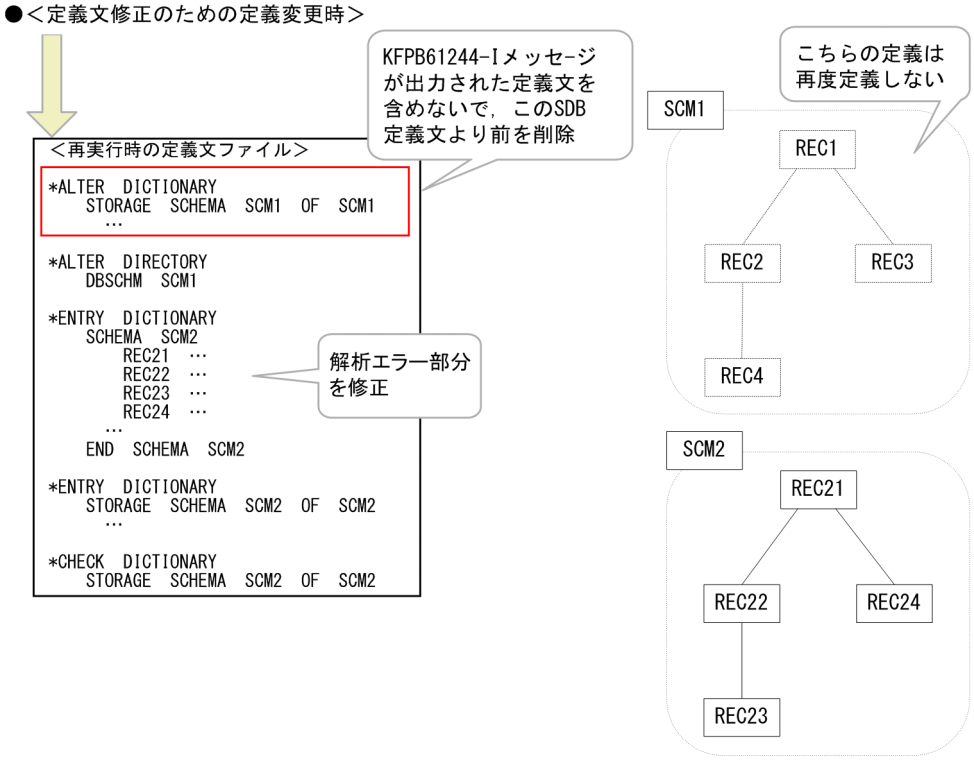
KFPB61244-Iメッセージ
出力

＜実行結果ファイル＞

```
STATEMENT    ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
         1    *ALTER  DICTIONARY
         2      STORAGE  SCHEMA  SCM1  OF  SCM1
               …
 2550 16:20:38 unt1 sds01    KFPB61202-I *ALTER DICTIONARY terminated, return code=0

STATEMENT    ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
         1    *ALTER  DIRECTORY
         2      DBSCHM  SCM1
 2560 16:20:40 unt1 sds01    KFPB61244-I *ALTER DIRECTORY transaction committed, code=2
 2560 16:20:40 unt1 sds01    KFPB61202-I *ALTER DIRECTORY terminated, return code=0

STATEMENT    ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
         1    *ENTRY  DICTIONARY
         2      SCHEMA  SCM2
         3        REC21  …
               …
       200      END  SCHEMA  SCM2
 2565 16:20:42 unt1 sds01    KFPB61597-E …
 2565 16:20:42 unt1 sds01    KFPB61202-I *ENTRY DICTIONARY terminated, return code=12
```

解析エラー

●＜定義文修正のための定義変更時＞



11.5　SDB データベース定義，SDB データベース格納定義【4V FMB】

4V FMB の SDB データベースを定義する場合の，SDB データベース定義および SDB データベース格納定義の指定形式と各オペランドの指定内容について説明します。

11.5.1　SDB データベース定義【4V FMB】

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定する SDB データベース定義について説明します。

注意事項

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定した SDB データベース定義は保存しておいてください。*ALTER DICTIONARY 文で SDB データベース定義を変更する際，変更個所だけを指定するのではなく，変更後の SDB データベース定義をすべて指定する必要があります。このとき，保存しておいた SDB データベース定義が必要になります。

(1)　機能

SDB データベースの論理構造を定義します。

(2)　形式

項番	形式	オペランドの追加可否※1
(a)	SCHEMA SDB データベース名	×
(b)	DBTYPE 4V FMB	×
(c)	RECORD レコード型名	○※2
(d)	{レベル番号｜ITEM}　構成要素名	
(e)	〔{CHARACTER 長さ ｜XCHARACTER 長さ ｜PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕 ｜INTEGER}	
(f)	TYPE データ種別 1，データ種別 2	
	〕…	
(g)	〔FUNCTION 〔REFER USE〕 〔ADD {USE ｜ NOUSE}〕 〔UPDATE {USE ｜ NOUSE}〕	

項番	形式	オペランドの追加可否※1
	〔ERASE {USE | NOUSE}〕 〔ALLERASE NOUSE〕 〕	
(h)	〔RECORDID レコード識別コード〕	
	…	
(i)	〔SET 親子集合型名	
(j)	OWNER 親レコード型名	
(k)	ORDER {LAST | SORTED DUPLICATES PROHIBITED}	
(l)	MEMBER 子レコード型名	
(m)	INSERTION AUTOMATIC	
(n)	RETENTION FIXED	
(o)	〔KEY {ASCENDING | DESCENDING} 構成要素名〕	
(p)	〔SETOPTION	
(q)	〔OCCURRENCE NUMBER 一連番号の最大値	
(r)	〔WARNING {一連番号使用比率 1 〔, 一連番号使用比率 2 〔, 一連番号使用比率 3〕〕 | 0}〕	
(s)	〔REUSE {YES | NO}〕 〕	
	〕 〕…	
(t)	〔SDBOPTION	×
(u)	〔FUNCTION 〔REFER USE〕 〔ADD {USE | NOUSE}〕 〔UPDATE {USE | NOUSE}〕 〔ERASE {USE | NOUSE}〕 〔ALLERASE NOUSE〕 〔DBLODUTL USE〕 〔FORMAT NOUSE〕 〕	
(v)	〔OCCURRENCE WARNING {一連番号使用比率 1 〔, 一連番号使用比率 2 〔, 一連番号使用比率 3〕〕 | 0}〕	○※3
	〕	

項番	形式	オペランドの追加可否※1
(w)	END SCHEMA	×

（凡例）

○：*ALTER DICTIONARY 文の実行時に追加できるオペランドです。

×：*ALTER DICTIONARY 文の実行時に追加できないオペランドです。

注※1

*ALTER DICTIONARY 文実行時のオペランド追加可否です。

注※2

- 最下位の子レコード型の追加とその親子集合の追加ができます。
- RECORD 句と SET 句は必ず両方指定してください。どちらか片方だけを指定することはできません。

注※3

ルートレコード型だけの定義に対して子レコード型を 1 個以上追加する場合は，追加指定できます。

参考

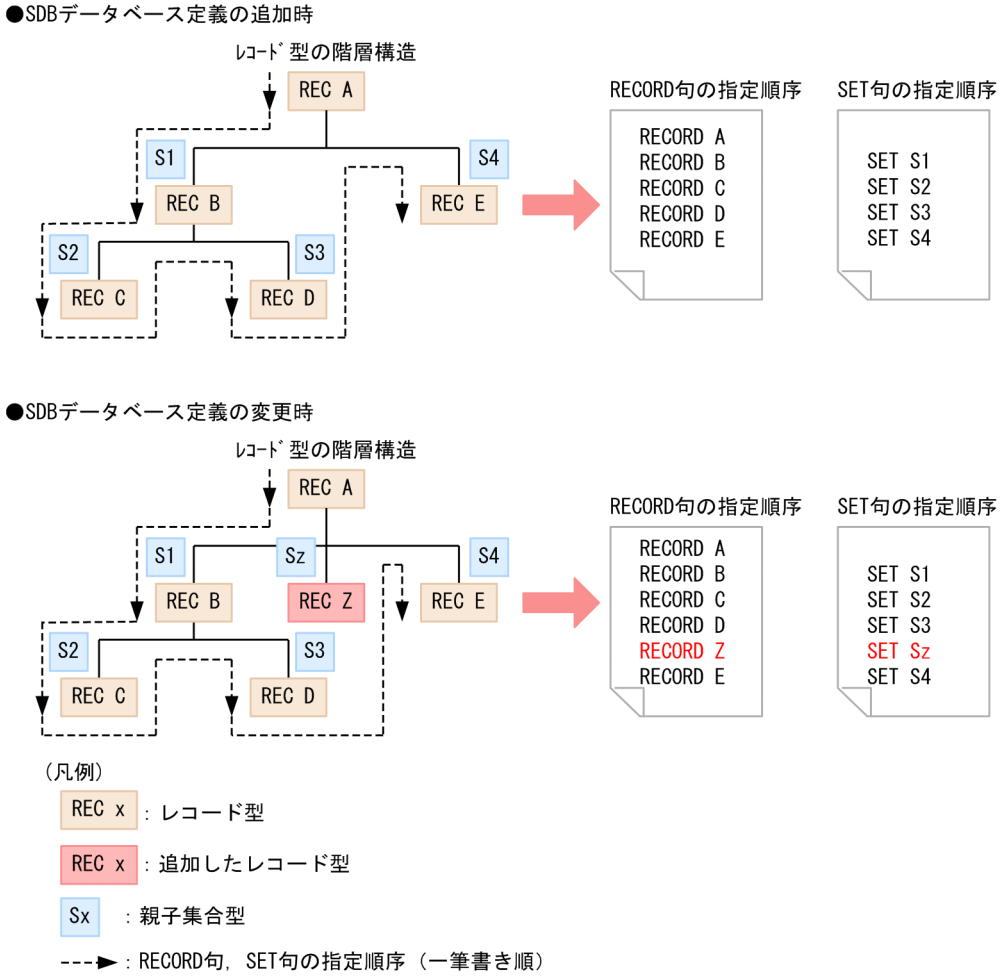
SDBOPTION 句下の READYMODE 句の指定は無視されるため，バージョン 09-66 のマニュアルから，上記の形式に READYMODE 句は記載していません。READYMODE 句を指定したままでもエラーにはなりません。また，READYMODE 句の指定値によってシステムの動作が変わることはありません。

(3) 定義句の指定順序

SDB データベース定義の定義句の指定順序には，次に示す規則があります。

- 必須の定義，または必須の定義と省略できる定義の間は，上記の形式で示す順序で指定してください。
- RECORD 句下の FUNCTION 句と RECORDID 句，および FUNCTION 句下の各句については指定順はありません。任意の順番で指定できます。
- RECORD 句および SET 句の指定順序は，次の図に示す規則（一筆書き順）に従ってください。

図 11-11　RECORD 句および SET 句の指定順序（一筆書き順）の規則

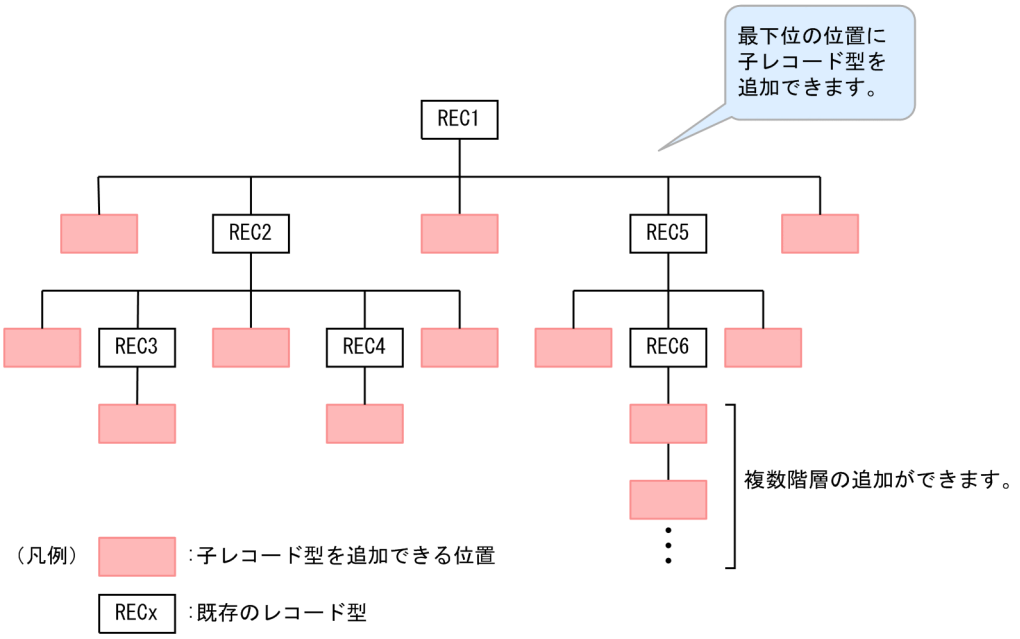


(4)　*ALTER DICTIONARY 文による SDB データベースの定義変更

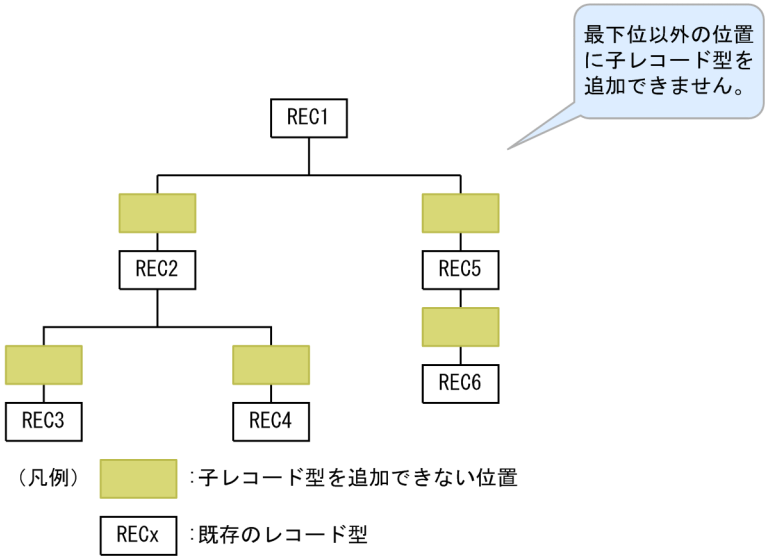
*ALTER DICTIONARY 文による SDB データベースの定義変更は，最下位の子レコード型の追加とその親子集合の追加ができます。子レコード型を追加できる位置と追加できない位置を次の図に示します。

図 11-12　子レコード型を追加できる位置と追加できない位置

■子レコード型を追加できる位置

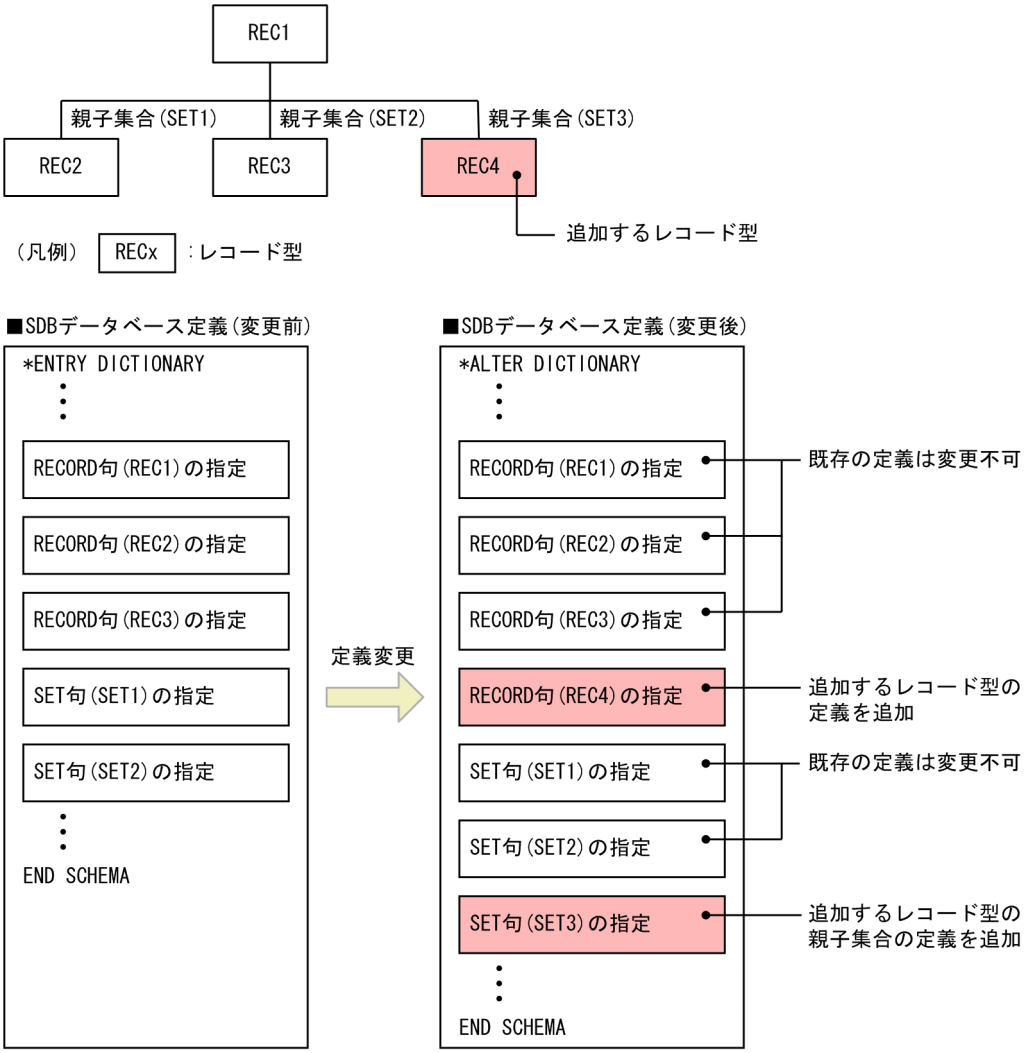


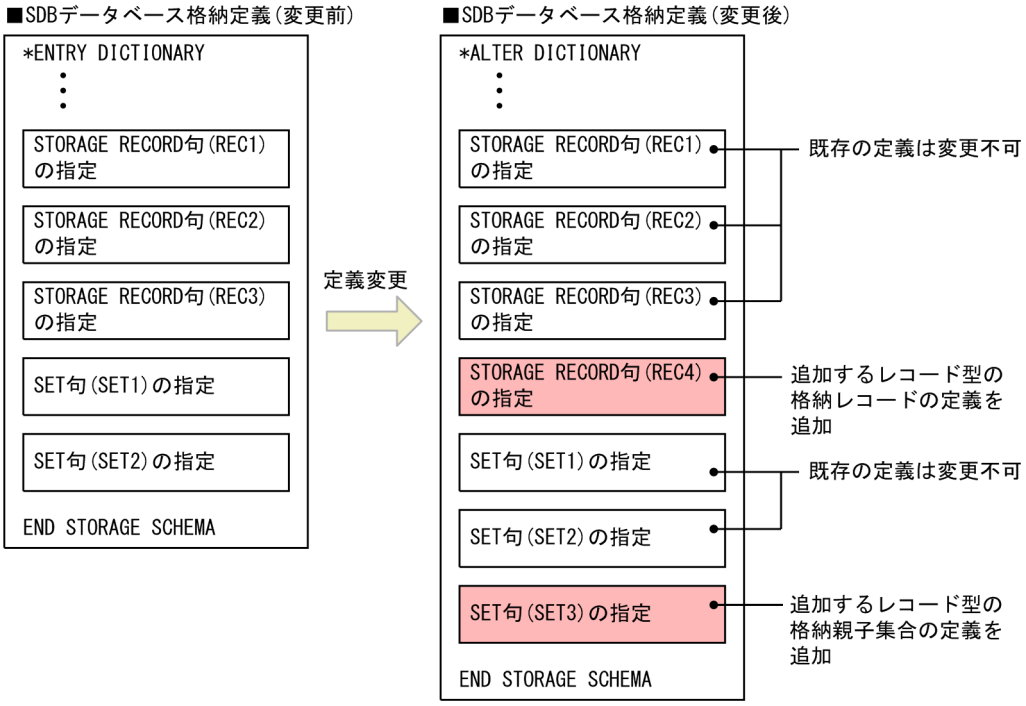
■子レコード型を追加できない位置



なお，*ALTER DICTIONARY 文では，既存のレコード型の定義は変更できません。*ALTER DICTIONARY 文による SDB データベースの定義変更の例を次の図に示します。

図 11-13　*ALTER DICTIONARY 文による SDB データベースの定義変更の例（4V FMB の SDB データベースの場合）





*ALTER DICTIONARY 文で実行できる SDB データベースの定義変更を次の表に示します。

表 11-10　*ALTER DICTIONARY 文で実行できる SDB データベースの定義変更

定義名	定義変更の内容	説明
SDB データベース定義	レコード型の追加（RECORD 句の追加）	次に示す条件をすべて満たす場合にレコード型を追加できます。 • 追加するレコード型が子レコード型である。 • 追加する子レコード型の親レコード型で管理している子レコード用の管理領域の予備領域がある（子レコード型を追加しても問題ないだけの予備領域がある）。 また，次に示す定義の追加または変更を行う必要があります。 • 既存のレコード型を親レコード型，追加するレコード型を子レコード型とする親子集合を追加する。 • SDB データベース格納定義の STORAGE RECORD 句を追加し，対応する格納レコード名を指定する。
	親子集合の追加（SET 句の追加）	追加した子レコード型の親子集合を追加できます。
	一連番号の使用状況の監視の追加（OCCURRENCE WARNING 句の追加）	ルートレコード型だけの定義に対して，子レコード型を 1 個以上追加する場合に限り，追加指定できます。
SDB データベース格納定義	格納レコードの追加（STORAGE RECORD 句の追加）	SDB データベース定義で子レコード型を追加した場合に，対応する格納レコード名を追加します。子レコード型を追加していない場合に，格納レコード名だけを追加することはできません。 なお，追加した子レコード型にデータ種別が K，N の構成要素が存在する場合は，K，N または U，K の構成要素の数まで二次インデクスを追加できます。二次インデクスの追加に伴い，二次インデクスを格納する RD エリアを追加できます。

定義名	定義変更の内容	説明
	格納親子集合の追加（SET 句の追加）	子レコード型を追加する際に追加する親子集合に対応する格納親子集合を追加できます。子レコード型を追加しないで，格納親子集合だけを追加することはできません。
	キー値の追加（KEYDEF 句の追加）	キー値の追加はできません。

(5)　オペランドの説明

SDB データベース定義の各オペランドの指定内容を説明します。

(a)　SCHEMA

SCHEMA SDB データベース名

～<識別子>((1～30 バイト))

SDB データベースの SDB データベース名を指定します。

すでに使用している SDB データベース名は指定できません。

(b)　DBTYPE

定義する SDB データベースの SDB データベース種別を指定します。

DBTYPE 4V FMB

SDB データベース種別が 4V FMB の SDB データベースを定義する場合は，DBTYPE 句に 4V FMB を指定します。

4V FMB の SDB データベースとは，階層構造を持つデータベースで，メインフレームの TMS-4V/SP がサポートしている FMB のデータベースに相当します。詳細については，「2.3.3　SDB データベース」を参照してください。

注意事項

SDB データベース種別が 4V（4V FMB または 4V AFM）の SDB データベースと SD FMB の SDB データベースをシステム内で混在して定義することはできません。混在させた場合，動作を保証しません。

(c)　RECORD

RECORD レコード型名

～<識別子>((1～30 バイト))

レコード型名を指定します。

《規則》

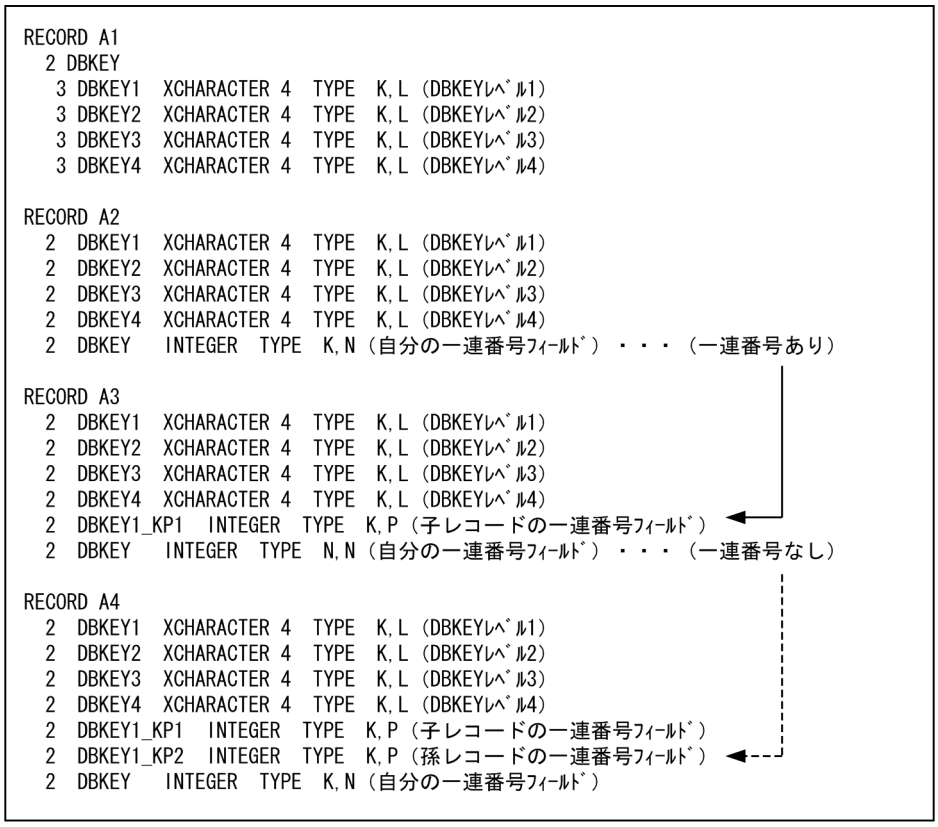
- レコード型名は，同一スキーマ内で一意になるように指定してください。

- 1 つの SDB データベース定義内で指定できる RECORD 句は，1～128 個です。

《定義例》

階層構造がルートレコード（RECORD A1），子レコード（RECORD A2），孫レコード（RECORD A3），ひ孫レコード（RECORD A4）の場合を例に，データベースキーの関係を次に示します。

```
ルートレコード：DBKEY
子レコード：ルートレコードのDBKEY＋一連番号1
孫レコード：ルートレコードのDBKEY＋一連番号1＋一連番号2
ひ孫レコード：ルートレコードのDBKEY＋一連番号1＋一連番号2＋一連番号3
```



(d) レベル番号，ITEM

{レベル番号｜ITEM}　構成要素名

構成要素に関する情報を指定します。1 つの RECORD 句内での構成要素の階層レベルを指定します。

- すべての構成要素が基本項目の場合，ITEM を指定できます。
- 1 つのレコード型内で，レベル番号と ITEM は同時に指定できません。

SDB データベースにアクセスするための API で「DBKEY」の名称を利用するため，構成要素名として「DBKEY」を指定する必要があります。指定方法の詳細については，「表 11-14　ルートレコード型の構成要素の指定」～「表 11-16　子レコード（3 階層以上）の構成要素の指定」を参照してください。

レベル番号

～<符号なし整数>((2～49))

1 つの RECORD 句内での構成要素の階層レベルを指定します。

階層が深くなるに従ってレベル番号を増やすように指定してください。レベル番号は連続した値である必要はありません。

階層が浅くなるときは，すでに指定したレベル番号にしてください。

ITEM

すべての構成要素が基本項目の場合に指定できます。

構成要素名

～<識別子>((1～30 バイト))

構成要素名を指定します。

《規則》

- 構成要素名は，RECORD 句内で一意にしてください。
- RECORD 句内に指定できる構成要素の合計長は最大 30,000 バイトです。また，RECORD 句内に指定できる構成要素の個数は最大 30,000 個です。
- 各レコード型の最初の基本項目の構成要素のデータ種別 1 には，K を指定してください。
- 上位レコードのキーの構成要素，および一連番号の構成要素は，下位レコードの構成要素として，データ型とデータ種別を同じように指定してください。
- ユーザデータ（データ種別 1，2（TYPE U,D））の構成要素は，幾つかの構成要素をまとめて，ユーザ任意の集団項目内に指定できます。

  ユーザキー（データ種別 1，2（TYPE U,K)），ユーザファイル通番（データ種別 1，2（TYPE U,F））の構成要素は，集団項目内に指定できません。

(e) データ型

{CHARACTER 長さ

｜XCHARACTER 長さ

｜PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕

｜INTEGER}

構成要素のデータ型を指定します。構成要素に指定できるデータ型を次の表に示します。

表 11-11　構成要素に指定できるデータ型

データ種別	データ型	データ形式
文字列	CHARACTER	固定長文字列
16 進形式の文字列	XCHARACTER	16 進固定長文字列
数値	PACKED 〔DECIMAL FIXED〕	符号付きパック形式 10 進データ
	INTEGER	整数

CHARACTER 長さ

～<符号なし整数>((1～30,000))

文字列データの長さを指定します。

XCHARACTER 長さ

～<符号なし整数>((1～30,000))

16 進形式の文字列データの長さを指定します。

PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕

整数部桁数

～<符号なし整数>((0～38))

小数部桁数

～<符号なし整数>((0～38))

符号付きパック形式の 10 進データの整数部桁数，および小数部桁数を指定します。

整数部桁数と小数部桁数の和は，1～38 になるようにしてください。

INTEGER

値の範囲が-2,147,483,648～2,147,483,647 の整数を指定します。長さは指定できません。

《規則》

- 一連番号の構成要素には，必ず INTEGER を指定してください。
- 基本項目の場合，データ型および TYPE 句を指定してください。
- 集団項目の場合，データ型および TYPE 句を省略してください。
- KEYDEF 句に指定する構成要素については，データの長さを 1～30 バイトで指定してください。KEYDEF 句については，「11.5.2(3)(v)　KEYDEF」を参照してください。

(f)　TYPE

TYPE データ種別 1，データ種別 2

構成要素のデータ種別を指定します。

データ種別 1

～<英字>((1 バイト))

データベースキーかユーザデータかを区別するための種別を指定します。

データ種別 2

～<英字>((1 バイト))

データ種別 1 をさらに分類するための詳細な種別を指定します。

TYPE 句で指定するデータ種別の指定値と意味を次の表に示します。

表 11-12　データ種別の指定値と意味

データ種別	指定値	意味
データ種別 1	K	データベースキーで，次に該当する場合の構成要素 • データベースキー • 上位レコードの一連番号

データ種別	指定値	意味
		• 自レコードの一連番号（一連番号の最大値（OCCURRENCE NUMBER 句の指定）が 1 以上の場合）
	N	自レコードの一連番号（一連番号の最大値（OCCURRENCE NUMBER 句の指定）が 0 の場合）
	U	ユーザデータ
データ種別 2	A	レコード型の RD エリア分割キーとする構成要素（SDB データベースを横分割する際，レコードの格納 RD エリアを決める構成要素）
	N	自レコードの一連番号の構成要素 • データ種別 1 が K の場合，一連番号属性あり • データ種別 1 が N の場合，一連番号属性なし
	P	上位レコードの一連番号の構成要素
	L	そのほかのデータベースキーの構成要素
	F	ユーザファイル通番の構成要素
	K	ユーザキーの構成要素
	D	そのほかのユーザデータの構成要素

《規則》

指定順序と記述の規則

- K，N，U の順に，必要な種別を指定してください。
- ユーザデータのあとにデータベースキーを指定することはできません。

そのほかの規則

- データ種別 1 が K，かつデータ種別 2 が A の構成要素は，ルートレコード型と子レコード型で同一名称にしてください。
- データ種別 1 が K，かつデータ種別 2 が A の構成要素は，データの長さを 1～30 バイトの範囲で指定してください。
- RECORD 句内では，データ種別 2 が A の構成要素は，それぞれ最大 1 つ指定できます。
- データ種別 1 が K，かつデータ種別 2 が N の場合，一連番号属性があります。OCCURRENCE NUMBER 句には 1 以上を指定するか，または指定を省略してください。
- データ種別 1 が N，かつデータ種別 2 が N の場合，一連番号属性がありません。OCCURRENCE NUMBER 句には 0 を指定してください。
- 上位レコードで一連番号属性なしの構成要素（データ種別 1，2（N，N））または一連番号属性ありの構成要素（データ種別 1，2（K，N））は，下位レコードの構成要素で，データ種別 1，2（K，P）を指定してください。
- データ種別 2 が F の構成要素はレコード型内の最後の構成要素に 1 つだけ指定できます。
- データ種別 2 が K の構成要素は 1 レコード型に最大 1 つのユーザキーを指定できます。

- 構成要素のデータ種別 1 とデータ種別 2 で，指定できる組み合わせを次の表に示します。

表 11-13　構成要素のデータ種別 1 とデータ種別 2 に指定できる組み合わせ

データ種別 1 の指定	データ種別 2 に指定できる値
K の場合	L
	A
	N※1
	P
N の場合	N※2
U の場合	D
	F※3
	K

注※1

OCCURRENCE NUMBER 句は省略するか，または 1 以上を指定してください。

注※2

OCCURRENCE NUMBER 句には 0 を指定してください。

注※3

レコード型内で最後に 1 つだけ指定できます。

- ルートレコード型と子レコード型（2 階層），および子レコード型（3 階層以上）の構成要素の指定を次の表に示します。

  指定順序や指定する個数，データ種別の組み合わせに誤りがある場合はエラーとなります。なお，1 つの項番の中で基本項目と集団項目を複数指定できる場合の，指定順序は任意です。

表 11-14　ルートレコード型の構成要素の指定

指定順	項目		説明	データ種別	
				1	2
1	「DBKEY」の構成要素名		• 構成要素名を「DBKEY」とします。 • 集団項目として指定します。	−	−
2	データベースキー 1～7	構成要素名	• 各データベースキーを，集団項目「DBKEY」下の基本項目として，同じレベル番号で指定します。 • 集団項目とはしないでください。	K	L
		個数	1～7		A
3	ユーザデータの構成要素名		ユーザデータの構成要素名を指定します。	−	−
				U	D
					F

（凡例）

－：該当しません。

表 11-15　子レコード型（2 階層）の構成要素の指定

指定順	項目		説明	データ種別	
				1	2
1	データベースキー 1～7	構成要素名	• ルートレコード型で指定したデータベースキーを，同じレベル番号の基本項目として指定します。 • 集団項目とはしないでください。	K	L
		個数	1～7		A
2	一連番号	OCCURRENCE NUMBER 句の指定値が 1 以上の場合	• 構成要素名を「DBKEY」とします。 • 基本項目として指定します。	K	N
		OCCURRENCE NUMBER 句の指定値が 0 の場合		N	N
3	ユーザデータの構成要素名		ユーザデータの構成要素名を指定します。	－	－
				U	D
					K
					F

（凡例）

－：該当しません。

表 11-16　子レコード（3 階層以上）の構成要素の指定

指定順	項目		説明	データ種別	
				1	2
1	データベースキー 1～7	構成要素名	• 上位の子レコード型で指定したデータベースキーを，同じレベル番号の基本項目として指定します。 • 集団項目とはしないでください。	K	L
		個数	1～7		A
2	上位の一連番号	構成要素名	• 構成要素名は「DBKEY」以外とします。 • 上位レコード型で指定した一連番号を，基本項目として指定します。	K	P
		個数	上位の子レコード型で指定した一連番号の構成要素数と同じになります（階層－2）。		
3	一連番号	OCCURRENCE NUMBER 句の指定値が 1 以上の場合	• 構成要素名を「DBKEY」とします。 • 基本項目として指定します。	K	N

指定順	項目		説明	データ種別	
				1	2
		OCCURRENCE NUMBER 句の指定値が 0 の場合		N	N
4	ユーザデータの構成要素名		ユーザデータの構成要素名を指定します。	－	－
				U	D
					K
					F

（凡例）

－：該当しません。

(g) FUNCTION

FUNCTION

〔REFER USE〕

〔ADD {USE | NOUSE}〕

〔UPDATE {USE | NOUSE}〕

〔ERASE {USE | NOUSE}〕

〔ALLERASE NOUSE〕

SDB データベースへのアクセス属性を指定します。

SDB データベースへのアクセス属性は，SDBOPTION 句下の FUNCTION 句でも指定できます。RECORD 句内で指定する場合は，各 RECORD 句にアクセス属性が指定できます。指定を省略した場合は，SDBOPTION 句下の FUNCTION 句の指定値，または省略値が仮定されます。

REFER USE

レコードの検索を許可します。

常に REFER USE が仮定されます（REFER 句を省略しても，REFER USE が仮定されます）。

ADD

レコードの追加を許可するかどうかを指定します。

USE：レコードの追加を許可します。

NOUSE：レコードの追加を許可しません。

RECORD 句下の ADD 句を省略した場合，SDBOPTION 句下の ADD 句の指定値が仮定されます。RECORD 句下の ADD 句と SDBOPTION 句下の ADD 句の両方を省略した場合，USE が仮定されます。

UPDATE

レコードの更新を許可するかどうかを指定します。

USE：レコードの更新を許可します。

NOUSE：レコードの更新を許可しません。

RECORD 句下の UPDATE 句を省略した場合，SDBOPTION 句下の UPDATE 句の指定値が仮定されます。RECORD 句下の UPDATE 句と SDBOPTION 句下の UPDATE 句の両方を省略した場合，USE が仮定されます。

ERASE

レコードの削除を許可するかどうかを指定します。

USE：レコードの削除を許可します。

NOUSE：レコードの削除を許可しません。

RECORD 句下の ERASE 句を省略した場合，SDBOPTION 句下の ERASE 句の指定値が仮定されます。RECORD 句下の UPDATE 句と SDBOPTION 句下の ERASE 句の両方を省略した場合，NOUSE が仮定されます。

ALLERASE NOUSE

レコードの一括削除を許可しません。

常に ALLERASE NOUSE が仮定されます（ALLERASE 句を省略しても，ALLERASE NOUSE が仮定されます）。

(h) RECORDID

RECORDID レコード識別コード

～X'＜16 進数字＞'((1 バイト))

レコード型を識別するコードとして，レコード識別コードを指定します。

《規則》

- レコード識別コードを 1 バイトで指定します。
- X'00'以外は重複できません。

(i) SET

SET 親子集合型名

～＜識別子＞((1～30 バイト))

親子集合型名を指定します。

《規則》

- 親子集合型名は，SDB データベース定義内で一意にしてください。
- ルートレコード型の数は 1 つにしてください。
- 親レコードから子レコードへの親子集合型をたどった場合に，元のレコードにたどり着くような親子集合型は定義できません。
- 1 つの SDB データベース定義内に指定できる親子集合型数は，0～127 個です。指定できる親子集合の階層は，1～15 です。

(j) OWNER

OWNER 親レコード型名

～<識別子>((1～30 バイト))

親子集合の親レコード型名を指定します。

《規則》

親レコード型名には，RECORD 句で定義したレコード型名を指定してください。

(k) ORDER

ORDER {LAST | SORTED DUPLICATES PROHIBITED}

API によるレコードの格納時（STORE），HiRDB/SD データベース作成ユティリティ（pdsdblod），または HiRDB/SD データベースアクセスユティリティ（pdsdbexe）で，子レコード実現値を親子集合内の任意の実現値に格納する際の位置（親子集合内の子レコードへの挿入順序）を指定します。

LAST

子レコード実現値群の最後に格納します。

SORTED DUPLICATES PROHIBITED

子レコード型で指定したキー項目（KEY 句に指定した構成要素）の値に従って，昇順または降順に格納します。

同じ親レコードに対して，同じ値のキー項目を持つ子レコードが格納済みの場合，格納できません。

データ種別 1，2（TYPE U,K）を指定している場合は，SORTED DUPLICATES PROHIBITED を指定してください。それ以外の場合は，LAST を指定してください。

《規則》

1 つの SET 句の指定には，必ず ORDER 句を指定してください。

(l) MEMBER

MEMBER 子レコード型名

～<識別子>((1～30 バイト))

親子集合内の子レコード型名を指定します。

《規則》

- 子レコード型名には，RECORD 句で定義したレコード型名を指定してください。ただし，親子集合の親レコード型名と重複する子レコード型名は指定できません。
- 1 つの SET 句には，必ず MEMBER 句を指定してください。

(m) INSERTION

INSERTION AUTOMATIC

子レコードを親子集合に格納するときの挿入方法を指定します。

AUTOMATIC

子レコードのレコード実現値がデータベースに格納されると，自動的に親レコードが属する親子集合型の親子集合実現値に格納されます。

(n) RETENTION

RETENTION FIXED

子レコードを親子集合から切り離すまでの属性を指定します。

FIXED

レコードが，ある親子集合の子レコードになったレコードは，データベースから削除されるまで，その親子集合の子レコードでなければなりません。

(o) KEY

KEY {ASCENDING｜DESCENDING} 構成要素名

ORDER 句で SORTED DUPLICATES PROHIBITED を指定した親子集合の子レコードを構成要素の値に従って，昇順に格納するか，または降順に格納するかを指定します。指定した構成要素をキー項目とします。

ASCENDING

子レコードを昇順に格納します。

DESCENDING

子レコードを降順に格納します。

構成要素名

～<識別子>((1～30 バイト))

格納時のキーになる構成要素名を指定します。

《規則》

- キーの構成要素長は 254 バイト以内としてください。
- 構成要素名に，集団項目の構成要素名は指定できません。
- 指定できる構成要素は，データ種別 1 がユーザデータ（U），データ種別 2 がユーザキー（K）の構成要素だけです。
- 構成要素名には，RECORD 句で定義した構成要素名を指定してください。
- 1 つの KEY 句に構成要素名を 2 つ以上指定できません。
- ORDER 句で LAST を指定した場合，KEY 句は指定できません。
- ORDER 句で SORTED DUPLICATES PROHIBITED を指定した場合，必ず KEY 句を指定してください。

(p) SETOPTION

SETOPTION

親子集合に関するオプションを指定します。

(q) OCCURRENCE NUMBER

OCCURRENCE NUMBER 一連番号の最大値

～<符号なし整数>((0～2,147,483,647))≪30,000≫

一連番号の最大値を指定します。

0 を指定した場合，非オカレンスであることを示します。

《規則》

- 構成要素のデータ種別 1，2 の指定が N，N の場合，OCCURRENCE NUMBER 句に 0 以外を指定，または省略することはできません。
- 構成要素のデータ種別 1，2 の指定が K，N の場合，0 は指定できません。
- WARNING 句または REUSE 句を指定する場合は，この句も指定する必要があります。

(r) WARNING

WARNING {一連番号使用比率 1〔，一連番号使用比率 2〔，一連番号使用比率 3〕〕 | 0}

一連番号の使用状況を監視する場合に，一連番号の使用比率を％単位で指定します。

一連番号使用比率 1，一連番号使用比率 2，一連番号使用比率 3

～<符号なし整数>((50～99))

割り当て済み一連番号の割合が一連番号使用比率 1～3 で指定した値に達した場合※に，メッセージを 1 回ずつ出力します。0 を指定した場合は，割り当て済みの一連番号の監視をしません。

注※　監視値＝↑一連番号の最大値×一連番号使用比率n↑

この句と SDBOPTION 句下の OCCURRENCE WARNING 句の両方を指定した場合，この句での指定が有効となります。この場合，一連番号使用比率の指定値の個数が異なっていても，双方で補ってメッセージを出力することはありません。

OCCURRENCE NUMBER 句の指定が 0 で，この句を省略した場合，割り当て済みの一連番号の監視をしません。

OCCURRENCE NUMBER 句の指定が 0 以外で，この句を省略した場合，SDBOPTION 句下の OCCURRENCE WARNING 句の指定が有効となります。

SDBOPTION 句下の OCCURRENCE WARNING 句も省略した場合，割り当て済みの一連番号の監視をしません。

一度に複数の一連番号使用比率に達した場合は，対応するメッセージがすべて出力されます。例えば，WARNING 句の指定を 60，70，80 とした場合に，1 回目のレコードの格納で 75％に達した場合は，指定値 60 と 70 に対応するメッセージが出力されます。

一連番号の最大値が小さいと，重複指定でなくても内部的に重複指定と同じような扱いになる場合があります。例えば，OCCURRENCE NUMBER 句が 10，WARNING 句が 97，98，99 と定義した場合，どの指定値も割り当て済み一連番号が 10 になった時点で，97（%）と 98（%）と 99（%）に対応するメッセージが出力されます。

一連番号の監視については，「5.25　一連番号の監視」を参照してください。

《規則》

- 一連番号使用比率を複数指定する場合，指定値が昇順になるように指定してください。
- OCCURRENCE NUMBER 句との組み合わせ可否を次の表に示します。

表 11-17　SETOPTION 句下の OCCURRENCE NUMBER 句と WARNING 句の組み合わせ可否

SETOPTION 句下の OCCURRENCE NUMBER 句の指定	SETOPTION 句下の WARNING 句の指定	
	0 以外	0
0 以外	○	○
0	×	○

（凡例）

○：WARNING 句が指定できます。

×：WARNING 句は指定できません。

表 11-18　SETOPTION 句下の OCCURRENCE NUMBER 句と SDBOPTION 句下の OCCURRENCE WARNING 句の組み合わせ可否

SETOPTION 句下の OCCURRENCE NUMBER 句の指定	SDBOPTION 句下の OCCURRENCE WARNING 句の指定	
	0 以外	0
すべて 0 以外	○	○
一部 0	○	○
すべて 0	×	○

（凡例）

○：OCCURRENCE WARNING 句が指定できます。

×：OCCURRENCE WARNING 句は指定できません。

- FUNCTION 句の ADD 指定との組み合わせとレコードの追加可否を次の表に示します。

表 11-19　FUNCTION 句の ADD 指定との組み合わせとレコードの追加可否

SDB データベース種別	SDBOPTION 句下の FUNCTION 句の ADD 指定								
	省略			U			N		
	子レコードの FUNCTION 句の ADD 指定								
	省略	U	N	省略	U	N	省略	U	N
4V FMB	○	○	×	○	○	×	×	○	×

（凡例）

U：USE 指定

N：NOUSE 指定

○：レコードの指定ができます。

×：レコードの指定ができません。

- 子レコードへの追加可否との組み合わせを次の表に示します。

表 11-20　子レコードへの追加可否と WARNING 句の組み合わせ可否

子レコードへの追加	SETOPTION 句下の WARNING 句の指定	
	0 以外	0
できる	○	○
できない	×	○

（凡例）

○：WARNING 句が指定できます。

×：WARNING 句は指定できません。

表 11-21　子レコードへの追加可否と SDBOPTION 句下の OCCURRENCE WARNING 句の組み合わせ可否

子レコードへの追加	SDBOPTION 句下の OCCURRENCE WARNING 句の指定	
	0 以外	0
すべての子レコードに対してできる	○	○
一部の子レコードに対してできる	○	○
すべての子レコードに対してできない	×	○

（凡例）

○：OCCURRENCE WARNING 句が指定できます。

×：OCCURRENCE WARNING 句は指定できません。

(s)　REUSE

REUSE　{YES｜NO}

削除された一連番号を再利用するかどうかを指定します。

レコードを一括削除した場合，または全件削除した場合は，REUSE 句の指定に関係なく，最初から一連番号を振り直します。

YES

一連番号を再利用します。

一連番号を再利用するには，ERASE 句に USE を指定している必要があります。

なお，OCCURRENCE NUMBER 句に 0 を指定してもエラーにはなりません。

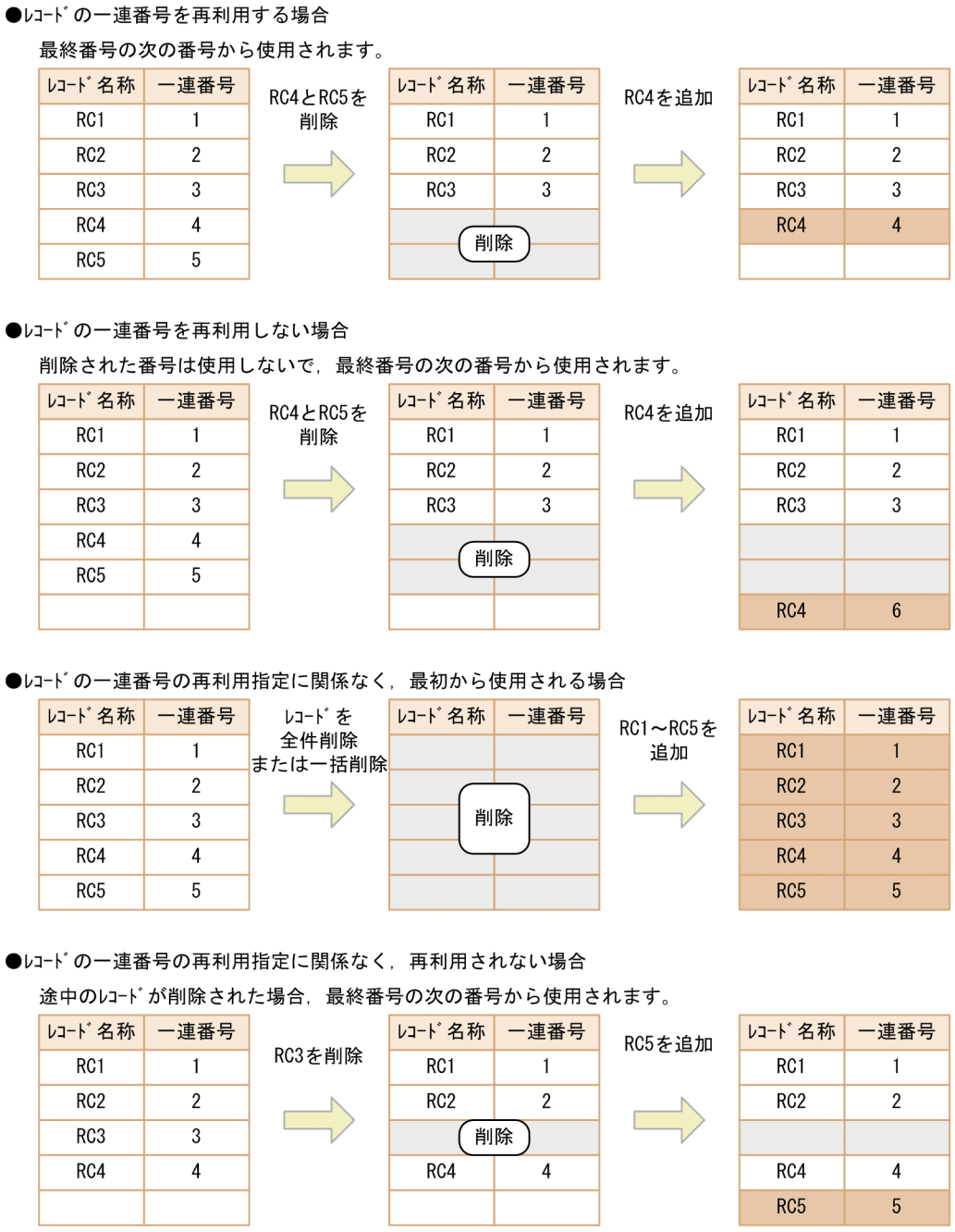
ORDER 句で SORTED DUPLICATES PROHIBITED を指定した場合は指定できません。

NO

削除された一連番号を再利用しません。

一連番号再利用の例を次の図に示します。

図 11-14　一連番号再利用の例



(t)　SDBOPTION

SDBOPTION

SDB データベースのオプション機能を定義します。

(u) FUNCTION

FUNCTION

〔REFER USE〕

〔ADD {USE | NOUSE}〕

〔UPDATE {USE | NOUSE}〕

〔ERASE {USE | NOUSE}〕

〔ALLERASE NOUSE〕

〔DBLODUTL USE〕

〔FORMAT NOUSE〕

SDB データベースへのアクセス属性を指定します。

REFER 句，ADD 句，UPDATE 句，ERASE 句，ALLERASE 句の詳細については，「11.5.1(5)(g) FUNCTION」を参照してください。

DBLODUTL USE

HiRDB/SD データベース作成ユティリティ（pdsdblod）のデータロード機能の実行を許可します。

常に DBLODUTL USE が仮定されます（DBLODUTL 句を省略しても，DBLODUTL USE が仮定されます）。

FORMAT NOUSE

HiRDB/SD データベース作成ユティリティ（pdsdblod）のデータベースへのフォーマットライト機能が使用できません。

常に FORMAT NOUSE が仮定されます（FORMAT 句を省略しても，FORMAT NOUSE が仮定されます）。

(v) OCCURRENCE WARNING

OCCURRENCE WARNING {一連番号使用比率 1〔，一連番号使用比率 2〔，一連番号使用比率 3〕〕 | 0}

一連番号の使用状況を監視する場合に，一連番号の使用比率を％単位で指定します。

一連番号使用比率 1，一連番号使用比率 2，一連番号使用比率 3

～＜符号なし整数＞((50～99))

割り当て済み一連番号の割合が，一連番号使用比率 1～3 で指定した値に達した場合に，メッセージを 1 回ずつ出力します。0 を指定した場合，またはこの句を省略した場合は，割り当て済み一連番号の監視はしません。

この指定はすべての子レコード型に対して有効となりますが，SETOPTION 句下で WARNING 句を指定した場合は，SETOPTION 句下の WARNING 句の指定が有効となります。この場合，一連番号使用比率の指定値の個数が異なっても，双方で補ってメッセージを出力することはありません。

一連番号の監視については，「5.25 一連番号の監視」を参照してください。

《規則》

- ルートレコード型だけを定義している場合は，このオペランドを指定できません。
- ルートレコード型だけの定義に対して，子レコード型を 1 個以上追加する場合は追加指定ができます。

上記以外の規則については，SETOPTION 句下の「(r)　WARNING」の規則を参照してください。

(w)　END SCHEMA

END SCHEMA

SDB データベース定義の終了を示します。

11.5.2　SDB データベース格納定義【4V FMB】

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定する SDB データベース格納定義について説明します。

注意事項

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定した SDB データベース格納定義は保存しておいてください。*ALTER DICTIONARY 文で SDB データベース格納定義を変更する際，変更個所だけを指定するのではなく，変更後の SDB データベース格納定義をすべて指定する必要があります。このとき，保存しておいた SDB データベース格納定義が必要になります。

(1)　機能

SDB データベースを格納する際の論理構造を定義します。

(2)　形式

項番	形式	オペランドの変更可否	
		レコード型の追加時※1	RD エリアの変更時※1
(a)	STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名	×	×
(b)	DBTYPE 4V	×	
(c)	STORAGE RECORD 格納レコード名	○※2	
(d)	{CLUSTERED 親子集合型名 　　| SEQUENTIAL シーケンシャルインデクス名		
(e)	FOR RECORD		
(f)	ORDER KEY ASCENDING 構成要素名〔，構成要素名〕…		

項番	形式	オペランドの変更可否	
		レコード型の追加時※1	RD エリアの変更時※1
(g)	〔PCTFREE ページ内の未使用領域の比率〕		
(h)	WITHIN {シーケンシャルインデクス用 RD エリア名 | (シーケンシャルインデクス用 RD エリア名) | ((シーケンシャルインデクス用 RD エリア名) 〔, (シーケンシャルインデクス用 RD エリア名)〕…)} }		○※3, ※4, ※6
(i)	〔SUBPAGE NUMBER サブページ分割数〕		×
(j)	〔PCTFREE {ページ内の未使用領域の比率 | ((〔ページ内の未使用領域の比率〕, セグメント内の空きページの比率)}〕		
(k)	〔DEPENDING ON 構成要素名〕		
(l)	WITHIN {格納レコード用 RD エリア名 | (格納レコード用 RD エリア名) | ((格納レコード用 RD エリア名) 〔格納条件〕 〔, (格納レコード用 RD エリア名) 格納条件〕…) | ((格納レコード用 RD エリア名) レコード型の RD エリア分割キー値 〔, (格納レコード用 RD エリア名) レコード型の RD エリア分割キー値…〕 , (格納レコード用 RD エリア名))}		○※3, ※4, ※6
(m)	〔POINTER AREA SIZE 管理領域の予備領域の長さ〕		×
(n)	〔SECONDARY INDEX 二次インデクス名		
(o)	ORDER KEY ASCENDING 構成要素名, 構成要素名〔, 構成要素名〕…		
(p)	〔PCTFREE ページ内の未使用領域の比率〕		
(q)	WITHIN {二次インデクス用 RD エリア名 | (二次インデクス用 RD エリア名) | ((二次インデクス用 RD エリア名) 〔, (二次インデクス用 RD エリア名)〕…)}		○※3, ※4, ※6
	〕…		
	…		
(r)	〔SET 親子集合型名		×
(s)	OWNER POINTER FOR {FIRST LAST | FIRST LAST USER} MEMBER		
(t)	MEMBER POINTER FOR NEXT PRIOR TENANT		

項番	形式	オペランドの変更可否	
		レコード型の追加時※1	RD エリアの変更時※1
	〕…		
(u)	〔SDBOPTION	×	
(v)	〔KEYDEF 構成要素名		
(w)	DATA データベースのキー値 …		○※5
	〕…		
	〕		
(x)	END STORAGE SCHEMA	×	×

（凡例）

○：*ALTER DICTIONARY 文の実行時に変更できるオペランドです。

×：*ALTER DICTIONARY 文の実行時に変更できないオペランドです。

注

- 定義句は，形式で記述している順序に従って指定します。必須の定義と省略できる定義の間は形式で記載されているとおりに指定してください。
- *ALTER DICTIONARY 文で子レコード型を追加した場合，POINTER AREA SIZE 句に指定していた管理領域の予備領域の長さが自動的に変更されることがあります。変更後の値については，KFPB61784-I メッセージに出力されます。

注※1

*ALTER DICTIONARY 文実行時のオペランド追加可否です。

注※2

- 最下位の子レコード型の追加とその親子集合の追加ができます。
- STORAGE RECORD 句と SET 句は必ず両方指定してください。どちらか片方だけを指定することはできません。
- 既存の格納レコード型の定義は変更できません。

注※3

格納レコード用 RD エリアとインデクス用 RD エリアはペアで変更します。

1 つの*ALTER DICTIONARY 文で実行できる格納レコード用 RD エリアの変更を次の表に示します。

表 11-22　1 つの*ALTER DICTIONARY 文で実行できる格納レコード用 RD エリアの変更

格納レコード用 RD エリア変更内容	横分割の方法	
	格納条件指定	境界値指定
1 つの RD エリアを追加または削除	○	−

格納レコード用 RD エリア変更内容	横分割の方法	
	格納条件指定	境界値指定
複数の RD エリアを追加または削除	○	−
1 つの RD エリアを複数の RD エリアに分割	○	○
複数の RD エリアを複数の RD エリアに分割	×	×
複数の RD エリアを 1 つの RD エリアに統合	○	○
複数の RD エリアを複数の RD エリアに統合	×	×
格納 RD エリア数を変更しないで，格納条件だけを追加，更新，または削除	×	×

（凡例）

○：実行できます。

×：実行できません。

−：該当しません。

なお，次の操作はできません。

- 1 つの RD エリアに格納していたレコード型を，複数の RD エリアに分割格納する変更
- 複数の RD エリアに分割格納していたレコード型を，1 つの RD エリアに格納する変更
- 横分割方法の変更

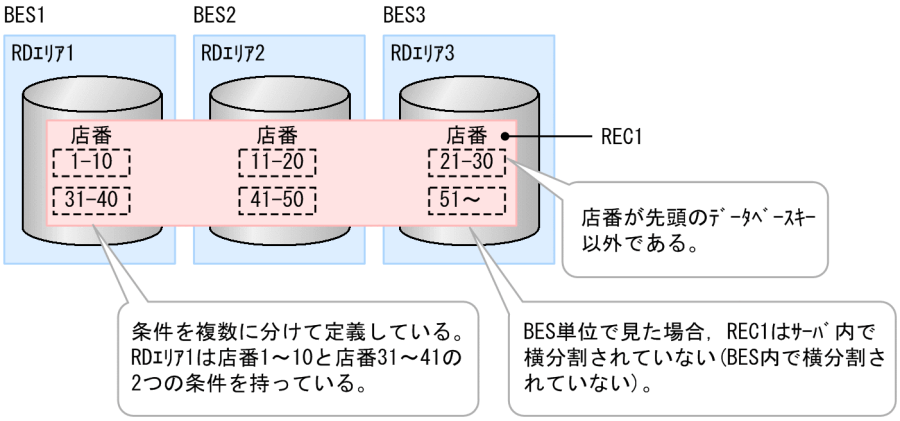
  格納条件指定で分割していた定義を，境界値指定の分割に変更することはできません。また，その逆もできません。
- 1 つの RD エリアを複数の RD エリアに分割する際，分割元の RD エリアを削除する変更
- 複数の RD エリアを 1 つの RD エリアに統合する際，統合先の RD エリアに統合元の RD エリアを使用しない変更

注※4

次の条件をすべて満たす場合，RD エリアの追加，削除はできません。

- 境界値分割指定である
- レコード型がサーバ内で横分割されていない
- レコード型の先頭のデータベースキーのデータ種別が K，A 以外である
- 同一 RD エリアに格納する条件を複数に分けて定義している

例を次に示します。



［説明］

- REC1 は親レコードを示しています。
- 店番は，REC1 の分割キーを示しています。
- REC1 は，3 つの BES および 3 つの RD エリアに境界値分割されています。

注※5

- RD エリアの追加および境界値分割の RD エリアの分割の場合に，レコード型の RD エリア分割キーの追加ができます。
- RD エリアの削除および境界値分割の RD エリアの統合の場合に，レコード型の RD エリア分割キーの削除ができます。

注※6

RD エリアの定義を変更した場合，変更対象の RD エリアに格納されているデータが削除されることがあります。RD エリアの定義変更時のデータの削除有無を次の 2 つの表に示します。

表 11-23　RD エリアの定義変更時のデータの削除有無（格納条件指定の横分割の場合）

分割格納条件の変更の種類	RD エリアごとの変更内容	データの削除有無
追加	追加	×
	格納条件を変更しない	×
削除	削除	○
	格納条件を変更しない	×
分割	追加	×
	格納条件を一部削除※2	○※1
	格納条件を変更しない	×
統合	格納条件を一部追加※3	×
	削除	○
	格納条件を変更しない	×

（凡例）

○：データが削除されます。

×：データは削除されません。

注※1

同期点処理が行われます。

注※2

分割元の RD エリアです。

注※3

統合先の RD エリアです。

表 11-24　RD エリアの定義変更時のデータの削除有無（境界値指定の横分割の場合）

分割格納条件の変更の種類	RD エリアごとの変更内容		データの削除有無
分割	追加		×
	境界値の範囲を変更	境界値の範囲を一部追加※2	×
		境界値の範囲を一部削除※3	○※1
	境界値の範囲を変更しない		×
統合	境界値の範囲を変更	境界値の範囲を一部追加※4	×
		境界値の範囲を一部削除※5	○※1
	削除		○
	境界値の範囲を変更しない		×

（凡例）

○：データが削除されます。

×：データは削除されません。

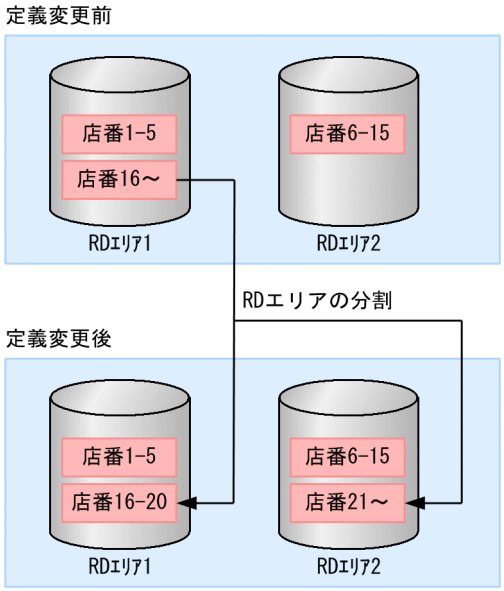
注※1

同期点処理が行われます。

注※2

すでに使用している RD エリアに対して，レコード型の RD エリア分割キー値の格納範囲を追加するケースが該当します。

（例）



上記の例の場合，RD エリア 2 がこのケースに該当します。RD エリア 1 の境界値の範囲のうち，店番 21～を RD エリア 2 に分割しています。

注※3

分割元の RD エリアです。

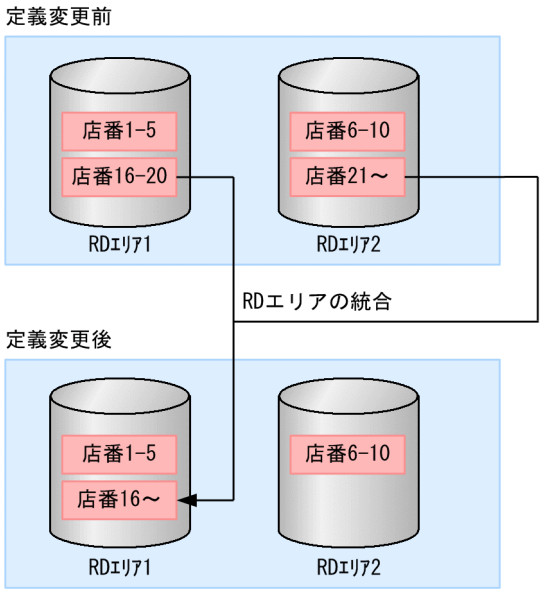
注※4

統合先の RD エリアです。

注※5

統合元の RD エリアのうち，レコード型の RD エリア分割キー値の格納範囲の一部が削除される RD エリアが該当します。

（例）



上記の例の場合，RD エリア 2 がこのケースに該当します。RD エリア 2 の境界値の範囲のうち，店番 21～を RD エリア 1 に統合しています。

(3) オペランドの説明

SDB データベース格納定義の各オペランドの指定内容を説明します。

(a) STORAGE SCHEMA

STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名

SDB データベース格納名と対応する SDB データベース名を指定します。

SDB データベース格納名

～<識別子>((1～30 バイト))

SDB データベース格納名を指定します。

すでに使用している SDB データベース格納名は指定できません。

SDB データベース名

～<識別子>((1～30 バイト))

SDB データベース名を指定します。

SDB データベース名に一致する名称が SDB データベース定義にない場合は，エラーとなります。

(b) DBTYPE

DBTYPE 4V

SDB データベース種別が 4V FMB の SDB データベースを定義します。

注意事項

SDB データベース種別が 4V（4V FMB または 4V AFM）の SDB データベースと SD FMB の SDB データベースをシステム内で混在して定義することはできません。混在させた場合，動作を保証しません。

(c) STORAGE RECORD

STORAGE RECORD 格納レコード名

格納レコード名

～<識別子>((1～30 バイト))

SDB データベース定義の RECORD 句で指定したレコード型名を，格納レコード名として指定します。

《規則》

- SDB データベース定義の RECORD 句で指定したレコード型名と同じ個数，同じ順番で指定してください。
- すでに使用している格納レコード名は指定できません。ただし，格納レコードの所有者（スキーマ）が異なる場合は，同じ格納レコード名を指定できます。

(d) CLUSTERED，SEQUENTIAL

CLUSTERED 親子集合型名｜SEQUENTIAL シーケンシャルインデクス名

格納するレコードの配置方法を指定します。

CLUSTERED 親子集合型名

親子集合型名

～<識別子>((1～30 バイト))

すべての子レコードを親レコードと同じ RD エリアに格納する場合に指定します。

《規則》

- STORAGE RECORD 句に指定したレコード型ごとに 1 つ指定できます。STORAGE RECORD 句に指定したレコード型が，ルートレコード型の場合は指定できません。
- SDB データベース格納定義の SET 句に指定する親子集合型名と同じ個数，同じ順番で指定してください。

SEQUENTIAL シーケンシャルインデクス名

インデクスのキーの値の順序に従って，親レコード実現値の格納位置を決めるときに指定します。SEQUENTIAL 句は，ルートレコード型に必ず指定します。

シーケンシャルインデクス名

～<識別子>((1～30 バイト))

STORAGE RECORD 句に指定したルートレコード型に対して 1 つ指定します。STORAGE RECORD 句に指定したレコード型が，ルートレコード型以外の場合は指定できません。

《規則》

- シーケンシャルインデクス名には，二次インデクス名とは別の名称を指定してください。
- シーケンシャルインデクス名は，同一スキーマ内で一意になるように指定してください。二次インデクス名も含めて一意になるようにしてください。

(e) FOR RECORD

FOR RECORD

レコード型単位にシーケンシャルインデクスを作成します。

キーの値が同じ場合，シーケンシャルインデクスへの格納を禁止します。

(f) ORDER KEY

ORDER KEY ASCENDING 構成要素名〔，構成要素名〕…

インデクスのキーとなる構成要素名を指定します。

指定した構成要素名をキーとして，昇順にシーケンシャルインデクスが作成されます。

ASCENDING

昇順にシーケンシャルインデクスを作成します。

構成要素名

～<識別子>((1～30 バイト))

インデクスのキーに指定する構成要素を次に示すように指定してください。

- SDB データベース定義内のルートレコード型のデータベースキーに指定した基本項目の構成要素名を，定義順にすべて指定します。ルートレコード型の構成要素の指定については，「表 11-14　ルートレコード型の構成要素の指定」を参照してください。SDB データベースの定義例については，「11.11.1　SDB データベースの定義例（4V FMB の場合）」を参照してください。

《規則》

- 構成要素名は，格納レコード名と対応する SDB データベース定義のレコード型名で指定した構成要素名としてください。
- シーケンシャルインデクス作成時に指定できる構成要素名の条件について，次の表に示します。

表 11-25　シーケンシャルインデクス作成時に指定できる構成要素名の条件

項目	内容
指定できる構成要素名の数	最大 7 個
インデクスのキー長の合計	255 バイト以内 インデクスのキーに指定する構成要素とキー長の関係については，「表 11-26　インデクスのキーに指定する構成要素とキー長の関係」を参照してください。
構成要素名	基本項目の構成要素名 集団項目が含まれる構成要素名は指定できません。

表 11-26　インデクスのキーに指定する構成要素とキー長の関係

データ型	インデクスのキー長（単位：バイト）	
	構成要素名の指定が 1 つの場合	構成要素名の指定が複数の場合
CHARACTER	SDB データベース定義で指定したデータ長	SDB データベース定義で指定したデータ長＋制御部（1 バイト）
XCHARACTER		
PACKED ［DECIMAL FIXED］ 整数部桁数［，小数部桁数］	↓（整数部桁数＋小数部桁数）÷2↓＋1	↓（整数部桁数＋小数部桁数）÷2↓＋2
INTEGER	4	5

(g)　PCTFREE

PCTFREE ページ内の未使用領域の比率

～<符号なし整数>((0～99))≪30≫

インデクスページ内の未使用領域の比率を指定します。

(h) WITHIN

WITHIN {シーケンシャルインデクス用 RD エリア名
| (シーケンシャルインデクス用 RD エリア名)
| ((シーケンシャルインデクス用 RD エリア名)
〔,(シーケンシャルインデクス用 RD エリア名)〕…)}

シーケンシャルインデクス用 RD エリア名

～< RD エリア名>((1～30 バイト))

シーケンシャルインデクスを格納する RD エリア名を指定します。

1 つの SEQUENTIAL 句に対して，WITHIN 句を必ず 1 つ指定してください。

《規則》

シーケンシャルインデクスを複数の RD エリアに格納する場合の規則

- シーケンシャルインデクス用 RD エリアとして指定できる RD エリアの数と SDB データベースの横分割の関係を次の表に示します。

表 11-27 シーケンシャルインデクス用 RD エリアとして指定できる RD エリア数と SDB データベースの横分割の関係

SDB データベースの横分割方法	指定できる RD エリア数
分割指定なし	1
格納条件分割	1～256
境界値分割	2～3,000（1,024）※

注※

WITHIN 句に指定できる RD エリア名の延べ数の最大値は 3,000 になります。1,024 は，WITHIN 句に RD エリア名を重複して指定する場合，RD エリア名の重複を排除して数えたときの最大値です。

- シーケンシャルインデクス用 RD エリアは，格納レコード用 RD エリアと同じ数を指定する必要があります。
- 1 つの WITHIN 句の中では，シーケンシャルインデクス用 RD エリア名は一意に指定してください。なお，境界値分割指定の場合，隣接する RD エリア名が重複している場合だけエラーとなります。
- 境界値分割指定の格納レコード用 RD エリアを重複して指定している場合は，シーケンシャルインデクス用 RD エリアも重複して指定してください。シーケンシャルインデクス用 RD エリアを重複して指定する場合の例を次の表に示します。

表 11-28 シーケンシャルインデクス用 RD エリアを重複して指定する場合の例

指定	格納レコード用 RD エリアの指定	シーケンシャルインデクス用 RD エリアの指定
RD エリアの重複なし	((RDAREAX) X'00FF', (RDAREA1) X'01FF', (RDAREA2) X'03FF',	((RDAREAX_I), (RDAREA1_I), (RDAREA2_I),

指定	格納レコード用 RD エリアの指定	シーケンシャルインデクス用 RD エリアの指定
	(RDAREA3))	(RDAREA3_I))
RD エリアの重複あり	((RDAREAX) X'00FF', (RDAREA1) X'01FF', (RDAREA2) X'03FF', (RDAREA1))	((RDAREAX_I), (RDAREA1_I), (RDAREA2_I), (RDAREA1_I))

- 格納レコード用 RD エリアと対応するシーケンシャルインデクス用 RD エリアは，同じサーバ内の RD エリアである必要があります。

指定するシーケンシャルインデクス用 RD エリアに関する規則

- 格納レコード用 RD エリアとして指定した RD エリアは指定できません。
- シーケンシャルインデクス用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- シーケンシャルインデクス用 RD エリアとして指定する RD エリアには，1 つの RD エリアにシーケンシャルインデクスと二次インデクスを合わせて，インデクスを最大 500 個指定できます。
- インナレプリカ機能を使用する場合，指定する RD エリアはインナレプリカ機能の使用条件を満たしている必要があります。インナレプリカ機能の使用条件については，「付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】」を参照してください。
- シーケンシャルインデクス用 RD エリアに定義できるリソースの組み合わせを次の表に示します。

表 11-29　RD エリアに定義できるリソースの組み合わせ

定義するリソース		RD エリアの定義		
		レコード型・インデクス定義なし	SDB データベース格納定義 A	
			レコード型定義あり	インデクス定義あり
SDB データベース格納定義 A	レコード型	○	○	×
	インデクス	○	×	○
SDB データベース格納定義 B	レコード型	○	×	×
	インデクス	○	×	○

（凡例）

○：定義できます。

×：定義できません。

(i)　SUBPAGE NUMBER

SUBPAGE NUMBER サブページ分割数

サブページ分割数

～<符号なし整数>((2～16))

1 データページ当たりのサブページ分割数を指定します。SUBPAGE NUMBER を省略した場合，サブページ分割されません。サブページ分割数の設計方針については，「3.2.4　サブページの設計」を参照してください。

《規則》

- ルートレコード型，およびすべての子レコード型で同じサブページ分割数を指定してください。
- サブページ分割数を指定する場合，格納レコード用 RD エリアのページ長が，4096 バイトである必要があります。

(j)　PCTFREE

PCTFREE　{ページ内の未使用領域の比率

｜（〔ページ内の未使用領域の比率〕，セグメント内の空きページの比率)}

ページ内の未使用領域の比率

～＜符号なし整数＞((0～99))

セグメント内の空きページの比率

～＜符号なし整数＞((0～50))

データベースの初期作成時に設定するページ内の未使用領域の比率を指定します。ページ内の未使用領域の比率は，0～99（%）の範囲で指定できます。ページ内の未使用領域の比率を省略すると，30（%）が仮定されます。

SUBPAGE NUMBER 句でサブページ分割数を指定している場合，ページ内の未使用領域の比率は，サブページ内の未使用領域の比率になります。例えば，30 を指定した場合，サブページ内の未使用領域の比率が 30%になります。

また，セグメント内の空きページの比率を 0～50（%）の範囲で指定できます。セグメント内の空きページの比率を省略すると，10（%）が仮定されます。

《規則》

ルートレコード型，およびすべての子レコード型で同じ値を指定してください。

(k)　DEPENDING ON

DEPENDING ON 構成要素名

構成要素名

～＜識別子＞((1～30 バイト))

SDB データベースを境界値で複数の RD エリアに横分割して格納する場合の，境界値分割のキーとなる構成要素名を指定します。

《規則》

- ルートレコード型にだけ指定できます。
- CLUSTERED 句を指定した格納レコードに対しては指定できません。

- 格納レコード名と対応する SDB データベース定義のレコード型名で指定した構成要素名である必要があります。
- 構成要素名には，データ種別 1 が K，データ種別 2 が A の構成要素名を指定します。
- SDB データベースの横分割方法が格納条件指定の場合は指定できません。

(l) WITHIN

WITHIN　{格納レコード用 RD エリア名
　｜（格納レコード用 RD エリア名）
　｜（（格納レコード用 RD エリア名）　〔格納条件〕
〔,（格納レコード用 RD エリア名）　格納条件〕…）
　｜（（格納レコード用 RD エリア名）　レコード型の RD エリア分割キー値
〔,（格納レコード用 RD エリア名）　レコード型の RD エリア分割キー値…〕
,（格納レコード用 RD エリア名））}

格納レコード用 RD エリア名

～< RD エリア名>((1～30))

格納条件

～構成要素名=（レコード型の RD エリア分割キー値〔,レコード型の RD エリア分割キー値〕…）

レコード型の RD エリア分割キー値

～X'< 16 進数字>'((1～30 バイト))

～'<英数字または下線（_）>'((1～30 バイト))※1

～< 10 進数定数>※2

～<整数定数>※3

注※1

先頭に下線（_）は指定できません。

注※2

10 進数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

整数部と小数部にすべて 0 を指定する場合，符号に-を指定しないでください。-0.を指定しても 0.として扱います。

なお，10 進数定数の値の範囲を超える値を指定できません。

注※3

整数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

-0 を指定しても 0 として扱います。

なお，整数定数の値の範囲を超える値を指定できません。

レコード実現値を格納する RD エリア名を指定します。

1 つの STORAGE RECORD 句に対して，WITHIN 句を必ず 1 つ指定してください。

データ型ごとのレコード型の RD エリア分割キー値の指定形式を次の表に示します。

表 11-30　データ型ごとのレコード型の RD エリア分割キー値の指定形式

レコード型の RD エリア分割キーのデータ型	レコード型の RD エリア分割キー値の指定形式			
	16 進数字	英数字，または下線 (_)	10 進数定数	整数定数
CHARACTER	○	○	×	×
XCHARACTER	○	○	×	×
PACKED ［DECIMAL FIXED］	×	×	○	×
INTEGER	×	×	×	○

（凡例）

○：指定できます。

×：指定できません。

《規則》

レコードを複数の RD エリアに格納する場合の規則

- 1 レコード型に対して指定できる格納レコード用 RD エリアの数と SDB データベースの横分割の関係を次の表に示します。

表 11-31　レコード型に対して指定できる格納レコード用 RD エリアの数と SDB データベースの横分割の関係

SDB データベースの横分割方法	指定できる RD エリア数
分割指定なし	1
格納条件分割	1～256
境界値分割	2～3,000（1,024）※

注※

WITHIN 句に指定できる RD エリア名の延べ数の最大値は 3,000 になります。1,024 は，WITHIN 句に RD エリア名を重複して指定する場合，RD エリア名の重複を排除して数えたときの最大値です。

- 格納条件分割で格納レコード用 RD エリア名を複数指定した場合は，すべての格納レコード用 RD エリア名に格納条件を指定してください。
- 同一 SDB データベース格納定義内の子レコード型に指定する格納レコード用 RD エリア名は，ルートレコード型と同じ数，同じ順番ですべて指定する必要があります。
- 1 つの WITHIN 句の中では，格納レコード用 RD エリア名は一意に指定してください。なお，境界値分割指定の場合，隣接する RD エリア名が重複している場合だけエラーとなります。
- レコード型の RD エリア分割キー値に指定できる範囲を次の表に示します。

表 11-32　レコード型の RD エリア分割キー値に指定できる範囲

レコード型の RD エリア分割キー値の指定形式	SDB データベースの横分割方法	レコード型の RD エリア分割キー値	
		最小	最大
16 進数字	格納条件分割	X'00…00'	X'FF…FF'
	境界値分割	X'00…00'	X'FF…FE'
10 進数定数	格納条件分割	-9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の負の数）	9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の正の数）
	境界値分割	-9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の負の数）	9…9.9…8 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数のうち，最後の桁を除くすべての桁が'9'かつ最後の桁が'8'の正の数）
整数定数	格納条件分割	−2147483648	2147483647
	境界値分割	−2147483648	2147483646

指定する格納レコード用 RD エリアに関する規則

- インデクスを格納する RD エリアは指定できません。
- 格納レコード用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- 指定する RD エリア名には，格納レコード長が当該 RD エリアのページ長（サブページ分割をしている場合はサブページ長）に収まる RD エリア名を指定してください。
- インナレプリカ機能を使用する場合，指定する RD エリアはインナレプリカ機能の使用条件を満たしている必要があります。インナレプリカ機能の使用条件については，「付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】」を参照してください。
- 格納レコード用 RD エリアに定義できるリソースの組み合わせについては，「表 11-29　RD エリアに定義できるリソースの組み合わせ」を参照してください。

レコードの格納条件に関する規則

- 格納条件は，ルートレコード型にだけ指定できます。

格納条件に指定する構成要素に関する規則

- 1 個の WITHIN 句の中では，単一の構成要素名を指定してください。複数種類の構成要素名は指定できません。
- 構成要素名には，SDB データベース定義で指定した構成要素名を指定してください。
- 構成要素名には，データ種別 1 が K，データ種別 2 が A の構成要素名を指定してください。

格納条件のレコード型の RD エリア分割キー値に関する規則

- 1 つの WITHIN 句に対するレコード型の RD エリア分割キー値は，最大 15,000 個指定できます。
- 1 つの WITHIN 句の中で，同じレコード型の RD エリア分割キー値は重複して指定できません。
- レコード型の RD エリア分割キーのデータ型が CHARACTER，または XCHARACTER の場合，指定したレコード型の RD エリア分割キー値の長さは，SDB データベース定義で指定した格納条件の構成要素の長さと一致させる必要があります。
- レコード型の RD エリア分割キーのデータ型が PACKED の場合，レコード型の RD エリア分割キーの整数部桁数，および小数部桁数の範囲内の値を指定してください。

境界値分割時（DEPENDING ON 句指定時）のレコード型の RD エリア分割キー値に関する規則

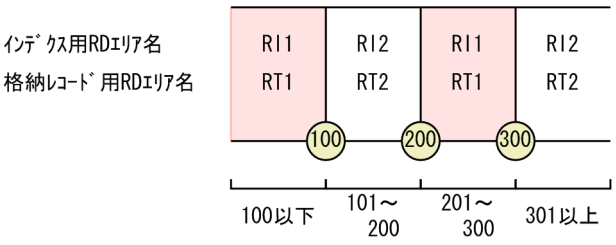
- 最初に境界値を指定する RD エリアには，指定する境界値以下の値のレコードが格納されます。

  2 番目以降（最後は除く）に指定する RD エリアには，その前に指定した境界値より大きく，かつその後に指定する境界値以下の値のレコードが格納されます。

  最後に指定する RD エリアには，その前に指定した境界値より大きい値のレコードが格納されます。

  境界値分割時の SDB データベースの横分割と SDB データベース格納定義の例を次の図に示します。

  図 11-15　境界値分割時の SDB データベースの横分割と SDB データベース格納定義の例



＜定義例＞

```
STORAGE SCHEMA SCM01 FOR SCM01
  DBTYPE 4V
  STORAGE RECORD RECA
    SEQUENTIAL IDX1 FOR RECORD
          :
      WITHIN ((RI1),(RI2),(RI1),(RI2))
    DEPENDING ON C1
      WITHIN ((RT1)100,(RT2)200,(RT1)300,(RT2))
          :
END STORAGE SCHEMA
```

- 境界値分割時には，格納条件の指定はできません（レコード型の RD エリア分割キー値の指定をする必要があります）。
- レコード型の RD エリア分割キーのデータ型が CHARACTER，または XCHARACTER の場合，レコード型の RD エリア分割キー値の長さは，DEPENDING ON 句で指定した構成要素の長さと一致させる必要があります。

- レコード型の RD エリア分割キー値は重複して指定できません。
- レコード型の RD エリア分割キー値は昇順に指定する必要があります。
- レコード型の RD エリア分割キー値を 16 進数字，英数字または下線（_）の形式で指定する場合，レコード型の RD エリア分割キー値の長さが 1 つ目の長さと異なるとエラーとなります。
- レコード型の RD エリア分割キーのデータ型が PACKED の場合，レコード型の RD エリア分割キーの整数部桁数，および小数部桁数の範囲内の値を指定してください。

参考

境界値分割を指定している場合，WITHIN 句では隣接する同一 RD エリア名に対する境界値の指定ができません。このため，TMS-4V/SP の境界値リストに相当する情報を KEYDEF 句下の DATA 句で指定できます。詳細については，「11.5.2(3)(v)　KEYDEF」を参照してください。

(m)　POINTER AREA SIZE

POINTER AREA SIZE 管理領域の予備領域の長さ

管理領域の予備領域の長さ

～<符号なし整数>((0～16,384))≪0≫

親レコードで管理している子レコード用の管理領域の予備領域の長さを指定します。

管理領域では，ポインタの情報や一連番号などを管理しています。

将来の親子集合の拡張を考慮して，親子集合 1 個当たり，約 40 バイトの予備領域を指定しておくことを推奨します。

(n)　SECONDARY INDEX

SECONDARY INDEX 二次インデクス名

二次インデクス名

～<識別子>((1～30 バイト))

二次インデクス名を指定します。

《規則》

- 二次インデクスはルートレコード型には指定できません。
- 二次インデクス名は，同一スキーマ内で一意になるように指定してください。シーケンシャルインデクス名も含めて一意になるようにしてください。
- 二次インデクス名には，シーケンシャルインデクス名とは別の名称を指定してください。
- SECONDARY 句は，1 つの STORAGE RECORD 句内で 2 つまで指定できます。詳細については，「表 11-33　二次インデクスの定義規則」を参照してください。

(o) ORDER KEY

ORDER KEY ASCENDING 構成要素名，構成要素名〔，構成要素名〕…

構成要素名

～＜識別子＞((1～30 バイト))

二次インデクスのキーとなる構成要素名を指定します。

ORDER KEY 句については，「11.5.2(3)(f) ORDER KEY」を参照してください。

《規則》

- 1 つの STORAGE RECORD 内に複数のインデクス（シーケンシャルインデクス，および二次インデクス）を定義する場合，すべての構成要素名の指定順序が同一のインデクスを複数定義できません。
- 二次インデクスについては，「表 11-33 二次インデクスの定義規則」に従って定義してください。
- キー項目を含む場合，SDB データベース定義の KEY 句に ASCENDING を指定してください。
- 1 つの二次インデクスに指定できる構成要素の数は，2～16 個です。
- 二次インデクスの構成要素の指定順序は，次の表の「指定する構成要素」の記述順に従ってください。

表 11-33 二次インデクスの定義規則

レコード型種別	一連番号最大値※1	ユーザキー※2	二次インデクスの指定	指定する構成要素
ルートレコード型	－	－	×	－
ルートレコード型以外	1 以上	なし	0～1 本	データベースキー 1～7，上位の一連番号群※3，一連番号
		あり	0～2 本	• データベースキー 1～7，上位の一連番号群※3，一連番号 • データベースキー 1～7，上位の一連番号群※3，ユーザキー
	0	－	×	－

（凡例）

×：指定できません。

－：該当しません。

注※1

SDB データベース定義の OCCURRENCE NUMBER 句の指定値，または省略値です。

注※2

SDB データベース定義の TYPE 句のデータ種別 1 に U，データ種別 2 に K を指定した構成要素です。

注※3

二次インデクスを指定したレコード型に含まれる，すべての上位の一連番号（データ種別 1 が K，かつデータ種別 2 が P の構成要素）を指定してください。なお，上位の一連番号は存在しない場合があります。

二次インデクスに指定する構成要素の定義例を次に示します。

- 上位の一連番号なしで，ユーザキーなしの場合

  SDB データベース定義

```
                              :
RECORD RECB
    2 DBKEY_L2  XCHARACTER  1  TYPE K,L
    2 DBKEY_L3  XCHARACTER  1  TYPE K,L
    2 DBKEY_L4  CHARACTER   3  TYPE K,L
    2 DBKEY     INTEGER        TYPE K,N
    2 USERDB0   CHARACTER   20 TYPE U,D
                              :
```

  SDB データベース格納定義

```
                              :
STORAGE RECORD RECB
    CLUSTERED SET1
    WITHIN  FMBX01
    SECONDARY INDEX  RECB_SI1
      ORDER KEY ASCENDING  DBKEY_L2, DBKEY_L3, DBKEY_L4, DBKEY
                              :
```

- 上位の一連番号あり，ユーザキーありで，ユーザキーに対する二次インデクスだけ指定する場合

  SDB データベース定義

```
                              :
RECORD RECD
    2 DBKEY_L2  XCHARACTER  1  TYPE K,L
    2 DBKEY_L3  XCHARACTER  1  TYPE K,L
    2 DBKEY_L4  CHARACTER   3  TYPE K,L
    2 DBKEY_P1  INTEGER        TYPE K,P
    2 DBKEY_P2  INTEGER        TYPE K,P
    2 DBKEY     INTEGER        TYPE K,N
    2 USERDD0   CHARACTER   10 TYPE U,K
    2 USERDB1   CHARACTER   20 TYPE U,D
                              :
```

  SDB データベース格納定義

```
                              :
STORAGE RECORD RECD
    CLUSTERED SET2
    WITHIN  FMBX01
    SECONDARY INDEX  RECD_SI1
      ORDER KEY ASCENDING  DBKEY_L2, DBKEY_L3, DBKEY_L4,
                           DBKEY_P1, DBKEY_P2, USERDD0
                              :
```

(p) PCTFREE

PCTFREE ページ内の未使用領域の比率

PCTFREE 句については，「11.5.2(3)(g) PCTFREE」を参照してください。

(q) WITHIN

WITHIN {二次インデクス用 RD エリア名
| (二次インデクス用 RD エリア名)
| ((二次インデクス用 RD エリア名)〔,(二次インデクス用 RD エリア名)〕…)}

二次インデクス用 RD エリア名

～<RD エリア名>((1～30 バイト))

二次インデクスを格納する RD エリア名を指定します。

1 つの SECONDARY INDEX 句で，WITHIN 句を必ず 1 つ指定してください。

《規則》

二次インデクスを複数 RD エリアに格納する場合の規則

- 1 つの二次インデクスに指定できる二次インデクス用 RD エリアの個数は，シーケンシャルインデクス用 RD エリアの個数と同じにしてください。シーケンシャルインデクス用 RD エリアの個数については，「表 11-27　シーケンシャルインデクス用 RD エリアとして指定できる RD エリア数と SDB データベースの横分割の関係」を参照してください。
- 二次インデクス用 RD エリアは，格納レコード用 RD エリアと同じ数を指定する必要があります。
- 1 つの WITHIN 句の中では，二次インデクス用 RD エリア名は一意に指定してください。なお，境界値分割指定の場合，隣接する RD エリア名が重複しているときだけエラーとなります。
- 境界値分割指定の格納レコード用 RD エリアを重複して指定している場合は，二次インデクス用 RD エリアも重複して指定する必要があります。
- 格納レコード用 RD エリアと対応する二次インデクス用 RD エリアは，同じサーバ内の RD エリアである必要があります。

指定する二次インデクス用 RD エリアに関する規則

- 格納レコード用 RD エリア名として指定した RD エリアは指定できません。
- 二次インデクス用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- 二次インデクス用 RD エリアとして指定する RD エリアには，1 つの RD エリアにシーケンシャルインデクスと二次インデクスを合わせて，インデクスを最大 500 個指定できます。
- インナレプリカ機能を使用する場合，指定する RD エリアはインナレプリカ機能の使用条件を満たしている必要があります。インナレプリカ機能の使用条件については，「付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】」を参照してください。
- 二次インデクス用 RD エリアに定義できるリソースの組み合わせについては，「表 11-29　RD エリアに定義できるリソースの組み合わせ」を参照してください。

(r) SET

SET 親子集合型名

～<識別子>((1～30 バイト))

親レコード型，子レコード型の親子集合型名を指定します。

《規則》

SDB データベース定義の SET 句で指定した親子集合型名と同じ個数，同じ順番で指定してください。

(s) OWNER POINTER FOR

OWNER POINTER FOR {FIRST LAST | FIRST LAST USER} MEMBER

親レコードに持たせるポインタを指定します。

FIRST LAST

親レコードに FIRST ポインタ，および LAST ポインタを持たせる場合に指定します。

FIRST LAST USER

親レコードに FIRST ポインタ，LAST ポインタ，および USER ポインタを持たせる場合に指定します。

(t) MEMBER POINTER FOR

MEMBER POINTER FOR NEXT PRIOR TENANT

子レコードに持たせるポインタを指定します。

NEXT PRIOR

子レコードに NEXT ポインタ，および PRIOR ポインタを持たせる場合に指定します。

(u) SDBOPTION

SDBOPTION

SDB データベース格納定義に関するオプションを指定します。

(v) KEYDEF

KEYDEF 構成要素名

構成要素名

～<識別子>((1～30 バイト))

キーの項目である構成要素名を指定します。

KEYDEF 句の用途を次に示します。

- 不正データをチェックするために指定できます。

- 境界値分割を指定している場合，横分割の指定方法が TMS-4V/SP と異なるため，TMS-4V/SP の境界値リストに相当する情報を返却するために利用できます。この場合（境界値分割の構成要素名を指定した場合），不正データのチェックはされません。

レコード型の RD エリア分割キーとして，シーケンシャルインデクスの先頭以外の構成要素を指定したレコード型に対して，RD エリアを指定した検索を行う場合，レコード型の RD エリア分割キーに指定した構成要素より前の構成要素をすべて KEYDEF 句に指定する必要があります。

《規則》

- KEYDEF 句は省略できます。
- 構成要素名は，SDB データベース定義のルートレコード型でデータベースキーとして定義されている必要があります。このため，1 つの SDB データベース格納定義で指定できる KEYDEF 句の最大数は，SDB データベース定義のルートレコード型で定義できるデータベースキーの基本項目数となります。
- KEYDEF 句を複数指定する場合は，SDB データベース定義の定義順に構成要素名を指定してください。
- KEYDEF 句を指定した場合は，DATA 句を指定する必要があります。
- SDB データベース種別と KEYDEF 句の指定の組み合わせについては，次の表を参照してください。

表 11-34　SDB データベース種別と KEYDEF 句の指定の組み合わせ

SDB データベース種別	KEYDEF 句，DATA 句の指定	用途
4V FMB	△※1	次の場合に使用します。 • 不正データのチェック時 • 境界値分割時（DEPENDING ON 句指定時）に TMS-4V/SP の境界値リスト相当の情報を取得する場合※2

（凡例）

△：省略できます。

注※1

レコード型の RD エリア分割キーとして，シーケンシャルインデクスの先頭以外の構成要素を指定したレコード型に対して，RD エリアを指定した検索を行う場合，レコード型の RD エリア分割キーに指定した構成要素より前の構成要素をすべて KEYDEF 句に指定する必要があります。

注※2

境界値分割を指定している場合，横分割の指定方法が TMS-4V/SP と異なるため，TMS-4V/SP の境界値リストに相当する情報を返却するために利用できます。この場合（境界値分割の構成要素名を指定した場合），不正データはチェックされません。

- レコード型の RD エリア分割キーとして，シーケンシャルインデクスの先頭以外の構成要素を指定したレコード型に対して，RD エリアを指定して検索をする場合は，次に示す条件で指定してください。

  ・レコード型の RD エリア分割キーに指定した構成要素よりも前にある構成要素をすべて KEYDEF 句に指定する。

  ・指定した KEYDEF 句ごとのデータベースのキー値は 1 つだけ指定する。

RD エリアを指定した検索については，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「店群順アクセス」を参照してください。

- 境界値分割を指定している場合，KEYDEF 句下の DATA 句に指定したすべてのデータベースのキー値（境界値）の情報を取得したい場合の KEYDEF 句の指定について説明します。

  ・WITHIN 句で指定した境界値とは別に，任意のデータベースのキーを指定できます。

  ・WITHIN 句で指定した境界値とのチェックは行われません。

  KEYDEF 句下の DATA 句に指定した，すべてのデータベースのキー値（境界値）の情報を取得したい場合の定義例を次に示します。

  [説明]

  TMS-4V/SP で，連続するエリアに対して複数の異なる境界値を指定して，次のように SDB データベースを横分割して格納していたとします。

エリア名	境界値	意味（レコードの格納範囲）
RDAREA1	X'0100'	X'0100'以上～X'01FF'まで
RDAREA2	X'0200'	X'0200'以上～X'02FF'まで
RDAREA2	X'0300'	X'0300'以上～X'03FF'まで
RDAREA3	X'0400'	X'0400'以上

  この例の場合，境界値 X'0100'，X'0200'，X'0300'，X'0400'の各値以上の値を持つレコードが各エリアに格納されます。

  これを SDB データベース格納定義で定義すると境界値（レコード型の RD エリア分割キー値）と RD エリアを次のように対応させる必要があります。

RD エリア名	境界値（レコード型の RD エリア分割キー値）	意味（レコードの格納範囲）
RDAREAX（新規）	X'00FF'	～X'00FF'以下
RDAREA1	X'01FF'	X'0100'～X'01FF'以下
RDAREA2	X'03FF'	X'0200'～X'03FF'以下
RDAREA3	なし	X'0400'～

  SDB データベース格納定義では，連続的に同じ RD エリアに対して異なるレコード型の RD エリア分割キー値を定義できません。このため，TMS-4V/SP で境界値として指定していた X'0300'のデータが不足してしまい，TMS-4V/SP と同じ境界値リストを返却できません。

  HiRDB/SD で TMS-4V/SP と同じ境界値リストを返却するためには，TMS-4V/SP で指定していたすべての境界値を KEYDEF 句に指定する必要があります。

  定義例を次に示します。

  定義例

```
STORAGE SCHEMA SCM01 FOR SCM01
  DBTYPE 4V
  STORAGE RECORD RECA
```

```
    SEQUENTIAL IDX1 FOR RECORD
         :
      WITHIN ((RDAREAX_I),( RDAREA1_I),( RDAREA2_I),( RDAREA3_I))
     DEPENDING ON C1
     WITHIN ( (RDAREAX) X'00FF' , (RDAREA1) X'01FF',
              (RDAREA2) X'03FF', (RDAREA3) )
           :
  SDBOPTION
    KEYDEF C1
      DATA  X'0100'
      DATA  X'0200'
      DATA  X'0300'
      DATA  X'0400'
END STORAGE SCHEMA
```

(w) DATA

DATA データベースのキー値 …

データベースのキー値

～X'＜16 進数字＞'((1～30 バイト))

～'＜英数字または下線（_）＞'((1～30 バイト))※1

～＜10 進数定数＞※2

～＜整数定数＞※3

注※1

先頭に下線（_）は指定できません。

注※2

10 進数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

整数部と小数部にすべて 0 を指定する場合，符号に-を指定しないでください。-0.を指定しても 0.として扱います。

なお，10 進数定数の値の範囲を超える値は指定できません。

注※3

整数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

-0 を指定しても 0 として扱います。

なお，整数定数の値の範囲を超える値は指定できません。

KEYDEF 句の構成要素名で指定した構成要素のデータ型が CHARACTER または XCHARACTER の場合，データベースのキー値には，SDB データベースのキー値を KEYDEF 句の構成要素名で指定した構成要素のデータ属性とバイト数を合わせて指定します。

KEYDEF 句の構成要素名で指定した構成要素のデータ型が PACKED の場合，データベースのキー値には KEYDEF 句の構成要素名で指定した構成要素の整数部桁数，および小数部桁数の範囲内の値を指定します。

KEYDEF 句下の DATA 句でのデータベースのキー値の指定方法を次の表に示します。

表 11-35　KEYDEF 句下の DATA 句でのデータベースのキー値の指定方法

データ型	データベースのキー値	KEYDEF 句下の DATA 句でのデータベースのキー値の指定例	SDB ディレクトリ情報の設定内容
CHARACTER(4)	ABCD	'ABCD'	41424344
CHARACTER(4)	abcd	'abcd'	61626364
CHARACTER(4)	1000	'1000'	31303030
XCHARACTER(4)	AB12	X'41423132'	41423132
PACKED(2,1)	15.5	15.5	155C※
PACKED(2,2)	15.5	15.5	01550C
PACKED(3,2)	15.5	15.5	01550C
INTEGER	10	10	0A000000

注※

正の値または 0 の場合は，符号部を X'C'で設定します。負の値の場合は，符号部を X'D'で設定します。

《規則》

データベースのキー値に関する規則

- 1 つの KEYDEF 句で，データベースのキー値が重複することはできません。
- データベースのキー値を 16 進数字で指定する場合，データベースのキー値の長さはバイト単位で指定してください。
- KEYDEF 句で指定した構成要素が，レコード型の RD エリア分割キーの構成要素として指定されている場合，格納条件のレコード型の RD エリア分割キー値で指定された値と一致させてください。
- DATA 句を複数指定する場合，1 つの KEYDEF 句内でデータベースのキー値を昇順に指定してください。データベースのキー値を昇順に指定する例を次の表に示します。

表 11-36　データベースのキー値を昇順に指定する例

データ型	指定順序	KEYDEF 句下の DATA 句で指定するデータベースのキー値	SDB ディレクトリ情報の設定内容	指定内容の意味
CHARACTER(4)	1	'ABCD'	41424344	文字列：ABCD※
	1	X'41424344'	41424344	文字列：ABCD※
	2	'abCD'	61624344	文字列：abCD
	3	'abcd'	61626364	文字列：abcd
XCHARACTER(2)	1	X'0030'	0030	文字列：0
	2	X'2B30'	2B30	文字列：＋0

データ型	指定順序	KEYDEF 句下の DATA 句で指定するデータベースのキー値	SDB ディレクトリ情報の設定内容	指定内容の意味
	3	X'2B31'	2B31	文字列：＋1
	4	X'2D30'	2D30	文字列：－0
	5	X'2D31'	2D31	文字列：－1
PACKED(2,2)	1	-11.15	01115D	数字：-11.15
	2	-10.15	01015D	数字：-10.15
	3	10.15	01015C	数字：10.15
	4	11.15	01115C	数字：11.15
INTEGER	1	-3	FDFFFFFF	数字：-3
	2	-1	FFFFFFFF	数字：-1
	3	0	00000000	数字：0
	4	1	01000000	数字：1
	5	3	03000000	数字：3

注※

指定内容の意味が同じとなるデータベースのキー値は指定できません。

そのほかの規則

- 1 つの KEYDEF 句で指定できる DATA 句は，最大 32,767 個です。

(x) END STORAGE SCHEMA

END STORAGE SCHEMA

SDB データベース格納定義の終了を示します。

11.6　SDB データベース定義，SDB データベース格納定義【4V AFM】

4V AFM の SDB データベースを定義する場合の，SDB データベース定義および SDB データベース格納定義の指定形式と各オペランドの指定内容について説明します。

11.6.1　SDB データベース定義【4V AFM】

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定する SDB データベース定義について説明します。

注意事項

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定した SDB データベース定義は保存しておいてください。*ALTER DICTIONARY 文で SDB データベース定義を変更する際，変更個所だけを指定するのではなく，変更後の SDB データベース定義をすべて指定する必要があります。このとき，保存しておいた SDB データベース定義が必要になります。

(1)　機能

SDB データベースの論理構造を定義します。

(2)　形式

項番	形式	オペランドの追加可否※1
(a)	SCHEMA SDB データベース名	×
(b)	DBTYPE 4V　{DAM ｜ MAM ｜ TAM ｜ SAM}	×
(c)	RECORD レコード型名	○※2
(d)	{レベル番号｜ ITEM}　構成要素名	
(e)	〔{CHARACTER 長さ 　　｜ XCHARACTER 長さ 　　｜ PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕 　　｜ INTEGER}	
(f)	TYPE データ種別 1，データ種別 2〔，データ種別 3〕	
	〕…	
(g)	〔FUNCTION 　　〔REFER　{USE ｜ NOUSE}〕 　　〔ADD　{USE ｜ NOUSE}〕※3 　　〔UPDATE　{USE ｜ NOUSE}〕※4	

項番	形式	オペランドの追加可否※1
	〔ERASE {USE | NOUSE}〕 〔ALLERASE {USE | NOUSE}〕 〕	
(h)	〔RECORDID レコード識別コード〕	
	…	
(i)	SET 親子集合型名	
(j)	OWNER 親レコード型名	
(k)	ORDER {LAST | SORTED DUPLICATES PROHIBITED}	
(l)	MEMBER 子レコード型名	
(m)	INSERTION AUTOMATIC	
(n)	RETENTION FIXED	
(o)	〔KEY {ASCENDING | DESCENDING} 構成要素名〕	
(p)	〔SETOPTION	
(q)	〔OCCURRENCE NUMBER 一連番号の最大値	
(r)	〔WARNING {一連番号使用比率 1 〔, 一連番号使用比率 2 〔, 一連番号使用比率 3〕〕 | 0}〕	
(s)	〔REUSE {YES | NO}〕 〕	
	〕 …	
(t)	〔SDBOPTION	×
(u)	〔FUNCTION 〔REFER {USE | NOUSE}〕 〔ADD {USE | NOUSE}〕※3 〔UPDATE {USE | NOUSE}〕※4 〔ERASE {USE | NOUSE}〕 〔ALLERASE {USE | NOUSE}〕 〔DBLODUTL {USE | NOUSE}〕※5 〔FORMAT {USE | NOUSE}〕 〕	
(v)	〔TAMMODE {NONPROTECTED RETRIEVE | SHARED RETRIEVE}〕	
(w)	〔OCCURRENCE WARNING {一連番号使用比率 1 〔, 一連番号使用比率 2	×

項番	形式	オペランドの追加可否※1
	〔，一連番号使用比率 3〕〕 ｜ 0}〕	
	〕	
(x)	END SCHEMA	×

（凡例）

○：*ALTER DICTIONARY 文の実行時に追加できるオペランドです。

×：*ALTER DICTIONARY 文の実行時に追加できないオペランドです。

注※1

*ALTER DICTIONARY 文実行時のオペランド追加可否です。

注※2

最下位の子レコード型の追加とその親子集合の追加ができます。

RECORD 句と SET 句は必ず両方指定してください。どちらか片方だけを指定することはできません。

注※3

省略値については，「表 11-43　ADD 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

注※4

省略値については，「表 11-44　UPDATE 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

注※5

省略値については，「表 11-52　DBLODUTL 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

参考

SDBOPTION 句下の READYMODE 句の指定は無視されるため，バージョン 09-66 のマニュアルから，上記の形式に READYMODE 句は記載していません。READYMODE 句を指定したままでもエラーにはなりません。また，READYMODE 句の指定値によってシステムの動作が変わることはありません。

(3)　定義句の指定順序

SDB データベース定義の定義句の指定順序には，次に示す規則があります。

- 必須の定義，または必須の定義と省略できる定義の間は，上記の形式で示す順序で指定してください。
- RECORD 句下の FUNCTION 句と RECORDID 句，および FUNCTION 句下の各句については指定順はありません。任意の順番で指定できます。
- RECORD 句および SET 句の指定順序は，一筆書き順の規則に従ってください。一筆書き順の規則については，「図 11-11　RECORD 句および SET 句の指定順序（一筆書き順）の規則」を参照してください。

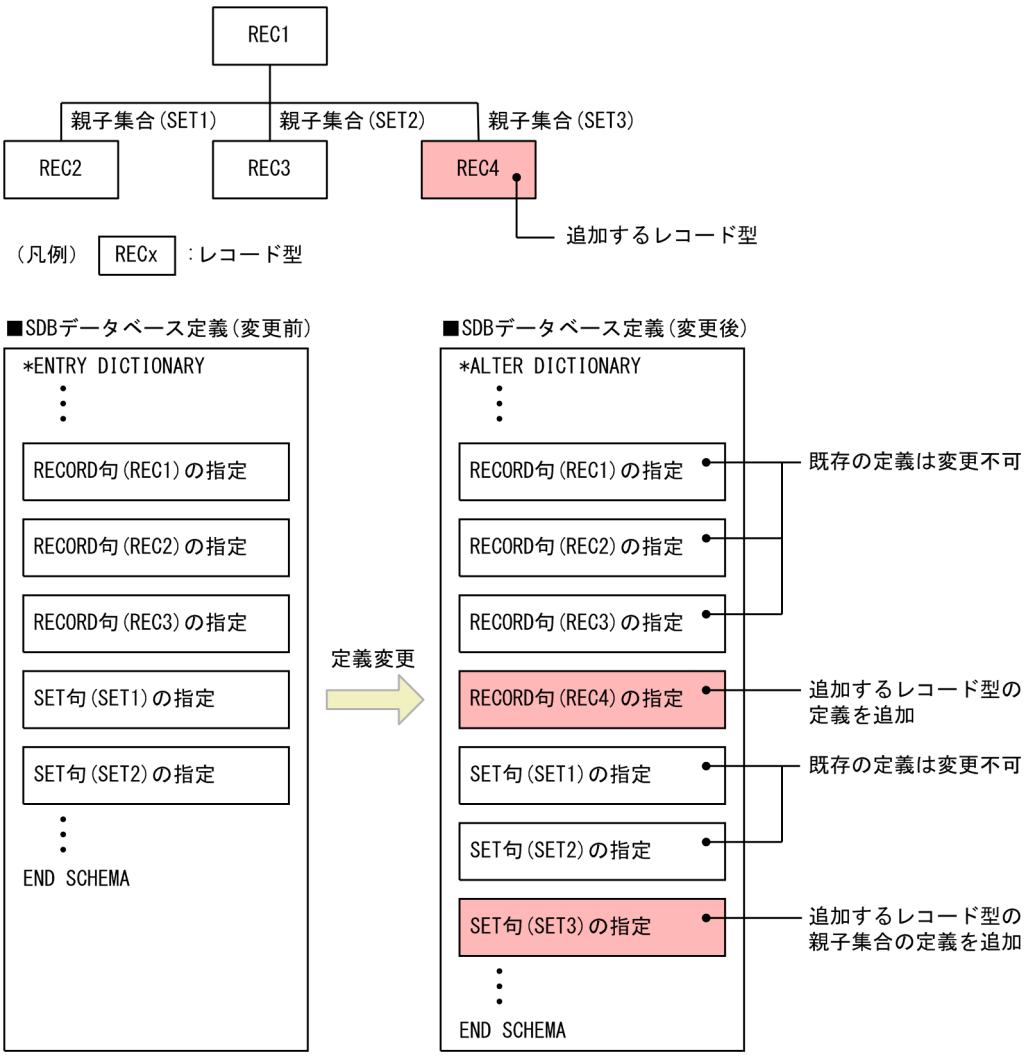
(4)　*ALTER DICTIONARY 文による SDB データベースの定義変更

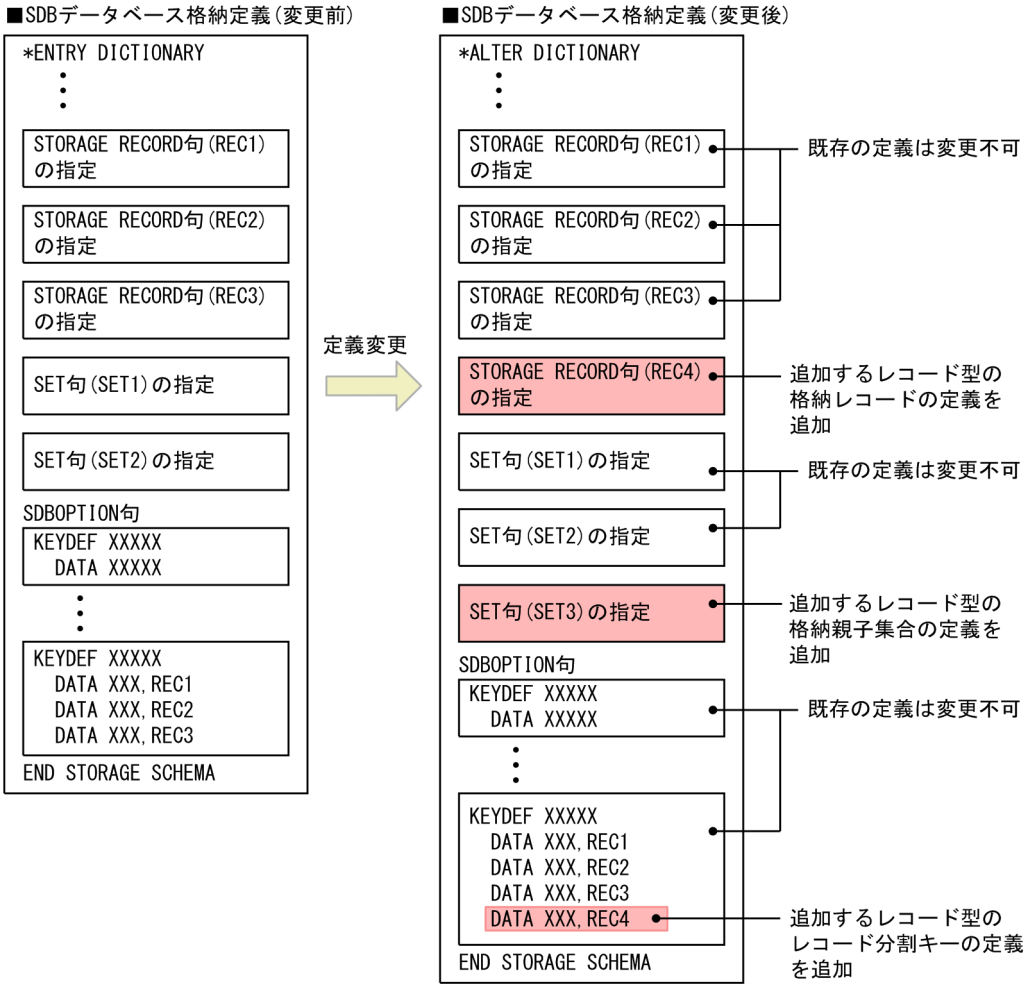
*ALTER DICTIONARY 文による SDB データベースの定義変更は，最下位の子レコード型の追加とその親子集合の追加ができます。子レコード型を追加できる位置と追加できない位置の例については「図 11-12　子レコード型を追加できる位置と追加できない位置」を参照してください。

なお，*ALTER DICTIONARY 文では，既存のレコード型の定義は変更できません。

*ALTER DICTIONARY 文による SDB データベースの定義変更の例を次の図に示します。

図 11-16　*ALTER DICTIONARY 文による SDB データベースの定義変更の例（4V AFM の SDB データベースの場合）





*ALTER DICTIONARY 文で実行できる SDB データベースの定義変更を次の表に示します。

表 11-37　*ALTER DICTIONARY 文で実行できる SDB データベースの定義変更

定義名	定義変更の内容	説明
SDB データベース定義	レコード型の追加（RECORD 句の追加）	次に示す条件をすべて満たす場合にレコードを追加できます。 • 追加するレコード型が子レコード型である。 • 仮想ルートレコード型の構成要素に，データ種別が K，R の構成要素がある。 • 仮想ルートレコード型で管理している子レコード用の管理領域の予備領域がある（子レコード型を追加しても問題ないだけの予備領域がある）。 また，次に示す定義の追加または変更を行う必要があります。 • 仮想ルートレコード型を親レコード型，追加するレコード型を子レコード型とする親子集合を追加する。 • SDB データベース格納定義の KEYDEF 句および DATA 句を追加し，データ種別が K，R の構成要素に対応するキー値と格納レコード名を指定する。 • SDB データベース格納定義の STORAGE RECORD 句を追加し，対応する格納レコード名を指定する。

定義名	定義変更の内容	説明
	親子集合の追加（SET 句の追加）	追加した子レコード型の親子集合を追加できます。 仮想ルートレコード型を親レコード型とし，追加するレコード型を子レコード型とする親子集合を追加してください。
SDB データベース格納定義	格納レコードの追加（STORAGE RECORD 句の追加）	SDB データベース定義で子レコード型を追加した場合に，対応する格納レコード名を追加します。子レコード型を追加していない場合に，格納レコード名だけを追加することはできません。 なお，追加した子レコード型にデータ種別が K，N の構成要素が存在する場合は，K，N または U，K の構成要素数分の二次インデクスを追加してください。二次インデクスの追加に伴い，二次インデクスを格納する RD エリアを追加できます。
	格納親子集合の追加（SET 句の追加）	子レコード型を追加する際に追加する親子集合に対応する格納親子集合を追加できます。子レコード型を追加しないで，格納親子集合だけを追加することはできません。
	キー値の追加（KEYDEF 句の追加）	■4V DAM，4V TAM，4V SAM の SDB データベースの場合 次に示す条件をすべて満たす場合は，追加する子レコード型ごとに，それぞれ 1 個以上のデータベースキーを追加できます。 • KEYDEF 句の構成要素に対応する SDB データベース定義の構成要素のデータ種別が K，R の場合 • 定義変更によって子レコード型を追加する場合
		■4V MAM の SDB データベースの場合 次に示す条件をすべて満たす場合は，追加する子レコード型ごとに，それぞれ 1 個のデータベースキーを追加できます。 • KEYDEF 句の構成要素に対応する SDB データベース定義の構成要素のデータ種別が K，R の場合 • 定義変更によって子レコード型を追加する場合

(5) オペランドの説明

SDB データベース定義の各オペランドの指定内容を説明します。

(a) SCHEMA

SCHEMA SDB データベース名

～<識別子>((1～30 バイト))

SDB データベースの SDB データベース名を指定します。

すでに使用している SDB データベース名は指定できません。

(b) DBTYPE

DBTYPE 4V {DAM | MAM | TAM | SAM}

定義する SDB データベースの SDB データベース種別を指定します。

DAM

SDB データベース種別が 4V DAM の SDB データベースを定義する場合に指定します。

MAM

SDB データベース種別が 4V MAM の SDB データベースを定義する場合に指定します。

TAM

SDB データベース種別が 4V TAM の SDB データベースを定義する場合に指定します。

SAM

SDB データベース種別が 4V SAM の SDB データベースを定義する場合に指定します。

4V DAM，4V MAM，4V TAM，または 4V SAM の SDB データベースとは，階層構造を持たないデータベースで，メインフレームの TMS-4V/SP がサポートしている DAM，MAM，TAM，または SAM のデータベースに相当します。詳細については，「2.3.3　SDB データベース」を参照してください。

注意事項

SDB データベース種別が 4V（4V FMB または 4V AFM）の SDB データベースと SD FMB の SDB データベースをシステム内で混在して定義することはできません。混在させた場合，動作を保証しません。

(c)　RECORD

RECORD レコード型名

～<識別子>((1～30 バイト))

レコード型名を指定します。

《規則》

- レコード型名は，同一スキーマ内で一意になるように指定してください。
- 1 つの SDB データベース定義内で指定できる RECORD 句は，2～128 個です。

(d)　レベル番号，ITEM

{レベル番号｜ITEM}　構成要素名

構成要素に関する情報を指定します。1 つの RECORD 句内での構成要素の階層レベルを指定します。

- すべての構成要素が基本項目の場合，ITEM を指定できます。
- 1 つのレコード型内で，レベル番号と ITEM は同時に指定できません。

データベースにアクセスするための API で「DBKEY」の名称を利用するため，構成要素名として「DBKEY」を指定する必要があります。指定方法の詳細については，「表 11-41　仮想ルートレコード型の構成要素の指定」～「表 11-42　子レコード型（2 階層）の構成要素の指定」を参照してください。

レベル番号

～<符号なし整数>((2～49))

1 つの RECORD 句内での構成要素の階層レベルを指定します。

階層が深くなるに従ってレベル番号を増やすように指定してください。レベル番号は連続した値である必要はありません。

階層が浅くなるときは，すでに指定したレベル番号にしてください。

ITEM

すべての構成要素が基本項目の場合に指定できます。

構成要素名

～<識別子>((1～30 バイト))

構成要素名を指定します。

《規則》

- 構成要素名は，RECORD 句内で一意にしてください。
- RECORD 句内に指定できる構成要素の合計長は最大 30,000 バイトです。また，RECORD 句内に指定できる構成要素の個数は最大 30,000 個です。
- 各レコード型の最初の基本項目の構成要素のデータ種別 1 には，D を指定してください。
- 上位レコードのキーの構成要素，および一連番号の構成要素は，下位レコードの構成要素として，データ型とデータ種別を同じように指定してください。
- ユーザデータ（データ種別 1，2（TYPE U,D））の構成要素は，幾つかの構成要素をまとめて，ユーザ任意の集団項目内に指定できます。

  ユーザキー（データ種別 1，2（TYPE U,K)），ユーザファイル通番（データ種別 1，2（TYPE U,F））の構成要素は，集団項目内に指定できません。

(e) データ型

{CHARACTER 長さ

｜XCHARACTER 長さ

｜PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕

｜INTEGER}

構成要素のデータ型を指定します。構成要素に指定できるデータ型を次の表に示します。

表 11-38　構成要素に指定できるデータ型

データ種別	データ型	データ形式
文字列	CHARACTER	固定長文字列
16 進形式の文字列	XCHARACTER	16 進固定長文字列
数値	PACKED 〔DECIMAL FIXED〕	符号付きパック形式 10 進データ
	INTEGER	整数

CHARACTER 長さ

～<符号なし整数>((1～30,000))

文字列データの長さを指定します。

XCHARACTER 長さ

～<符号なし整数>((1～30,000))

16 進形式の文字列データの長さを指定します。

PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕

整数部桁数

～<符号なし整数>((0～38))

小数部桁数

～<符号なし整数>((0～38))

符号付きパック形式の 10 進データの整数部桁数，および小数部桁数を指定します。

整数部桁数と小数部桁数の和は，1～38 になるようにしてください。

INTEGER

値の範囲が-2,147,483,648～2,147,483,647 の整数を指定します。長さは指定できません。

《規則》

- 一連番号の構成要素には，必ず INTEGER を指定してください。
- 基本項目の場合，データ型および TYPE 句を指定してください。
- 集団項目の場合，データ型および TYPE 句を省略してください。
- KEYDEF 句に指定する構成要素については，データの長さを 1～30 バイトで指定してください。KEYDEF 句については，「11.6.2(3)(u)　KEYDEF」を参照してください。

(f)　TYPE

TYPE データ種別 1，データ種別 2〔，データ種別 3〕

構成要素のデータ種別を指定します。

データ種別 1

～<英字>((1 バイト))

データベースキーかユーザデータかを区別するための種別を指定します。

データ種別 2

～<英字>((1 バイト))

データ種別 1 をさらに分類するための詳細な種別を指定します。

データ種別 3

～X'< 16 進数字>'((2 バイト))

{X'ユーザ情報'｜X'0000'}

ユーザ情報を指定します。

TYPE 句で指定するデータ種別の指定値と意味を次の表に示します。

表 11-39　データ種別の指定値と意味

データ種別	指定値	意味
データ種別 1	D	データベース名称を識別する構成要素
	K	データベースキーで，次に該当する場合の構成要素 • データベースキー • 上位レコードの一連番号 • 自レコードの一連番号（一連番号の最大値（OCCURRENCE NUMBER 句の指定）が 1 以上の場合）
	N	自レコードの一連番号（一連番号の最大値（OCCURRENCE NUMBER 句の指定）が 0 の場合）
	U	ユーザデータ
データ種別 2	A※1	レコード型の RD エリア分割キーとする構成要素（SDB データベースを横分割する際，レコードの格納 RD エリアを決める構成要素）
	R※2	レコード分割キーとする構成要素
	M※3	A かつ R の構成要素
	N	自レコードの一連番号の構成要素 • データ種別 1 が K の場合，一連番号属性あり • データ種別 1 が N の場合，一連番号属性なし
	L	そのほかのデータベースキーの構成要素
	F	ユーザファイル通番の構成要素
	K	ユーザキーの構成要素
	D	そのほかのユーザデータの構成要素
データ種別 3	2 バイトの 16 進数字	ユーザが独自に設定する属性

注※1

データ種別 1 が D の場合，SDB データベースの横分割がされていないことを示します。

注※2

データ種別 1 が D の場合，レコード分割がされていないことを示します。

注※3

データ種別 1 が D の場合，SDB データベースの横分割およびレコード分割がされていないことを示します。

《規則》

共通の規則

- D，K，N，U の順に，必要な種別を指定してください。
- ユーザデータのあとにデータベースキーを指定することはできません。
- レコード型の先頭にデータ種別 1 が D の構成要素を指定してください。

- 子レコード型が 1 つのときはデータ種別 1 が K かつ，データ種別 2 が M の構成要素を指定できません。
- 子レコード型が 2 つ以上のときは，データ種別 1 が K かつ，データ種別 2 が R または M の構成要素の指定が必要になります。

4V MAM の SDB データベースの規則

- データ種別 2 が K の構成要素を子レコード型に指定してください。
- データ種別 1 が K，かつデータ種別 2 が A の構成要素に対して，2 つ以上のレコード型の RD エリア分割キー値を指定できません。

4V TAM の SDB データベースの規則

- データ種別 2 が K の構成要素を子レコード型に指定してください。
- データ種別 1 が K，かつデータ種別 2 が A，M の構成要素は指定できません。

4V DAM の SDB データベースの規則

- データ種別 2 が K の構成要素は子レコード型には指定できません。

4V SAM の SDB データベースの規則

- データ種別 2 が K の構成要素は子レコード型には指定できません。
- データ種別 1 が K，かつデータ種別 2 が A，M の構成要素は指定できません。

そのほかの規則

- データ種別 1 が K，かつデータ種別 2 が A，M の構成要素は，仮想ルートレコード型と子レコード型で同一名称にしてください。
- データ種別 1 が K，かつデータ種別 2 が A，M の構成要素は，データの長さを 1～30 バイトの範囲で指定してください。
- RECORD 句内では，データ種別 1 の指定に関係なく，データ種別 2 が A，R，または M の構成要素は，それぞれ最大 1 つ指定できます。
- RECORD 句内では，データ種別 1 の指定に関係なく，データ種別 2 が M の構成要素を指定した場合，データ種別 2 が A，R の構成要素は指定できません。
- データ種別 1 が K，かつデータ種別 2 が N の場合，一連番号属性があります。OCCURRENCE NUMBER 句には 1 以上を指定するか，または指定を省略してください。
- データ種別 1 が N，かつデータ種別 2 が N の場合，一連番号属性がありません。OCCURRENCE NUMBER 句には 0 を指定してください。
- データ種別 2 が F の構成要素はレコード型内の最後の構成要素に 1 つだけ指定できます。
- データ種別 2 が K の構成要素は 1 レコード型に最大 1 つのユーザキーを指定できます。
- データ種別 1 が K，かつデータ種別 2 が M の構成要素を指定した場合，*ALTER DICTIONARY 文による SDB データベースの定義変更はできません。
- 構成要素のデータ種別 1，データ種別 2，データ種別 3 で指定できる組み合わせを次の表に示します。

表 11-40　データ種別 1，データ種別 2，データ種別 3 で指定できる組み合わせ

データ種別 1	データ種別 2	データ種別 3
D	L	−
	A	
	R	
	M	
K	L	
	A※1	
	R	
	M※1	
	N※2	
N	N※3	−
U	D	○
		−
	F※4	○
		−
	K※5，※6	○
		−

（凡例）

○：指定あり

−：指定なし

注※1

4V TAM または 4V SAM の SDB データベースの場合は指定できません。

注※2

OCCURRENCE NUMBER 句は省略するか，または 1 以上を指定してください。

注※3

OCCURRENCE NUMBER 句には 0 を指定してください。

注※4

レコード型内で最後に 1 つだけ指定できます。

注※5

フォーマットライト機能は使用できません。FUNCTION 句の FORMAT を省略するか，NOUSE を指定してください。

注※6

レコード型内で最大 1 つだけ指定できます。

- 仮想ルートレコード型と子レコード型の構成要素の指定を次の表に示します。指定順序や指定する個数，データ種別の組み合わせに誤りがある場合は，エラーとなります。なお，1 つの項番の中で基本項目と集団項目を複数指定できる場合，指定順序は任意です。

表 11-41　仮想ルートレコード型の構成要素の指定

指定順	項目		説明	データ種別		
				1	2	3
1	データベースキー 1	データベース名称を識別する構成要素名	• 構成要素名を「DBKEY」以外とします。 • レコードの先頭の構成要素として指定します。 • 基本項目として指定します。	D	L	－
					A	
					R	
		個数	1		M	
2	データベースキー 2～7※4	構成要素名	• 構成要素名を「DBKEY」以外とします。 • データベースキー 2～7 を，基本項目として指定します。	K	L	－
					A※1	
					R※3	
		個数	0～6		M※1，※2，※3	

（凡例）

－：該当しません。

注※1

4V TAM または 4V SAM の SDB データベースの場合，指定するとエラーになります。

注※2

子レコード型が 1 つの場合，指定するとエラーになります。

注※3

子レコード型が 2 つ以上の場合，データ種別 1 が K，かつデータ種別 2 が R または M の構成要素がないとエラーになります。

注※4

一連番号の最大値（OCCURRENCE NUMBER 句の指定値）が 0 の場合，データベースキー 2～7 が最低 1 つ必要です。

表 11-42　子レコード型（2 階層）の構成要素の指定

指定順	項目		説明	データ種別		
				1	2	3
1	データベースキー 1	データベース名称を識別する構成要素名	仮想ルートレコード型で指定したデータベースキー 1（データベースを識別する構成要素名）を，基本項目として指定します。	D	L	－
		個数	1		A	

指定順	項目		説明	データ種別		
				1	2	3
					R	
					M	
2	「DBKEY」の構成要素名		次のどちらかの場合，構成要素名「DBKEY」の集団項目を指定します。 • データベースキー 2~7 がある。 • 一連番号の最大値（OCCURRENCE NUMBER 句の指定値）が 1 以上である。	−		
3	データベースキー 2~7	構成要素名	仮想ルートレコード型で指定したデータベースキー 2~7 を集団項目「DBKEY」下の基本項目として指定します。	K	L	−
					A	
					R	
		個数	0~6		M	
4	一連番号	OCCURRENCE NUMBER 句の指定値が 1 以上の場合	集団項目「DBKEY」下の基本項目として指定します。	K	N	−
		OCCURRENCE NUMBER 句の指定値が 0 の場合	• 構成要素名を「DBKEY」以外とします。 • 基本項目として指定します。※	N	N	
5	ユーザデータの構成要素名		ユーザデータの構成要素名を指定します。	−	−	−
				U	D	
					K	
					F	

（凡例）

−：該当しません。

注※

集団項目「DBKEY」下の基本項目として指定するとエラーになります。

(g) FUNCTION

FUNCTION

〔REFER {USE | NOUSE}〕

〔ADD {USE | NOUSE}〕※1

〔UPDATE {USE | NOUSE}〕※2

〔ERASE {USE | NOUSE}〕

〔ALLERASE　{USE ｜ NOUSE}〕

注※1

省略値については，「表 11-43　ADD 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

注※2

省略値については，「表 11-44　UPDATE 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

SDB データベースへのアクセス属性を指定します。

SDB データベースへのアクセス属性は，SDBOPTION 句下の FUNCTION 句でも指定できます。RECORD 句内で指定する場合は，各 RECORD 句にアクセス属性が指定できます。指定を省略した場合は，SDBOPTION 句下の FUNCTION 句の指定値，または省略値が仮定されます。

REFER

レコードの検索を許可するかどうかを指定します。

USE：レコードの検索を許可します。

NOUSE：レコードの検索を許可しません。

ADD

レコードの追加を許可するかどうかを指定します。

USE：レコードの追加を許可します。

NOUSE：レコードの追加を許可しません。

表 11-43　ADD 句の省略値（4V AFM の SDB データベースの場合）

SDB データベース種別	レコード型の指定条件		
	RECORD 句下の FUNCTION 句が省略された場合		SDBOPTION 句下の FUNCTION 句が省略された場合
	仮想ルートレコード型	仮想ルートレコード型以外	
4V AFM	USE	SDBOPTION 句下の FUNCTION 句の指定値，または省略値	NOUSE

UPDATE

レコードの更新を許可するかどうかを指定します。

USE：レコードの更新を許可します。

NOUSE：レコードの更新を許可しません。

表 11-44　UPDATE 句の省略値（4V AFM の SDB データベースの場合）

SDB データベース種別	レコード型の指定条件		
	RECORD 句下の FUNCTION 句が省略された場合		SDBOPTION 句下の FUNCTION 句が省略された場合
	仮想ルートレコード型	仮想ルートレコード型以外	
4V DAM 4V MAM 4V TAM	USE	SDBOPTION 句下の FUNCTION 句の指定値，または省略値	USE
4V SAM	USE	SDBOPTION 句下の FUNCTION 句の指定値，または省略値	NOUSE

ERASE

レコードの削除を許可するかどうかを指定します。

USE：レコードの削除を許可します。

NOUSE：レコードの削除を許可しません。

ALLERASE

レコードの一括削除を許可するかどうかを指定します。

USE：レコードの一括削除を許可します。

NOUSE：レコードの一括削除を許可しません。

《規則》

- SDB データベース種別ごとの，仮想ルートレコード型に対する FUNCTION 句の指定可否を次の表に示します。

表 11-45　仮想ルートレコード型に対する FUNCTION 句の指定可否

SDB データベース種別	RECORD 句下の FUNCTION 句の指定														
	REFER 句			ADD 句			UPDATE 句			ERASE 句			ALLERASE 句		
	U	N	省略※	U	N	省略※	U	N	省略※	U	N	省略※	U	N	省略※
4V DAM	×	×	○	×	×	○	×	×	○	×	×	○	×	×	○
4V MAM	×	×	○	×	×	○	×	×	○	×	×	○	×	×	○
4V TAM	×	×	○	×	×	○	×	×	○	×	×	○	×	×	○
4V SAM	×	×	○	×	×	○	×	×	○	×	×	○	×	×	○

（凡例）

U：USE 指定

N：NOUSE 指定

○：指定できます。

×：指定できません。

注※

次の値が仮定されます。

REFER：USE

ADD：USE

UPDATE：USE

ERASE：NOUSE

ALLERASE：NOUSE

- SDB データベース種別ごとの，子レコード型に対する FUNCTION 句の指定可否を次の表に示します。

表 11-46　子レコード型に対する FUNCTION 句の指定可否

SDB データベース種別	RECORD 句下の FUNCTION 句の指定														
	REFER 句			ADD 句			UPDATE 句			ERASE 句			ALLERASE 句		
	U	N	省略※	U	N	省略※	U	N	省略※	U	N	省略※	U	N	省略※
4V DAM	○	○	○	○	○	○	○	○	○	×	○	○	○	○	○
4V MAM	○	×	○	○	○	○	○	○	○	○	○	○	×	○	○
4V TAM	○	×	○	×	○	○	○	○	○	×	○	○	×	○	○
4V SAM	○	×	○	○	○	○	×	○	○	×	○	○	○	○	○

（凡例）

U：USE 指定

N：NOUSE 指定

○：指定できます。

×：指定できません。

注※

SDBOPTION 句下の FUNCTION 句の指定値，または省略値が仮定されます。

(h)　RECORDID

RECORDID レコード識別コード

～X'< 16 進数字>'((2 バイト))

レコード型を識別するコードとして，レコード識別コードを指定します。

《規則》

- レコード識別コードを 2 バイトで指定します。
- X'0000'以外は重複できません。
- 仮想ルートレコード型に対しては，このオペランドは指定できません。
- 4V MAM の SDB データベースの子レコード型に対しては，このオペランドを必ず指定してください。

(i) SET

SET 親子集合型名

～<識別子>((1～30 バイト))

親子集合型名を指定します。

《規則》

- 親子集合型名は，SDB データベース定義内で一意にしてください。
- 仮想ルートレコード型の数は 1 つにしてください。
- 親レコードから子レコードへの親子集合型をたどった場合に，元のレコードにたどり着くような親子集合型は定義できません。
- 1 つの SDB データベース定義内に指定できる親子集合型数は，1～127 個です。指定できる親子集合の階層は，2 です。

(j) OWNER

OWNER 親レコード型名

～<識別子>((1～30 バイト))

親子集合の親レコード型名を指定します。

《規則》

親レコード型名には，RECORD 句で定義したレコード型名を指定してください。

(k) ORDER

ORDER {LAST | SORTED DUPLICATES PROHIBITED}

API によるレコードの格納時（STORE），HiRDB/SD データベース作成ユティリティ（pdsdblod），または HiRDB/SD データベースアクセスユティリティ（pdsdbexe）で，子レコード実現値を親子集合内の任意の実現値に格納する際の位置（親子集合内の子レコードへの挿入順序）を指定します。

LAST

子レコード実現値群の最後に格納します。

SORTED DUPLICATES PROHIBITED

子レコード型で指定したキー項目（KEY 句に指定した構成要素）の値に従って，昇順または降順に格納します。

同じ親レコードに対して，同じ値のキー項目を持つ子レコードが格納済みの場合，格納できません。

データ種別 1，2（TYPE U,K）を指定している場合は，SORTED DUPLICATES PROHIBITED を指定してください。それ以外の場合は，LAST を指定してください。

《規則》

1 つの SET 句の指定には，必ず ORDER 句を指定してください。

(l) MEMBER

MEMBER 子レコード型名

～<識別子>((1～30 バイト))

親子集合内の子レコード型名を指定します。

《規則》

- 子レコード型名には，RECORD 句で定義したレコード型名を指定してください。ただし，親子集合の親レコード型名と重複する子レコード型名は指定できません。
- 1 つの SET 句には，必ず MEMBER 句を指定してください。

(m) INSERTION

INSERTION AUTOMATIC

子レコードを親子集合に格納するときの挿入方法を指定します。

AUTOMATIC

子レコードのレコード実現値がデータベースに格納されると，自動的に親レコードが属する親子集合型の親子集合実現値に格納されます。

(n) RETENTION

RETENTION FIXED

子レコードを親子集合から切り離すまでの属性を指定します。

FIXED

レコードが，ある親子集合の子レコードになったレコードは，データベースから削除されるまで，その親子集合の子レコードでなければなりません。

(o) KEY

KEY {ASCENDING｜DESCENDING} 構成要素名

ORDER 句で SORTED DUPLICATES PROHIBITED を指定した親子集合の子レコードを構成要素の値に従って，昇順に格納するか，または降順に格納するかを指定します。指定した構成要素をキー項目とします。

ASCENDING

子レコードを昇順に格納します。

DESCENDING

子レコードを降順に格納します。

構成要素名

～<識別子>((1～30 バイト))

格納時のキーになる構成要素名を指定します。

《規則》

- キーの構成要素長は 254 バイト以内としてください。
- 構成要素名に，集団項目の構成要素名は指定できません。
- 指定できる構成要素は，データ種別 1 がユーザデータ（U），データ種別 2 がユーザキー（K）の構成要素だけです。
- 構成要素名には，RECORD 句で定義した構成要素名を指定してください。
- 1 つの KEY 句に構成要素名を 2 つ以上指定できません。
- ORDER 句で LAST を指定した場合，KEY 句は指定できません。
- ORDER 句で SORTED DUPLICATES PROHIBITED を指定した場合，必ず KEY 句を指定してください。

(p) SETOPTION

SETOPTION

親子集合に関するオプションを指定します。

(q) OCCURRENCE NUMBER

OCCURRENCE NUMBER 一連番号の最大値

～<符号なし整数>((0～2,147,483,647))≪30,000≫

一連番号の最大値を指定します。

0 を指定した場合，非オカレンスであることを示します。

FUNCTION 句下の FORMAT 句に USE を指定した場合（フォーマットライト機能を使用する場合），KEYDEF 句の組み合わせ数×OCCURRENCE NUMBER 句の指定値分のレコードが作成されます。このため，この句の値が大きいと，フォーマットライトに時間が掛かるため，注意してください。

《規則》

- 構成要素のデータ種別 1，2 の指定が N，N の場合，OCCURRENCE NUMBER 句に 0 以外を指定，または省略することはできません。
- 構成要素のデータ種別 1，2 の指定が K，N の場合，0 は指定できません。
- 1 つの SDB データベース定義内で一連番号の最大値の値に 0 と 1 以上（OCCURRENCE NUMBER 句の指定を省略する場合も含む）を混在させることはできません。
- WARNING 句または REUSE 句を指定する場合は，この句も指定する必要があります。

(r) WARNING

WARNING {一連番号使用比率 1〔，一連番号使用比率 2〔，一連番号使用比率 3〕〕 | 0}

一連番号の使用状況を監視する場合に，一連番号の使用比率を％単位で指定します。

一連番号使用比率 1，一連番号使用比率 2，一連番号使用比率 3
～<符号なし整数>((50～99))

割り当て済み一連番号の割合が一連番号使用比率 1～3 で指定した値に達した場合※に，メッセージを 1 回ずつ出力します。0 を指定した場合は，割り当て済みの一連番号の監視をしません。

注※　監視値＝↑一連番号の最大値×一連番号使用比率n↑

この句と SDBOPTION 句下の OCCURRENCE WARNING 句の両方を指定した場合，この句での指定が有効となります。この場合，一連番号使用比率の指定値の個数が異なっていても，双方で補ってメッセージを出力することはありません。

OCCURRENCE NUMBER 句の指定が 0 で，この句を省略した場合，割り当て済みの一連番号の監視をしません。

OCCURRENCE NUMBER 句の指定が 0 以外で，この句を省略した場合，SDBOPTION 句下の OCCURRENCE WARNING 句の指定が有効となります。

SDBOPTION 句下の OCCURRENCE WARNING 句も省略した場合，割り当て済みの一連番号の監視をしません。

一度に複数の一連番号使用比率に達した場合は，対応するメッセージがすべて出力されます。例えば，WARNING 句の指定を 60，70，80 とした場合に，1 回目のレコードの格納で 75%に達した場合は，指定値 60 と 70 に対応するメッセージが出力されます。

一連番号の最大値が小さいと，重複指定でなくても内部的に重複指定と同じような扱いになる場合があります。例えば，OCCURRENCE NUMBER 句が 10，WARNING 句が 97，98，99 と定義した場合，どの指定値も割り当て済み一連番号が 10 になった時点で，97（%）と 98（%）と 99（%）に対応するメッセージが出力されます。

一連番号の監視については，「5.25　一連番号の監視」を参照してください。

《規則》

- 4V TAM の SDB データベースの場合は指定できません。
- 一連番号使用比率を複数指定する場合，指定値が昇順になるように指定してください。
- SDBOPTION 句下の FUNCTION で FORMAT USE を指定している場合は指定できません。
- OCCURRENCE NUMBER 句との組み合わせ可否を次の表に示します。

表 11-47　SETOPTION 句下の OCCURRENCE NUMBER 句と WARNING 句の組み合わせ可否

SETOPTION 句下の OCCURRENCE NUMBER 句の指定	SETOPTION 句下の WARNING 句の指定	
	0 以外	0
0 以外	○	○
0	×	○

（凡例）
○：WARNING 句が指定できます。
×：WARNING 句は指定できません。

表 11-48　SETOPTION 句下の OCCURRENCE NUMBER 句と SDBOPTION 句下の OCCURRENCE WARNING 句の組み合わせ可否

SETOPTION 句下の OCCURRENCE NUMBER 句の指定	SDBOPTION 句下の OCCURRENCE WARNING 句の指定	
	0 以外	0
すべて 0 以外	○	○
一部 0	○	○
すべて 0	×	○

（凡例）

○：OCCURRENCE WARNING 句が指定できます。

×：OCCURRENCE WARNING 句は指定できません。

- FUNCTION 句の ADD 指定との組み合わせとレコードの追加可否を次の表に示します。

表 11-49　FUNCTION 句の ADD 指定との組み合わせとレコードの追加可否

SDB データベース種別	SDBOPTION 句下の FUNCTION 句の ADD 指定								
	省略			U			N		
	子レコードの FUNCTION 句の ADD 指定								
	省略	U	N	省略	U	N	省略	U	N
4V DAM 4V MAM 4V SAM	×	○	×	○	○	×	×	○	×
4V TAM	×	−	×	−	−	−	×	−	×

（凡例）

U：USE 指定

N：NOUSE 指定

○：レコードの指定ができます。

×：レコードの指定ができません。

−：該当しません。

- 子レコードへの追加可否との組み合わせを次の表に示します。

表 11-50　子レコードへの追加可否と WARNING 句の組み合わせ可否

子レコードへの追加	SETOPTION 句下の WARNING 句の指定	
	0 以外	0
できる	○	○
できない	×	○

（凡例）

○：WARNING 句が指定できます。

×：WARNING 句は指定できません。

表 11-51　子レコードへの追加可否と SDBOPTION 句下の OCCURRENCE WARNING 句の組み合わせ可否

子レコードへの追加	SDBOPTION 句下の OCCURRENCE WARNING 句の指定	
	0 以外	0
すべての子レコードに対してできる	○	○
一部の子レコードに対してできる	○	○
すべての子レコードに対してできない	×	○

（凡例）

○：OCCURRENCE WARNING 句が指定できます。

×：OCCURRENCE WARNING 句は指定できません。

(s)　REUSE

REUSE　{YES｜NO}

削除された一連番号を再利用するかどうかを指定します。

レコードを一括削除した場合，または全件削除した場合は，REUSE 句の指定に関係なく，最初から一連番号を振り直します。

YES

一連番号を再利用します。

4V DAM または 4V SAM の SDB データベースで，ALLERASE 句に USE を指定した場合に，一連番号を再利用できます。

なお，OCCURRENCE NUMBER 句に 0 を指定してもエラーにはなりません。

ORDER 句で SORTED DUPLICATES PROHIBITED を指定した場合は指定できません。

NO

削除された一連番号を再利用しません。

一連番号の再利用の例については，「図 11-14　一連番号再利用の例」を参照してください。

(t)　SDBOPTION

SDBOPTION

SDB データベースのオプション機能を定義します。

(u)　FUNCTION

FUNCTION

〔REFER　{USE ｜ NOUSE}〕

〔ADD　{USE ｜ NOUSE}〕※1

〔UPDATE　{USE ｜ NOUSE}〕※2

〔ERASE　{USE ｜ NOUSE}〕
〔ALLERASE　{USE ｜ NOUSE}〕
〔DBLODUTL　{USE ｜ NOUSE}〕※3
〔FORMAT　{USE ｜ NOUSE}〕

注※1

省略値については，「表 11-43　ADD 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

注※2

省略値については，「表 11-44　UPDATE 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

注※3

省略値については，「表 11-52　DBLODUTL 句の省略値（4V AFM の SDB データベースの場合）」を参照してください。

SDB データベースへのアクセス属性を指定します。

REFER 句，ADD 句，UPDATE 句，ERASE 句，ALLERASE 句の説明については，「11.6.1(5)(g) FUNCTION」を参照してください。

DBLODUTL

HiRDB/SD データベース作成ユティリティ（pdsdblod）のデータロード機能の実行を許可するかどうかを指定します。

USE：HiRDB/SD データベース作成ユティリティ（pdsdblod）のデータロード機能の実行を許可します。

NOUSE：HiRDB/SD データベース作成ユティリティ（pdsdblod）のデータロード機能の実行を許可しません。

表 11-52　DBLODUTL 句の省略値（4V AFM の SDB データベースの場合）

SDB データベース種別	FUNCTION 句下の ADD 句の指定	
	USE	NOUSE
4V DAM	NOUSE	NOUSE
4V MAM 4V SAM	NOUSE	USE
4V TAM	－	USE

（凡例）

－：4V TAM の SDB データベースの場合，ADD 句に USE は指定できません。

FORMAT

HiRDB/SD データベース作成ユティリティ（pdsdblod）のフォーマットライト機能の実行を許可するかどうかを指定します。

SET 句下の OCCURRENCE NUMBER 句の指定値が大きいとフォーマットライトに時間が掛かるため，注意してください。

USE：HiRDB/SD データベース作成ユティリティ（pdsdblod）のフォーマットライト機能の実行を許可します。

NOUSE：HiRDB/SD データベース作成ユティリティ（pdsdblod）のフォーマットライト機能の実行を許可しません。

《規則》

- 4V DAM の SDB データベースの場合，次の指定をするとデータベースにデータを格納できないため，指定しないでください。
  ・ADD 句，DBLODUTL 句，および FORMAT 句の指定をすべて省略する。
  ・ADD 句，DBLODUTL 句，および FORMAT 句にすべて NOUSE を指定する。
- SDB データベース種別と SDBOPTION 句下の FUNCTION 句（ADD 句，DBLODUTL 句，FORMAT 句）の指定の組み合わせ可否を次の表に示します。

表 11-53　SDBOPTION 句下の FUNCTION 句の指定の組み合わせ可否

SDB データベース種別	ADD 句の指定							
	USE				NOUSE			
	DBLODUTL 句の指定							
	USE		NOUSE		USE		NOUSE	
	FORMAT 句の指定							
	USE	NOUSE	USE	NOUSE	USE	NOUSE	USE	NOUSE
4V DAM	×	○	×	○	○	○	○	○※
4V MAM 4V SAM	×	○	×	○	×	○	×	×
4V TAM	−	−	−	−	×	○	×	×

（凡例）

○：指定できます。

×：指定できません。

−：4V TAM の SDB データベースの場合，ADD 句に USE は指定できません。

注※

データベースにデータを格納できない組み合わせの指定のため，指定しないでください。

(v)　TAMMODE

TAMMODE　{NONPROTECTED RETRIEVE | SHARED RETRIEVE}

4V TAM の SDB データベースの場合に，TAM のデータベースの無排他検索機能を適用し，ページまたはサブページに排他を掛けないで検索するかどうかを指定します。

この指定は，個別開始時に指定する SDB データベースを操作する API で，排他モード 1 に'S'（共用モード）を指定した場合に有効となります。

詳細については，「2.9.6　TAM のデータベースの無排他検索機能【4V TAM】」を参照してください。

なお，4V TAM の SDB データベース以外の場合に，この指定をするとエラーになります。

NONPROTECTED RETRIEVE

TAM のデータベースの無排他検索機能を適用し，4V TAM の SDB データベースのページまたはサブページに排他を掛けないで検索します。

SHARED RETRIEVE

TAM のデータベースの無排他検索機能を適用しません（共用モードで，4V TAM の SDB データベースのレコードを検索します）。

(w)　OCCURRENCE WARNING

OCCURRENCE WARNING　{一連番号使用比率 1〔，一連番号使用比率 2〔，一連番号使用比率 3〕〕 | 0}

一連番号の使用状況を監視する場合に，一連番号の使用比率を％単位で指定します。

一連番号使用比率 1，一連番号使用比率 2，一連番号使用比率 3

～<符号なし整数>((50～99))

割り当て済み一連番号の割合が，一連番号使用比率 1～3 で指定した値に達した場合に，メッセージを 1 回ずつ出力します。0 を指定した場合，またはこの句を省略した場合は，割り当て済み一連番号の監視はしません。

この指定はすべての子レコード型に対して有効となりますが，SETOPTION 句下で WARNING 句を指定した場合は，SETOPTION 句下の WARNING 句の指定が有効となります。この場合，一連番号使用比率の指定値の個数が異なっても，双方で補ってメッセージを出力することはありません。

一連番号の監視については，「5.25　一連番号の監視」を参照してください。

《規則》

- このオペランドの指定規則については，「(r)　WARNING」の規則を参照してください。

(x)　END SCHEMA

END SCHEMA

SDB データベース定義の終了を示します。

11.6.2　SDB データベース格納定義【4V AFM】

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定する SDB データベース格納定義について説明します。

注意事項

*ENTRY DICTIONARY 文または*ALTER DICTIONARY 文に指定した SDB データベース格納定義は保存しておいてください。*ALTER DICTIONARY 文で SDB データベース格納定義を変更する際，変更個所だけを指定するのではなく，変更後の SDB データベース格納定義をすべて指定する必要があります。このとき，保存しておいた SDB データベース格納定義が必要になります。

(1) 機能

SDB データベースを格納する際の論理構造を定義します。

(2) 形式

項番	形式	オペランドの変更可否	
		レコード型の追加時※1	RD エリアの変更時※1
(a)	STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名	×	×
(b)	DBTYPE 4V	×	
(c)	STORAGE RECORD 格納レコード名	○※2	
(d)	{CLUSTERED 親子集合型名 | SEQUENTIAL シーケンシャルインデクス名		
(e)	FOR RECORD		
(f)	ORDER KEY ASCENDING 構成要素名〔，構成要素名〕…		
(g)	〔PCTFREE ページ内の未使用領域の比率〕		
(h)	WITHIN {シーケンシャルインデクス用 RD エリア名 | (シーケンシャルインデクス用 RD エリア名) | ((シーケンシャルインデクス用 RD エリア名) 〔，(シーケンシャルインデクス用 RD エリア名)〕…)} }		○※4，※5，※7
(i)	〔SUBPAGE NUMBER サブページ分割数〕		×
(j)	〔PCTFREE {ページ内の未使用領域の比率 | ((〔ページ内の未使用領域の比率〕，セグメント内の空きページの比率)}〕		
(k)	WITHIN {格納レコード用 RD エリア名 | (格納レコード用 RD エリア名) | ((格納レコード用 RD エリア名) 〔格納条件〕 〔，(格納レコード用 RD エリア名) 格納条件〕…)}		○※4，※5，※7
(l)	〔POINTER AREA SIZE 管理領域の予備領域の長さ〕		×

項番	形式	オペランドの変更可否	
		レコード型の追加時※1	RD エリアの変更時※1
(m)	〔SECONDARY INDEX 二次インデクス名		
(n)	ORDER KEY ASCENDING 構成要素名，構成要素名〔，構成要素名〕…		
(o)	〔PCTFREE ページ内の未使用領域の比率〕		
(p)	WITHIN {二次インデクス用 RD エリア名 | (二次インデクス用 RD エリア名) | ((二次インデクス用 RD エリア名) 〔，(二次インデクス用 RD エリア名)〕…)}		○※4，※5，※7
	〕…		
	…		
(q)	SET 親子集合型名		×
(r)	OWNER POINTER FOR FIRST LAST MEMBER		
(s)	MEMBER POINTER FOR NEXT PRIOR TENANT		
	…		
(t)	〔SDBOPTION	○※3	
(u)	〔KEYDEF 構成要素名		
(v)	DATA データベースのキー値〔，格納レコード名〕…		○※6
	〕…		
	〕		
(w)	END STORAGE SCHEMA	×	×

（凡例）

○：*ALTER DICTIONARY 文の実行時に変更できるオペランドです。

×：*ALTER DICTIONARY 文の実行時に変更できないオペランドです。

注

- 定義句は，形式で記述している順序に従って指定します。必須の定義と省略できる定義の間は形式で記載されているとおりに指定してください。
- *ALTER DICTIONARY 文で子レコード型を追加した場合，POINTER AREA SIZE 句に指定していた管理領域の予備領域の長さが自動的に変更されることがあります。変更後の値については，KFPB61784-I メッセージに出力されます。

注※1

*ALTER DICTIONARY 文実行時のオペランド追加可否です。

注※2

- 最下位の子レコード型の追加とその親子集合の追加ができます。
- STORAGE RECORD 句と SET 句は必ず両方指定してください。どちらか片方だけを指定することはできません。
- 既存の格納レコード型の定義は変更できません。

注※3

レコード分割キーの追加ができます。

注※4

格納レコード用 RD エリアとインデクス用 RD エリアはペアで変更します。

1 つの*ALTER DICTIONARY 文で実行できる格納レコード用 RD エリアの変更を次の表に示します。

表 11-54　1 つの*ALTER DICTIONARY 文で実行できる格納レコード用 RD エリアの変更

格納レコード用 RD エリア変更内容	横分割の方法
	格納条件指定
1 つの RD エリアを追加または削除	○
複数の RD エリアを追加または削除	○
1 つの RD エリアを複数の RD エリアに分割	○
複数の RD エリアを複数の RD エリアに分割	×
複数の RD エリアを 1 つの RD エリアに統合	○
複数の RD エリアを複数の RD エリアに統合	×
格納 RD エリア数を変更しないで，格納条件だけを追加，更新，または削除	×

（凡例）

○：実行できます。

×：実行できません。

なお，次の操作はできません。

- 1 つの RD エリアに格納していたレコード型を，複数の RD エリアに分割格納する変更
- 複数の RD エリアに分割格納していたレコード型を，1 つの RD エリアに格納する変更
- 1 つの RD エリアを複数の RD エリアに分割する際，分割元の RD エリアを削除する変更
- 複数の RD エリアを 1 つの RD エリアに統合する際，統合先の RD エリアに統合元の RD エリアを使用しない変更

注※5

次の条件をすべて満たす場合，RD エリアの追加，削除はできません。

- レコード型がサーバ内で横分割されていない
- レコード型の先頭のデータベースキーのデータ種別が K，A 以外である

- 同一 RD エリアに格納する条件を複数に分けて定義している

注※6

- RD エリアの追加の場合に，レコード型の RD エリア分割キーの追加ができます。
- RD エリアの削除の場合に，レコード型の RD エリア分割キーの削除ができます。

注※7

RD エリアの定義を変更した場合，変更対象の RD エリアに格納されているデータが削除されることがあります。RD エリアの定義変更時のデータの削除有無については，「表 11-23　RD エリアの定義変更時のデータの削除有無（格納条件指定の横分割の場合）」を参照してください。

(3)　オペランドの説明

SDB データベース格納定義の各オペランドの指定内容を説明します。

(a)　STORAGE SCHEMA

STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名

SDB データベース格納名と対応する SDB データベース名を指定します。

SDB データベース格納名

～<識別子>((1～30 バイト))

SDB データベース格納名を指定します。

すでに使用している SDB データベース格納名は指定できません。

SDB データベース名

～<識別子>((1～30 バイト))

SDB データベース名を指定します。

SDB データベース名に一致する名称が SDB データベース定義にない場合は，エラーとなります。

(b)　DBTYPE

DBTYPE 4V

SDB データベース種別が 4V AFM の SDB データベースを定義します。

注意事項

SDB データベース種別が 4V（4V FMB または 4V AFM）の SDB データベースと SD FMB の SDB データベースをシステム内で混在して定義することはできません。混在させた場合，動作を保証しません。

(c) STORAGE RECORD

STORAGE RECORD 格納レコード名

格納レコード名

～<識別子>((1～30 バイト))

SDB データベース定義の RECORD 句で指定したレコード型名を，格納レコード名として指定します。

《規則》

- SDB データベース定義の RECORD 句で指定したレコード型名と同じ個数，同じ順番で指定してください。
- すでに使用している格納レコード名は指定できません。ただし，格納レコードの所有者（スキーマ）が異なる場合は，同じ格納レコード名を指定できます。

(d) CLUSTERED，SEQUENTIAL

CLUSTERED 親子集合型名｜SEQUENTIAL シーケンシャルインデクス名

格納するレコードの配置方法を指定します。

CLUSTERED 親子集合型名

親子集合型名

～<識別子>((1～30 バイト))

すべての子レコードを親レコードと同じ RD エリアに格納する場合に指定します。

《規則》

- STORAGE RECORD 句に指定したレコード型ごとに 1 つ指定できます。STORAGE RECORD 句に指定したレコード型が，仮想ルートレコード型の場合は指定できません。
- SDB データベース格納定義の SET 句に指定する親子集合型名と同じ個数，同じ順番で指定してください。

SEQUENTIAL シーケンシャルインデクス名

インデクスのキーの値の順序に従って，親レコード実現値の格納位置を決めるときに指定します。SEQUENTIAL 句は，仮想ルートレコード型に必ず指定します。

シーケンシャルインデクス名

～<識別子>((1～30 バイト))

STORAGE RECORD 句に指定した仮想ルートレコード型に対して 1 つ指定します。STORAGE RECORD 句に指定したレコード型が，仮想ルートレコード型以外の場合は指定できません。

《規則》

- シーケンシャルインデクス名には，二次インデクス名とは別の名称を指定してください。
- シーケンシャルインデクス名は，同一スキーマ内で一意になるように指定してください。二次インデクス名も含めて一意になるようにしてください。

(e) FOR RECORD

FOR RECORD

レコード型単位にシーケンシャルインデクスを作成します。

キーの値が同じ場合，シーケンシャルインデクスへの格納を禁止します。

(f) ORDER KEY

ORDER KEY ASCENDING 構成要素名〔，構成要素名〕…

インデクスのキーとなる構成要素名を指定します。

指定した構成要素名をキーとして，昇順にシーケンシャルインデクスが作成されます。

ASCENDING

昇順にシーケンシャルインデクスを作成します。

構成要素名

～<識別子>((1～30 バイト))

インデクスのキーに指定する構成要素を次に示すように指定してください。

- 仮想ルートレコード型に指定した基本項目の構成要素名を，定義順にすべて指定します。仮想ルートレコード型の構成要素の指定については，「表 11-41　仮想ルートレコード型の構成要素の指定」を参照してください。4V DAM の SDB データベースの定義例については，「11.11.2 SDB データベースの定義例（4V DAM の場合）」を参照してください。

《規則》

- 構成要素名は，格納レコード名と対応する SDB データベース定義のレコード型名で指定した構成要素名としてください。
- シーケンシャルインデクス作成時に指定できる構成要素名の条件について，次の表に示します。

表 11-55　シーケンシャルインデクス作成時に指定できる構成要素名の条件

項目	内容
指定できる構成要素名の数	最大 7 個
インデクスのキー長の合計	249 バイト以内 インデクスのキーに指定する構成要素とキー長の関係については，「表 11-56　インデクスのキーに指定する構成要素とキー長の関係」を参照してください。
構成要素名	基本項目の構成要素名 集団項目が含まれる構成要素名は指定できません。

表 11-56　インデクスのキーに指定する構成要素とキー長の関係

データ型	インデクスのキー長（単位：バイト）	
	構成要素名の指定が 1 つの場合	構成要素名の指定が複数の場合
CHARACTER	SDB データベース定義で指定したデータ長	SDB データベース定義で指定したデータ長＋制御部（1 バイト）
XCHARACTER		
PACKED〔DECIMAL FIXED〕整数部桁数〔，小数部桁数〕	↓（整数部桁数＋小数部桁数）÷2↓＋1	↓（整数部桁数＋小数部桁数）÷2↓＋2
INTEGER	4	5

(g)　PCTFREE

PCTFREE ページ内の未使用領域の比率

～<符号なし整数>((0～99))≪30≫

インデクスページ内の未使用領域の比率を指定します。

(h)　WITHIN

WITHIN　{シーケンシャルインデクス用 RD エリア名

｜(シーケンシャルインデクス用 RD エリア名)

｜((シーケンシャルインデクス用 RD エリア名)

〔,(シーケンシャルインデクス用 RD エリア名)〕…)}

シーケンシャルインデクス用 RD エリア名

～< RD エリア名>((1～30 バイト))

シーケンシャルインデクスを格納する RD エリア名を指定します。

1 つの SEQUENTIAL 句に対して，WITHIN 句を必ず 1 つ指定してください。

《規則》

シーケンシャルインデクスを複数の RD エリアに格納する場合の規則

- シーケンシャルインデクス用 RD エリアとして指定できる RD エリアの数と SDB データベースの横分割の関係を次の表に示します。

表 11-57　シーケンシャルインデクス用 RD エリアとして指定できる RD エリア数と SDB データベースの横分割の関係

SDB データベースの横分割方法	指定できる RD エリア数
分割指定なし	1
格納条件分割	1～256

- シーケンシャルインデクス用 RD エリアは，格納レコード用 RD エリアと同じ数を指定する必要があります。

- 1 つの WITHIN 句の中では，シーケンシャルインデクス用 RD エリア名は一意に指定してください。
- 格納レコード用 RD エリアと対応するシーケンシャルインデクス用 RD エリアは，同じサーバ内の RD エリアである必要があります。

指定するシーケンシャルインデクス用 RD エリアに関する規則

- 格納レコード用 RD エリアとして指定した RD エリアは指定できません。
- シーケンシャルインデクス用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- シーケンシャルインデクス用 RD エリアとして指定する RD エリアには，1 つの RD エリアにシーケンシャルインデクスと二次インデクスを合わせて，インデクスを最大 500 個指定できます。
- インナレプリカ機能を使用する場合，指定する RD エリアはインナレプリカ機能の使用条件を満たしている必要があります。インナレプリカ機能の使用条件については，「付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】」を参照してください。
- シーケンシャルインデクス用 RD エリアに定義できるリソースの組み合わせを次の表に示します。

表 11-58　RD エリアに定義できるリソースの組み合わせ

定義するリソース		RD エリアの定義		
		レコード型・インデクス定義なし	SDB データベース格納定義 A	
			レコード型定義あり	インデクス定義あり
SDB データベース格納定義 A	レコード型	○	○	×
	インデクス	○	×	○
SDB データベース格納定義 B	レコード型	○	×	×
	インデクス	○	×	○

（凡例）

○：定義できます。

×：定義できません。

(i)　SUBPAGE NUMBER

SUBPAGE NUMBER サブページ分割数

サブページ分割数

～<符号なし整数>((2～16))

1 データページ当たりのサブページ分割数を指定します。SUBPAGE NUMBER を省略した場合，サブページ分割されません。サブページ分割数の設計方針については，「3.2.4　サブページの設計」を参照してください。

《規則》

- 仮想ルートレコード型，およびすべての子レコード型で同じサブページ分割数を指定してください。
- サブページ分割数を指定する場合，格納レコード用 RD エリアのページ長が，4096 バイトである必要があります。

(j) PCTFREE

PCTFREE {ページ内の未使用領域の比率
| ({ページ内の未使用領域の比率}，セグメント内の空きページの比率)}

ページ内の未使用領域の比率
～<符号なし整数>((0～99))

セグメント内の空きページの比率
～<符号なし整数>((0～50))

データベースの初期作成時に設定するページ内の未使用領域の比率を指定します。ページ内の未使用領域の比率は，0～99（%）の範囲で指定できます。ページ内の未使用領域の比率を省略すると，30（%）が仮定されます。

SUBPAGE NUMBER 句でサブページ分割数を指定している場合，ページ内の未使用領域の比率は，サブページ内の未使用領域の比率になります。例えば，30 を指定した場合，サブページ内の未使用領域の比率が 30%になります。

また，セグメント内の空きページの比率を 0～50（%）の範囲で指定できます。セグメント内の空きページの比率を省略すると，10（%）が仮定されます。

《規則》
仮想ルートレコード型，およびすべての子レコード型で同じ値を指定してください。

(k) WITHIN

WITHIN {格納レコード用 RD エリア名
| (格納レコード用 RD エリア名)
| ((格納レコード用 RD エリア名) {格納条件}
{, (格納レコード用 RD エリア名) 格納条件} …)}

格納レコード用 RD エリア名
～< RD エリア名>((1～30))

格納条件
～構成要素名=（レコード型の RD エリア分割キー値{，レコード型の RD エリア分割キー値} …）

レコード型の RD エリア分割キー値
～X'< 16 進数字>'((1～30 バイト))
～'<英数字または下線（_）>'((1～30 バイト))※1
～< 10 進数定数>※2

～＜整数定数＞※3

注※1

先頭に下線（_）は指定できません。

注※2

10 進数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

整数部と小数部にすべて 0 を指定する場合，符号に-を指定しないでください。-0.を指定しても 0. として扱います。

なお，10 進数定数の値の範囲を超える値を指定できません。

注※3

整数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

-0 を指定しても 0 として扱います。

なお，整数定数の値の範囲を超える値を指定できません。

レコード実現値を格納する RD エリア名を指定します。

1 つの STORAGE RECORD 句に対して，WITHIN 句を必ず 1 つ指定してください。

データ型ごとのレコード型の RD エリア分割キー値の指定形式を次の表に示します。

表 11-59　データ型ごとのレコード型の RD エリア分割キー値の指定形式

レコード型の RD エリア分割キーのデータ型	レコード型の RD エリア分割キー値の指定形式			
	16 進数字	英数字，または下線(_)	10 進数定数	整数定数
CHARACTER	○	○	×	×
XCHARACTER	○	○	×	×
PACKED ［DECIMAL FIXED］	×	×	○	×
INTEGER	×	×	×	○

（凡例）

○：指定できます。

×：指定できません。

《規則》

レコードを複数の RD エリアに格納する場合の規則

- 1 レコード型に対して指定できる格納レコード用 RD エリアの数と SDB データベースの横分割の関係を次の表に示します。

表 11-60　レコード型に対して指定できる格納レコード用 RD エリアの数と SDB データベースの横分割の関係

SDB データベースの横分割方法	指定できる RD エリア数
分割指定なし	1

SDB データベースの横分割方法	指定できる RD エリア数
格納条件分割	1～256

- 格納条件分割で格納レコード用 RD エリア名を複数指定した場合は，すべての格納レコード用 RD エリア名に格納条件を指定してください。
- 同一 SDB データベース格納定義内の子レコード型に指定する格納レコード用 RD エリア名は，仮想ルートレコード型と同じ数，同じ順番ですべて指定する必要があります。
- 1 つの WITHIN 句の中では，格納レコード用 RD エリア名は一意に指定してください。
- レコード型の RD エリア分割キー値に指定できる範囲を次の表に示します。

表 11-61　レコード型の RD エリア分割キー値に指定できる範囲

レコード型の RD エリア分割キー値の指定形式	レコード型の RD エリア分割キー値	
	最小	最大
16 進数字	X'00…00'	X'FF…FF'
10 進数定数	-9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の負の数）	9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の正の数）
整数定数	－2147483648	2147483647

指定する格納レコード用 RD エリアに関する規則

- インデクスを格納する RD エリアは指定できません。
- 格納レコード用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- 指定する RD エリア名には，格納レコード長が当該 RD エリアのページ長（サブページ分割をしている場合はサブページ長）に収まる RD エリア名を指定してください。
- インナレプリカ機能を使用する場合，指定する RD エリアはインナレプリカ機能の使用条件を満たしている必要があります。インナレプリカ機能の使用条件については，「付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】」を参照してください。
- 格納レコード用 RD エリアに定義できるリソースの組み合わせについては，「表 11-58　RD エリアに定義できるリソースの組み合わせ」を参照してください。

レコードの格納条件に関する規則

- 格納条件は，仮想ルートレコード型にだけ指定できます。

格納条件に指定する構成要素に関する規則

- 1 個の WITHIN 句の中では，単一の構成要素名を指定してください。複数種類の構成要素名は指定できません。
- 構成要素名には，SDB データベース定義で指定した構成要素名を指定してください。

- 構成要素名には，データ種別 1 が K，データ種別 2 が A または M の構成要素名を指定してください。
- 4V DAM または 4V MAM の SDB データベースで，データ種別 1 が K，データ種別 2 が A または M の構成要素が存在する場合，その構成要素を格納条件の構成要素として指定してください。

格納条件のレコード型の RD エリア分割キー値に関する規則

- 1 つの WITHIN 句に対するレコード型の RD エリア分割キー値は，最大 4,000 個指定できます。
- 1 つの WITHIN 句の中で同じレコード型の RD エリア分割キー値は重複して指定できません。
- レコード型の RD エリア分割キーのデータ型が CHARACTER，または XCHARACTER の場合，指定したレコード型の RD エリア分割キー値の長さは，SDB データベース定義で指定した格納条件の構成要素の長さと一致させる必要があります。
- レコード型の RD エリア分割キーのデータ型が PACKED の場合，レコード型の RD エリア分割キーの整数部桁数，および小数部桁数の範囲内の値を指定してください。

(l) POINTER AREA SIZE

POINTER AREA SIZE 管理領域の予備領域の長さ

管理領域の予備領域の長さ

～<符号なし整数>((0～16,384))≪0≫

親レコードで管理している子レコード用の管理領域の予備領域の長さを指定します。

管理領域では，ポインタの情報や一連番号などを管理しています。

将来の親子集合の拡張を考慮して，親子集合 1 個当たり，約 40 バイトの予備領域を指定しておくことを推奨します。

(m) SECONDARY INDEX

SECONDARY INDEX 二次インデクス名

二次インデクス名

～<識別子>((1～30 バイト))

二次インデクス名を指定します。

《規則》

- 二次インデクスは，仮想ルートレコード型には指定できません。
- 二次インデクス名は，同一スキーマ内で一意になるように指定してください。シーケンシャルインデクス名も含めて一意になるようにしてください。
- 二次インデクス名には，シーケンシャルインデクス名とは別の名称を指定してください。
- SECONDARY 句は，1 つの STORAGE RECORD 句内で 2 つまで指定できます。詳細については，「表 11-62　二次インデクスの定義規則」を参照してください。

(n)　ORDER KEY

ORDER KEY ASCENDING 構成要素名，構成要素名〔，構成要素名〕…

構成要素名

～＜識別子＞((1～30 バイト))

二次インデクスのキーとなる構成要素名を指定します。

ORDER KEY 句については，「11.6.2(3)(f)　ORDER KEY」を参照してください。

《規則》

- 1 つの STORAGE RECORD 内に複数のインデクス（シーケンシャルインデクス，および二次インデクス）を定義する場合，すべての構成要素名の指定順序が同一のインデクスを複数定義できません。
- 二次インデクスについては，「表 11-62　二次インデクスの定義規則」に従って定義してください。
- キー項目を含む場合，SDB データベース定義の KEY 句に ASCENDING を指定してください。
- 1 つの二次インデクスに指定できる構成要素の数は，2～8 個です。
- 二次インデクスの構成要素の指定順序は，次の表の「指定する構成要素」の記述順に従ってください。

表 11-62　二次インデクスの定義規則

レコード型種別	一連番号最大値※1	ユーザキー※2	二次インデクスの指定	指定する構成要素
仮想ルートレコード型	－	－	×	－
子レコード型	1 以上	なし	1 本必須	データベースキー 1～7，一連番号
		あり	2 本必須	• データベースキー 1～7，一連番号 • データベースキー 1～7，ユーザキー
	0	－	×	－

（凡例）

×：指定できません。

－：該当しません。

注※1

SDB データベース定義の OCCURRENCE NUMBER 句の指定値，または省略値です。

注※2

SDB データベース定義の TYPE 句のデータ種別 1 に U，データ種別 2 に K を指定した構成要素です。

二次インデクスに指定する構成要素の定義例を次に示します。

- 4V MAM の SDB データベースで，ユーザキーありの場合

SDB データベース定義

```
                              :
RECORD REC1
    2 DBKEY_L1   CHARACTER  5     TYPE D,L
    2 DBKEY
      3 DBKEY_L2   XCHARACTER 1   TYPE K,L
      3 DBKEY_L3   XCHARACTER 1   TYPE K,L
      3 DBKEY_L4   CHARACTER  3   TYPE K,R
      3 DBKEY_LN   INTEGER        TYPE K,N
    2 USERD10      CHARACTER  4   TYEP U,K
    2 USERD11      CHARACTER  10  TYPE U,D
                              :
```

SDB データベース格納定義

```
                              :
STORAGE RECORD REC1
    CLUSTERED SET1
    WITHIN  MAMX01
    SECONDARY INDEX  REC1_SI1
      ORDER KEY ASCENDING  DBKEY_L1, DBKEY_L2, DBKEY_L3,
                           DBKEY_L4, DBKEY_LN
      WITHIN MAMX01_I
    SECONDARY INDEX  REC1_SI2
      ORDER KEY ASCENDING  DBKEY_L1, DBKEY_L2, DBKEY_L3,
                           DBKEY_L4, USERD10
      WITHIN MAMX01_I
                              :
```

(o) PCTFREE

PCTFREE ページ内の未使用領域の比率

PCTFREE 句については，「11.6.2(3)(g) PCTFREE」を参照してください。

(p) WITHIN

WITHIN {二次インデクス用 RD エリア名
| (二次インデクス用 RD エリア名)
| ((二次インデクス用 RD エリア名)〔, (二次インデクス用 RD エリア名)〕…)}

二次インデクス用 RD エリア名

～< RD エリア名>((1～30 バイト))

二次インデクスを格納する RD エリア名を指定します。

1 つの SECONDARY INDEX 句で，WITHIN 句を必ず 1 つ指定してください。

《規則》

二次インデクスを複数 RD エリアに格納する場合の規則

- 1 つの二次インデクスに指定できる二次インデクス用 RD エリアの個数は，シーケンシャルインデクス用 RD エリアの個数と同じにしてください。シーケンシャルインデクス用 RD エリアの個数については，「表 11-57 シーケンシャルインデクス用 RD エリアとして指定できる RD エリア数と SDB データベースの横分割の関係」を参照してください。

- 二次インデクス用 RD エリアは，格納レコード用 RD エリアと同じ数を指定する必要があります。
- 1 つの WITHIN 句の中では，二次インデクス用 RD エリア名は一意に指定してください。
- 格納レコード用 RD エリアと対応する二次インデクス用 RD エリアは，同じサーバ内の RD エリアである必要があります。

指定する二次インデクス用 RD エリアに関する規則

- 格納レコード用 RD エリア名として指定した RD エリアは指定できません。
- 二次インデクス用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- 二次インデクス用 RD エリアとして指定する RD エリアには，1 つの RD エリアにシーケンシャルインデクスと二次インデクスを合わせて，インデクスを最大 500 個指定できます。
- インナレプリカ機能を使用する場合，指定する RD エリアはインナレプリカ機能の使用条件を満たしている必要があります。インナレプリカ機能の使用条件については，「付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】」を参照してください。
- 二次インデクス用 RD エリアに定義できるリソースの組み合わせについては，「表 11-58　RD エリアに定義できるリソースの組み合わせ」を参照してください。

(q)　SET

SET 親子集合型名

～<識別子>((1～30 バイト))

親レコード型，子レコード型の親子集合型名を指定します。

《規則》

SDB データベース定義の SET 句で指定した親子集合型名と同じ個数，同じ順番で指定してください。

(r)　OWNER POINTER FOR

OWNER POINTER FOR FIRST LAST MEMBER

親レコードに持たせるポインタを指定します。

親レコードに FIRST ポインタ，および LAST ポインタを持たせる場合に指定します。

(s)　MEMBER POINTER FOR

MEMBER POINTER FOR NEXT PRIOR TENANT

子レコードに持たせるポインタを指定します。

NEXT PRIOR

子レコードに NEXT ポインタ，および PRIOR ポインタを持たせる場合に指定します。

(t) SDBOPTION

SDBOPTION

SDB データベース格納定義に関するオプションを指定します。

(u) KEYDEF

KEYDEF 構成要素名

構成要素名

～<識別子>((1～30 バイト))

キーの項目である構成要素名を指定します。

KEYDEF 句の用途を次に示します。

- API（レコードの検索，またはレコードの格納）の実行時，または HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時に，データのキー値を基にアクセス対象のレコードを決めることができます。
- 不正データをチェックするために指定できます。

《規則》

- 仮想ルートレコード型内のデータ種別 1 が D の構成要素を必ず指定してください。
- 構成要素名は，SDB データベース定義の仮想ルートレコード型でデータベースキーとして定義されている必要があります。このため，1 つの SDB データベース格納定義で指定できる KEYDEF 句の最大数は，SDB データベース定義の仮想ルートレコード型で定義できるデータベースキーの基本項目数となります。
- KEYDEF 句を複数指定する場合は，SDB データベース定義の定義順に構成要素名を指定してください。
- KEYDEF 句を指定した場合は，DATA 句を指定する必要があります。
- 4V MAM の SDB データベースの場合，データベースキーの構成要素名を SDB データベース定義の定義順にすべて指定する必要があります。
- SDB データベース定義の SDBOPTION 句下の FUNCTION 句に FORMAT USE を指定して，フォーマットライト機能を使用する場合，データベースキーの構成要素名を定義順にすべて指定する必要があります。
- SDB データベース種別と KEYDEF 句の指定の組み合わせについては，次の表を参照してください。

表 11-63　SDB データベース種別と KEYDEF 句の指定の組み合わせ

SDB データベース種別	KEYDEF 句，DATA 句の指定	レコード型名	用途
4V AFM	○	△※	次の場合に使用します。 • フォーマットライト機能使用時

SDB データベース種別	KEYDEF 句，DATA 句の指定	レコード型名	用途
			• データロード時 • API によるデータ操作時 • 不正データのチェック時

（凡例）

○：必須で指定します。

△：省略できます。

注※

仮想ルートレコード型を除いて，2 個以上の格納レコードがある場合，どちらかの KEYDEF 句内で最低 1 つは指定する必要があります。

- レコード型の RD エリア分割キーとして，シーケンシャルインデクスの先頭以外の構成要素を指定したレコード型に対して，RD エリアを指定して検索をする場合は，次に示す条件で指定してください。

  ・レコード型の RD エリア分割キーに指定した構成要素よりも前にある構成要素をすべて KEYDEF 句に指定する。

  ・指定した KEYDEF 句ごとのデータベースのキー値は 1 つだけ指定する。

  RD エリアを指定した検索については，マニュアル「TP1/Financial Service Platform 使用の手引」の「SDB ハンドラ機能」の「店群順アクセス」および「DAM 順アクセス」を参照してください。

- 構成要素のデータ種別と KEYDEF 句の指定の組み合わせを次の表に示します。

表 11-64　構成要素のデータ種別と KEYDEF 句の指定の組み合わせ

データ種別 1	データ種別 2	KEYDEF 句下の DATA 句の格納レコード名（子レコード型）の指定
D	L	×
	A	×
	R	×
	M	×
K	L	×
	A	×
	R	○
	M	○
	上記以外	×
上記以外		×

（凡例）

○：指定が必要です。

×：指定できません。

(v) DATA

DATA データベースのキー値〔，格納レコード名〕…

データベースのキー値

～X'＜16 進数字＞'((1～30 バイト))

～'＜英数字または下線（_）＞'((1～30 バイト))※1

～＜10 進数定数＞※2

～＜整数定数＞※3

注※1

先頭に下線（_）は指定できません。

注※2

10 進数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

整数部と小数部にすべて 0 を指定する場合，符号に-を指定しないでください。-0.を指定しても 0.として扱います。

なお，10 進数定数の値の範囲を超える値は指定できません。

注※3

整数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

-0 を指定しても 0 として扱います。

なお，整数定数の値の範囲を超える値は指定できません。

格納レコード名

～＜識別子＞((1～30 バイト))

データベースのキー値と格納先の格納レコード名を指定します。

KEYDEF 句の構成要素名で指定した，構成要素のデータ型ごとのデータベースのキー値の指定形式を次の表に示します。

表 11-65　KEYDEF 句の構成要素名で指定した，構成要素のデータ型ごとのデータベースのキー値の指定形式

KEYDEF 句の構成要素名で指定した構成要素のデータ型	データベースのキー値の指定形式			
	16 進数字	英数字，または下線（_）	10 進数定数	整数定数
CHARACTER	○	○	×	×
XCHARACTER	○	○	×	×
PACKED 〔DECIMAL FIXED〕	×	×	○	×
INTEGER	×	×	×	○

（凡例）

○：指定できます。

×：指定できません。

KEYDEF 句の構成要素名で指定した構成要素のデータ型が CHARACTER または XCHARACTER の場合，データベースのキー値には，SDB データベースのキー値を KEYDEF 句の構成要素名で指定した構成要素のデータ属性とバイト数を合わせて指定します。

KEYDEF 句の構成要素名で指定した構成要素のデータ型が PACKED の場合，データベースのキー値には KEYDEF 句の構成要素名で指定した構成要素の整数部桁数，および小数部桁数の範囲内の値を指定します。

SDB データベース定義の SDBOPTION 句下の FUNCTION 句に FORMAT USE を指定して，フォーマットライト機能を使用する場合，フォーマットライト機能ではすべての組み合わせで，データベースのフォーマットライトを行います。

格納レコード名は，SDB データベースのキー値で格納先の子レコード型名を決める場合に指定します。

KEYDEF 句下の DATA 句でのデータベースのキー値の指定方法を次の表に示します。

表 11-66　KEYDEF 句下の DATA 句でのデータベースのキー値の指定方法

データ型	データベースのキー値	KEYDEF 句下の DATA 句でのデータベースのキー値の指定例	SDB ディレクトリ情報の設定内容
CHARACTER(4)	ABCD	'ABCD'	41424344
CHARACTER(4)	abcd	'abcd'	61626364
CHARACTER(4)	1000	'1000'	31303030
XCHARACTER(4)	AB12	X'41423132'	41423132
PACKED(2,1)	15.5	15.5	155C※
PACKED(2,2)	15.5	15.5	01550C
PACKED(3,2)	15.5	15.5	01550C
INTEGER	10	10	0A000000

注※

正の値または 0 の場合は，符号部を X'C'で設定します。負の値の場合は，符号部を X'D'で設定します。

《規則》

データベースのキー値に関する規則

- 4V MAM の SDB データベースの場合，レコード分割キー以外のデータベースキーのキー値は 1 つだけ指定できます。
- 1 つの KEYDEF 句で，データベースのキー値が重複することはできません。
- データベースのキー値を 16 進数字で指定する場合，データベースのキー値の長さはバイト単位で指定してください。
- KEYDEF 句で指定した構成要素が，レコード型の RD エリア分割キーの構成要素として指定されている場合，格納条件のレコード型の RD エリア分割キー値で指定された値と一致させてください。

- DATA 句を複数指定する場合，1 つの KEYDEF 句内でデータベースのキー値を昇順に指定してください。データベースのキー値を昇順に指定する例を次の表に示します。

表 11-67　データベースのキー値を昇順に指定する例

データ型	指定順序	KEYDEF 句下の DATA 句で指定するデータベースのキー値	SDB ディレクトリ情報の設定内容	指定内容の意味
CHARACTER(4)	1	'ABCD'	41424344	文字列：ABCD※
	1	X'41424344'	41424344	文字列：ABCD※
	2	'abCD'	61624344	文字列：abCD
	3	'abcd'	61626364	文字列：abcd
XCHARACTER(2)	1	X'0030'	0030	文字列：0
	2	X'2B30'	2B30	文字列：＋0
	3	X'2B31'	2B31	文字列：＋1
	4	X'2D30'	2D30	文字列：－0
	5	X'2D31'	2D31	文字列：－1
PACKED(2,2)	1	-11.15	01115D	数字：-11.15
	2	-10.15	01015D	数字：-10.15
	3	10.15	01015C	数字：10.15
	4	11.15	01115C	数字：11.15
INTEGER	1	-3	FDFFFFFF	数字：-3
	2	-1	FFFFFFFF	数字：-1
	3	0	00000000	数字：0
	4	1	01000000	数字：1
	5	3	03000000	数字：3

注※

指定内容の意味が同じとなるデータベースのキー値は指定できません。

格納レコード名に関する規則

- 1 つの KEYDEF 句に対して，格納レコード名が存在する DATA 句と存在しない DATA 句を同時に指定することはできません。
- 格納レコード名は，1 つの SDB データベース定義に対して 1 つの KEYDEF 句にまとめて指定してください。
- 格納レコード名には，子レコード型名を指定してください。
- データ種別 1 が K で，データ種別 2 が R または M の構成要素を指定した KEYDEF 句内で，すべての子レコード型名を指定する必要があります。

- 4V MAM の SDB データベースの場合，複数のキー値に対して同じ格納レコード名を指定できません。

そのほかの規則

- 1 つの KEYDEF 句で指定できる DATA 句は，最大 32,767 個です。
- KEYDEF 句の構成要素名で指定した構成要素が，データ種別 1 が D の構成要素の場合，DATA 句は 1 つだけ指定してください。

(w) END STORAGE SCHEMA

END STORAGE SCHEMA

SDB データベース格納定義の終了を示します。

11.7　SDB データベース定義，SDB データベース格納定義【SD FMB】

SD FMB の SDB データベースを定義する場合の，SDB データベース定義および SDB データベース格納定義の指定形式と各オペランドの指定内容について説明します。

11.7.1　SDB データベース定義【SD FMB】

*ENTRY DICTIONARY 文に指定する SDB データベース定義について説明します。

(1)　機能

SDB データベースの論理構造を定義します。

(2)　形式

項番	形式
(a)	SCHEMA SDB データベース名
(b)	DBTYPE SD FMB
(c)	RECORD レコード型名
(d)	{レベル番号｜ITEM}　構成要素名
(e)	〔{CHARACTER 長さ ｜XCHARACTER 長さ ｜PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕 ｜INTEGER ｜SMALLINT}
(f)	TYPE データ種別 1，データ種別 2
(g)	〔OCCURS 繰り返し回数〕
	〕…
(h)	〔RECORDID レコード識別コード〕
	…
(i)	〔SET 親子集合型名
(j)	OWNER 親レコード型名
(k)	ORDER　{FIRST ｜LAST ｜SORTED DUPLICATES　{PROHIBITED ｜ FIRST ｜ LAST}}
(l)	MEMBER 子レコード型名
(m)	INSERTION AUTOMATIC

項番	形式
(n)	RETENTION FIXED
(o)	〔KEY {ASCENDING | DESCENDING} 構成要素名〔, 構成要素名〕… 〔 {ASCENDING | DESCENDING} 構成要素名〔, 構成要素名〕…〕…〕
(p)	〔SETOPTION
(q)	〔OCCURRENCE NUMBER 一連番号の最大値
(r)	〔WARNING {一連番号使用比率 1 〔, 一連番号使用比率 2 〔, 一連番号使用比率 3〕〕 | 0}〕
	〕 〕 〕…
(s)	〔SDBOPTION
(t)	〔OCCURRENCE WARNING {一連番号使用比率 1 〔, 一連番号使用比率 2 〔, 一連番号使用比率 3〕〕 | 0}〕
	〕
(u)	END SCHEMA

(3) 定義句の指定順序

SDB データベース定義の定義句の指定順序には，次に示す規則があります。

- 必須の定義，または必須の定義と省略できる定義の間は，上記の形式で示す順序で指定してください。
- RECORD 句および SET 句の指定順序は，一筆書き順の規則に従ってください。一筆書き順の規則については，「図 11-11 RECORD 句および SET 句の指定順序（一筆書き順）の規則」を参照してください。

(4) オペランドの説明

SDB データベース定義の各オペランドの指定内容を説明します。

(a) SCHEMA

SCHEMA SDB データベース名

～<識別子>((1～30 バイト))

SDB データベースの SDB データベース名を指定します。

すでに使用している SDB データベース名は指定できません。

(b) DBTYPE

定義する SDB データベースの SDB データベース種別を指定します。

DBTYPE SD FMB

SDB データベース種別が SD FMB の SDB データベースを定義する場合は，DBTYPE 句に SD FMB を指定します。

SD FMB の SDB データベースとは，階層構造を持つデータベースで，メインフレームの XDM/SD がサポートしている階層型モデルのデータベースに相当します。詳細については，「2.3.3 SDB データベース」を参照してください。

注意事項

SDB データベース種別が 4V（4V FMB または 4V AFM）の SDB データベースと SD FMB の SDB データベースをシステム内で混在して定義することはできません。混在させた場合，動作を保証しません。

(c) RECORD

RECORD レコード型名

～<識別子>((1～30 バイト))

レコード型名を指定します。

《規則》

- レコード型名は，同一スキーマ内で一意になるように指定してください。
- 1 つの SDB データベース定義内で指定できる RECORD 句は，1～128 個です。

(d) レベル番号，ITEM

{レベル番号｜ITEM}　構成要素名

構成要素に関する情報を指定します。1 つの RECORD 句内での構成要素の階層レベルを指定します。

- すべての構成要素が基本項目の場合，ITEM を指定できます。
- 1 つのレコード型内で，レベル番号と ITEM は同時に指定できません。

指定方法の詳細については，「表 11-71 ルートレコード型の構成要素の指定」～「表 11-73 子レコード（3 階層以上）の構成要素の指定」を参照してください。

レベル番号

～<符号なし整数>((2～49))

1 つの RECORD 句内での構成要素の階層レベルを指定します。

階層が深くなるに従ってレベル番号を増やすように指定してください。レベル番号は連続した値である必要はありません。

階層が浅くなるときは，すでに指定したレベル番号にしてください。

ITEM

すべての構成要素が基本項目の場合に指定できます。

構成要素名

～<識別子>((1～30 バイト))

構成要素名を指定します。

《規則》

- 構成要素名は，RECORD 句内で一意にしてください。
- OCCURS 句を指定した場合，繰り返し回数分の構成要素を HiRDB/SD が生成します。このとき，次に示す形式に従った構成要素名が付けられます。この構成要素名の長さが 31 バイト以上になった場合，pdsdbdef コマンドがエラーになります。

  構成要素名(nnnn1〔,nnnn2〔,nnnn3〔,nnnn4〕〕〕)

  nnnn1～nnnn4：0～9 で構成される 4 桁の文字列です。0001 から繰り返し回数まで 1 ずつ増加します。

  OCCURS 句を指定した構成要素の下位の階層レベルの構成要素にも OCCURS 句を指定した場合，下位の階層の構成要素の繰り返し回数がコンマ（,）で区切って付加されます。
- RECORD 句内に指定できる構成要素の合計長は最大 30,000 バイトです。また，RECORD 句内に指定できる構成要素の個数は最大 30,000 個です。

  なお，OCCURS 句を指定した場合，繰り返し回数分生成された構成要素の長さと個数も含まれます。
- 各レコード型の最初の基本項目の構成要素のデータ種別 1 には，K を指定してください。
- 上位レコードのキーの構成要素，および一連番号の構成要素は，下位レコードの構成要素として，データ型とデータ種別を同じように指定してください。
- ユーザデータ（データ種別 1，2（TYPE U,D））の構成要素は，幾つかの構成要素をまとめて，ユーザ任意の集団項目内に指定できます。

  ユーザファイル通番（データ種別 1，2（TYPE U,F））の構成要素は，集団項目内に指定できません。

(e) データ型

{CHARACTER 長さ

｜XCHARACTER 長さ

｜PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕

｜INTEGER

｜SMALLINT}

構成要素のデータ型を指定します。構成要素に指定できるデータ型を次の表に示します。

表 11-68　構成要素に指定できるデータ型

データ種別	データ型	データ形式
文字列	CHARACTER	固定長文字列
16 進形式の文字列	XCHARACTER	16 進固定長文字列
数値	PACKED 〔DECIMAL FIXED〕	符号付きパック形式 10 進データ
	INTEGER	整数（4 バイトの 2 進形式）
	SMALLINT	整数（2 バイトの 2 進形式）

CHARACTER 長さ

～<符号なし整数>((1～30,000))

文字列データの長さを指定します。

XCHARACTER 長さ

～<符号なし整数>((1～30,000))

16 進形式の文字列データの長さを指定します。

PACKED 〔DECIMAL FIXED〕 整数部桁数〔，小数部桁数〕

整数部桁数

～<符号なし整数>((0～38))

小数部桁数

～<符号なし整数>((0～38))

符号付きパック形式の 10 進データの整数部桁数，および小数部桁数を指定します。

整数部桁数と小数部桁数の和は，1～38 になるようにしてください。

INTEGER

値の範囲が-2,147,483,648～2,147,483,647 の整数を指定します。長さは指定できません。

SMALLINT

値の範囲が-32,768～32,767 の整数を指定します。長さは指定できません。

《規則》

- 一連番号の構成要素には，必ず INTEGER を指定してください。
- 基本項目の場合，データ型および TYPE 句を指定してください。
- 集団項目の場合，データ型および TYPE 句を省略してください。

(f)　TYPE

TYPE データ種別 1，データ種別 2

構成要素のデータ種別を指定します。

データ種別 1

～<英字>((1 バイト))

データベースキーかユーザデータかを区別するための種別を指定します。

データ種別 2

～<英字>((1 バイト))

データ種別 1 をさらに分類するための詳細な種別を指定します。

TYPE 句で指定するデータ種別の指定値と意味を次の表に示します。

表 11-69　データ種別の指定値と意味

データ種別	指定値	意味
データ種別 1	K	データベースキーで，次に該当する場合の構成要素 • データベースキー • 上位レコードの一連番号 • 自レコードの一連番号
	U	ユーザデータ
データ種別 2	A	レコード型の RD エリア分割キーとする構成要素（SDB データベースを横分割する際，レコードの格納 RD エリアを決める構成要素）
	N	自レコードの一連番号の構成要素 • データ種別 1 が K の場合，一連番号属性あり
	P	上位レコードの一連番号の構成要素
	L	そのほかのデータベースキーの構成要素
	F	ユーザファイル通番の構成要素
	D	そのほかのユーザデータの構成要素

《規則》

指定順序と記述の規則

- K，U の順に，必要な種別を指定してください。
- ユーザデータのあとにデータベースキーを指定することはできません。

そのほかの規則

- データ種別 1 が K で，データ種別 2 が A の構成要素は，ルートレコード型と子レコード型で同一名称にしてください。
- データ種別 1 が K で，データ種別 2 が A の構成要素は，データの長さを 1～30 バイトの範囲で指定してください。
- データ種別 1 が K で，データ種別 2 が A の構成要素のデータ型には，SMALLINT を指定できません。
- RECORD 句内では，データ種別 1 の指定に関係なく，データ種別 2 が A の構成要素は，それぞれ最大 1 つ指定できます。
- データ種別 1 が K で，データ種別 2 が N の場合，一連番号属性があります。

- 上位レコードで一連番号属性ありの構成要素（データ種別 1，2（K，N））は，下位レコードの構成要素で，データ種別 1，2（K，P）を指定してください。
- データ種別 2 が F の構成要素はレコード型内の最後の構成要素に 1 つだけ指定できます。
- 構成要素のデータ種別 1 とデータ種別 2 で，指定できる組み合わせを次の表に示します。

表 11-70　構成要素のデータ種別 1 とデータ種別 2 に指定できる組み合わせ

データ種別 1 の指定	データ種別 2 に指定できる値
K の場合	L
	A
	N
	P
U の場合	D
	F※

注※

データ種別 1，2 の指定が U，F の場合，レコード型内で最後に指定してください。

- ルートレコード型と子レコード型（2 階層），および子レコード型（3 階層以上）の構成要素の指定を次の表に示します。

  指定順序や指定する個数，データ種別の組み合わせに誤りがある場合はエラーとなります。なお，1 つの項番の中で基本項目と集団項目を複数指定できる場合の，指定順序は任意です。

表 11-71　ルートレコード型の構成要素の指定

指定順	項目		説明	データ種別	
				1	2
1	データベースキーの構成要素名		集団項目として指定します。	－	－
2	データベースキー 1～7	構成要素名	• 各データベースキーを，指定順 1 で指定した集団項目下の基本項目として，同じレベル番号で指定します。 • 集団項目とはしないでください。	K	L
		個数	1～7		A※
3	ユーザデータの構成要素名		ユーザデータの構成要素名を指定します。	－	－
				U	D
					F

（凡例）

－：該当しません。

注※

データベースキー 1 の構成要素にだけ指定できます。データベースキー 1 以外の構成要素には指定できません。

表 11-72　子レコード型（2 階層）の構成要素の指定

指定順	項目		説明	データ種別	
				1	2
1	データベースキー 1～7	構成要素名	• ルートレコード型で指定したデータベースキーを，同じレベル番号の基本項目として指定します。 • 集団項目とはしないでください。	K	L
		個数	1～7		A※
2	一連番号		• 構成要素名を「DBKEY」とします。 • 基本項目として指定します。	K	N
3	ユーザデータの構成要素名		ユーザデータの構成要素名を指定します。	－	－
				U	D
					F

（凡例）

－：該当しません。

注※

データベースキー 1 の構成要素にだけ指定できます。データベースキー 1 以外の構成要素には指定できません。

表 11-73　子レコード（3 階層以上）の構成要素の指定

指定順	項目		説明	データ種別	
				1	2
1	データベースキー 1～7	構成要素名	• 上位の子レコード型で指定したデータベースキーを，同じレベル番号の基本項目として指定します。 • 集団項目とはしないでください。	K	L
		個数	1～7		A※
2	上位の一連番号	構成要素名	• 構成要素名は「DBKEY」以外とします。 • 上位レコード型で指定した一連番号を，基本項目として指定します。	K	P
		個数	上位の子レコード型で指定した一連番号の構成要素数と同じになります（階層－2）。		
3	一連番号		• 構成要素名を「DBKEY」とします。 • 基本項目として指定します。	K	N
4	ユーザデータの構成要素名		ユーザデータの構成要素名を指定します。	－	－
				U	D
					F

（凡例）

－：該当しません。

注※

データベースキー 1 の構成要素にだけ指定できます。データベースキー 1 以外の構成要素には指定できません。

(g) OCCURS

OCCURS 繰り返し回数

～<符号なし整数>((1～4,000))

構成要素の繰り返し回数を指定します。

OCCURS 句を指定した場合，ここで指定した繰り返し回数分の構成要素が生成されます。

《規則》

- OCCURS 句は，ユーザデータの構成要素を指定した集団項目またはユーザデータ（データ種別 1，2（TYPE U,D））である基本項目に指定できます。
- OCCURS 句を指定した構成要素の下位の構成要素に OCCURS 句を指定する場合，OCCURS 句は 4 次元まで指定できます。

  （例）

```
03 UDATA1 OCCURS 2                                          ←1次元
   05 UDATA1A OCCURS 3                                      ←2次元
      07 UDATA1A1 CHARACTER 1 TYPE U,D
      07 UDATA1A2 OCCURS 3                                  ←3次元
         09 UDATA1A21 CHARACTER 1 TYPE U,D  OCCURS 2        ←4次元
03 UDATA2 CHARACTER 76 TYPE U,D
```

- OCCURS 句を指定したときの構成要素名の生成例を次に示します。

  （例）

■SDBデータベース定義文の指定

```
SCHEMA FMBO1
RECORD RECA
03 KEYDATA
     05 KEYDATA1 CHARACTER 4 TYPE K,L
     05 KEYDATA2 INTEGER     TYPE K,L
03 UDATA1 OCCURS 2
     05 UDATA1A CHARACTER 2 TYPE U,D OCCURS 4
     05 UDATA1B
          07 UDATA1B1 CHARACTER 1 TYPE U,D
          07 UDATA1B2 CHARACTER 1 TYPE U,D
03 UDATA2 CHARACTER 76 TYPE U,D
```

■生成されるSDBデータベース定義

```
SCHEMA FMBO1
RECORD RECA
03 KEYDATA
     05 KEYDATA1 CHARACTER 4 TYPE K,L
     05 KEYDATA2 INTEGER     TYPE K,L
03 UDATA1(0001)
     05 UDATA1A(0001,0001) CHARACTER 2 TYPE U,D   <- UDATA1Aを4回繰り返す
     05 UDATA1A(0001,0002) CHARACTER 2 TYPE U,D
     05 UDATA1A(0001,0003) CHARACTER 2 TYPE U,D
     05 UDATA1A(0001,0004) CHARACTER 2 TYPE U,D
     05 UDATA1B(0001)
          07 UDATA1B1(0001) CHARACTER 1 TYPE U,D
          07 UDATA1B2(0001) CHARACTER 1 TYPE U,D
03 UDATA1(0002)                                  UDATA1を2回繰り返す
     05 UDATA1A(0002,0001) CHARACTER 2 TYPE U,D   <- UDATA1Aを4回繰り返す
     05 UDATA1A(0002,0002) CHARACTER 2 TYPE U,D
     05 UDATA1A(0002,0003) CHARACTER 2 TYPE U,D
     05 UDATA1A(0002,0004) CHARACTER 2 TYPE U,D
     05 UDATA1B(0002)
          07 UDATA1B1(0002) CHARACTER 1 TYPE U,D
          07 UDATA1B2(0002) CHARACTER 1 TYPE U,D
03 UDATA2 CHARACTER 76 TYPE U,D
```

(h) RECORDID

RECORDID レコード識別コード

～X'＜16進数字＞'((1バイト))

レコード型を識別するコードとして，レコード識別コードを指定します。

《規則》

- レコード識別コードを1バイトで指定します。
- X'00'以外は重複できません。

(i) SET

SET 親子集合型名

～＜識別子＞((1～30バイト))

親子集合型名を指定します。

《規則》

- 親子集合型名は，SDBデータベース定義内で一意にしてください。
- ルートレコード型の数は1つにしてください。

- 親レコードから子レコードへの親子集合型をたどった場合に，元のレコードにたどり着くような親子集合型は定義できません。
- 1 つの SDB データベース定義内に指定できる親子集合型数は，0～127 個です。指定できる親子集合の階層は，1～15 です。

(j) OWNER

OWNER 親レコード型名

～<識別子>((1～30 バイト))

親子集合の親レコード型名を指定します。

《規則》

親レコード型名には，RECORD 句で定義したレコード型名を指定してください。

(k) ORDER

ORDER {FIRST | LAST | SORTED DUPLICATES {PROHIBITED | FIRST | LAST}}

DML によるレコードの格納時（STORE），または HiRDB/SD データベース作成ユティリティ（pdsdblod）で，子レコード実現値を親子集合内の任意の実現値に格納する際の位置（親子集合内の子レコードへの挿入順序）を指定します。

FIRST

子レコード実現値群の先頭に格納します。

LAST

子レコード実現値群の最後に格納します。

SORTED DUPLICATES PROHIBITED

子レコード型で指定したキー項目（KEY 句に指定した構成要素）の値に従って，昇順または降順に格納します。

同じ親レコードに対して，同じ値のキー項目を持つ子レコードが格納済みの場合，格納できません。

SORTED DUPLICATES FIRST

子レコード型で指定したキー項目（KEY 句に指定した構成要素）の値に従って，昇順または降順に格納します。

同じ親レコードに対して，同じ値のキー項目を持つ子レコードが格納済みの場合，重複している子レコード実現値群の先頭に格納します。

SORTED DUPLICATES LAST

子レコード型で指定したキー項目（KEY 句に指定した構成要素）の値に従って，昇順または降順に格納します。

同じ値のキー項目を持つ子レコードが格納済みの場合，重複している子レコード実現値群の最後に格納します。

《規則》

- 1 つの SET 句の指定には，必ず ORDER 句を指定してください。
- ORDER 句の指定によって，削除された一連番号を再利用するかどうかをシステムが決定します。一連番号の再利用可否を次の表に示します。一連番号の再利用については，「11.5.1(5)(s) REUSE」を参照してください。

表 11-74　一連番号の再利用可否

ORDER 句の指定値	一連番号の再利用可否
ORDER FIRST	再利用しない
ORDER LAST	再利用する
ORDER SORTED DUPLICATES PROHIBITED	再利用しない
ORDER SORTED DUPLICATES FIRST	
ORDER SORTED DUPLICATES LAST	

《注意事項》

SORTED DUPLICATES FIRST，または SORTED DUPLICATES LAST に指定した構成要素を二次インデクスの構成要素に含む場合，インデクスキー値の重複エラー（SQLCODE= -803）によって，重複したデータが格納できないことがあります。

(l)　MEMBER

MEMBER 子レコード型名

～<識別子>((1～30 バイト))

親子集合内の子レコード型名を指定します。

《規則》

- 子レコード型名には，RECORD 句で定義したレコード型名を指定してください。ただし，親子集合の親レコード型名と重複する子レコード型名は指定できません。
- 1 つの SET 句には，必ず MEMBER 句を指定してください。

(m)　INSERTION

INSERTION AUTOMATIC

子レコードを親子集合に格納するときの挿入方法を指定します。

AUTOMATIC

子レコードのレコード実現値がデータベースに格納されると，自動的に親レコードが属する親子集合型の親子集合実現値に格納されます。

(n) RETENTION

RETENTION FIXED

子レコードを親子集合から切り離すまでの属性を指定します。

FIXED

レコードが，ある親子集合の子レコードになったレコードは，データベースから削除されるまで，その親子集合の子レコードでなければなりません。

(o) KEY

KEY {ASCENDING｜DESCENDING} 構成要素名〔，構成要素名〕…

〔{ASCENDING ｜ DESCENDING} 構成要素名〔,構成要素名〕…〕…

ORDER 句で SORTED DUPLICATES を指定した親子集合の子レコードを構成要素の値に従って，昇順に格納するか，または降順に格納するかを指定します。指定した構成要素をキー項目とします。

ASCENDING

子レコードを昇順に格納します。

DESCENDING

子レコードを降順に格納します。

構成要素名

～<識別子>((1～30 バイト))

格納時のキーになる構成要素名を指定します。

《規則》

- キーの構成要素長の合計は 254 バイト以内としてください。
- 指定できる構成要素は，データ種別 1 がユーザデータ（U），データ種別 2 がその他のユーザデータ（D）の構成要素だけです。
- 構成要素名には，RECORD 句で定義した構成要素名を指定してください。
- 1 つの KEY 句に構成要素名を 17 個以上指定できません。
- KEY 句に同じ構成要素名を重複して指定できません。
- 指定した構成要素，または指定した構成要素の上位の構成要素に OCCURS 句を指定できません。
- ORDER 句で FIRST または LAST を指定した場合，KEY 句は指定できません。
- ORDER 句で SORTED DUPLICATES を指定した場合，必ず KEY 句を指定してください。

(p) SETOPTION

SETOPTION

親子集合に関するオプションを指定します。

(q) OCCURRENCE NUMBER

OCCURRENCE NUMBER 一連番号の最大値

～<符号なし整数>((1～2,147,483,647))≪2,147,483,647≫

一連番号の最大値を指定します。

《規則》

WARNING 句を指定する場合は，この句も指定する必要があります。

(r) WARNING

WARNING {一連番号使用比率 1〔，一連番号使用比率 2〔，一連番号使用比率 3〕〕 | 0}

一連番号の使用状況を監視する場合に，一連番号の使用比率を％単位で指定します。

一連番号使用比率 1，一連番号使用比率 2，一連番号使用比率 3

～<符号なし整数>((50～99))

割り当て済み一連番号の割合が一連番号使用比率 1～3 で指定した値に達した場合※に，メッセージを 1 回ずつ出力します。0 を指定した場合は，割り当て済みの一連番号の監視をしません。

注※　監視値＝↑一連番号の最大値×一連番号使用比率n↑

この句と SDBOPTION 句下の OCCURRENCE WARNING 句の両方を指定した場合，この句での指定が有効となります。この場合，一連番号使用比率の指定値の個数が異なっていても，双方で補ってメッセージを出力することはありません。

この句を省略した場合，SDBOPTION 句下の OCCURRENCE WARNING 句の指定が有効となります。

一度に複数の一連番号使用比率に達した場合は，対応するメッセージがすべて出力されます。例えば，WARNING 句の指定を 60，70，80 とした場合に，1 回目のレコードの格納で 75％に達した場合は，指定値 60 と 70 に対応するメッセージが出力されます。

一連番号の最大値が小さいと，重複指定でなくても内部的に重複指定と同じような扱いになる場合があります。例えば，OCCURRENCE NUMBER 句が 10，WARNING 句が 97，98，99 と定義した場合，どの指定値も割り当て済み一連番号が 10 になった時点で，97（％）と 98（％）と 99（％）に対応するメッセージが出力されます。

一連番号の監視については，「5.25　一連番号の監視」を参照してください。

《規則》

- 一連番号使用比率を複数指定する場合，指定値が昇順になるように指定してください。

(s) SDBOPTION

SDBOPTION

SDB データベースのオプション機能を定義します。

(t) OCCURRENCE WARNING

OCCURRENCE WARNING {一連番号使用比率 1〔, 一連番号使用比率 2〔, 一連番号使用比率 3〕〕 | 0}

一連番号の使用状況を監視する場合に，一連番号の使用比率を％単位で指定します。

一連番号使用比率 1，一連番号使用比率 2，一連番号使用比率 3

～<符号なし整数>((50～99))

割り当て済み一連番号の割合が，一連番号使用比率 1～3 で指定した値に達した場合に，メッセージを 1 回ずつ出力します。0 を指定した場合は，割り当て済み一連番号の監視はしません。

この指定はすべての子レコード型に対して有効となりますが，SETOPTION 句下で WARNING 句を指定した場合は，SETOPTION 句下の WARNING 句の指定が有効となります。この場合，一連番号使用比率の指定値の個数が異なっても，双方で補ってメッセージを出力することはありません。

一連番号の監視については，「5.25　一連番号の監視」を参照してください。

《規則》

- この句の指定を省略した場合，一連番号使用比率 1 には 80 が，一連番号使用比率 2 には 90 が，一連番号使用比率 3 には 95 が仮定されます。
- ルートレコード型だけを定義している場合は，このオペランドを指定できません。

上記以外の規則については，SETOPTION 句下の「11.7.1(4)(r)　WARNING」の規則を参照してください。

(u) END SCHEMA

END SCHEMA

SDB データベース定義の終了を示します。

11.7.2 SDB データベース格納定義【SD FMB】

*ENTRY DICTIONARY 文に指定する SDB データベース格納定義について説明します。

(1) 機能

SDB データベースを格納する際の論理構造を定義します。

(2) 形式

項番	形式
(a)	STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名
(b)	DBTYPE SD

項番	形式
(c)	STORAGE RECORD 格納レコード名
(d)	{CLUSTERED 親子集合型名 | SEQUENTIAL シーケンシャルインデクス名
(e)	FOR RECORD
(f)	ORDER KEY { ASCENDING | DESCENDING } 構成要素名〔, 構成要素名〕… 〔 { ASCENDING | DESCENDING } 構成要素名〔, 構成要素名〕…〕…
(g)	〔PCTFREE ページ内の未使用領域の比率〕
(h)	WITHIN {シーケンシャルインデクス用 RD エリア名 | (シーケンシャルインデクス用 RD エリア名) | ((シーケンシャルインデクス用 RD エリア名) 〔, (シーケンシャルインデクス用 RD エリア名)〕…)} }
(i)	〔SUBPAGE NUMBER サブページ分割数〕
(j)	〔PCTFREE {ページ内の未使用領域の比率 | (〔ページ内の未使用領域の比率〕, セグメント内の空きページの比率)}〕
(k)	〔DEPENDING ON 構成要素名〕
(l)	WITHIN {格納レコード用 RD エリア名 | (格納レコード用 RD エリア名) | ((格納レコード用 RD エリア名) 〔格納条件〕 〔, (格納レコード用 RD エリア名) 格納条件〕…) | ((格納レコード用 RD エリア名) レコード型の RD エリア分割キー値 〔, (格納レコード用 RD エリア名) レコード型の RD エリア分割キー値…〕 , (格納レコード用 RD エリア名))}
(m)	〔SECONDARY INDEX 二次インデクス名
(n)	USED FOR RECORD
(o)	ORDER KEY { ASCENDING | DESCENDING } 構成要素名〔, 構成要素名〕… 〔{ ASCENDING | DESCENDING } 構成要素名〔, 構成要素名〕…〕…
(p)	〔PCTFREE ページ内の未使用領域の比率〕
(q)	〔DEPENDING ON 構成要素名〕
(r)	WITHIN {二次インデクス用 RD エリア名 | (二次インデクス用 RD エリア名) | ((二次インデクス用 RD エリア名) 二次インデクスの RD エリア分割キー値 〔, (二次インデクス用 RD エリア名) 二次インデクスの RD エリア分割キー値〕… , (二次インデクス用 RD エリア名))}
	〕…

項番	形式
	…
(s)	〔SET 親子集合型名
(t)	OWNER POINTER FOR FIRST LAST MEMBER
(u)	MEMBER POINTER FOR {NEXT PRIOR｜OWNER NEXT PRIOR} TENANT
	〕…
−	〔SDBOPTION〕※
(v)	END STORAGE SCHEMA

注

定義句は，形式で記述している順序に従って指定します。必須の定義と省略できる定義の間は形式で記載されているとおりに指定してください。

注※

SDBOPTION 句の指定有無によってシステムの動作に変わりはありません。そのため，SDBOPTION 句は省略してください。

(3) オペランドの説明

SDB データベース格納定義の各オペランドの指定内容を説明します。

(a) STORAGE SCHEMA

STORAGE SCHEMA SDB データベース格納名 FOR SDB データベース名

SDB データベース格納名と対応する SDB データベース名を指定します。

SDB データベース格納名

～＜識別子＞((1～30 バイト))

SDB データベース格納名を指定します。

すでに使用している SDB データベース格納名は指定できません。

SDB データベース名

～＜識別子＞((1～30 バイト))

SDB データベース名を指定します。

SDB データベース名に一致する名称が SDB データベース定義にない場合は，エラーとなります。

(b) DBTYPE

DBTYPE SD

SDB データベース種別が SD FMB の SDB データベースを定義します。

注意事項

SDB データベース種別が 4V（4V FMB または 4V AFM）の SDB データベースと SD FMB の SDB データベースをシステム内で混在して定義することはできません。混在させた場合，動作を保証しません。

(c) STORAGE RECORD

STORAGE RECORD 格納レコード名

格納レコード名

～<識別子>((1～30 バイト))

SDB データベース定義の RECORD 句で指定したレコード型名を，格納レコード名として指定します。

《規則》

- SDB データベース定義の RECORD 句で指定したレコード型名と同じ個数，同じ順番で指定してください。
- すでに使用している格納レコード名は指定できません。ただし，格納レコードの所有者（スキーマ）が異なる場合は，同じ格納レコード名を指定できます。

(d) CLUSTERED，SEQUENTIAL

CLUSTERED 親子集合型名｜SEQUENTIAL シーケンシャルインデクス名

格納するレコードの配置方法を指定します。

CLUSTERED 親子集合型名

親子集合型名

～<識別子>((1～30 バイト))

すべての子レコードを親レコードと同じ RD エリアに格納する場合に指定します。

《規則》

- STORAGE RECORD 句に指定したレコード型ごとに 1 つ指定できます。STORAGE RECORD 句に指定したレコード型が，ルートレコード型の場合は指定できません。
- SDB データベース格納定義の SET 句に指定する親子集合型名と同じ個数，同じ順番で指定してください。

SEQUENTIAL シーケンシャルインデクス名

インデクスのキーの値の順序に従って，親レコード実現値の格納位置を決めるときに指定します。SEQUENTIAL 句は，ルートレコード型に必ず指定します。

シーケンシャルインデクス名

～<識別子>((1～30 バイト))

STORAGE RECORD 句に指定したルートレコード型に対して 1 つ指定します。STORAGE RECORD 句に指定したレコード型が，ルートレコード型以外の場合は指定できません。

《規則》

- シーケンシャルインデクス名は，同一スキーマ内で一意になるように指定してください。

(e) FOR RECORD

FOR RECORD

レコード型単位にシーケンシャルインデクスを作成します。

キーの値が同じ場合，シーケンシャルインデクスへの格納を禁止します。

(f) ORDER KEY

ORDER KEY ｛ASCENDING ｜ DESCENDING｝構成要素名〔，構成要素名〕…

〔｛ASCENDING ｜ DESCENDING｝構成要素名〔，構成要素名〕…〕…

インデクスのキーとなる構成要素名を指定します。

指定した構成要素名をキーとして，昇順または降順にシーケンシャルインデクスが作成されます。

ASCENDING

昇順にシーケンシャルインデクスを作成します。

DESCENDING

降順にシーケンシャルインデクスを作成します。

構成要素名

～＜識別子＞((1～30 バイト))

インデクスのキーに指定する構成要素を次に示すように指定してください。

- SDB データベース定義内のルートレコード型のデータベースキーに指定した基本項目の構成要素名を，定義順にすべて指定します。ルートレコード型の構成要素の指定については，「表 11-71　ルートレコード型の構成要素の指定」を参照してください。SDB データベースの定義例については，「11.11.3　SDB データベースの定義例（SD FMB の場合）」を参照してください。

《規則》

- 構成要素名は，格納レコード名と対応する SDB データベース定義のレコード型名で指定した構成要素名としてください。
- シーケンシャルインデクス作成時に指定できる構成要素名の条件について，次の表に示します。
- 1 つの STORAGE RECORD 内に同じ構成のインデクス（シーケンシャルインデクス，および二次インデクス）を複数定義できません。同じ構成とは，次の条件をすべて満たす場合です。
  ・すべての構成要素名の指定順序が同一
  ・昇順，降順の指定がすべて一致している，またはすべて逆
- レコード型の RD エリア分割キーとする構成要素の場合，ASCENDING だけを指定できます。

表 11-75　シーケンシャルインデクス作成時に指定できる構成要素名の条件

項目	内容
指定できる構成要素名の数	最大 7 個
インデクスのキー長の合計	255 バイト以内 インデクスのキーに指定する構成要素とキー長の関係については，「表 11-76　インデクスのキーに指定する構成要素とキー長の関係」を参照してください。
構成要素名	基本項目の構成要素名 集団項目が含まれる構成要素名は指定できません。

表 11-76　インデクスのキーに指定する構成要素とキー長の関係

データ型	インデクスのキー長（単位：バイト）	
	構成要素名の指定が 1 つの場合	構成要素名の指定が複数の場合
CHARACTER	SDB データベース定義で指定したデータ長	SDB データベース定義で指定したデータ長＋制御部（1 バイト）
XCHARACTER		
PACKED〔DECIMAL FIXED〕整数部桁数〔，小数部桁数〕	↓（整数部桁数＋小数部桁数）÷2↓＋1	↓（整数部桁数＋小数部桁数）÷2↓＋2
INTEGER	4	5
SMALLINT	2	3

(g)　PCTFREE

PCTFREE ページ内の未使用領域の比率

～<符号なし整数>((0～99))≪30≫

インデクスページ内の未使用領域の比率を指定します。

(h)　WITHIN

WITHIN　{シーケンシャルインデクス用 RD エリア名

｜（シーケンシャルインデクス用 RD エリア名)

｜（(シーケンシャルインデクス用 RD エリア名)

〔,（シーケンシャルインデクス用 RD エリア名)〕…)}

シーケンシャルインデクス用 RD エリア名

～< RD エリア名>((1～30 バイト))

シーケンシャルインデクスを格納する RD エリア名を指定します。

1 つの SEQUENTIAL 句に対して，WITHIN 句を必ず 1 つ指定してください。

《規則》

シーケンシャルインデクスを複数の RD エリアに格納する場合の規則

- シーケンシャルインデクス用 RD エリアとして指定できる RD エリアの数と SDB データベースの横分割の関係を次の表に示します。

表 11-77　シーケンシャルインデクス用 RD エリアとして指定できる RD エリア数と SDB データベースの横分割の関係

SDB データベースの横分割方法	指定できる RD エリア数
分割指定なし	1
格納条件分割	1～256
境界値分割	2～3,000（1,024）※

注※

WITHIN 句に指定できる RD エリア名の延べ数の最大値は 3,000 になります。1,024 は，WITHIN 句に RD エリア名を重複して指定する場合，RD エリア名の重複を排除して数えたときの最大値です。

- シーケンシャルインデクス用 RD エリアは，格納レコード用 RD エリアと同じ数を指定する必要があります。
- 1 つの WITHIN 句の中では，シーケンシャルインデクス用 RD エリア名は一意に指定してください。なお，境界値分割指定の場合，隣接する RD エリア名が重複している場合だけエラーとなります。
- 境界値分割指定の格納レコード用 RD エリアを重複して指定している場合は，シーケンシャルインデクス用 RD エリアも重複して指定してください。シーケンシャルインデクス用 RD エリアを重複して指定する場合の例を次の表に示します。

表 11-78　シーケンシャルインデクス用 RD エリアを重複して指定する場合の例

指定	格納レコード用 RD エリアの指定	シーケンシャルインデクス用 RD エリアの指定
RD エリアの重複なし	((RDAREAX) X'00FF', (RDAREA1) X'01FF', (RDAREA2) X'03FF', (RDAREA3))	((RDAREAX_I), (RDAREA1_I), (RDAREA2_I), (RDAREA3_I))
RD エリアの重複あり	((RDAREAX) X'00FF', (RDAREA1) X'01FF', (RDAREA2) X'03FF', (RDAREA1))	((RDAREAX_I), (RDAREA1_I), (RDAREA2_I), (RDAREA1_I))

- 格納レコード用 RD エリアと対応するシーケンシャルインデクス用 RD エリアは，同じサーバ内の RD エリアである必要があります。

指定するシーケンシャルインデクス用 RD エリアに関する規則

- 格納レコード用 RD エリアとして指定した RD エリアは指定できません。
- シーケンシャルインデクス用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。

- シーケンシャルインデクス用 RD エリアとして指定する RD エリアには，1 つの RD エリアにシーケンシャルインデクスを最大 500 個指定できます。
- シーケンシャルインデクス用 RD エリアに定義できるリソースの組み合わせを次の表に示します。

表 11-79　RD エリアに定義できるリソースの組み合わせ

定義するリソース		RD エリアの定義		
		レコード型・インデクス定義なし	SDB データベース格納定義 A	
			レコード型定義あり	インデクス定義あり
SDB データベース格納定義 A	レコード型	○	○	×
	インデクス	○	×	○
SDB データベース格納定義 B	レコード型	○	×	×
	インデクス	○	×	○

（凡例）
○：定義できます。
×：定義できません。

(i)　SUBPAGE NUMBER

SUBPAGE NUMBER サブページ分割数

サブページ分割数

～<符号なし整数>((2～16))

1 データページ当たりのサブページ分割数を指定します。SUBPAGE NUMBER を省略した場合，サブページ分割されません。サブページ分割数の設計方針については，「3.2.4　サブページの設計」を参照してください。

《規則》

- ルートレコード型，およびすべての子レコード型で同じサブページ分割数を指定してください。
- サブページ分割数を指定する場合，格納レコード用 RD エリアのページ長が，4096 バイトである必要があります。

(j)　PCTFREE

PCTFREE　{ページ内の未使用領域の比率
　｜（〔ページ内の未使用領域の比率〕，セグメント内の空きページの比率）}

ページ内の未使用領域の比率

～<符号なし整数>((0～99))

セグメント内の空きページの比率

～<符号なし整数>((0～50))

データベースの初期作成時に設定するページ内の未使用領域の比率を指定します。ページ内の未使用領域の比率は，0～99（%）の範囲で指定できます。ページ内の未使用領域の比率を省略すると，30（%）が仮定されます。

SUBPAGE NUMBER 句でサブページ分割数を指定している場合，ページ内の未使用領域の比率は，サブページ内の未使用領域の比率になります。例えば，30 を指定した場合，サブページ内の未使用領域の比率が 30%になります。

また，セグメント内の空きページの比率を 0～50（%）の範囲で指定できます。セグメント内の空きページの比率を省略すると，10（%）が仮定されます。

《規則》

ルートレコード型，およびすべての子レコード型で同じ値を指定してください。

(k) DEPENDING ON

DEPENDING ON 構成要素名

構成要素名

～<識別子>((1～30 バイト))

SDB データベースを境界値で複数の RD エリアに横分割して格納する場合の，境界値分割のキーとなる構成要素名を指定します。

《規則》

- ルートレコード型にだけ指定できます。
- CLUSTERED 句を指定した格納レコードに対しては指定できません。
- 格納レコード名と対応する SDB データベース定義のレコード型名で指定した構成要素名である必要があります。
- 構成要素名には，データ種別 1 が K，データ種別 2 が A の構成要素名を指定します。
- SDB データベースの横分割方法が格納条件指定の場合は指定できません。

(l) WITHIN

WITHIN {格納レコード用 RD エリア名

| (格納レコード用 RD エリア名)

| ((格納レコード用 RD エリア名) 〔格納条件〕

〔,(格納レコード用 RD エリア名) 格納条件〕…)

| ((格納レコード用 RD エリア名) レコード型の RD エリア分割キー値

〔,(格納レコード用 RD エリア名) レコード型の RD エリア分割キー値…〕

,(格納レコード用 RD エリア名))}

格納レコード用 RD エリア名

～< RD エリア名>((1～30))

格納条件

～構成要素名＝（レコード型の RD エリア分割キー値〔，レコード型の RD エリア分割キー値〕…）

レコード型の RD エリア分割キー値

～X'＜16 進数字＞'((1～30 バイト))

～'＜英数字または下線（_）＞'((1～30 バイト))※1

～＜10 進数定数＞※2

～＜整数定数＞※3

注※1

先頭に下線（_）は指定できません。

注※2

10 進数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

整数部と小数部にすべて 0 を指定する場合，符号に-を指定しないでください。-0.を指定しても 0. として扱います。

なお，10 進数定数の値の範囲を超える値を指定できません。

注※3

整数定数については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

-0 を指定しても 0 として扱います。

なお，整数定数の値の範囲を超える値を指定できません。

レコード実現値を格納する RD エリア名を指定します。

1 つの STORAGE RECORD 句に対して，WITHIN 句を必ず 1 つ指定してください。

データ型ごとのレコード型の RD エリア分割キー値の指定形式を次の表に示します。

表 11-80　データ型ごとのレコード型の RD エリア分割キー値の指定形式

レコード型の RD エリア分割キーのデータ型	レコード型の RD エリア分割キー値の指定形式			
	16 進数字	英数字，または下線(_)	10 進数定数	整数定数
CHARACTER	○	○	×	×
XCHARACTER	○	○	×	×
PACKED 〔DECIMAL FIXED〕	×	×	○	×
INTEGER	×	×	×	○

（凡例）

○：指定できます。

×：指定できません。

《規則》

レコードを複数の RD エリアに格納する場合の規則

- 1 レコード型に対して指定できる格納レコード用 RD エリアの数と SDB データベースの横分割の関係を次の表に示します。

表 11-81　レコード型に対して指定できる格納レコード用 RD エリアの数と SDB データベースの横分割の関係

SDB データベースの横分割方法	指定できる RD エリア数
分割指定なし	1
格納条件分割	1～256
境界値分割	2～3,000（1,024）※

注※

WITHIN 句に指定できる RD エリア名の延べ数の最大値は 3,000 になります。1,024 は，WITHIN 句に RD エリア名を重複して指定する場合，RD エリア名の重複を排除して数えたときの最大値です。

- 格納条件分割で格納レコード用 RD エリア名を複数指定した場合は，すべての格納レコード用 RD エリア名に格納条件を指定してください。
- 同一 SDB データベース格納定義内の子レコード型に指定する格納レコード用 RD エリア名は，ルートレコード型と同じ数，同じ順番ですべて指定する必要があります。
- 1 つの WITHIN 句の中では，格納レコード用 RD エリア名は一意に指定してください。なお，境界値分割指定の場合，隣接する RD エリア名が重複している場合だけエラーとなります。
- SECONDARY INDEX 句を指定する場合，格納条件は指定できません。
- レコード型の RD エリア分割キー値に指定できる範囲を次の表に示します。

表 11-82　レコード型の RD エリア分割キー値に指定できる範囲

レコード型の RD エリア分割キー値の指定形式	SDB データベースの横分割方法	レコード型の RD エリア分割キー値	
		最小	最大
16 進数字	格納条件分割	X'00…00'	X'FF…FF'
	境界値分割	X'00…00'	X'FF…FE'
10 進数定数	格納条件分割	-9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の負の数）	9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の正の数）
	境界値分割	-9…9.9…9 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数分'9'の負の数）	9…9.9…8 （整数部と小数部がレコード型の RD エリア分割キーの全体桁数のうち，最後の桁を除くすべての桁が'9'かつ最後の桁が'8'の正の数）
整数定数	格納条件分割	－2147483648	2147483647
	境界値分割	－2147483648	2147483646

指定する格納レコード用 RD エリアに関する規則

- インデクスを格納する RD エリアは指定できません。
- 格納レコード用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- 指定する RD エリア名には，格納レコード長が当該 RD エリアのページ長（サブページ分割をしている場合はサブページ長）に収まる RD エリア名を指定してください。
- 格納レコード用 RD エリアに定義できるリソースの組み合わせについては，「表 11-79　RD エリアに定義できるリソースの組み合わせ」を参照してください。

レコードの格納条件に関する規則

- 格納条件は，ルートレコード型にだけ指定できます。

格納条件に指定する構成要素に関する規則

- 1 個の WITHIN 句の中では，単一の構成要素名を指定してください。複数種類の構成要素名は指定できません。
- 構成要素名には，SDB データベース定義で指定した構成要素名を指定してください。
- 構成要素名には，データ種別 1 が K，データ種別 2 が A の構成要素名を指定してください。

格納条件のレコード型の RD エリア分割キー値に関する規則

- 1 つの WITHIN 句に対するレコード型の RD エリア分割キー値は，最大 15,000 個指定できます。
- 1 つの WITHIN 句の中で同じレコード型の RD エリア分割キー値は重複して指定できません。
- レコード型の RD エリア分割キーのデータ型が CHARACTER，または XCHARACTER の場合，指定したレコード型の RD エリア分割キー値の長さは，SDB データベース定義で指定した格納条件の構成要素の長さと一致させる必要があります。
- レコード型の RD エリア分割キーのデータ型が PACKED の場合，RD エリア分割キーの整数部桁数，および小数部桁数の範囲内の値を指定してください。

境界値分割時（DEPENDING ON 句指定時）のレコード型の RD エリア分割キー値に関する規則

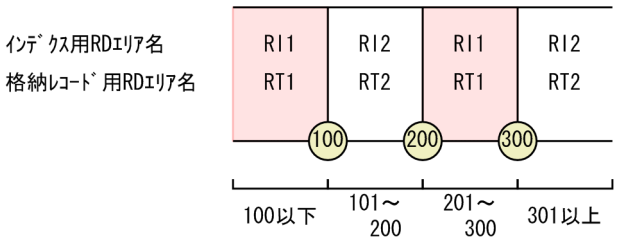
- 最初に境界値を指定する RD エリアには，指定する境界値以下の値のレコードが格納されます。

  2 番目以降（最後は除く）に指定する RD エリアには，その前に指定した境界値より大きく，かつその後に指定する境界値以下の値のレコードが格納されます。

  最後に指定する RD エリアには，その前に指定した境界値より大きい値のレコードが格納されます。

  境界値分割時の SDB データベースの横分割と SDB データベース格納定義の例を次の図に示します。

図 11-17　境界値分割時の SDB データベースの横分割と SDB データベース格納定義の例



＜定義例＞

```
STORAGE SCHEMA SCM01 FOR SCM01
  DBTYPE SD
  STORAGE RECORD RECA
    SEQUENTIAL IDX1 FOR RECORD
         :
      WITHIN ((RI1),(RI2),(RI1),(RI2))
    DEPENDING ON C1
      WITHIN ((RT1)100,(RT2)200,(RT1)300,(RT2))
         :
END STORAGE SCHEMA
```

- 境界値分割時には，格納条件の指定はできません（レコード型の RD エリア分割キー値の指定をする必要があります）。
- レコード型の RD エリア分割キーのデータ型が CHARACTER，または XCHARACTER の場合，レコード型の RD エリア分割キー値の長さは，DEPENDING ON 句で指定した構成要素の長さと一致させる必要があります。
- レコード型の RD エリア分割キー値は重複して指定できません。
- レコード型の RD エリア分割キー値は昇順に指定する必要があります。
- レコード型の RD エリア分割キー値を 16 進数字，英数字または下線（_）の形式で指定する場合，レコード型の RD エリア分割キー値の長さが 1 つ目の長さと異なるとエラーとなります。
- レコード型の RD エリア分割キーのデータ型が PACKED の場合，レコード型の RD エリア分割キーの整数部桁数，および小数部桁数の範囲内の値を指定してください。

(m)　SECONDARY INDEX

SECONDARY INDEX 二次インデクス名

二次インデクス名

～＜識別子＞((1～30 バイト))

二次インデクス名を指定します。

《規則》

- SECONDARY INDEX 句を指定する場合は，HiRDB Structured Data Access Facility Extension for XDM/SD type をインストールする必要があります。

- 二次インデクス名は，同一スキーマ内で一意になるように指定してください。シーケンシャルインデクス名も含めて一意になるようにしてください。
- 二次インデクス名には，シーケンシャルインデクス名とは別の名称を指定してください。
- SECONDARY 句は，1 つの STORAGE RECORD 句内で 16 個まで指定できます（ルートレコード型の場合は，シーケンシャルインデクスを含みます）。詳細については，「表 11-83　二次インデクスの定義規則」を参照してください。
- 3 階層以上のレコード型には指定できません。
- 子レコードに SECONDARY INDEX 句を指定した場合，MEMBER POINTER FOR 句に OWNER を必ず指定してください。
- SECONDARY INDEX 句を指定した場合，格納レコード用 RD エリア名を指定する WITHIN 句の RD エリア指定には格納条件を指定しないでください。

(n)　USED FOR

USED FOR RECORD

二次インデクスの種別を指定します。

RECORD

レコード型内二次インデクスを定義します。

《規則》

SECONDARY INDEX 句を指定した場合，このオペランドを必ず指定してください。

(o)　ORDER KEY

ORDER KEY ｛ASCENDING ｜ DESCENDING｝構成要素名〔，構成要素名〕…

〔｛ASCENDING ｜ DESCENDING｝構成要素名〔，構成要素名〕…〕…

インデクスのキーとなる構成要素名を指定します。

指定した構成要素名をキーとして，昇順または降順に二次インデクスが作成されます。

ASCENDING

昇順に二次インデクスを作成します。

DESCENDING

降順に二次インデクスを作成します。

構成要素名

～＜識別子＞((1～30 バイト))

二次インデクスのキーとなる構成要素名を指定します。

ORDER KEY 句については，「11.7.2(3)(f)　ORDER KEY」を参照してください。

《規則》

- 1 つの STORAGE RECORD 内に同じ構成のインデクス（シーケンシャルインデクス，および二次インデクス）を複数定義できません。同じ構成とは，次の条件をすべて満たす場合です。
  ・すべての構成要素名の指定順序が同一
  ・昇順，降順の指定がすべて一致している，またはすべて逆
- 二次インデクスについては，「表 11-83　二次インデクスの定義規則」に従って定義してください。
- 構成要素に，集団項目の構成要素は指定できません。
- 同じ構成要素名を重複して指定できません。
- 1 つの二次インデクスに指定できる構成要素の数は，1～16 個です。
- 二次インデクスのキー長の合計値の上限はシーケンシャルインデクスのキー長の合計の上限値と同じです。詳細については，「表 11-75　シーケンシャルインデクス作成時に指定できる構成要素名の条件」の「インデクスのキー長の合計」を参照してください。
- 指定した構成要素，または指定した構成要素の上位の構成要素に OCCURS 句は指定できません。
- 二次インデクスの構成要素の指定順序は任意です。
- 二次インデクスを複数の RD エリアに格納する場合，二次インデクスの構成要素にレコード型の RD エリア分割キーと同じ名称の構成要素を含めて指定してください。
- 二次インデクスを複数の RD エリアに格納する場合，インデクスの最初の構成要素には，データ型が CHARACTER または XCHARACTER の構成要素だけを指定できます。
- 二次インデクスを複数の RD エリアに格納する場合，インデクスの最初の構成要素には，ASCENDING だけを指定できます。

二次インデクスの定義規則を次に示します。

表 11-83　二次インデクスの定義規則

レコード型種別	二次インデクスの指定	指定する構成要素
ルートレコード型	0～15 本	「表 11-84　指定できる構成要素のデータ種別」を参照してください。
子レコード型 （2 階層）	0～16 本	
子レコード型 （3 階層以上）	×	－

（凡例）
×：指定できません。
－：該当しません。

表 11-84　指定できる構成要素のデータ種別

項番	データ種別 1	データ種別 2	構成要素の指定可否
1	K	L	○
2		A	○

項番	データ種別 1	データ種別 2	構成要素の指定可否
3		N	○
4		P	×
5	U	D	○
6		F	×

（凡例）

○：指定できます。

×：指定できません。

二次インデクスに指定する構成要素の定義例を次に示します。

- ルートレコード型の場合

  SDB データベース定義

```
                              :
RECORD  RECA
    2 KEYDATA
      3 DBKEY_L2  XCHARACTER  1  TYPE K,L
      3 DBKEY_L3  XCHARACTER  1  TYPE K,L
      3 DBKEY_L4  CHARACTER   3  TYPE K,L
    2 USERDA0     CHARACTER   30 TYPE U,D
    2 USERDAN0    CHARACTER   10 TYPE U,D
                              :
```

  SDB データベース格納定義

```
                              :
STORAGE RECORD  RECA
    SEQUENTIAL RECA_SI FOR RECORD
      ORDER KEY ASCENDING  DBKEY_L2, DBKEY_L3, DBKEY_L4
      WITHIN  FMBX01
    WITHIN  FMBX02
    SECONDARY INDEX  RECA_SI1
      USED FOR RECORD
      ORDER KEY ASCENDING  DBKEY_L2, DBKEY_L3, USERDA0
      WITHIN  FMBX01       :
```

- 第 2 階層のレコード型の場合

  SDB データベース定義

```
                              :
RECORD  RECB
    2 DBKEY_L2  XCHARACTER  1  TYPE K,L
    2 DBKEY_L3  XCHARACTER  1  TYPE K,L
    2 DBKEY_L4  CHARACTER   3  TYPE K,L
    2 DBKEY     INTEGER        TYPE K,N
    2 USERDB0   CHARACTER   20 TYPE U,D
    2 USERDBN0  CHARACTER   10 TYPE U,D
                              :
```

  SDB データベース格納定義

```
                          :
STORAGE RECORD  RECB
    CLUSTERED SET1
    WITHIN  FMBX02
    SECONDARY INDEX  RECB_SI1
      USED FOR RECORD
      ORDER KEY ASCENDING  USERDB0
      WITHIN  FMBX01
                          :
```

(p) PCTFREE

PCTFREE ページ内の未使用領域の比率

PCTFREE 句については，「11.7.2(3)(g) PCTFREE」を参照してください。

(q) DEPENDING ON

DEPENDING ON 構成要素名

構成要素名

～<識別子>((1～30 バイト))

二次インデクスを境界値として，複数の RD エリアに SDB データベースを横分割する場合，境界値分割のキーとなる構成要素名を指定します。

《規則》

- 次の条件をすべて満たす場合は，DEPENDING ON 句を必ず指定してください。

  ・ルートレコードを境界値分割している。

  ・二次インデクスの最初の構成要素名がレコード型の RD エリア分割キーと異なる。

  上記の条件をすべて満たさない場合に DEPENDING ON 句を指定すると，pdsdbdef コマンドがエラーになります。
- 二次インデクスの最初の構成要素名を指定します。
- 構成要素の長さは 128 バイト以内としてください。

(r) WITHIN

WITHIN {二次インデクス用 RD エリア名

| (二次インデクス用 RD エリア名)

| ((二次インデクス用 RD エリア名) 二次インデクスの RD エリア分割キー値

〔,(二次インデクス用 RD エリア名) 二次インデクスの RD エリア分割キー値〕…

,(二次インデクス用 RD エリア名))}

二次インデクス用 RD エリア名

～< RD エリア名>((1～30 バイト))

二次インデクスの RD エリア分割キー値

～X'＜16 進数字＞'((1～128 バイト))

～'＜英数字または下線（_）＞'((1～128 バイト))※

注※　先頭に下線（_）は指定できません。

二次インデクスを格納する RD エリア名を指定します。

1 つの SECONDARY INDEX 句で，WITHIN 句を必ず 1 つ指定してください。

《規則》

二次インデクスを複数 RD エリアに格納する場合の規則

- 1 つの二次インデクスに指定できる二次インデクス用 RD エリアの個数は，シーケンシャルインデクス用 RD エリアの個数と同じにしてください。シーケンシャルインデクス用 RD エリアの個数については，「表 11-77　シーケンシャルインデクス用 RD エリアとして指定できる RD エリア数と SDB データベースの横分割の関係」を参照してください。
- 二次インデクス用 RD エリアは，格納レコード用 RD エリアと同じ数を指定する必要があります。
- 隣接する二次インデクス用 RD エリア名を重複して指定している場合，エラーとなります。
- 格納レコード用 RD エリアを重複して指定している場合は，二次インデクス用 RD エリアも重複して指定する必要があります。
- 格納レコード用 RD エリアと対応する二次インデクス用 RD エリアは，同じバックエンドサーバ内の RD エリアである必要があります。
- 二次インデクスに DEPENDING ON 句を指定している場合は，二次インデクスの RD エリア分割キー値を指定する必要があります。
- 二次インデクスに DEPENDING ON 句を指定していない場合は，二次インデクスの RD エリア分割キー値を指定できません。
- 二次インデクスの RD エリア分割キー値に指定できる範囲を次の表に示します。

表 11-85　二次インデクスの RD エリア分割キー値に指定できる範囲

二次インデクスの RD エリア分割キー値の指定形式	二次インデクスの RD エリア分割キー値	
	最小	最大
16 進数字	X'00…00'	X'FF…FF'

指定する二次インデクス用 RD エリアに関する規則

- 格納レコード用 RD エリア名として指定した RD エリアは指定できません。
- 二次インデクス用 RD エリアには，create rdarea 文の data model オペランドに structured を指定した RD エリアを指定してください。
- 二次インデクス用 RD エリアとして指定する RD エリアには，1 つの RD エリアにシーケンシャルインデクスと二次インデクスを合わせて，インデクスを最大 500 個指定できます。
- 二次インデクス用 RD エリアに定義できるリソースの組み合わせについては，「表 11-79　RD エリアに定義できるリソースの組み合わせ」を参照してください。

二次インデクスの RD エリア分割キー値に関する規則

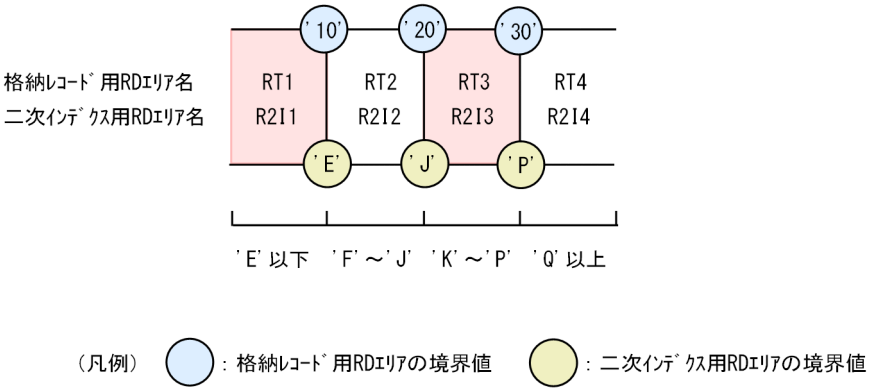
- 最初に境界値を指定する RD エリアには，指定する境界値以下の値のインデクスが格納されます。

  2 番目以降（最後は除く）に指定する RD エリアには，その前に指定した境界値より大きく，かつそのあとに指定する境界値以下の値のインデクスが格納されます。

  最後に指定する RD エリアには，その前に指定した境界値より大きい値のインデクスが格納されます。

  境界値分割時の SDB データベースの横分割と SDB データベース格納定義の例を次の図に示します。

  図 11-18　二次インデクスの横分割と SDB データベース格納定義の例



  ＜定義例＞

```
STORAGE SCHEMA SCM01 FOR SCM01
  DBTYPE SD
  STORAGE RECORD RECA
        :
    DEPENDING ON C1
      WITHIN ((RT1)'10',(RT2)'20',(RT3)'30',(RT4))
        :
    SECONDARY INDEX IDX2
        :
      WITHIN ((R2I1)'E',(R2I2)'J',(R2I3)'P',(R2I4))
        :
END STORAGE SCHEMA
```

- レコードの格納（STORE），レコードの更新（MODIFY），またはデータロードの実行時，格納または更新しようとしているレコードの二次インデクスキー値が，範囲外だとエラーとなります。例えば，上記の図では，最初の RD エリアに格納するレコードの二次インデクスキー値に，'E'より大きい値を指定すると，エラーとなります。
- 二次インデクスの RD エリア分割キー値の長さは，DEPENDING ON 句で指定した構成要素の長さと一致させる必要があります。
- 二次インデクスの RD エリア分割キー値は，重複して指定できません。
- 二次インデクスの RD エリア分割キー値は，昇順に指定する必要があります。
- 二次インデクスの RD エリア分割キー値の長さが，1 つ目の長さと異なるとエラーとなります。

(s) SET

SET 親子集合型名

～<識別子>((1～30 バイト))

親レコード型，子レコード型の親子集合型名を指定します。

《規則》

SDB データベース定義の SET 句で指定した親子集合型名と同じ個数，同じ順番で指定してください。

(t) OWNER POINTER FOR

OWNER POINTER FOR FIRST LAST MEMBER

親レコードに持たせるポインタを指定します。

FIRST LAST

親レコードに FIRST ポインタ，および LAST ポインタを持たせる場合に指定します。

(u) MEMBER POINTER FOR

MEMBER POINTER FOR {NEXT PRIOR | OWNER NEXT PRIOR} TENANT

子レコードに持たせるポインタを指定します。

NEXT PRIOR

子レコードに NEXT ポインタ，および PRIOR ポインタを持たせる場合に指定します。

OWNER NEXT PRIOR

子レコードに OWNER ポインタ，NEXT ポインタ，および PRIOR ポインタを持たせる場合に指定します。

《規則》

- 親子集合の子レコードに対して SECONDARY INDEX 句を指定した場合，OWNER NEXT PRIOR を必ず指定してください。
- 親子集合の子レコードに対して位置指示子指定の検索を実行する場合，OWNER NEXT PRIOR を必ず指定してください。

OWNER NEXT PRIOR の指定規則の詳細については，「17.4.3 FIND 文」の「(4) 共通規則」の「位置指示子指定の検索で，検索対象レコードがルートレコードではない場合」の説明を参照してください。

(v) END STORAGE SCHEMA

END STORAGE SCHEMA

SDB データベース格納定義の終了を示します。

11.8　規則および注意事項

pdsdbdef コマンド実行時の規則および注意事項について説明します。

11.8.1　規則

(1)　pdsdbdef コマンドの実行条件

- pdsdbdef コマンドは，HiRDB が稼働中のときだけ実行できます。
- pdsdbdef コマンドは，システムマネジャがあるサーバマシンで実行してください。
- pdsdbdef コマンド実行時には，次のサーバが稼働している必要があります。
  - システムマネジャ
  - 1 つ以上のフロントエンドサーバ
  - ディクショナリサーバ
  - SDB データベースを定義する RD エリアを管理しているバックエンドサーバ

(2)　コマンドの実行者

HiRDB 管理者が pdsdbdef コマンドを実行できます。

(3)　pdsdbdef コマンドの実行可否

pdsdbdef コマンドを実行する前に，RD エリアの初期設定が完了している必要があります。また，次に示す RD エリアが閉塞なしオープン状態である必要があります。

- マスタディレクトリ用 RD エリア
- データディクショナリ用 RD エリア
- データディレクトリ用 RD エリア
- SDB データベース格納定義の WITHIN 句に指定する RD エリア

RD エリアの状態による pdsdbdef コマンドの実行可否については，「付録 I.1　RD エリアの状態によるユティリティの実行可否」を参照してください。

(4)　pdsdbdef コマンドの最大同時実行数

pdsdbdef コマンドの最大同時実行数は，システム共通定義の pd_utl_exec_mode オペランドの指定値に関係なく 1 です。

注意事項

HiRDB/SD データベース作成ユティリティ，HiRDB/SD データベース再編成ユティリティやほかの UAP と同時に実行することはできません。ほかのユティリティや UAP が終了してから pdsdbdef コマンドを実行してください。

(5) 環境変数の設定

- 環境変数 LANG を設定してください。OS がサポートしていない文字コードを使用する場合，クライアント環境定義の PDLANG オペランドを設定する必要があります。

  環境変数 LANG，およびクライアント環境定義の PDLANG オペランドの設定については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。
- 環境変数 LANG またはクライアント環境定義の PDLANG オペランドで，使用する文字コードに UTF-8 を指定した場合，pdsdbdef コマンドの入力ファイルに BOM 付きファイルを使用できます。なお，pdsdbdef コマンドの実行結果として出力されるファイルには BOM は付きません。

(6) クライアント環境定義の設定

pdsdbdef コマンドを実行する場合，クライアント環境定義の PDUSER オペランドに，スキーマ所有者の認可識別子を指定してください。

このとき，OS ログインユーザの簡易認証機能は使用できません。

(7) pdsdbdef コマンドで使用する文字コード

SDB 制御文や SDB 定義文に記述する文字コードは，pdsetup コマンドで指定する文字コードと同じ文字コードにしてください。

(8) スキーマと SDB データベース定義の関係

pdsdbdef コマンドを実行する前に，クライアント環境定義の PDUSER オペランドに指定した認可識別子に対して，スキーマを定義しておく必要があります。

HiRDB/SD では，1 つの SDB データベース定義内に複数のレコード型を定義できます。HiRDB/SD では，「スキーマ名（＝認可識別子）＋レコード型」として，アクセスするレコード型を一意に識別するため，レコード型の定義時には，次の点に注意してください。

- 1 スキーマ内のすべての SDB データベース定義で，レコード型名を重複させることはできません。なお，1 スキーマ内で，表とレコード型の両方を定義できますが，表名とレコード型名を重複させることもできません。
- 1 スキーマ内のすべての SDB データベース格納定義で，インデクス名を重複させることはできません。また，1 スキーマ内で，表とレコード型の両方にインデクスを定義できますが，表に定義したインデクス名と SDB データベース格納定義で定義したインデクス名を重複させることもできません。

- レコード型名やインデクス名を重複させる場合は，別スキーマとしてください。

(9) SDB データベース格納定義で指定できる RD エリア数

1 つの SDB データベース格納定義で指定できる RD エリア名の数は最大 32,766 個です（RD エリア名が重複する場合は 1 つと数えます）。

(10) pdsdbdef コマンドで使用するファイルおよびディレクトリ

pdsdbdef コマンドで使用するファイル，およびファイルを出力するディレクトリは，事前に作成しておいてください。また，パーミッションを次のように設定してください。

- SDB 制御文ファイル：644
- SDB 定義文ファイル：644
- 実行結果ファイルおよび実行結果ファイルの出力先ディレクトリ：666
- SDB ディレクトリ情報ファイルの格納先ディレクトリ：755

11.8.2 TP1/FSP の SDB ハンドラ機能使用時の注意事項【4V FMB，4V AFM】

TP1/FSP で SDB ハンドラ機能を使用する場合の，SDB データベース定義時の注意事項を次に示します。TP1/FSP の機能については，マニュアル「TP1/Financial Service Platform 使用の手引」を参照してください。

- 構成要素名は 8 バイト以下で定義してください。

  TP1/FSP では，構成要素名として 8 バイトまでしか指定できません。そのため，8 バイトを超える構成要素名の構成要素が存在する場合は，意図しない構成要素にアクセスするおそれがあります。この場合の動作は保証されません。
- 構成要素名に空白を含めて定義しないでください。

  TP1/FSP では，空白を構成要素名の区切りとして扱います。そのため，構成要素名中に空白を含む構成要素が存在する場合，意図しない構成要素にアクセスするおそれがあります。この場合の動作は保証されません。
- 4V AFM の SDB データベースでは，仮想ルートレコード型を除くすべての子レコード型でデータベースキーの構成（属性とサイズ）とデータ種別を同じ構成で定義してください。構成要素名は異なってもかまいません。同じ構成でない場合の動作は保証されません。
- UAP で 4V MAM の SDB データベースを検索する条件式に"%ENTRY△△"を指定する場合は，次に示すどちらかの構成で 4V MAM の SDB データベースを定義してください。次に示す構成以外のデータベースキーを定義した 4V MAM の SDB データベースに対して，条件式に"%ENTRY△△"を指定した場合の動作は保証されません。
  - レコード分割キー（TYPE K,R）と一連番号（TYPE K,N）を定義する構成

- レコード分割キー（TYPE K,R）を定義しないで一連番号（TYPE K,N）を定義する構成

レコード分割キーと一連番号を定義する構成で 4V MAM の SDB データベースを定義する場合は，レコード識別コード（RECORDID 句の指定値）と同じ値をレコード分割キー値として KEYDEF 句で指定してください。指定しない場合の動作は保証されません。

レコード分割キーと一連番号を定義する構成の場合の，SDB データベース定義の例を次に示します。

SDB データベース定義の例

```
SCHEMA SDBデータベース名
        :（省略）
RECORD A1    ……・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・仮想ルートレコード型
  02 DBKEY_L1    CHARACTER  8 TYPE D,L
  02 DBKEY_ID    XCHARACTER 2 TYPE K,R
RECORD A2
  02 DBKEY_L1    CHARACTER  8 TYPE D,L
  02 DBKEY
    03 DBKEY_ID  XCHARACTER 2 TYPE K,R
    03 DBKEY_LN  INTEGER      TYPE K,N
        :（ユーザデータ部は省略しています）
  RECORDID X'AAAA'    ……・・・・・・・・・・・・・・・・・・・・・・・・・・・【A】
RECORD A3
  02 DBKEY_L1    CHARACTER  8 TYPE D,L
  02 DBKEY
    03 DBKEY_ID  XCHARACTER 2 TYPE K,R
    03 DBKEY_LN  INTEGER      TYPE K,N
        :（ユーザデータ部は省略しています）
  RECORDID X'BBBB'    ……・・・・・・・・・・・・・・・・・・・・・・・・・・・【B】

STORAGE SCHEMA SDBデータベース格納名 FOR SDBデータベース名
        :（省略）
SDBOPTION
  KEYDEF DBKEY_ID
    DATA X'AAAA',A2 ……レコード識別コード【A】と同じ値にする必要があります
    DATA X'BBBB',A3 ……レコード識別コード【B】と同じ値にする必要があります
```

11.9　リターンコード

pdsdbdef コマンドのリターンコードの意味および対処を次の表に示します。

表 11-86　pdsdbdef コマンドのリターンコードの意味および対処

リターンコード	意味	対処
0	pdsdbdef コマンドが正常終了しました。	特にありません。
8	pdsdbdef コマンドの処理でエラーが発生しました。 メッセージが出力され，処理は継続されます。 dirinf 文を指定している場合，dirinf 文の処理はスキップされます。	エラーの原因を取り除き，pdsdbdef コマンドを再実行してください。
12	pdsdbdef コマンドの処理が続行できないエラーが発生しました。 メッセージが出力され，処理は打ち切られます。	エラーの原因を取り除き，pdsdbdef コマンドを再実行してください。

11.10　実行結果ファイルの出力形式

実行結果ファイルの出力例を次に示します。

実行結果ファイルの出力例

```
STATEMENT   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
--8
         1   *ENTRY DICTIONARY
         2   SCHEMA SCM01
         3    DBTYPE 4V FMB
         4    RECORD RECA
         5     2 DBKEY
         6       3 DBKEY_L2 XCHARACTER   2 TYPE K,L
         7       3 DBKEY_L3 XCHARACTER   5 TYPE K,L
         8     2 USR_0001 XCHARACTER   74 TYPE U,D
         :         :
       802   END SCHEMA
 プロセスID hh:mm:ss ユニット名 サーバ名   KFPB61202-I *ENTRY DICTIONARY terminated, retu
rn code=0
```

出力例の説明

- ヘッダ情報（スケール）は，SDB 定義文の機能ごと（*ENTRY DICTIONARY 文などの制御文ごと）に出力されます。80 バイトの目盛りが出力されます。ただし，*ENTRY DICTIONARY 文などの制御文が行の先頭でない場合は出力されません。また，定義情報の行長が 600 バイトを超えている場合，ヘッダ情報は出力されません。
- SDB 定義文の機能ごと（*ENTRY DICTIONARY 文などの制御文ごと）に行番号が付加されます。コメント行や区切り文字だけの行には行番号は付加されません。また，行長が 600 バイトを超えている行には行番号は付加されません。*ENTRY DICTIONARY 文などの制御文が行の先頭でない場合，行番号は継続されます。
- 行番号は 1～2,147,483,647 の範囲で設定されます。2,147,483,647 の次はまた 1 に戻ります。
- SDB 定義文に誤りがある場合は，誤りのある行の後にエラーメッセージが出力されます。
- SDB 定義文内の組み合わせに誤りがある場合は，最下行にエラーメッセージが出力されます。エラーがない場合は，SDB 定義文に対するメッセージが出力されます。

11.11　使用例

pdsdbdef コマンドの使用例を次に示します。

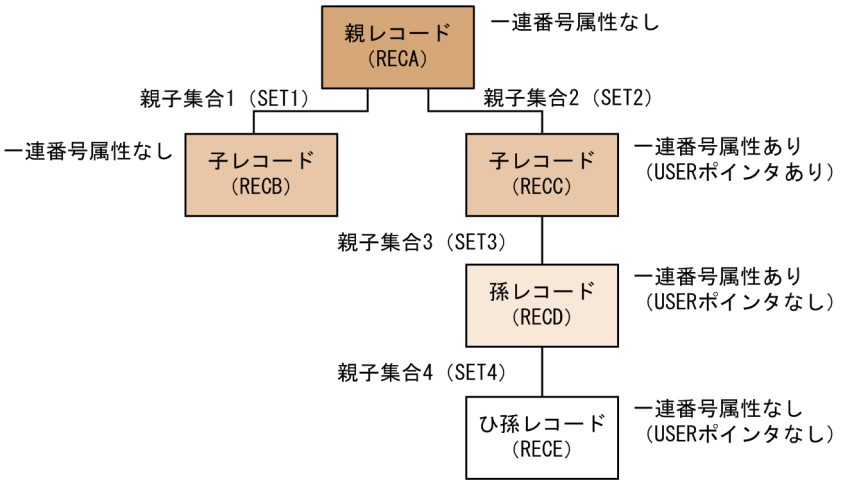
11.11.1　SDB データベースの定義例（4V FMB の場合）

4V FMB の SDB データベースを定義する場合の定義例を次に示します。

(1)　データベースの階層構造

作成する 4V FMB の SDB データベースの階層構造の概要を次の図に示します。

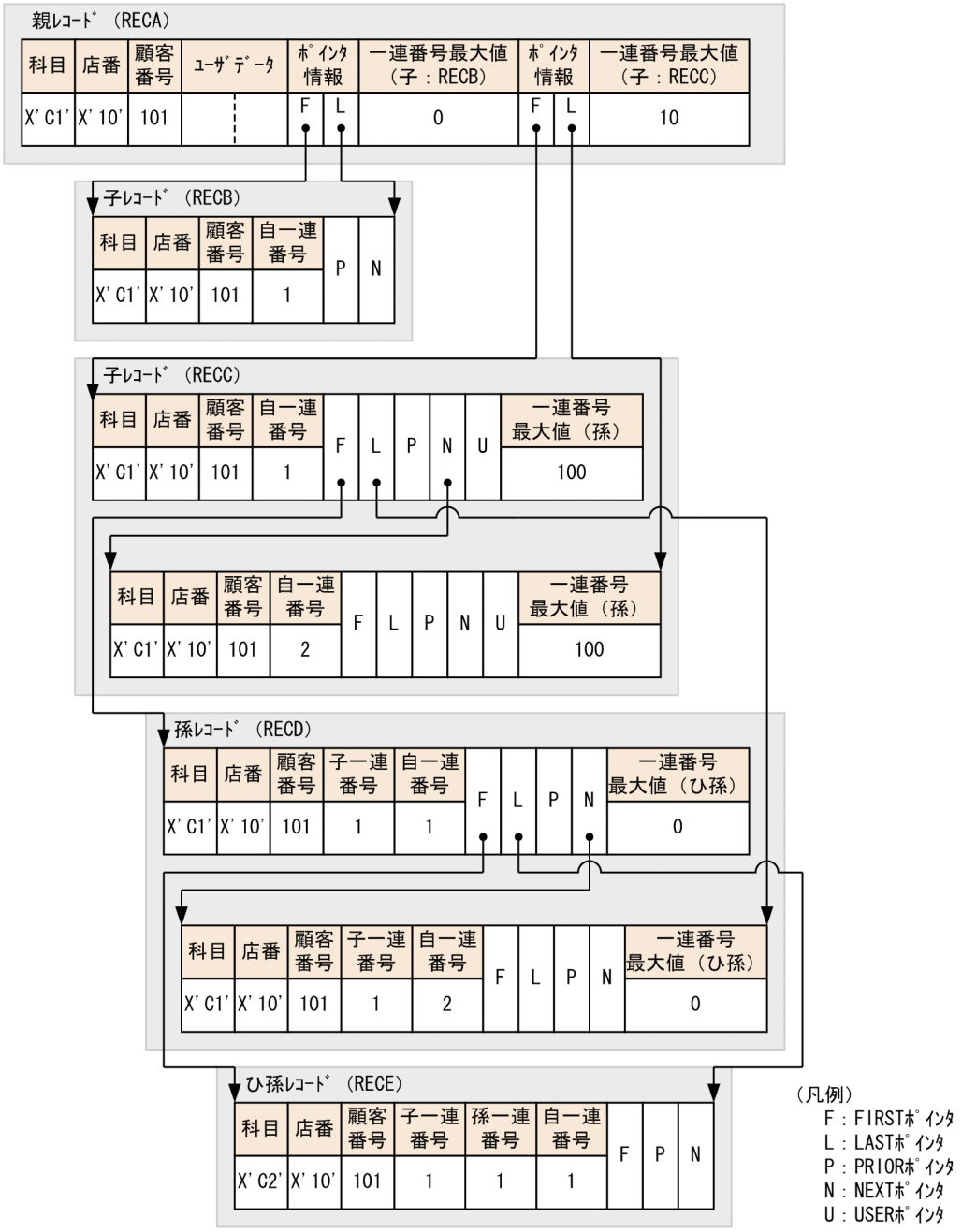
図 11-19　作成する 4V FMB の SDB データベースの階層構造



(2)　データベースの格納構造

作成する 4V FMB の SDB データベースの格納構造の概要を次の図に示します。

図 11-20　作成する 4V FMB の SDB データベースの格納構造



注1　ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽは省略しています。
注2　ﾕｰｻﾞﾃﾞｰﾀの構成要素は省略しています。

(3)　各レコード型のキーおよびキー構成

作成する 4V FMB の SDB データベースの，各レコード型のキーおよびキー構成を次の図に示します。

図 11-21　作成する 4V FMB の SDB データベースの，各レコード型のキーおよびキー構成

●親ﾚｺｰﾄﾞ：RECA

キー			ﾕｰｻﾞﾃﾞｰﾀ		管理情報			
構成要素定義必要			構成要素定義必要		子ﾚｺｰﾄﾞ：RECB		子ﾚｺｰﾄﾞ：RECC	
科目	店番	顧客番号	ﾕｰｻﾞﾃﾞｰﾀ	ﾕｰｻﾞﾌｧｲﾙ通番	各ﾎﾟｲﾝﾀ	一連番号最大値（子：RECB）	各ﾎﾟｲﾝﾀ	一連番号最大値（子：RECC）
DBKEY								

●子ﾚｺｰﾄﾞ：RECB

キー				ﾕｰｻﾞﾃﾞｰﾀ	管理情報
構成要素定義必要				構成要素定義必要	管理情報
科目	店番	顧客番号	自一連番号	ﾕｰｻﾞﾃﾞｰﾀ	各ﾎﾟｲﾝﾀ
			DBKEY		

●子ﾚｺｰﾄﾞ：RECC

キー				ﾕｰｻﾞﾃﾞｰﾀ			管理情報	
構成要素定義必要				構成要素定義必要			管理情報	
科目	店番	顧客番号	自一連番号	ﾕｰｻﾞﾃﾞｰﾀ	ﾕｰｻﾞｷｰ	ﾕｰｻﾞﾃﾞｰﾀ	各ﾎﾟｲﾝﾀ	一連番号最大値（孫：RECD）
			DBKEY					

●孫ﾚｺｰﾄﾞ：RECD

キー					ﾕｰｻﾞﾃﾞｰﾀ			管理情報	
構成要素定義必要					構成要素定義必要			管理情報	
科目	店番	顧客番号	子ﾚｺｰﾄﾞ（RECC）の一連番号	自一連番号	ﾕｰｻﾞﾃﾞｰﾀ	ﾕｰｻﾞｷｰ	ﾕｰｻﾞﾃﾞｰﾀ	各ﾎﾟｲﾝﾀ	一連番号最大値（ひ孫：RECE）
				DBKEY					

●ひ孫ﾚｺｰﾄﾞ：RECE

キー						ﾕｰｻﾞﾃﾞｰﾀ		管理情報
構成要素定義必要						構成要素定義必要		管理情報
科目	店番	顧客番号	子ﾚｺｰﾄﾞ（RECC）の一連番号	孫ﾚｺｰﾄﾞ（RECD）の一連番号	自一連番号	ﾕｰｻﾞﾃﾞｰﾀ	ﾕｰｻﾞﾃﾞｰﾀ	各ﾎﾟｲﾝﾀ
					DBKEY			

(4)　SDB データベース定義の定義例

SDB データベース定義の定義例（*ENTRY DICTIONARY 文の指定例）を次に示します。

```
*ENTRY DICTIONARY
 SCHEMA FMB01                                    /* SDBデータベース名(FMB01)*/
 DBTYPE 4V FMB                                   /* SDBデータベース種別     */
   RECORD RECA                                   /* レコード型名(RECA)      */
     2 DBKEY
       3 DBKEY_L2  XCHARACTER 1  TYPE  K,L      /* 科目     ┐※  */
       3 DBKEY_L3  XCHARACTER 1  TYPE  K,L      /* 店番     │    */
       3 DBKEY_L4  CHARACTER  3  TYPE  K,L      /* 顧客番号┘    */
```

```
    2 USERDA0     CHARACTER 30  TYPE  U,D    /* ユーザデータ             */
    2 USERDAN0    CHARACTER 10  TYPE  U,F    /* ユーザファイル通番       */
    FUNCTION
      REFER     USE                          /* 検索の指定               */
      ADD       USE                          /* 追加の指定               */
      UPDATE    USE                          /* 更新の指定               */
      ERASE     NOUSE                        /* 削除の指定               */
      ALLERASE  NOUSE                        /* 一括削除の指定           */
  RECORD RECB                                /* レコード型名(RECB)       */
    2 DBKEY_L2  XCHARACTER 1    TYPE  K,L    /* 科目                     */
    2 DBKEY_L3  XCHARACTER 1    TYPE  K,L    /* 店番                     */
    2 DBKEY_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                 */
    2 DBKEY     INTEGER         TYPE  N,N    /* 一連番号属性なし         */
    2 USERDB0   CHARACTER 20    TYPE  U,D    /* ユーザデータ             */
    FUNCTION
      REFER     USE                          /* 検索の指定               */
      ADD       USE                          /* 追加の指定               */
      UPDATE    USE                          /* 更新の指定               */
      ERASE     NOUSE                        /* 削除の指定               */
      ALLERASE  NOUSE                        /* 一括削除の指定           */
  RECORD RECC                                /* レコード型名(RECC)       */
    2 DBKEY_L2  XCHARACTER 1    TYPE  K,L    /* 科目                     */
    2 DBKEY_L3  XCHARACTER 1    TYPE  K,L    /* 店番                     */
    2 DBKEY_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                 */
    2 DBKEY     INTEGER         TYPE  K,N    /* 一連番号属性あり         */
    2 USERDC0   CHARACTER 10    TYPE  U,D    /* ユーザデータ             */
    2 USERDC1   CHARACTER 10    TYPE  U,K    /* ユーザキー               */
    2 USERDC2   CHARACTER 10    TYPE  U,D    /* ユーザデータ             */
    FUNCTION
      REFER     USE                          /* 検索の指定               */
      ADD       USE                          /* 追加の指定               */
      UPDATE    USE                          /* 更新の指定               */
      ERASE     NOUSE                        /* 削除の指定               */
      ALLERASE  NOUSE                        /* 一括削除の指定           */
  RECORD RECD                                /* レコード型名(RECD)       */
    2 DBKEY_L2  XCHARACTER 1    TYPE  K,L    /* 科目                     */
    2 DBKEY_L3  XCHARACTER 1    TYPE  K,L    /* 店番                     */
    2 DBKEY_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                 */
    2 DBKEY_P2  INTEGER         TYPE  K,P    /* 上位の一連番号(RECC用)   */
    2 DBKEY     INTEGER         TYPE  K,N    /* 一連番号属性あり         */
    2 USERDD0   CHARACTER 10    TYPE  U,D    /* ユーザデータ             */
    2 USERDD1   CHARACTER 10    TYPE  U,K    /* ユーザキー               */
    2 USERDD2   CHARACTER 10    TYPE  U,D    /* ユーザデータ             */
    FUNCTION
      REFER     USE                          /* 検索の指定               */
      ADD       USE                          /* 追加の指定               */
      UPDATE    USE                          /* 更新の指定               */
      ERASE     NOUSE                        /* 削除の指定               */
      ALLERASE  NOUSE                        /* 一括削除の指定           */
  RECORD RECE                                /* レコード型名(RECE)       */
    2 DBKEY_L2  XCHARACTER 1    TYPE  K,L    /* 科目                     */
    2 DBKEY_L3  XCHARACTER 1    TYPE  K,L    /* 店番                     */
    2 DBKEY_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                 */
    2 DBKEY_P2  INTEGER         TYPE  K,P    /* 上位の一連番号(RECC用)   */
    2 DBKEY_P3  INTEGER         TYPE  K,P    /* 上位の一連番号(RECD用)   */
    2 DBKEY     INTEGER         TYPE  N,N    /* 一連番号属性なし         */
    2 USERDD0   CHARACTER 10    TYPE  U,D    /* ユーザデータ             */
    2 USERDD2   CHARACTER 10    TYPE  U,D    /* ユーザデータ             */
```

```
    FUNCTION
        REFER   USE                          /* 検索の指定              */
        ADD       USE                        /* 追加の指定              */
        UPDATE    USE                        /* 更新の指定              */
        ERASE     NOUSE                      /* 削除の指定              */
        ALLERASE  NOUSE                      /* 一括削除の指定          */
  SET SET1                                   /* 親子集合(SET1)          */
    OWNER RECA                               /* 親レコードの指定        */
      ORDER LAST
    MEMBER RECB                              /* 子レコードの指定        */
      INSERTION AUTOMATIC
      RETENTION FIXED
    SETOPTION                                /* SETオプション           */
      OCCURRENCE NUMBER 0                    /* 一連番号属性なし        */

  SET SET2                                   /* 親子集合(SET2)          */
    OWNER RECA                               /* 親レコードの指定        */
      ORDER SORTED DUPLICATES PROHIBITED
    MEMBER RECC                              /* 子レコードの指定        */
      INSERTION AUTOMATIC
      RETENTION FIXED
      KEY ASCENDING USERDC1
    SETOPTION                                /* SETオプション           */
      OCCURRENCE NUMBER 10                   /* 一連番号属性あり        */
  SET SET3                                   /* 親子集合(SET3)          */
    OWNER RECC                               /* 子レコードの指定        */
      ORDER SORTED DUPLICATES PROHIBITED
    MEMBER RECD                              /* 孫レコードの指定        */
      INSERTION AUTOMATIC
      RETENTION FIXED
      KEY ASCENDING  USERDD1
    SETOPTION                                /* SETオプション           */
      OCCURRENCE NUMBER 100                  /* 一連番号属性あり        */
  SET SET4                                   /* 親子集合(SET4)          */
    OWNER RECD                               /* 孫レコードの指定        */
      ORDER LAST
    MEMBER RECE                              /* ひ孫レコードの指定      */
      INSERTION AUTOMATIC
      RETENTION FIXED
    SETOPTION                                /* SETオプション           */
      OCCURRENCE NUMBER 0                    /* 一連番号属性なし        */
  SDBOPTION                                  /* SDBオプション           */
    FUNCTION
      DBLODUTL USE            /* HiRDB/SDデータベース作成ユティリティ    */
END SCHEMA
```

注※

シーケンシャルインデクスの構成要素となります。

(5) SDB データベース格納定義の定義例

SDB データベース格納定義例（*ENTRY DICTIONARY 文の指定例）を次に示します。

```
*ENTRY DICTIONARY
 STORAGE SCHEMA FMB01 FOR FMB01              /* SDBデータベース格納名(FMB01)*/
```

```
                                          /* FOR SDBデータベース名(FMB01)*/
 DBTYPE 4V                                /* SDBデータベース種別         */
   STORAGE RECORD RECA                    /* 格納レコード名(RECA)        */
     SEQUENTIAL RECA_SI FOR RECORD  /* シーケンシャルインデクス名(RECA_SI)   */
       ORDER KEY ASCENDING  DBKEY_L2,DBKEY_L3,DBKEY_L4
       WITHIN FMBX01_I              /* シーケンシャルインデクス用RDエリア名  */
     WITHIN FMBX01                        /* 格納レコード用RDエリア名    */
   STORAGE RECORD RECB                    /* 格納レコード名(RECB)        */
     CLUSTERED SET1
     WITHIN FMBX01                        /* 格納レコード用RDエリア      */
   STORAGE RECORD RECC                    /* 格納レコード名(RECC)        */
     CLUSTERED SET2
     WITHIN FMBX01                        /* 格納レコード用RDエリア      */
   STORAGE RECORD RECD                    /* 格納レコード名(RECD)        */
     CLUSTERED SET3
     WITHIN FMBX01                        /* 格納レコード用RDエリア      */
   STORAGE RECORD RECE                    /* 格納レコード名(RECE)        */
     CLUSTERED SET4
     WITHIN FMBX01                        /* 格納レコード用RDエリア      */
   SET SET1                               /* 親子集合(SET1)              */
     OWNER POINTER FOR FIRST LAST MEMBER  /* 子(RECB)へのポインタ        */
     MEMBER POINTER FOR NEXT PRIOR TENANT /* 子の兄弟ポインタ            */
   SET SET2                               /* 親子集合(SET2)              */
     OWNER POINTER FOR FIRST LAST MEMBER  /* 子(RECC)へのポインタ        */
     MEMBER POINTER FOR NEXT PRIOR TENANT /* 子の兄弟ポインタ            */
   SET SET3                               /* 親子集合(SET3)              */
     OWNER POINTER FOR FIRST LAST USER MEMBER /* 孫(RECD)へのポインタ    */
     MEMBER POINTER FOR NEXT PRIOR TENANT /* 孫の兄弟ポインタ定義        */
   SET SET4                               /* 親子集合(SET4)              */
     OWNER POINTER FOR FIRST LAST MEMBER  /* ひ孫(RECE)へのポインタ      */
     MEMBER POINTER FOR NEXT PRIOR TENANT /* ひ孫の兄弟ポインタ          */
   SDBOPTION                              /* SDBオプション               */
END STORAGE SCHEMA
```

(6) SDB ディレクトリ情報の追加例

SDB ディレクトリ情報の追加例（*ENTRY DIRECTORY 文の指定例）を次に示します。

```
*ENTRY DIRECTORY
DBSCHM FMB01
```

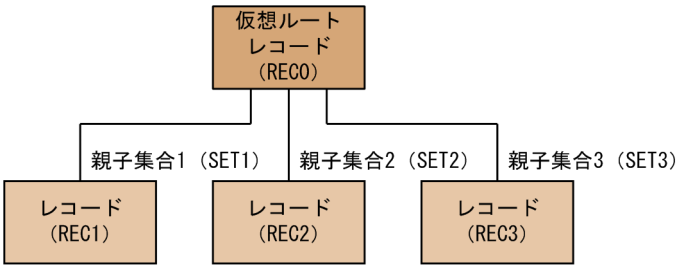
11.11.2 SDB データベースの定義例（4V DAM の場合）

4V DAM の SDB データベースを定義する場合の定義例を次に示します。

(1) データベースの階層構造

作成する 4V DAM の SDB データベースの階層構造の概要を次の図に示します。

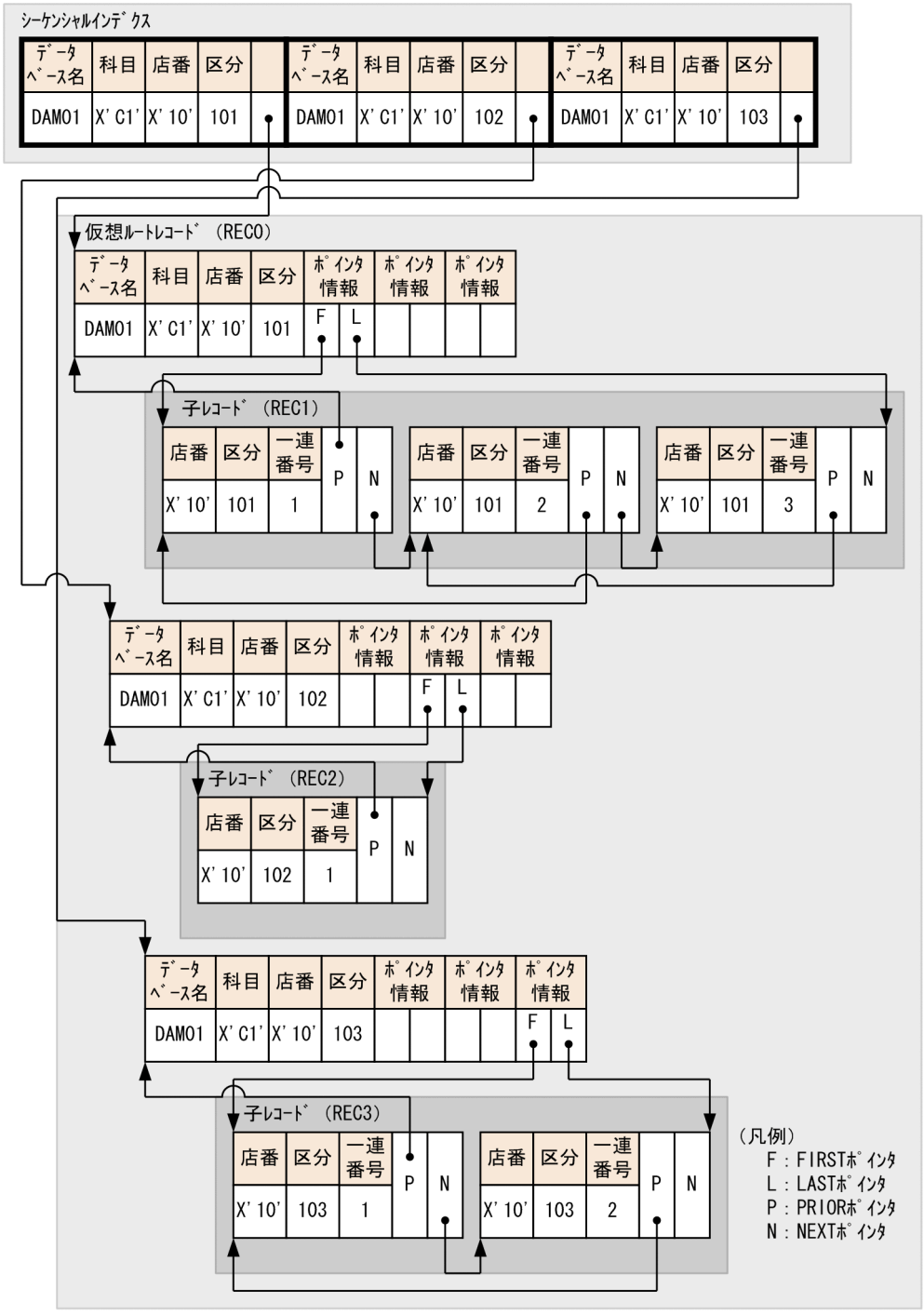
図 11-22　作成する 4V DAM の SDB データベースの階層構造



(2)　データベースの格納構造

作成する 4V DAM の SDB データベースの格納構造の概要を次の図に示します。

図 11-23　作成する 4V DAM の SDB データベースの格納構造



注1　二次ｲﾝﾃﾞｸｽは省略しています。
注2　子ﾚｺｰﾄﾞのﾃﾞｰﾀﾍﾞｰｽ名，科目，およびﾕｰｻﾞﾃﾞｰﾀの構成要素は省略しています。

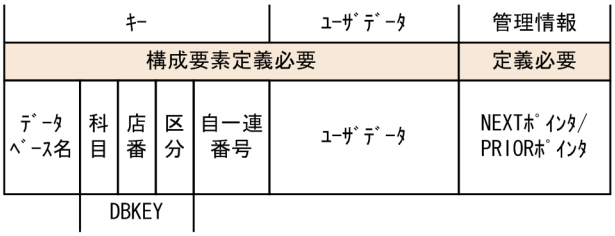
(3)　各レコード型のキーおよびキー構成

作成する 4V DAM の SDB データベースの，各レコード型のキーおよびキー構成を次の図に示します。

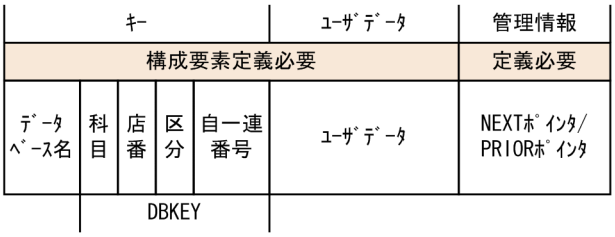
図 11-24　作成する 4V DAM の SDB データベースの，各レコード型のキーおよびキー構成

●仮想ﾙｰﾄﾚｺｰﾄﾞ：REC0

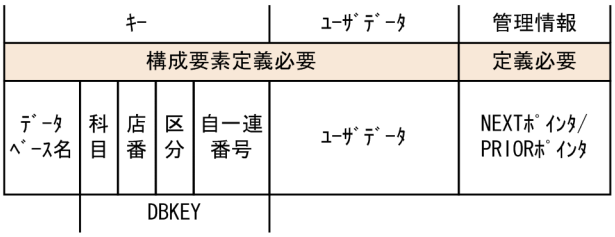
●子ﾚｺｰﾄﾞ1：REC1



●子ﾚｺｰﾄﾞ2：REC2



●子ﾚｺｰﾄﾞ3：REC3



(4)　SDB データベース定義の定義例

SDB データベース定義の定義例（*ENTRY DICTIONARY 文の指定例）を次に示します。

```
*ENTRY DICTIONARY
 SCHEMA DAM01                                    /* データベース名(DAM01)       */
 DBTYPE 4V DAM                                   /* SDBデータベース種別         */
   RECORD REC0                                  /* 仮想ルートレコード型名(REC0)*/
     2 DBKEY_L1   CHARACTER  5    TYPE  D,L     /* DB名     ┐                  */
     2 DBKEY_L2   XCHARACTER 1    TYPE  K,L     /* 科目     │※                */
     2 DBKEY_L3   XCHARACTER 1    TYPE  K,L     /* 店番     │                  */
     2 DBKEY_L4   CHARACTER  3    TYPE  K,R     /* 区分     ┘                  */
   RECORD REC1                                   /* レコード型名(REC1)          */
     2 DBKEY_L1   CHARACTER  5    TYPE  D,L     /* DB名                        */
     2 DBKEY                                    /* インデクス対応のキー情報    */
       3 DBKEY_L2   XCHARACTER 1  TYPE  K,L     /* 科目                        */
       3 DBKEY_L3   XCHARACTER 1  TYPE  K,L     /* 店番                        */
```

```
      3 DBKEY_L4  CHARACTER  3  TYPE  K,R    /* 区分(レコード分割キー)     */
      3 DBKEY_LN  INTEGER       TYPE  K,N    /* 一連番号あり               */
    2 USERD10     CHARACTER 20  TYPE  U,D    /* ユーザデータ               */
  RECORD REC2                                /* レコード型名(REC2)         */
    2 DBKEY_L1  CHARACTER 5     TYPE  D,L    /* DB名                       */
    2 DBKEY                                  /* インデクス対応のキー情報   */
      3 DBKEY_L2  XCHARACTER 1  TYPE  K,L    /* 科目                       */
      3 DBKEY_L3  XCHARACTER 1  TYPE  K,L    /* 店番                       */
      3 DBKEY_L4  CHARACTER  3  TYPE  K,R    /* 区分(レコード分割キー)     */
      3 DBKEY_LN  INTEGER       TYPE  K,N    /* 自一連番号あり             */
    2 USERD20  CHARACTER 20     TYPE  U,D    /* ユーザデータ               */
  RECORD REC3                                /* レコード型名(REC3)         */
    2 DBKEY_L1  CHARACTER 5     TYPE  D,L    /* DB名                       */
    2 DBKEY                                  /* インデクス対応のキー情報   */
      3 DBKEY_L2  XCHARACTER 1  TYPE  K,L    /* 科目                       */
      3 DBKEY_L3  XCHARACTER 1  TYPE  K,L    /* 店番                       */
      3 DBKEY_L4  CHARACTER  3  TYPE  K,R    /* 区分(レコード分割キー)     */
      3 DBKEY_LN  INTEGER       TYPE  K,N    /* 自一連番号あり             */
    2 USERD30     CHARACTER 10  TYPE  U,D    /* ユーザデータ               */
  SET SET1                                   /* 親子集合(SET1)             */
    OWNER REC0                               /* 親レコードの指定           */
      ORDER LAST
    MEMBER REC1                              /* 子レコードの指定           */
      INSERTION AUTOMATIC
      RETENTION FIXED
    SETOPTION                                /* SETオプション              */
      OCCURRENCE NUMBER 20                   /* 一連番号あり               */
  SET SET2                                   /* 親子集合(SET2)             */
    OWNER REC0                               /* 親レコードの指定           */
      ORDER LAST
    MEMBER REC2                              /* 子レコードの指定           */
      INSERTION AUTOMATIC
      RETENTION FIXED
    SETOPTION                                /* SETオプション              */
      OCCURRENCE NUMBER 20                   /* 一連番号あり               */
  SET SET3                                   /* 親子集合(SET3)             */
    OWNER REC0                               /* 親レコードの指定           */
      ORDER LAST
    MEMBER REC3                              /* 子レコードの指定           */
      INSERTION AUTOMATIC
      RETENTION FIXED
    SETOPTION                                /* SETオプション              */
      OCCURRENCE NUMBER 20                   /* 一連番号あり               */
  SDBOPTION                                  /* SDBオプション              */
    FUNCTION
      REFER     USE                          /* 検索の指定                 */
      ADD       USE                          /* 追加の指定                 */
      UPDATE    USE                          /* 更新の指定                 */
      ERASE     NOUSE                        /* 削除の指定                 */
      ALLERASE  NOUSE                        /* 一括削除の指定             */
      DBLODUTL  USE             /* HiRDB/SDデータベース作成ユティリティ   */
      FORMAT    NOUSE                        /* フォーマットライトの指定   */
END SCHEMA
```

注※

シーケンシャルインデクスの構成要素となります。

(5) SDB データベース格納定義の定義例

SDB データベース格納定義例（*ENTRY DICTIONARY 文の指定例）を次に示します。

```
*ENTRY DICTIONARY
 STORAGE SCHEMA DAM01 FOR DAM01              /* SDBデータベース格納名(DAM01) */
                                             /* FOR SDBデータベース名(DAM01) */
 DBTYPE 4V                                   /* SDBデータベース種別          */
   STORAGE RECORD REC0                       /* 格納レコード名(REC0)         */
     SEQUENTIAL REC0_SI FOR RECORD  /* シーケンシャルインデクス名(REC0_SI)  */
       ORDER KEY ASCENDING  DBKEY_L1, DBKEY_L2, DBKEY_L3, DBKEY_L4
       WITHIN DAMX01_I              /* シーケンシャルインデクス用RDエリア   */
     WITHIN DAMX01                           /* 格納レコード用RDエリア       */
   STORAGE RECORD REC1                       /* 格納レコード名(REC1)         */
     CLUSTERED SET1
     WITHIN DAMX01                           /* 格納レコード用RDエリア       */
     SECONDARY INDEX REC1_SI1                /* 二次インデクス名(REC1_SI1)   */
       ORDER KEY ASCENDING  DBKEY_L1, DBKEY_L2, DBKEY_L3,
                            DBKEY_L4, DBKEY_LN
       WITHIN DAMX01_I                       /* 二次インデクス用RDエリア     */
   STORAGE RECORD REC2                       /* 格納レコード名(REC2)         */
     CLUSTERED SET2
     WITHIN DAMX01                           /* 格納レコード用RDエリア       */
     SECONDARY INDEX REC2_SI1                /* 二次インデクス名(REC2_SI1)   */
       ORDER KEY ASCENDING  DBKEY_L1, DBKEY_L2, DBKEY_L3,
                            DBKEY_L4, DBKEY_LN
       WITHIN DAMX01_I                       /* 二次インデクス用RDエリア     */
   STORAGE RECORD REC3                       /* 格納レコード名(REC3)         */
     CLUSTERED SET3
     WITHIN DAMX01                           /* 格納レコード用RDエリア       */
     SECONDARY INDEX REC3_SI1                /* 二次インデクス名(REC3_SI1)   */
       ORDER KEY ASCENDING  DBKEY_L1, DBKEY_L2, DBKEY_L3,
                            DBKEY_L4, DBKEY_LN
       WITHIN DAMX01_I                       /* 二次インデクス用RDエリア     */
   SET SET1                                  /* 親子集合(SET1)               */
     OWNER POINTER FOR FIRST LAST MEMBER     /* 子(REC1)へのポインタ定義     */
     MEMBER POINTER FOR NEXT PRIOR TENANT    /* 子の兄弟ポインタ定義         */
   SET SET2                                  /* 親子集合(SET2)               */
     OWNER POINTER FOR FIRST LAST MEMBER     /* 子(REC2)へのポインタ定義     */
     MEMBER POINTER FOR NEXT PRIOR TENANT    /* 子の兄弟ポインタ定義         */
   SET SET3                                  /* 親子集合(SET3)               */
     OWNER POINTER FOR FIRST LAST MEMBER     /* 子(REC3)へのポインタ定義     */
     MEMBER POINTER FOR NEXT PRIOR TENANT    /* 子の兄弟ポインタ定義         */
   SDBOPTION                                 /* SDBオプション                */
     KEYDEF DBKEY_L1                         /* キー値(区分)とレコードの     */
       DATA 'DAM01'                          /* 対応関係を指定する。         */
     KEYDEF DBKEY_L2                         /* 仮想ルートレコード型のキーと */
       DATA X'C1'                            /* キー値を指定する             */
     KEYDEF DBKEY_L3
       DATA X'10'
     KEYDEF DBKEY_L4                         /* レコード分割キー             */
       DATA '101',REC1
       DATA '102',REC2
       DATA '103',REC3
END STORAGE SCHEMA
```

(6) SDB ディレクトリ情報の追加例

SDB ディレクトリ情報の追加例（*ENTRY DIRECTORY 文の指定例）を次に示します。

```
*ENTRY DIRECTORY
DBSCHM DAM01
```

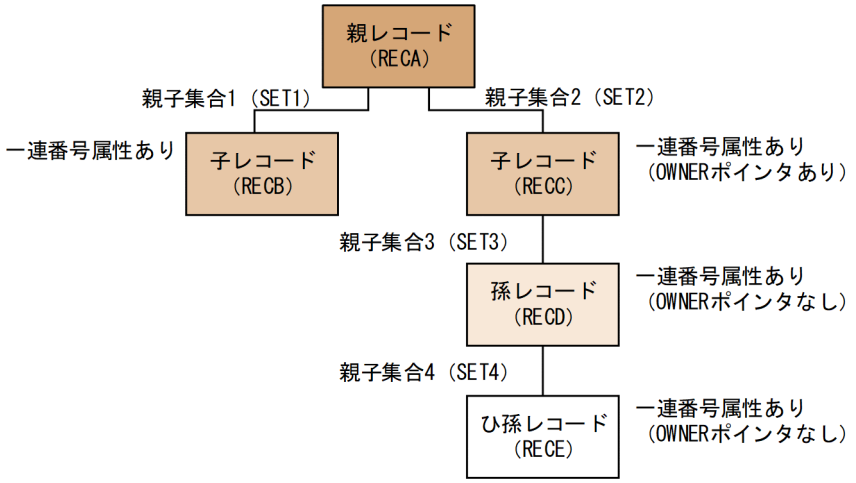
11.11.3 SDB データベースの定義例（SD FMB の場合）

SD FMB の SDB データベースを定義する場合の定義例を次に示します。

(1) データベースの階層構造

作成する SD FMB の SDB データベースの階層構造の概要を次の図に示します。

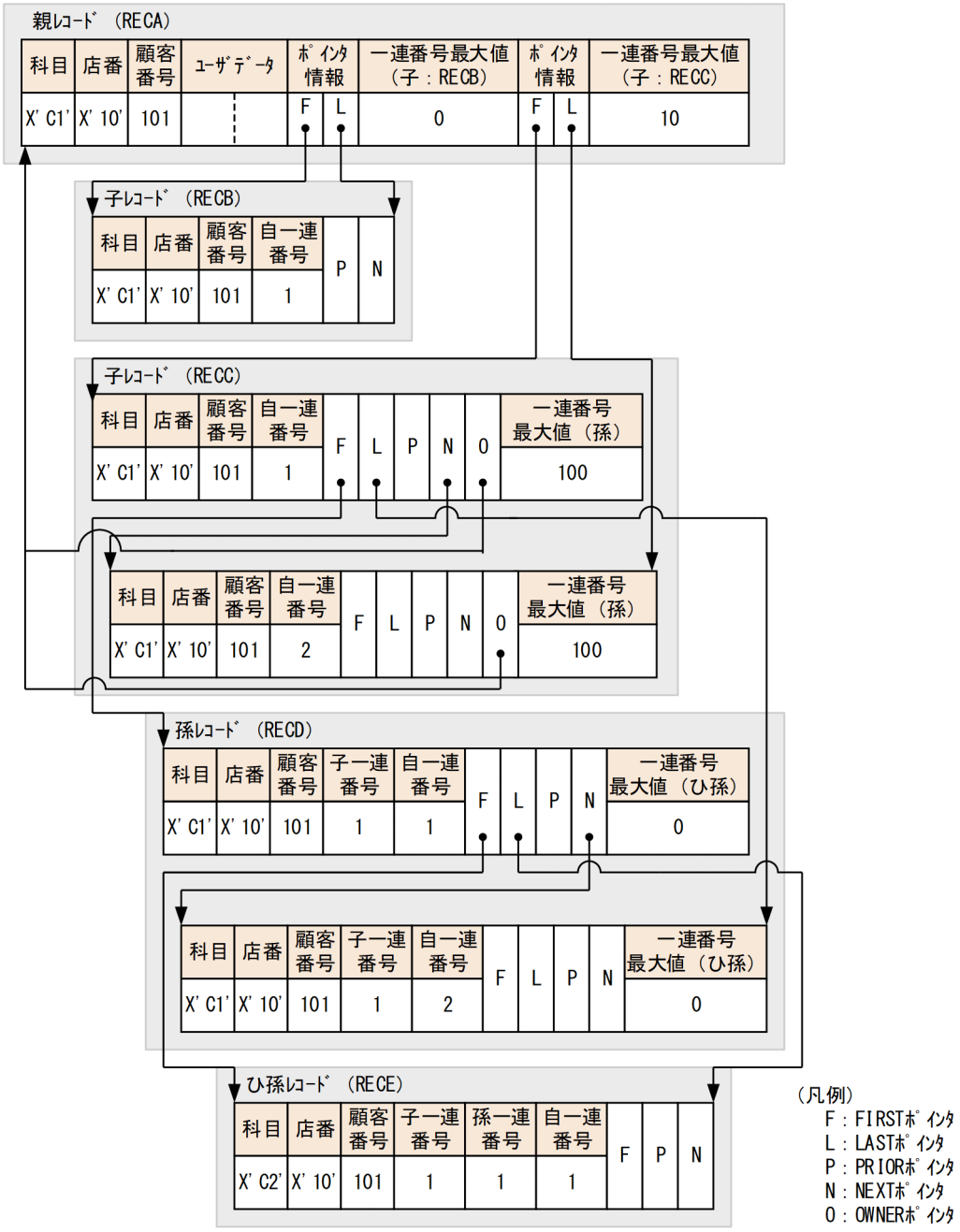
図 11-25 作成する SD FMB の SDB データベースの階層構造



(2)　データベースの格納構造

作成する SD FMB の SDB データベースの格納構造の概要を次の図に示します。

図 11-26　作成する SD FMB の SDB データベースの格納構造

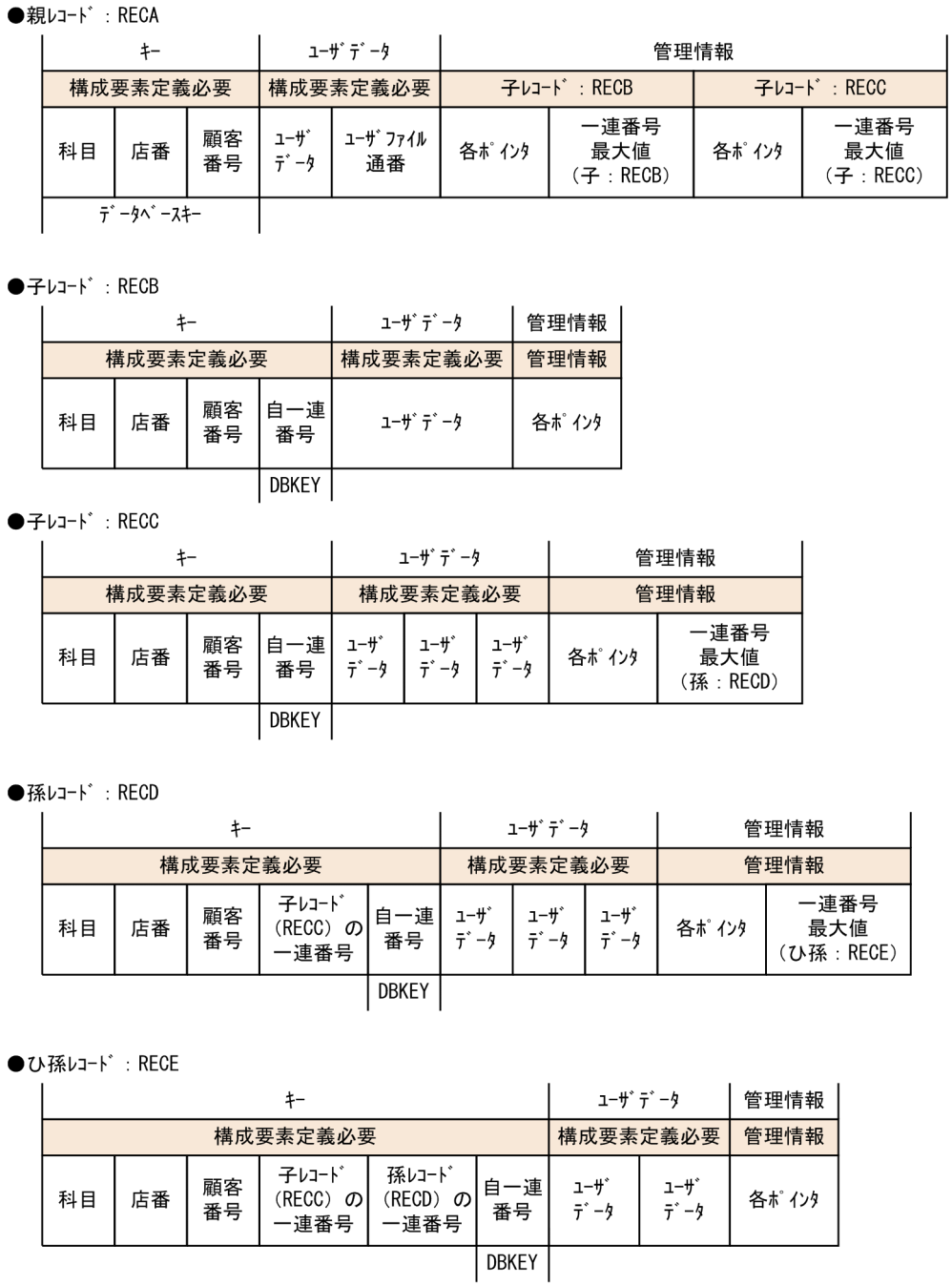


注1　ｼｰｹﾝｼｬﾙｲﾝﾃﾞｸｽは省略しています。
注2　ﾕｰｻﾞﾃﾞｰﾀの構成要素は省略しています。

(3)　各レコード型のキーおよびキー構成

作成する SD FMB の SDB データベースの，各レコード型のキーおよびキー構成を次の図に示します。

図 11-27　作成する SD FMB の SDB データベースの，各レコード型のキーおよびキー構成



(4)　SDB データベース定義の定義例

SDB データベース定義の定義例（*ENTRY DICTIONARY 文の指定例）を次に示します。

```
*ENTRY DICTIONARY
 SCHEMA FMB01                                    /* SDBデータベース名(FMB01)*/
 DBTYPE SD FMB                                   /* SDBデータベース種別     */
   RECORD RECA                                   /* レコード型名(RECA)      */
     2 KEYDATA
       3 KEYDATA_L2  XCHARACTER 1  TYPE  K,L     /* 科目     ┐※  */
       3 KEYDATA_L3  XCHARACTER 1  TYPE  K,L     /* 店番     │    */
       3 KEYDATA_L4  CHARACTER  3  TYPE  K,L     /* 顧客番号┘    */
```

```
  2 USERDA0     CHARACTER 30  TYPE  U,D    /* ユーザデータ            */
  2 USERDAN0    CHARACTER 10  TYPE  U,D    /* ユーザデータ            */
RECORD RECB                                /* レコード型名(RECB)      */
  2 KEYDATA_L2  XCHARACTER 1    TYPE  K,L    /* 科目                    */
  2 KEYDATA_L3  XCHARACTER 1    TYPE  K,L    /* 店番                    */
  2 KEYDATA_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                */
  2 DBKEY     INTEGER         TYPE  K,N    /* 一連番号属性あり        */
  2 USERDB0   CHARACTER 20    TYPE  U,D    /* ユーザデータ            */
RECORD RECC                                /* レコード型名(RECC)      */
  2 KEYDATA_L2  XCHARACTER 1    TYPE  K,L    /* 科目                    */
  2 KEYDATA_L3  XCHARACTER 1    TYPE  K,L    /* 店番                    */
  2 KEYDATA_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                */
  2 DBKEY     INTEGER         TYPE  K,N    /* 一連番号属性あり        */
  2 USERDC0   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
  2 USERDC1   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
  2 USERDC2   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
RECORD RECD                                /* レコード型名(RECD)      */
  2 KEYDATA_L2  XCHARACTER 1    TYPE  K,L    /* 科目                    */
  2 KEYDATA_L3  XCHARACTER 1    TYPE  K,L    /* 店番                    */
  2 KEYDATA_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                */
  2 DBKEY_P2  INTEGER         TYPE  K,P    /* 上位の一連番号(RECC用)  */
  2 DBKEY     INTEGER         TYPE  K,N    /* 一連番号属性あり        */
  2 USERDD0   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
  2 USERDD1   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
  2 USERDD2   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
RECORD RECE                                /* レコード型名(RECE)      */
  2 KEYDATA_L2  XCHARACTER 1    TYPE  K,L    /* 科目                    */
  2 KEYDATA_L3  XCHARACTER 1    TYPE  K,L    /* 店番                    */
  2 KEYDATA_L4  CHARACTER  3    TYPE  K,L    /* 顧客番号                */
  2 DBKEY_P2  INTEGER         TYPE  K,P    /* 上位の一連番号(RECC用)  */
  2 DBKEY_P3  INTEGER         TYPE  K,P    /* 上位の一連番号(RECD用)  */
  2 DBKEY     INTEGER         TYPE  K,N    /* 一連番号属性あり        */
  2 USERDE0   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
  2 USERDE2   CHARACTER 10    TYPE  U,D    /* ユーザデータ            */
SET SET1                                   /* 親子集合(SET1)          */
  OWNER RECA                               /* 親レコードの指定        */
    ORDER SORTED DUPLICATES PROHIBITED
  MEMBER RECB                              /* 子レコードの指定        */
    INSERTION AUTOMATIC
    RETENTION FIXED
    KEY ASCENDING USERDB0
SET SET2                                   /* 親子集合(SET2)          */
  OWNER RECA                               /* 親レコードの指定        */
    ORDER LAST
  MEMBER RECC                              /* 子レコードの指定        */
    INSERTION AUTOMATIC
    RETENTION FIXED
SET SET3                                   /* 親子集合(SET3)          */
  OWNER RECC                               /* 子レコードの指定        */
    ORDER LAST
  MEMBER RECD                              /* 孫レコードの指定        */
    INSERTION AUTOMATIC
    RETENTION FIXED
SET SET4                                   /* 親子集合(SET4)          */
  OWNER RECD                               /* 孫レコードの指定        */
    ORDER LAST
  MEMBER RECE                              /* ひ孫レコードの指定      */
    INSERTION AUTOMATIC
```

```
       RETENTION FIXED
END SCHEMA
```

注※

シーケンシャルインデクスの構成要素となります。

(5) SDB データベース格納定義の定義例

SDB データベース格納定義例（*ENTRY DICTIONARY 文の指定例）を次に示します。

```
*ENTRY DICTIONARY
 STORAGE SCHEMA FMB01 FOR FMB01                /* SDBデータベース格納名(FMB01)*/
                                               /* FOR SDBデータベース名(FMB01)*/
 DBTYPE SD                                     /* SDBデータベース種別         */
   STORAGE RECORD RECA                         /* 格納レコード名(RECA)        */
     SEQUENTIAL RECA_SI FOR RECORD  /* シーケンシャルインデクス名(RECA_SI)   */
       ORDER KEY ASCENDING  KEYDATA_L2,KEYDATA_L3,KEYDATA_L4
       WITHIN FMBX01_I              /* シーケンシャルインデクス用RDエリア名  */
     WITHIN FMBX01                             /* 格納レコード用RDエリア名    */
   STORAGE RECORD RECB                         /* 格納レコード名(RECB)        */
     CLUSTERED SET1
     WITHIN FMBX01                             /* 格納レコード用RDエリア      */
   STORAGE RECORD RECC                         /* 格納レコード名(RECC)        */
     CLUSTERED SET2
     WITHIN FMBX01                             /* 格納レコード用RDエリア      */
     SECONDARY INDEX RECC_SI1                  /* 二次インデクス名(REC1_SI1)  */
     USED FOR RECORD
     ORDER KEY ASCENDING KEYDATA_L2, KEYDATA_L3, USERDC1
     WITHIN FMBX01_I                           /* 二次インデクス用RDエリア    */
   STORAGE RECORD RECD                         /* 格納レコード名(RECD)        */
     CLUSTERED SET3
     WITHIN FMBX01                             /* 格納レコード用RDエリア      */
   STORAGE RECORD RECE                         /* 格納レコード名(RECE)        */
     CLUSTERED SET4
     WITHIN FMBX01                             /* 格納レコード用RDエリア      */
   SET SET1                                    /* 親子集合(SET1)              */
     OWNER POINTER FOR FIRST LAST MEMBER       /* 子(RECB)へのポインタ        */
     MEMBER POINTER FOR NEXT PRIOR TENANT      /* 子の兄弟ポインタ            */
   SET SET2                                    /* 親子集合(SET2)              */
     OWNER POINTER FOR FIRST LAST MEMBER       /* 子(RECC)へのポインタ        */
     MEMBER POINTER FOR OWNER NEXT PRIOR TENANT /* 親ポインタおよび子の兄弟ポインタ */
   SET SET3                                    /* 親子集合(SET3)              */
     OWNER POINTER FOR FIRST LAST MEMBER       /* 孫(RECD)へのポインタ        */
     MEMBER POINTER FOR NEXT PRIOR TENANT      /* 孫の兄弟ポインタ定義        */
   SET SET4                                    /* 親子集合(SET4)              */
     OWNER POINTER FOR FIRST LAST MEMBER       /* ひ孫(RECE)へのポインタ      */
     MEMBER POINTER FOR NEXT PRIOR TENANT      /* ひ孫の兄弟ポインタ          */
END STORAGE SCHEMA
```

(6) SDB ディレクトリ情報の追加例

SDB ディレクトリ情報の追加例（*ENTRY DIRECTORY 文の指定例）を次に示します。

```
*ENTRY DIRECTORY
DBSCHM FMB01
```

12 HiRDB/SD データベース作成ユティリティ (pdsdblod)

この章では，HiRDB/SD データベース作成ユティリティの機能と使い方について説明します。

なお，この章では，HiRDB/SD データベース作成ユティリティを pdsdblod コマンドと表記しています。

12.1　機能

pdsdblod コマンドには次の機能があります。

- データロード
- フォーマットライト
- インデクスの再作成

各機能について説明します。

12.1.1　データロード

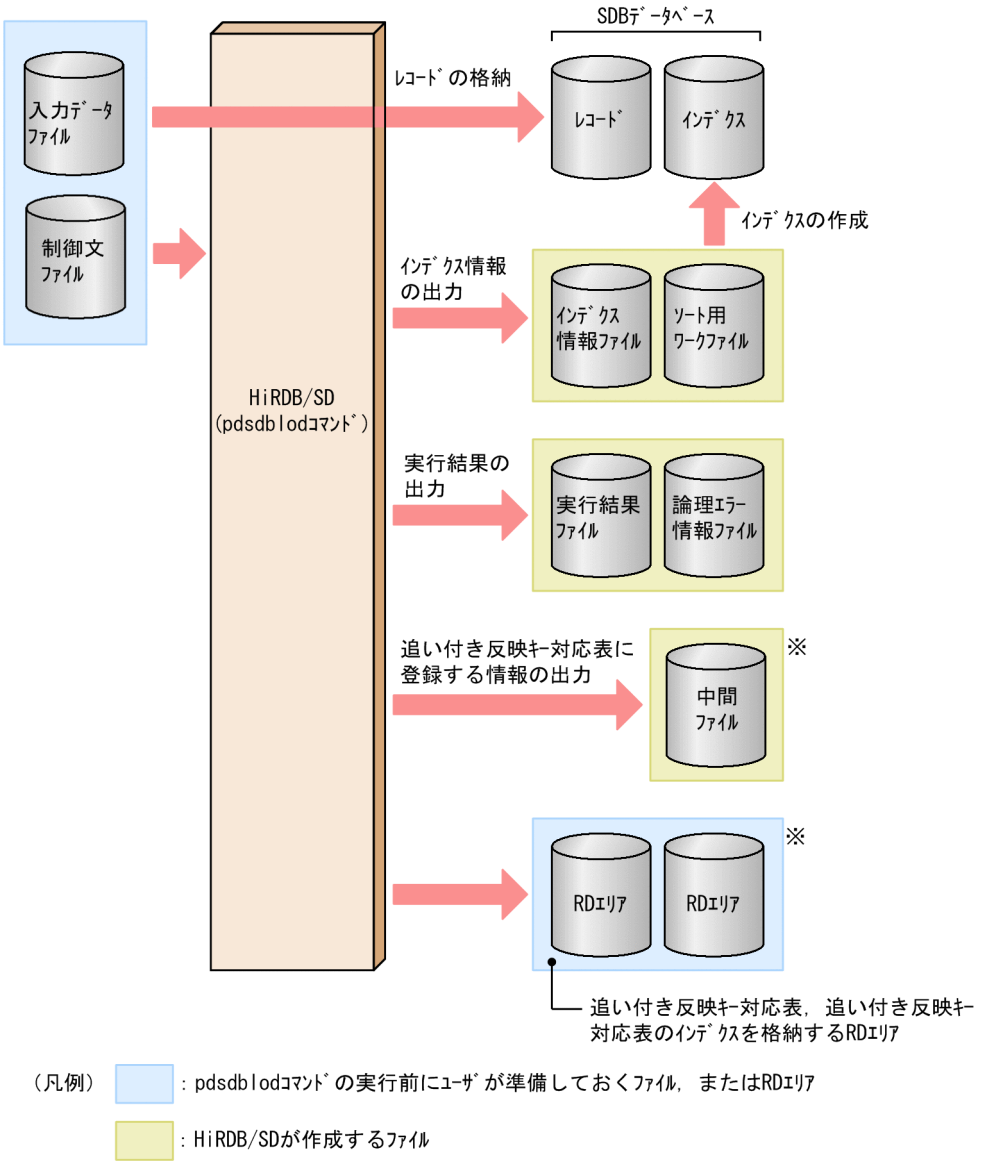
pdsdbdef コマンドで定義したデータベースにレコードを格納し，インデクスを作成します。この機能をデータロードといいます。データロードには，次の表に示す 2 種類があります。

表 12-1　データロードの種類

データロードの種類	説明
初期データロード	environment 文の purge オペランドに yes を指定して実行するデータロードのことです。この場合，格納済みのレコードを削除してから，データロードが実行されます。
追加データロード	environment 文の purge オペランドに no を指定するか，または purge オペランドを省略して実行するデータロードのことです。この場合，格納済みのレコードは削除されません。入力データファイルのデータが追加でデータロードされます。 追加データロードは，4V FMB または SD FMB の SDB データベースの場合に実行できます。

データロードの概要を次の図に示します。

図 12-1　データロードの概要



注※

更新可能なオンライン再編成を実行する場合に必要となる RD エリア，および作成されるファイルです。

[説明]

- 入力データファイル中のレコードをデータベースに格納します。
- インデクスに関する情報がインデクス情報ファイルに出力され，ソート用ワークファイルでインデクスのデータをソートしてインデクスが作成されます。
- 更新可能なオンライン再編成を実行する場合，追い付き反映キー対応表に登録する情報を中間ファイルに一時的に出力します。その後，追い付き反映キー対応表にデータが登録されます。

SDB データベース定義の DBLODUTL 句に USE を指定したデータベースに対してだけデータロードを実行できます。

(1)　データロードの際に準備するファイル

データロードを実行する前に，次のファイルを準備しておきます。これらのファイルは通常ファイル上に作成してください。

- 入力データファイル

  データベースに格納するレコードを格納しているファイルです。入力データファイルは決まった形式で作成する必要があります。入力データファイルの形式については，「12.6　入力データファイルの形式」を参照してください。

- 制御文ファイル

  pdsdblod コマンドの動作条件を指定した pdsdblod 制御文を格納したファイルです。pdsdblod 制御文については，「12.3　pdsdblod 制御文」を参照してください。

(2)　データロードの際に準備する RD エリア【4V FMB】

更新可能なオンライン再編成を実行する前に，次の RD エリアを準備しておきます。

- 追い付き反映キー対応表を格納する RD エリア
- 追い付き反映キー対応表のインデクスを格納する RD エリア

詳細については，「5.13.3(6)　追い付き反映キー対応表，追い付き状態管理表を格納する RD エリアの作成（pdmod コマンド）」を参照してください。

(3)　データロードの際に作成されるファイル

データロードの際，HiRDB/SD は次に示すファイルを作成します。これらのファイルは通常ファイル上に作成されます。

- インデクス情報ファイル，ソート用ワークファイル

  インデクスを一括作成する際に一時的に作成されるファイルです。pdsdblod コマンドが正常終了した場合，これらのファイルは自動的に削除されます。

  追加データロードのインデクス一括作成モードの場合は，レコード格納前に既存レコードのインデクス情報をインデクス情報ファイルに出力します。その後，追加するレコードのインデクス情報をこのインデクス情報ファイルに追加します。そのため，レコード格納後には，追加するレコードだけでなく，SDB データベースに格納済みのレコードを含めた全レコードのインデクス情報ファイルが作成されます。

- 実行結果ファイル

  pdsdblod コマンドの実行結果を出力するファイルです。実行結果ファイルに出力される情報については，「12.7　実行結果ファイルの出力形式」を参照してください。

- 論理エラー情報ファイル

  入力データファイル中のレコードに論理エラーがあった場合，論理エラーに関する情報を出力するファイルです。

HiRDB/SD はデータベースの定義に合っていない不整合なレコードを論理エラーのレコードと判定します。データロードの際は，正常なレコードだけが格納され，論理エラーと判定されたレコードは格納されません。論理エラーが発生した場合，どのようなエラーで，レコード中のどの個所にエラーがあったかという情報が論理エラー情報ファイルに出力されます。論理エラー情報ファイルに出力される情報については，「12.8.2　論理エラー情報ファイルの出力形式」を参照してください。

- **中間ファイル**

  追い付き反映キー対応表に登録する情報を一時的に格納するファイルです。更新可能なオンライン再編成の実行時に作成されます。

(4)　インデクスの作成モードの選択

データロードを実行する際，インデクスの作成方法（作成モード）を選択します。次に示すどちらかのモードを選択してください。

- インデクス一括作成モード
- インデクス更新モード

インデクス一括作成モードを選択した場合，すべてのレコードの格納処理が完了したあとに，まとめてインデクスが作成されます。インデクス更新モードを選択した場合，1 レコードを格納するたびにそのレコードのインデクスが作成されます。

ポイント

通常は，性能面で優れているインデクス一括作成モードを選択してください。ただし，データロード時にシーケンシャルインデクスのキー値の重複が検出された場合（KFPB63481-E メッセージが出力された場合）は，インデクス更新モードでデータロードを再実行してください。

理由：

インデクス一括作成モードでデータロードを実行した場合，シーケンシャルインデクスのキー値の重複情報が論理エラー情報として出力されないため，どのキー値が重複しているのかを特定できません。一方，インデクス更新モードでデータロードを実行した場合，シーケンシャルインデクスのキー値の重複情報が論理エラー情報として出力されるため，どのキー値が重複しているのかを特定できます。

pdsdblod 制御文の load 文の idxmode オペランドでインデクスの作成モードを選択します。

(5)　SDB データベースが横分割されている場合のデータロード

SDB データベースが複数の RD エリアに横分割されている場合のデータロードについて説明します。

- **SDB データベースがサーバ内横分割されている場合**

複数の RD エリアに対して一括してデータロードが実行できます（pdsdblod コマンドの一度の実行でデータロードが完了します）。また，RD エリア単位にデータロードを実行することもできます。この場合，pdsdblod コマンドを複数回実行します。

- SDB データベースがサーバ間横分割されている場合

  データロードはバックエンドサーバ単位でしか実行できないため，SDB データベースがサーバ間横分割されている場合は，バックエンドサーバ単位にデータロードを実行します。

(6) 一連番号の扱い

一連番号は入力データ順に自動的に 1 から採番され，付与されます。

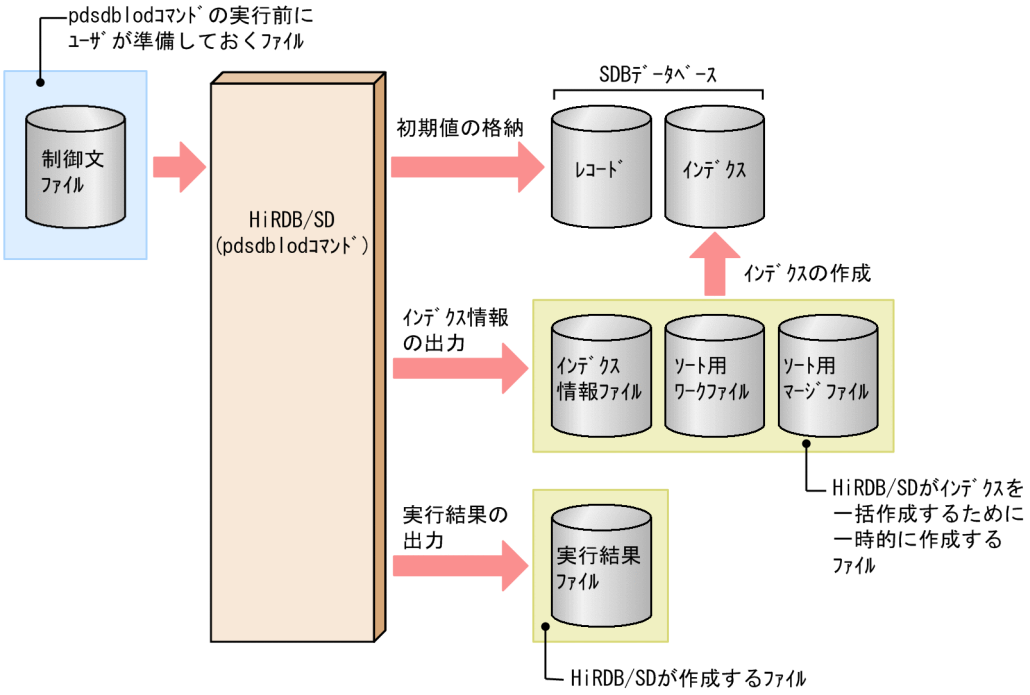
(7) 追加データロード【4V FMB，SD FMB】

格納されていないデータベースキーを持つルートレコードのファミリ単位で追加データロードができます。

12.1.2 フォーマットライト【4V DAM】

pdsdbdef コマンドで定義したデータベースに初期値を格納し，インデクスを作成します。この機能をフォーマットライトといいます。初期値は HiRDB/SD が自動的に作成します。データロードとは異なり，HiRDB/SD の利用者が格納データを準備する必要はありません。フォーマットライトの概要を次の図に示します。

図 12-2 フォーマットライトの概要



［説明］

- データベースの定義（SDB データベース定義および SDB データベース格納定義）に従って HiRDB/SD が初期値を作成してデータベースに格納します。
- インデクスに関する情報がインデクス情報ファイルに出力され，ソート用ワークファイルでインデクスのデータをソートしてインデクスが作成されます。

SDB データベース定義の FORMAT 句に USE を指定したレコード型に対してだけフォーマットライトを実行できます。

フォーマットライトは，4V DAM の SDB データベースの場合に実行できます。

(1)　フォーマットライトの際に準備するファイルと作成されるファイル

フォーマットライトの際に準備するファイルと作成されるファイルについては，「12.1.1(1)　データロードの際に準備するファイル」および「12.1.1(3)　データロードの際に作成されるファイル」を参照してください。

(2)　インデクスの作成モードの選択

インデクスの作成モードもデータロードと同じく，次に示す 2 つのモードがあります。

- インデクス一括作成モード
- インデクス更新モード

ポイント

フォーマットライトの場合，シーケンシャルインデクスのキー値が重複することがありません。そのため，性能面で優れているインデクス一括作成モードを使用してください。

(3)　SDB データベースが横分割されている場合のフォーマットライト

SDB データベースが複数の RD エリアに横分割されている場合のフォーマットライトの方法を説明します。

- SDB データベースがサーバ内横分割されている場合

  複数の RD エリアに対して一括してフォーマットライトが実行できます（pdsdblod コマンドの一度の実行でフォーマットライトが完了します）。また，RD エリア単位にフォーマットライトを実行することもできます。この場合，pdsdblod コマンドを複数回実行します。
- SDB データベースがサーバ間横分割されている場合

  フォーマットライトはバックエンドサーバ単位でしか実行できないため，SDB データベースがサーバ間横分割されている場合は，バックエンドサーバ単位にフォーマットライトを実行します。

(4) 初期値として格納されるデータ

初期値として格納されるデータは，データベースの定義（SDB データベース定義および SDB データベース格納定義）に従って HiRDB/SD が自動的に作成します。HiRDB/SD が作成する初期値の例を次に示します。

SDB データベース定義

```
    :
  SCHEMA DAM1
  DBTYPE 4V DAM
  RECORD REC0  …仮想ルートレコード
    2 DBKDBMEI          CHARACTER 4   TYPE D,L       …データベース名
    2 DBKHIDUKE         XCHARACTER 1  TYPE K,L       …日付
    2 DBKTENBAN         XCHARACTER 3  TYPE K,L       …店番
    2 DBKKUBUN          XCHARACTER 4  TYPE K,R       …区分
  RECORD REC1
    2 DBKDBMEI          CHARACTER 4   TYPE D,L       …データベース名
    2 DBKEY
     3 DBKHIDUKE        XCHARACTER 1  TYPE K,L       …日付
     3 DBKTENBAN        XCHARACTER 3  TYPE K,L       …店番
     3 DBKKUBUN         XCHARACTER 4  TYPE K,R       …区分
     3 DBKITIRENBANGOU INTEGER        TYPE K,N       …一連番号
    2 USERDA1           CHARACTER 5   TYPE U,D       …ユーザデータ
  RECORD REC2
    2 DBKDBMEI          CHARACTER 4   TYPE D,L       …データベース名
    2 DBKEY
     3 DBKHIDUKE        XCHARACTER 1  TYPE K,L       …日付
     3 DBKTENBAN        XCHARACTER 3  TYPE K,L       …店番
     3 DBKKUBUN         XCHARACTER 4  TYPE K,R       …区分
     3 DBKITIRENBANGOU INTEGER        TYPE K,N       …一連番号
    2 USERDA2             CHARACTER 10  TYPE U,D     …ユーザデータ
  SET 親子集合1
      :
    MEMBER REC1
      :
    OCCURRENCE NUMBER 10  …一連番号の最大値
  SET 親子集合2
      :
    MEMBER REC2
      :
    OCCURRENCE NUMBER 20  …一連番号の最大値
          :
```

SDB データベース格納定義

```
        :
STORAGE SCHEMA DAM1 FOR DAM1
        :
  SDBOPTION
    KEYDEF DBKDBMEI                  …データベース名
      DATA DAM1
    KEYDEF DBKHIDUKE                 …日付
      DATA X'C1'
      DATA X'C2'
```

```
  KEYDEF DBKTENBAN                  …店番
    DATA X'000011'
    DATA X'000012'
  KEYDEF DBKKUBUN                   …区分
    DATA X'00000001', REC1
    DATA X'00000002', REC2
      :
```

初期値として格納されるデータ

キー部					ユーザデータ部
データベース名	日付	店番	区分	一連番号	ユーザデータ※1
DAM1	X'C1'	X'000011'	X'00000001'	1～10※2	X'0000000000'
DAM1	X'C1'	X'000011'	X'00000002'	1～20※2	X'00000000000000000000'
DAM1	X'C1'	X'000012'	X'00000001'	1～10※2	X'0000000000'
DAM1	X'C1'	X'000012'	X'00000002'	1～20※2	X'00000000000000000000'
DAM1	X'C2'	X'000011'	X'00000001'	1～10※2	X'0000000000'
DAM1	X'C2'	X'000011'	X'00000002'	1～20※2	X'00000000000000000000'
DAM1	X'C2'	X'000012'	X'00000001'	1～10※2	X'0000000000'
DAM1	X'C2'	X'000012'	X'00000002'	1～20※2	X'00000000000000000000'

注※1

対応するレコードごとに構成要素を切り替えます。

注※2

SDB データベース定義の OCCURRENCE NUMBER 句に指定した一連番号の最大値の数だけ，レコードが作成されます。

キー部の作成仕様

- SDB データベース定義の RECORD 句に指定した DBKEY に対応する構成要素に対して，SDB データベース格納定義の KEYDEF 句に指定したすべてのキー値の組み合わせが作成されます。
- SDB データベース定義の RECORD 句に指定した DBKEY に対応する構成要素に対して，下位の構成要素から先頭の構成要素の順番に，キー値を組み替えながらすべての組み合わせが作成されます。
- 1 つの構成要素に設定するキー値は，対応する SDB データベース格納定義の KEYDEF 句に指定したキー値の指定順に設定されます。
- 一連番号の構成要素は，1 から SDB データベース定義の OCCURRENCE NUMBER 句に指定した一連番号の最大値（指定値が 0 の場合は 1）までが採番されます。

- SDB データベース格納定義の KEYDEF 句で指定したキー値の定義順にキー部が作成されます。作成されたキー部（データベースキー）が SEQUENTIAL 句で定義した並びと異なる場合，論理エラーとなるため注意してください。

ユーザデータ部の作成仕様

- ユーザデータ部に格納される初期値は，構成要素のデータ型によって決まります。構成要素のデータ型とユーザデータ部に格納される初期値を次の表に示します。

表 12-2　構成要素のデータ型とユーザデータ部に格納される初期値

構成要素のデータ型	データ長（単位：バイト）	ユーザデータ部に格納される初期値
CHARACTER n	n	X'00....00'（0 を代入）
XCHARACTER n	n	X'00....00'（0 を代入）
PACKED DECIMAL FIXED p,s	↓（p ＋ s）÷2↓＋ 1	X'00....0C' （最終 8 ビットを X'0C'とし，残りは 0 を代入）
INTEGER	4	X'00....00'（0 を代入）

格納されるレコード数の仕様

フォーマットライトによって格納されるレコードの最大数は次のようになります。

（A÷B）×C

A：SDB データベース格納定義の KEYDEF 句で指定したすべてのキー値の組み合わせ数

B：格納先のレコード数

C：SDB データベース定義の OCCURRENCE NUMBER 句に指定した一連番号の最大値の総数

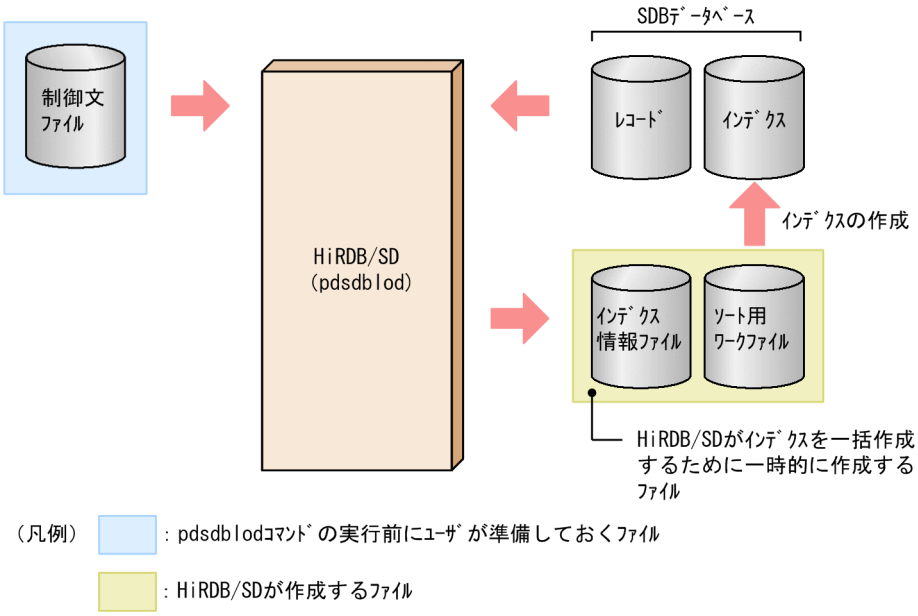
(5)　注意事項

- SDB データベース格納定義の KEYDEF 句で，すべてのキーに対するキー値を過不足なく定義しておいてください。過不足があった場合でも，定義情報に従ってフォーマットライトが実行されます。
- ユーザキー（SDB データベース定義の RECORD 句で構成要素のデータ種別を指定する TYPE 句に U,K を指定）がある場合，フォーマットライト機能は実行できません。この場合，フォーマットライトしたい値で入力データファイルを作成し，データロードを実行してください。

12.1.3　インデクスの再作成

レコード格納用 RD エリアのレコード型中のデータから，インデクスを作成することができます。この機能をインデクスの再作成といいます。インデクスの再作成の概要を次の図に示します。

図 12-3　インデクスの再作成の概要



[説明]

- レコード格納用 RD エリアのレコード型中のデータの内容からインデクスの作成に必要な情報を取得します。
- インデクスに関する情報がインデクス情報ファイルに出力され，ソート用ワークファイルでインデクスのデータをソートしてインデクスが作成されます。

SDB データベース定義の DBLODUTL 句に USE を指定した SDB データベースに対してだけ，インデクスを再作成できます。

(1)　インデクスを再作成する際に準備するファイル

インデクスを再作成する前に，次のファイルを準備しておきます。これらのファイルは通常ファイル上に作成してください。

- **制御文ファイル**

  pdsdblod コマンドの動作条件を指定した pdsdblod 制御文を格納したファイルです。pdsdblod 制御文については，「12.3　pdsdblod 制御文」を参照してください。

(2)　インデクスを再作成する際に作成されるファイル

インデクスを再作成する際，HiRDB/SD は次に示すファイルを作成します。これらのファイルは通常ファイル上に作成されます。

- **インデクス情報ファイル，ソート用ワークファイル**

  インデクスを再作成する際に一時的に作成されるファイルです。pdsdblod コマンドが正常終了した場合，これらのファイルは自動的に削除されます。

(3) インデクスを再作成する際のインデクス作成単位

インデクス格納用 RD エリアに格納されているインデクスごとに，インデクスを再作成します。

SDB データベースを複数の RD エリアに横分割している場合は，インデクス格納用 RD エリアごとにインデクスを再作成します。

12.2　コマンドの形式

```
pdsdblod 制御文ファイル名
```

制御文ファイル名：

pdsdblod 制御文を指定した制御文ファイルの名称を絶対パスまたは相対パスで指定します。pdsdblod 制御文については，次の「12.3　pdsdblod 制御文」で説明します。

12.3　pdsdblod 制御文

pdsdblod 制御文には，pdsdblod コマンドの動作条件を指定します。pdsdblod 制御文は，environment 文，load 文，dbinf 文，index 文，idxload 文，および oreload 文で構成されています。ここでは，各文の指定形式と各文に指定するオペランドについて説明します。

注意事項

- pdsdblod 制御文は次の表に示す順序で指定してください。

  表 12-3　pdsdblod 制御文の指定順序

実行する操作	pdsdblod 制御文の指定順序
データロードの場合	• environment 文，load 文，dbinf 文，idxload 文，oreload 文の順に指定してください。 • environment 文と load 文は必ず指定してください。
フォーマットライトの場合	• environment 文，load 文，dbinf 文，idxload 文の順に指定してください。 • environment 文と load 文は必ず指定してください。
インデクスの再作成の場合	• environment 文，index 文，idxload 文の順に指定してください。 • environment 文と index 文は必ず指定してください。

- pdsdblod 制御文はテキストファイルに作成します。テキストファイルを保存する際，pdsetup コマンド実行時に指定した文字コード種別で保存してください。
- 環境変数 LANG またはクライアント環境定義の PDLANG オペランドで，使用する文字コードに UTF-8 を指定した場合，pdsdblod 制御文を記述したファイルには BOM を持つファイルが指定できます（BOM を読み飛ばします）。ただし，入力データファイルには BOM を付加しないでください。
- pdsdblod 制御文中にコメントは記述できません。

12.3.1　environment 文

environment 文には，データロードまたはフォーマットライトの基本的な動作条件を指定します。

指定形式（データロードの場合）

```
environment
      schema = SDBデータベース名
      logmode =  {all | no}
     〔purge =  {yes | no} 〕
     〔exectime = pdsdblodコマンドの実行時間の上限〕
```

```
     〔recnomsg = 処理経過メッセージの出力間隔〕

     〔infmsglvl =  {lvl0 | lvl1 | lvl2} 〕

     〔recfree = レコード型名, （〔ページ内の未使用領域の比率〕
                , セグメント内の空きページの比率）〕

     〔divermsg =  {put | noput} 〕

     〔generation = 世代番号〕
```

指定形式（フォーマットライトの場合）

```
environment

      schema = SDBデータベース名

      logmode =  {all | no}

     〔purge =  {yes | no} 〕

     〔exectime = pdsdblodコマンドの実行時間の上限〕

     〔recnomsg = 処理経過メッセージの出力間隔〕

     〔infmsglvl =  {lvl0 | lvl1 | lvl2} 〕


     〔recfree = レコード型名, （〔ページ内の未使用領域の比率〕
                , セグメント内の空きページの比率）〕

     〔generation = 世代番号〕
```

●schema = SDB データベース名

～<識別子>((1～30 バイト))

データロード，インデクスの再作成，またはフォーマットライトを実行する SDB データベースの名称を指定します。SDB データベース定義の SCHEMA 句に指定した SDB データベース名を指定します。

SDB データベース名を引用符（''）で囲んだ場合，英大文字と英小文字が区別されます。引用符（''）で囲まない場合，すべて英大文字として扱われます。

●logmode = {all | no}

データロード，インデクスの再作成，またはフォーマットライトを実行する際，システムログを取得するかどうかを指定します。

all：システムログを取得します（ログ取得モード）。

no：システムログを取得しません（ログレスモード）。

システムログを取得するかどうかによる運用差異を次の表に示します。

表 12-4　システムログを取得するかどうかによる運用差異

項番	比較項目	ログ取得モードの場合	ログレスモードの場合
1	データロードまたはフォーマットライトの処理に掛かる時間	システムログを取得する分，データロードまたはフォーマットライトに掛かる時間が長くなります。	システムログを取得しない分，データロードまたはフォーマットライトに掛かる時間が短くなります。
2	pdsdblod コマンドが異常終了したときの RD エリアの状態	ロールバックするため，RD エリアの状態は変わりません（障害閉塞しません）。 障害対策後，pdsdblod コマンドを再実行します。	ロールバックしないため，RD エリアは障害閉塞します。障害閉塞を解除するには，pdmod コマンドで，RD エリアを再初期化します。 RD エリアを再初期化し，障害対策後，pdsdblod コマンドを再実行します。

《留意事項》

ログレスモードの場合，SDB データベースに格納するレコード数やインデクス数に比例した大量のシステムログは取得しません。ただし，当該 RD エリアをログレスモードで更新したことを記録するシステムログを RD エリア数に比例して取得します。システムログ量の計算式については，「3.5.1(5) HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のシステムログ量」を参照してください。

更新可能なオンライン再編成の実行時，追い付き反映キー対応表については，logmode = no を指定してもシステムログが取得されます。

●purge = {yes | no}

データロードまたはフォーマットライトを実行する際，格納済みの既存レコードを削除するかどうかを指定します。

yes：

既存レコードを削除してから，データロードまたはフォーマットライトを実行します。初期データロードとして動作します。

no：

既存レコードを削除しないで，データロードまたはフォーマットライトを実行します。追加データロードとして動作します。

SDB データベースの再編成のリロードとして実行する場合は，yes を指定してください。追加データロードの場合は，no を指定してください。

初期データロードなど，レコードが格納されていない SDB データベースに対してデータロードまたはフォーマットライトする場合は，このオペランドを指定する必要はありません。デフォルト値の no でデータロードまたはフォーマットライトできます。

なお，このオペランドに no を指定，または省略した場合，次の指定はできません。

- oreload 文
- load 文の type オペランドに single
- load 文の write オペランドに use

参考

格納済みのデータがない場合でもこのオペランドに no を指定できますが，dupkeyck オペランドに yes を指定すると，格納済みのデータがなくても重複キーのチェック処理が行われ，その分処理が遅くなります。そのため，格納済みのデータがない場合は，dupkeyck オペランドに no を指定することを推奨します。

●exectime = pdsdblod コマンドの実行時間の上限

～<符号なし整数>((0～35,791,394))（単位：分）

pdsdblod コマンドの実行時間の上限を分単位で指定します。このオペランドに指定した時間を超えても pdsdblod コマンドの処理が完了しない場合，pdsdblod コマンドを強制終了します（異常終了の扱いになります）。このとき，無応答障害の原因調査のための障害情報が$PDDIR/spool/save ディレクトリ下に出力されます。出力される障害情報を次に示します。

- コアファイル
- .deb ファイル
- コマンド名<障害情報の出力日時>プロセス ID.txt ファイル※

注※

このファイルには，次に示すコマンドの実行結果が出力されます。

・pdls -d lck

・pdls -d rpc -a

なお，このオペランドは無応答障害の検知を目的としています。したがって，pdsdblod コマンドの予想実行時間に対して，十分に余裕をもたせた時間を指定してください。例えば，pdsdblod コマンドの予想実行時間が 7 分程度の場合，このオペランドには 20 分ぐらいの時間を指定してください。

データロードを夜間バッチとして実行するときなどにこのオペランドの指定を推奨します。このときにこのオペランドを指定しておくと，通信障害（一時的な障害を含む）やディスク障害などが発生して，pdsdblod コマンドが無応答状態になった場合，pdsdblod コマンドを自動的に強制終了させることができます。これによって，pdsdblod コマンドの無応答状態が，UAP やほかのコマンドの実行に影響が及ぶのを防ぐことができます。

《留意事項》

- このオペランドに 0 を指定した場合は，pdsdblod コマンドの実行時間の上限は設定されません。
- このオペランドを省略した場合，システム定義の pd_utl_exec_time オペランド，または pd_cmd_exec_time オペランドに指定した時間が，pdsdblod コマンドの実行時間の上限になります。pd_utl_exec_time オペランド，および pd_cmd_exec_time オペランドについては，マニュアル「HiRDB システム定義」を参照してください。

  オペランドの優先順位は高い順に次のとおりです（優先順位は 1 の指定がいちばん上位です）。

優先順位	指定個所	オペランド名
1	pdsdblod 制御文の environment 文	exectime

優先順位	指定個所	オペランド名
2	システム定義	pd_utl_exec_time
3	システム定義	pd_cmd_exec_time

●recnomsg = 処理経過メッセージの出力間隔

～<符号なし整数>((1～1,000))《10》（単位：1 万件）

データロードまたはフォーマットライトの処理経過メッセージの出力間隔を，1 万件（レコード）単位で指定します。例えば，このオペランドに 5 を指定すると，データロードが 5 万件（レコード）終わるごとに経過メッセージ（KFPB63011-I）が出力されます。

●infmsglvl = {lvl0 | lvl1 | lvl2}

データロード，インデクスの再作成，またはフォーマットライト時に出力される次に示すメッセージを出力するかどうかを指定します。

- KFPB63035-I（インデクス情報ファイルに関するメッセージ）
- KFPB63036-I（インデクスの一括作成の開始を知らせるメッセージ）
- KFPB63037-I（インデクスの一括作成の終了を知らせるメッセージ）
- KFPB63038-I（インデクスの一括作成の開始を知らせるメッセージ）
- KFPB63039-I（インデクスの一括作成の終了を知らせるメッセージ）
- KFPB63040-I（インデクス情報ファイルの削除を知らせるメッセージ）
- KFPB63044-I（インデクスの抽出開始を知らせるメッセージ）
- KFPB63045-I（インデクスの抽出終了を知らせるメッセージ）
- KFPB63046-I（インデクスの抽出開始を知らせるメッセージ）
- KFPB63047-I（インデクスの抽出終了を知らせるメッセージ）

インデクス数や，インデクス格納用 RD エリア数に比例して，上記のメッセージの出力回数が多くなります。

lvl0：

上記のメッセージのうち，次のメッセージを除いて，標準出力，syslogfile，およびメッセージログファイルに出力します。

- KFPB63036-I（インデクスの一括作成の開始を知らせるメッセージ）
- KFPB63037-I（インデクスの一括作成の終了を知らせるメッセージ）
- KFPB63044-I（インデクスの抽出開始を知らせるメッセージ）
- KFPB63045-I（インデクスの抽出終了を知らせるメッセージ）

lvl1：

上記のメッセージを syslogfile およびメッセージログファイルに出力します。

ただし，次のメッセージは，標準出力にも出力します（次のメッセージ以外は標準出力には出力しません）。

- KFPB63036-I（インデクスの一括作成の開始を知らせるメッセージ）
- KFPB63037-I（インデクスの一括作成の終了を知らせるメッセージ）
- KFPB63038-I（インデクスの一括作成の開始を知らせるメッセージ）
- KFPB63039-I（インデクスの一括作成の終了を知らせるメッセージ）
- KFPB63044-I（インデクスの抽出開始を知らせるメッセージ）
- KFPB63045-I（インデクスの抽出終了を知らせるメッセージ）

lvl2：

上記のメッセージは，標準出力，syslogfile，およびメッセージログファイルに出力しません。代わりにワークファイルに上記のメッセージを出力します。

ただし，次のメッセージについては，標準出力，syslogfile，メッセージログファイル，およびワークファイルに出力します。

- KFPB63036-I（インデクスの一括作成の開始を知らせるメッセージ）
- KFPB63037-I（インデクスの一括作成の終了を知らせるメッセージ）
- KFPB63044-I（インデクスの抽出開始を知らせるメッセージ）
- KFPB63045-I（インデクスの抽出終了を知らせるメッセージ）

通常は，このオペランドを省略するか，または lvl1 を指定してください。lvl2 を指定してワークファイルにメッセージを出力するのは，トラブルシュート目的（pdsdblod コマンドの動作を確認するため）のときだけにしてください。

《ワークファイルの出力先》

ワークファイルの出力先ディレクトリは，次に示す優先順位（1 の指定がいちばん上位）に従って決定されます。

1. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
2. 環境変数 TMPDIR に指定したディレクトリ
3. /tmp ディレクトリ

1 のディレクトリにワークファイルを出力することを推奨します。1 のディレクトリにワークファイルを出力すると，pdcspool コマンドでワークファイルを一括削除できるため，ファイル削除の手間が少なく済みます。

処理対象の RD エリアがあるサーバマシンのディレクトリにワークファイルが出力されます。ファイル名は，PDSDBLOAD-aa....aa-bb....bb になります。aa....aa はサーバ名，bb....bb はファイル作成時間およびプロセス ID になります。

参考

ファイルオープンエラーや入出力エラーなどによってワークファイルにメッセージが出力できない場合でも pdsdblod コマンドの処理は続行されます。

《ワークファイルの出力形式》

ワークファイルの出力形式を次に示します。

```
AAAAA hh:mm:ss BBBB CC....CC KFPB63nnn-I XX....XX
```

AAAAA：プロセス ID

hh:mm:ss：メッセージの出力時刻

BBBB：ユニット識別子

CC....CC：メッセージの出力要求元のサーバ名

KFPB63nnn-I：メッセージ ID

XX....XX ：メッセージテキスト

ユニット識別子とメッセージの出力要求元のサーバ名は，出力されないことがあります。

●recfree = レコード型名，（〔ページ内の未使用領域の比率〕，セグメント内の空きページの比率）

SDB データベース格納定義の PCTFREE オペランドの指定値を変更して，データロードまたはフォーマットライトを実行する場合にこのオペランドを指定します。このオペランドを省略した場合は，SDB データベース格納定義の PCTFREE オペランドの指定値に従って，ページ内の未使用領域の比率やセグメント内の空きページの比率が設定されます。

ページ内の未使用領域の比率およびセグメント内の空きページの比率については，「3.2.5　レコード格納時の空き領域の作成（サブページ分割をしない場合）」または「3.2.6　レコード格納時の空き領域の作成（サブページ分割をする場合）」を参照してください。

レコード型名：

～＜識別子＞((1～30 バイト))

4V FMB または SD FMB の SDB データベースの場合は，ページ内の未使用領域の比率の変更を行うルートレコードのレコード型名を指定します。ルートレコードのページ内の未使用領域の比率は，子レコードにも適用されます。

4V AFM の SDB データベースの場合は，ページ内の未使用領域の比率の変更を行う仮想ルートレコードのレコード型名を指定します。仮想ルートレコードのページ内の未使用領域の比率は，子レコードにも適用されます。

ページ内の未使用領域の比率：

～＜符号なし整数＞((0～99))（単位：%）

ページ内の未使用領域の比率を%単位で指定します。

なお，サブページ分割をしている場合，ページ内の未使用領域の比率は，サブページ内の未使用領域の比率になります。例えば，30 を指定した場合，サブページ内の未使用領域の比率が 30%になります。

セグメント内の空きページの比率：

～＜符号なし整数＞((0～50))（単位：%）

セグメント内の空きページの比率を%単位で指定します。

ページ内の未使用領域の比率およびセグメント内の空きページの比率の指定値を小さくすると，RD エリアに格納できるレコード件数が多くなります。逆に，指定値を大きくすると，RD エリアに格納できるレコード件数が少なくなります。したがって，データロード時の RD エリア容量不足を一時的に回避

したい場合は，SDB データベース格納定義の PCTFREE オペランドで指定した値より小さい値を指定してください。

●divermsg = {put | noput} 【データロード限定オペランド】

入力データファイル中に処理対象 RD エリア以外に格納されるレコードがある場合，そのレコードを論理エラーとするかどうかを指定します。RD エリア単位のデータロードを実行する場合にこのオペランドの指定を検討してください。

put：

入力データファイル中に格納対象外のレコードがあった場合，そのレコードを論理エラーとして扱います。論理エラーの情報は，実行結果ファイルおよび論理エラー情報ファイルに出力されます。

noput：

入力データファイル中に格納対象外のレコードがあった場合でも，そのレコードを論理エラーとして扱いません。

入力データファイル中に格納対象外のレコードが大量にあるときに put を指定すると，論理エラーの情報が実行結果ファイルに大量に出力されます。このような場合は noput を指定してください。

参考

put または noput のどちらを指定しても，格納対象外のレコードはデータベースに格納されません。

●generation = 世代番号

～<符号なし整数>((0～10))

処理対象の RD エリアの世代番号を指定します。このオペランドは，インナレプリカ機能使用時に指定します。

なお，SD FMB の SDB データベースを使用している場合，このオペランドは省略してください。

0：オリジナル RD エリアを処理対象とする場合に指定します。

1～10：指定した世代のレプリカ RD エリアを処理対象とする場合に指定します。

- 更新可能なオンライン再編成時のデータロードの場合

  このオペランドには，0（オリジナル RD エリア）を指定してください。0 以外の値（レプリカ RD エリア）を指定すると，エラーになります。

  また，オリジナル RD エリアがオンライン再編成閉塞でない場合も，エラーになります。

- 上記以外の場合

  このオペランドを省略した場合，カレント RD エリアが処理対象になります。カレント RD エリア以外の RD エリアを処理対象とする場合に，このオペランドを指定してください。

《運用上の注意事項》

- レプリカ RD エリアに対して RD エリア単位のデータロードまたはフォーマットライトを実行する場合，load 文の area オペランドにはオリジナル RD エリア名を指定してください。このオペランドには処理対象の RD エリアの世代番号を指定してください。

- pdsdbrog コマンドでレコードをアンロードし，pdsdblod コマンドでデータロードする場合，アンロードとデータロードの実行時に同じ世代番号を指定してください。異なる世代番号を指定してもチェックアウトされないため，指定された世代番号の RD エリア（アンロードを実行した RD エリアとは異なる RD エリア）にデータロードされます。

> **参考**
>
> SDB データベースが横分割されているため，処理対象の RD エリアが複数ある場合，HiRDB/SD は次に示すチェックを行います。チェックの結果，不一致がある場合，KFPB63306-E メッセージを出力してリターンコード 8 で pdsdblod コマンドが終了します。
>
> - RD エリアの世代番号のチェック：
>
> このオペランドを指定した場合は，指定した世代番号の RD エリアがあるかどうかをチェックします。レコードが格納されている RD エリアとインデクスが格納されている RD エリアの世代番号をチェックします。
>
> このオペランドを省略した場合は，レコードが格納されている RD エリアとインデクスが格納されている RD エリアのカレント RD エリアの世代番号が同じかどうかをチェックします。
> - RD エリアのレプリカステータスのチェック：
>
> 処理対象の RD エリアにカレント RD エリアとカレント RD エリア以外が混在していないか，各 RD エリアのレプリカステータスをチェックします。

12.3.2 load 文

load 文には，入力データをデータベースに格納するときの情報や処理条件を指定します。

なお，load 文は，index 文と同時に指定できません。

指定形式（データロードの場合）

```
load
     type =  {tree | single}
     record = レコード型名
    〔server = バックエンドサーバ名〕
    〔area = 処理対象のRDエリア名〕
     data = 入力データファイル名
    〔storinf = 実行結果ファイル名〕
    〔errdata = {（論理エラー情報ファイル名〔，出力件数の上限〕）
                | 論理エラー情報ファイル名}〕
```

```
     prefix = プリフィクス部の長さ
    〔idxmode =  {create | sync〕
    〔seqkeyck =  {yes | no} 〕
    〔dupkeyck =  {yes | no} 〕
```

指定形式（フォーマットライトの場合）

```
load
     write = use
     record = レコード型名
    〔server = バックエンドサーバ名〕
    〔area = 処理対象のRDエリア名〕
    〔storinf = 実行結果ファイル名〕
    〔idxmode =  {create | sync〕
```

●type = {tree | single} 【データロード限定オペランド】

入力データファイルの種類を指定します。

tree：入力データファイルが木構造ファイルの場合に指定します。

single：入力データファイルが単純階層ファイルの場合に指定します。

4V FMB または SD FMB の SDB データベースの場合は tree を，4V AFM の SDB データベースの場合は single を指定してください。

●write = use 【フォーマットライト限定オペランド】

フォーマットライトを実行する場合に指定します。

environment 文の purge オペランドに no を指定するか，または省略した場合，このオペランドは指定できません。

●record = レコード型名

～<識別子>((1～30 バイト))

データロードまたはフォーマットライトを実行するレコードのレコード型名を指定します。SDB データベース定義の RECORD 句に指定したレコード型名を指定します。

4V FMB または SD FMB の SDB データベースの場合は，ルートレコードのレコード型名を指定してください。4V AFM の SDB データベースの場合は，仮想ルートレコードのレコード型名を指定してください。

●server = バックエンドサーバ名

～<識別子>((1～8 バイト))

データロードまたはフォーマットライトの処理対象 RD エリアを管理しているバックエンドサーバの名称を指定します。

英大文字と英小文字の指定が区別されます。

このオペランドは，次に示す条件をすべて満たす場合に指定してください。

- SDB データベースを横分割している場合
- BES 内の全対象 RD エリアに対して，一括でデータロードまたはフォーマットライトを実行する場合

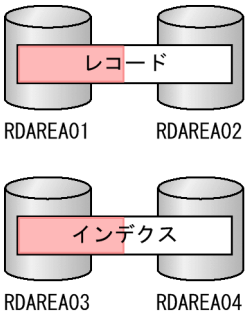
上記の条件を満たさない場合は，このオペランドを省略してもかまいません。

●area = 処理対象の RD エリア名

～<識別子>((1～30 バイト))

SDB データベースが横分割されている場合，RD エリア単位にデータロードまたはフォーマットライトを実行するときにこのオペランドを指定します。

例えば，次に示すように SDB データベースが横分割されている場合，このオペランドに RDAREA01 を指定すると，RDAREA01 にレコードが格納され，RDAREA03 にインデクスが格納されます。



《インナレプリカ機能使用時の注意事項》

レプリカ RD エリアに対してデータロードまたはフォーマットライトを実行する場合，このオペランドにはオリジナル RD エリア名を指定してください。そして，environment 文の generation オペランドに，処理対象 RD エリアの世代番号を指定してください。

●data = 入力データファイル名 【データロード限定オペランド】

～<パス名>((1～1,023 バイト))

入力データファイル名を絶対パスで指定します。

入力データファイルは，処理対象 RD エリアがあるサーバマシンに格納してください。そのサーバマシンのパス名を指定します。

《注意事項》

このオペランドに指定するファイルを格納するディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与してください。権限がない場合，pdsdblod コマンドがエラーになります。

> **参考**
>
> 入力データファイルの入出力時に使用されるバッファ長は，システム定義の pd_utl_file_buff_size オペランドの値が適用されます。

●storinf = 実行結果ファイル名

～<パス名>((1～1,023 バイト))

pdsdblod コマンドの処理が完了すると，実行結果が実行結果ファイルに出力されます。このオペランドには，実行結果ファイルの名称を絶対パスで指定します。実行結果ファイルは，処理対象 RD エリアがあるサーバマシンに作成されます。

このオペランドを省略した場合，次に示すどれかのディレクトリ下に実行結果ファイルが作成されます（優先順位は 1 の指定がいちばん上位です）。

1. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
2. 環境変数 TMPDIR に指定したディレクトリ
3. /tmp ディレクトリ

実行結果ファイルの名称は SDBLOD-xxxxxxxxx になります。xxxxxxxxx は，ファイル作成時間，プロセス ID などが含まれた文字列になります。作成されたファイルの名称は KFPB63030-I メッセージで確認できます。

《注意事項》

- このオペランドに指定するファイルのディレクトリは事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与してください。ディレクトリがない場合，または権限がない場合は，pdsdblod コマンドがエラーになります。
- このオペランドに指定したファイルがすでに存在する場合，そのファイルは上書きされます。

実行結果ファイルに出力される情報については，「12.7　実行結果ファイルの出力形式」を参照してください。

●errdata = {(論理エラー情報ファイル名〔, 出力件数の上限〕) | 論理エラー情報ファイル名} 【データロード限定オペランド】

論理エラー情報の出力先と，出力件数の上限を指定します。このオペランドを省略した場合，論理エラー情報は出力されません。

論理エラー情報ファイル名：

～<パス名>((1～1,023 バイト))

論理エラー情報ファイル名を絶対パスで指定します。入力データの論理エラーが発生した場合，ここで指定したファイルに論理エラー情報が出力されます。論理エラー情報ファイルは，処理対象の RD エリアがあるサーバマシンに作成されます。

出力件数の上限：

～<符号なし整数>((1～4,294,967,295))《100》

論理エラー情報の出力件数の上限を指定します。論理エラー情報の件数が，このオペランドの指定値を超えても処理は継続されますが，論理エラー情報は指定した件数までしか出力されません。

《注意事項》

- このオペランドに指定するファイルのディレクトリは事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与

してください。ディレクトリがない場合，または権限がない場合は，pdsdblod コマンドがエラーになります。

- このオペランドに指定したファイルがすでに存在する場合，そのファイルは上書きされます。

●prefix = プリフィクス部の長さ 【データロード限定オペランド】

～<符号なし整数>((0～32,767))（単位：バイト）

入力データのレコードのプリフィクス部の長さをバイト単位で指定します。プリフィクス部がない場合は 0 を指定してください。

入力データのプリフィクス部については，「12.6.1(3) プリフィクス部【4V FMB，4V AFM】」で説明されている L1 の値を参照して，このオペランドに指定します。

●idxmode = {create | sync}

インデクスの作成モードを指定します。

create：インデクス一括作成モードでインデクスを作成します。

sync：インデクス更新モードでインデクスを作成します。

●seqkeyck = {yes | no} 【データロード限定オペランド】

シーケンシャルインデクスが定義されているデータベースに対して，キー値の順番をチェックするかどうかを指定します。4V FMB または SD FMB の SDB データベースにデータロードする場合にこのオペランドの指定を検討してください。

yes：

キー値の順番をチェックします。この場合，キー値が順番に並んでいるレコードだけがデータベースに格納されます。キー値が順番に並んでいないレコードは論理エラーとなり，データベースに格納されません。

no：

キー値の順番をチェックしません。この場合，キー値の順番に関係なく，すべてのレコードがデータベースに格納されます。

yes を指定するとキー値の順番が保証されるため，次に示すメリットがあります。

- 各インデクスページにインデクスキーが均等に配分されるため，インデクスを使用した検索処理時間のばらつきが少なくなります。
- キー値の順番にレコードがデータページ上に格納されるため，キー順に検索する場合，グローバルバッファのヒット率が向上します。

ただし，yes を指定した場合，順番のチェック処理の分だけ，データロードまたはフォーマットライトの処理時間が長くなります。したがって，キー値の順番が保証されている入力データの場合は no を指定してください。

《注意事項》

- シーケンシャルインデクスのデータ型が PACKED DECIMAL FIXED の場合，DECIMAL 型の符号正規化機能を使用しない環境下では，次のチェック条件となります。

  ・符号が 0xC と 0xF で異なる場合でも同じデータとして扱います。

・＋0（0x0c）と－0（0x0d）は異なるデータと見なし，＋0は－0より大きいデータとして扱います。

- 4V AFMのSDBデータベースの場合，親レコード型に対する入力データの作成が不要のため，このオペランドの指定に関係なく順番のチェックは行われません。
- SDBデータベース定義のORDER句にSORTED DUPLICATESを指定した子レコード型に対する順番のチェックは，このオペランドの指定に関係なく実施されます。入力データのユーザキーは，KEY句で指定した順序（ASCENDINGは昇順，DESCENDINGは降順）で作成してください。並び順が不正な場合，論理エラーになります。また，ORDER句にSORTED DUPLICATES PROHIBITEDを指定した場合，ユーザキーの重複も論理エラーになります。

●dupkeyck＝ {yes | no} 【データロード限定オペランド】

インデクス一括作成モードを指定した場合，入力データファイルのルートレコードに対し，SDBデータベースに格納されているレコードのキー重複チェックをするかどうかを指定します。4V FMBまたはSD FMBのSDBデータベースにデータロードをする場合は，このオペランドの指定を検討してください。

yes：

キー値の重複をチェックします。キー値が重複していないレコードだけがSDBデータベースに格納されます。キー値が重複しているレコードは論理エラーとなり，SDBデータベースに格納されません。

no：

キー値の重複をチェックしません。キー値の重複に関係なく，すべてのレコードがSDBデータベースに格納されます。

yesを指定すると，同じキー値のシーケンシャルインデクスの格納を防げます。ただし，yesを指定した場合，重複チェック処理の分だけ，追加データロードの処理時間が長くなります。したがって，キー値の重複がないことが保証されている入力データの場合は，このオペランドにnoを指定してください。

なお，インデクス一括作成モードのときにこのオペランドにnoを指定した場合，同じインデクスキーであってもチェックされません。この場合，インデクスの一括作成時にキー重複エラーが発生し，インデクス不整合となるため注意してください。

《注意事項》

- 4V AFMのSDBデータベースの場合，このオペランドの指定は無視されます。
- 4V FMBまたはSD FMBのSDBデータベースのインデクス更新モードの場合，yesが仮定されます。
- environment文のpurgeオペランドにyesを指定した場合，noが仮定されます。

12.3.3 dbinf文【4V FMB，4V AFM】

dbinf文には，入力データの各レコードのプリフィクス部に格納されているUSERポインタフラグ，ページ切り替えフラグ，および事前割り当てページ数の位置を指定します。dbinf文は次に示す場合に指定します。

- 4V FMB の SDB データベースの場合
  - USER ポインタを使って親子集合を対応させる場合
  - ページ切り替え，またはサブページ切り替えを行いながらデータロードする場合
  - 事前ページ割り当てを行いながらデータロードする場合
- 4V AFM の SDB データベースの場合
  - ページ切り替え，またはサブページ切り替えを行いながらデータロードする場合
  - ページ切り替え，またはサブページ切り替えを行いながらフォーマットライトする場合

SD FMB の SDB データベースの場合，dbinf 文は指定できません。

《規則》

dbinf 文は，index 文と同時に指定できません。

指定形式（4V FMB の SDB データベースの場合）

```
dbinf
     〔userpflg = （USERポインタフラグの開始位置）〕
     〔pagecflg = （ページ切り替えフラグの開始位置）〕
     〔pagenum  = （事前割り当てページ数の開始位置）〕
```

指定形式（4V AFM の SDB データベースのデータロードの場合）

```
dbinf
      〔pagecflg = （ページ切り替えフラグの開始位置）〕
```

指定形式（フォーマットライトの場合）

```
dbinf
      〔pagecflg = {change | occupation} 〕
```

●userpflg = （USER ポインタフラグの開始位置）【4V FMB 限定オペランド】

～<符号なし整数>((0～32,766))

入力データの各レコードのプリフィクス部に格納されている USER ポインタフラグの開始位置を指定します。プリフィクス部の先頭を 0 とし，USER ポインタフラグの開始位置を指定してください。

プリフィクス部と USER ポインタフラグの開始位置の関係については，「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」を参照してください。

●pagecflg = （ページ切り替えフラグの開始位置）【データロード限定オペランド】

～<符号なし整数>((0～32,766))

入力データの各レコードのプリフィクス部に格納されているページ切り替えフラグの開始位置を指定します。プリフィクス部の先頭を 0 とし，ページ切り替えフラグの開始位置を指定してください。

ページ切り替えによる格納配置制御については，「2.6.2(3)　レコード格納時のページ切り替え【4V FMB，4V AFM】」を参照してください。サブページ分割をしている場合は，「2.7.2(3)　レコード格納時のサブページ切り替え【4V FMB，4V AFM】」を参照してください。

プリフィクス部とページ切り替えフラグの開始位置の関係については，「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」を参照してください。

●pagenum = （事前割り当てページ数の開始位置）【4V FMB のデータロード限定オペランド】

～<符号なし整数>((0～32,763))

入力データの各ルートレコードのプリフィクス部に格納されている事前割り当てページ数の開始位置を指定します。プリフィクス部の先頭を 0 とし，事前割り当てページ数の開始位置を指定してください。指定された開始位置から 4 バイトを事前割り当てページ数の指定値として扱います。

プリフィクス部と事前割り当てページ数の開始位置の関係については，「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」を参照してください。

●pagecflg = {change | occupation}　【4V DAM 限定オペランド】

子レコードの格納方法を指定します。このオペランドを指定する際は，「2.6.2(3)　レコード格納時のページ切り替え【4V FMB，4V AFM】」を参照してください。サブページ分割をしている場合は，「2.7.2(3)　レコード格納時のサブページ切り替え【4V FMB，4V AFM】」を参照してください。

change：

フォーマットライトの際，1 つの子レコードを 1 ページに格納します。レコードがあとから追加された際，そのページに追加レコードを格納します。

サブページ分割をしている場合は，フォーマットライトの際，1 つの子レコードを 1 サブページに格納します。レコードがあとから追加された際，そのサブページに追加レコードを格納します。

occupation：

フォーマットライトの際，1 つの子レコードを 1 ページに格納します。そのページ内には，その子レコードだけを格納し，ほかのレコードは格納しません。

サブページ分割をしている場合は，フォーマットライトの際，1 つの子レコードを 1 サブページに格納します。そのサブページ内には，その子レコードだけを格納し，ほかのレコードは格納しません。

12.3.4　index 文

インデクスの再作成を実行する場合に index 文を指定します。index 文には，再作成するインデクスの情報と処理条件を指定します。

《規則》

- index 文は制御情報ファイル中に 1 つだけ指定できます。
- index 文と load 文は，同時に指定できません。

指定形式

```
index
    idxremode = recrt
    idxarea = インデクス格納用RDエリア名
    idxname = インデクス名
```

●idxremode = recrt

インデクスを再作成する際の処理モードを指定します。

recrt：

インデクス再作成モードでインデクスを再作成します。

●idxarea = インデクス格納用 RD エリア名

～<識別子>((1～30 バイト))

再作成対象のインデクスが格納されているインデクス格納用 RD エリア名を指定します。

●idxname = インデクス名

～<識別子>((1～30 バイト))

再作成対象のインデクス名を指定します。

12.3.5 idxload 文

idxload 文には，インデクスを作成する際に作成される一時ファイルの出力先を指定します。idxload 文は次に示す場合に指定します。

- インデクス情報ファイルの出力先を変更する場合
- ソート用ワークファイルの出力先，またはソート用バッファサイズを変更する場合

《規則》

idxload 文は，次に示す場合に指定が有効になります。

- load 文の idxmode オペランドに create を指定した場合
- index 文の idxremode オペランドに recrt を指定した場合

参考

- idxload 文を省略した場合，次に示すどれかのディレクトリ下にインデクス情報ファイル，ソート用ワークファイルが出力されます（優先順位は 1 の指定がいちばん上位です）。
  1. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
  2. 環境変数 TMPDIR に指定したディレクトリ

3. /tmp ディレクトリ

- インデクス更新モードでインデクスを作成する場合（load 文で idxmode = sync を指定した場合）に idxload 文を指定すると，pdsdblod コマンドがエラーになります。

指定形式

```
idxload

    〔workdir = インデクス情報ファイルの出力先ディレクトリ〕

    〔sortdir = ソート用ワークファイルの出力先ディレクトリ
          〔, ソート用バッファサイズ〕〕
```

●workdir = インデクス情報ファイルの出力先ディレクトリ

～<パス名>((1～1,023 バイト))

インデクス情報ファイルの出力先ディレクトリを絶対パスで指定します。

《注意事項》

- このオペランドに指定するディレクトリは事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与してください。ディレクトリがない場合，または権限がない場合は，pdsdblod コマンドがエラーになります。
- NFS のディレクトリを指定しないでください。指定した場合，正常に動作しないことがあります。
- パス名の長さの上限の 1,023 バイトには，インデクス情報ファイルの名称長も含んでいます。インデクス情報ファイルの名称規則については，「12.9.1(2)　注意事項」を参照してください。

●sortdir = ソート用ワークファイルの出力先ディレクトリ〔, ソート用バッファサイズ〕

一時的に作成されるソート用ワークファイルの出力先ディレクトリと，ソート処理で使用するソート用バッファサイズを指定します。

ソート用ワークファイルの出力先ディレクトリ

～<パス名>((1～1,023 バイト))

ソート用ワークファイルの出力先ディレクトリを絶対パスで指定します。

《注意事項》

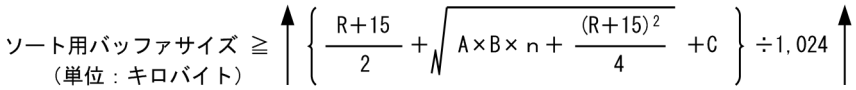
- このオペランドに指定するディレクトリは事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与してください。ディレクトリがない場合，または権限がない場合は，pdsdblod コマンドがエラーになります。
- NFS のディレクトリを指定しないでください。指定した場合，正常に動作しないことがあります。

ソート用バッファサイズ

～<符号なし整数>((128～2,000,000))《1,024》（単位：キロバイト）

ソート処理で使用するソート用バッファサイズをキロバイト単位で指定します。指定値の目安を次に示します。

- メモリに余裕がない場合や，ほとんどのレコードがインデクスキー順に並んでいる場合は，デフォルト値で pdsdblod コマンドを実行してください。
- 基本的にバッファサイズを大きくするほど性能向上が見込まれます。
- このオペランドの指定値の見積もり式を次に示します。



A：R + key + 76

B：R + key + 124

C：key + 2292

n：レコード件数

R：key + 10

key：インデクスのキー長の最大値

インデクスのキー長については，「3.5.5(8)　インデクスを格納するページ数の見積もり」を参照してください。

12.3.6　oreload 文【4V FMB】

oreload 文には，追い付き反映キー対応表の中間ファイルの格納ディレクトリ名を指定します。oreload 文は，更新可能なオンライン再編成を実行する場合に指定してください。

4V AFM または SD FMB の SDB データベースの場合，oreload 文は指定できません。

《規則》

- environment 文の purge オペランドに no を指定するか，または省略した場合，oreload 文は指定できません。
- oreload 文は，index 文と同時に指定できません。

指定形式

```
oreload
    〔midfile＝追い付き反映キー対応表の中間ファイル名〕
```

●midfile ＝追い付き反映キー対応表の中間ファイル名

～<パス名>((1～1,023 バイト))

追い付き反映キー対応表の中間ファイルを格納するファイル名を，絶対パス名で指定します。このオペランドを省略した場合，次の名称の中間ファイルが作成されます。

- SDBLOD-ORE-ファイル作成時間-プロセスID

このオペランドを省略した場合，次に示すどれかのディレクトリ下に中間ファイルが作成されます（優先順位は 1 の指定がいちばん上位です）。

1. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
2. 環境変数 TMPDIR に指定したディレクトリ
3. /tmp ディレクトリ

中間ファイルの容量見積もりについては，「3.6.1(1)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量」を参照してください。

参考

中間ファイルの入出力時に使用されるバッファ長は，システム定義の pd_utl_file_buff_size オペランドの値が適用されます。

《注意事項》

- このオペランドに指定するディレクトリは事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与してください。ディレクトリがない場合，または権限がない場合は，pdsdblod コマンドがエラーになります。
- データロードの対象 RD エリアがオンライン再編成閉塞でない場合に oreload 文を指定すると，pdsdblod コマンドがエラーになります。
- 追い付き反映キー対応表の登録処理が正常終了した場合，中間ファイルは自動的に削除されます。登録処理がエラーとなった場合，中間ファイルは削除されないため，必要に応じて削除してください。作成された中間ファイルの名称は，KFPB63041-I メッセージで確認できます。

12.3.7　pdsdblod 制御文の各オペランドの指定可否

pdsdblod 制御文の各オペランドの指定可否（4V FMB および 4V AFM の SDB データベースの場合）と，pdsdblod 制御文の各オペランドの指定可否（SD FMB の SDB データベースの場合）を，以下のそれぞれの表に示します。

表 12–5　pdsdblod 制御文の各オペランドの指定可否（4V FMB および 4V AFM の SDB データベースの場合）

項番	オペランド		操作					備考
			データロード		フォーマットライト	インデクスの再作成		
			4V FMB	4V AFM	4V DAM	4V FMB	4V AFM	
1	environment 文	schema	○	○	○	○	○	environment 文は必ず指定してください。
2		logmode	○	○	○	○	○	

項番	オペランド		操作					備考
			データロード		フォーマットライト	インデクスの再作成		
			4V FMB	4V AFM	4V DAM	4V FMB	4V AFM	
3		purge	△	△	△	×	×	
4		exectime	△	△	△	△	△	
5		recnomsg	△	△	△	×	×	
6		infmsglvl	△	△	△	△	△	
7		recfree	△	△	△	×	×	
8		divermsg	△	△	×	×	×	
9		generation	△	△	△	△	△	
10	load 文	type	○	○	×	×	×	データロードを実行する場合，load 文は必ず指定してください。
11		write	△	△	○	×	×	
12		record	○	○	○	×	×	
13		server	△	△	△	×	×	
14		area	△	△	△	×	×	
15		data	○	○	×	×	×	
16		storinf	△	△	△	×	×	
17		errdata	△	△	×	×	×	
18		prefix	○	○	×	×	×	
19		idxmode	△	△	△	×	×	
20		seqkeyck	△	△	×	×	×	
21		dupkeyck	△	△※	△※	×	×	
22	dbinf 文	userpflg	△	×	×	×	×	親子集合を対応させる場合，またはページ切り替えをする場合に dbinf 文を指定します。
23		pagecflg	△	△	△	×	×	
24		pagenum	△	×	×	×	×	
25	index 文	idxremode	×	×	×	○	○	インデクス再作成機能を実行する場合，index 文は必ず指定してください。
26		idxarea	×	×	×	○	○	
27		idxname	×	×	×	○	○	
28	idxload 文	workdir	△	△	△	△	△	インデクス一括作成モードでインデクスを作成する場合に idxload 文の指定を検討してください。
29		sortdir	△	△	△	△	△	

項番	オペランド		操作					備考
			データロード		フォーマットライト	インデクスの再作成		
			4V FMB	4V AFM	4V DAM	4V FMB	4V AFM	
30	oreload 文	midfile	△	×	×	×	×	更新可能なオンライン再編成を実行する場合に指定します。

（凡例）

○：必ず指定するオペランドです。

△：指定を検討するオペランドです。

×：指定不要なオペランドです。

注※

オペランドを指定しても，指定は無視されます。

表 12-6　pdsdblod 制御文の各オペランドの指定可否（SD FMB の SDB データベースの場合）

項番	オペランド		操作		備考
			データロード	インデクスの再作成	
			SD FMB	SD FMB	
1	environment 文	schema	○	○	environment 文は必ず指定してください。
2		logmode	○	○	
3		purge	△	×	
4		exectime	△	△	
5		recnomsg	△	×	
6		infmsglvl	△	△	
7		recfree	△	×	
8		divermsg	△	×	
9		generation	△	△	
10	load 文	type	○	×	データロードを実行する場合，load 文は必ず指定してください。
11		write	△	×	
12		record	○	×	
13		server	△	×	
14		area	△	×	
15		data	○	×	
16		storinf	△	×	

項番	オペランド		操作		備考
			データロード	インデクスの再作成	
			SD FMB	SD FMB	
17		errdata	△	×	
18		prefix	○	×	
19		idxmode	△	×	
20		seqkeyck	△	×	
21		dupkeyck	△	×	
22	dbinf 文	userpflg	×	×	SD FMB の SDB データベースの場合，dbinf 文は指定できません。
23		pagecflg	×	×	
24		pagenum	×	×	
25	index 文	idxremode	×	○	インデクス再作成機能を実行する場合，index 文は必ず指定してください。
26		idxarea	×	○	
27		idxname	×	○	
28	idxload 文	workdir	△	△	インデクス一括作成モードでインデクスを作成する場合に idxload 文の指定を検討してください。
29		sortdir	△	△	
30	oreload 文	midfile	×	×	SD FMB の SDB データベースの場合，oreload 文は指定できません。

（凡例）

○：必ず指定するオペランドです。

△：指定を検討するオペランドです。

×：指定不要なオペランドです。

12.3.8 pdsdblod 制御文の記述規則

pdsdblod 制御文の記述規則を次の表に示します。

表 12-7 pdsdblod 制御文の記述規則

項目	記述規則
pdsdblod 制御文の構成	pdsdblod 制御文は，pdsdblod 制御文の種類や定義句を表すキーワードとユーザ指定値から成ります。 1 つの pdsdblod 制御文内のキーワードとキーワードの間は，区切り文字で区切ります。 ただし，これらの位置に，分離記号または囲み文字がある場合，区切り文字を省略できます。
1 行の長さ	1 行の長さに制限はありません。また，改行位置は任意です。

項目	記述規則
	ただし，制御文ファイルのファイルサイズが大き過ぎるとpdsdblodコマンドがエラーとなることがあります。このため，大量の空白行などは含めないようにしてください。
改行コード	改行コードには次の文字を指定できます。 • NL（X'0a'） • CR + NL（X'0d'+ X'0a'）
分離記号	分離記号は，キーワードとユーザ指定値，またはユーザ指定値とユーザ指定値を区切る文字です。分離記号には次の文字を指定できます。 • =（X'3d'） • ,（X'2c'） • (（X'28'） •)（X'29'）
区切り文字	区切り文字には次の文字を指定できます。 • 空白（X'20'） • タブ（X'09'） • NL（X'0a'） • CR（X'0d'）
区切り文字を挿入してもよい位置	区切り文字を挿入してもよい位置を次に示します。 • 区切り文字および分離記号の前後で，かつ区切り文字の挿入が禁止されていない位置
区切り文字を挿入してはいけない位置	区切り文字を挿入してはいけない位置を次に示します。 • キーワードの中 • 引用符（"）で囲まれていないユーザ指定値の中 • 数値の中 引用符（"）で囲まれたユーザ指定値の中の区切り文字および分離記号はユーザ指定値として扱われます。
囲み文字	囲み文字には次の文字を指定します。 • "（X'22'）
コメントの記述	pdsdblod制御文中にコメントは記述できません。
パス名	• パス名に空白は指定できません。 • パス名の指定はOSの規則に従う必要があります。OSの規則に従っていない場合は，オペランドの指定規則内であってもシステムコールエラーとなることがあります。
識別子	• 空白を含む場合は，引用符（"）で囲んでください。 • 引用符（"）で囲んだ場合，半角英大文字と半角英小文字が区別されます。引用符（"）で囲まない場合，半角英小文字は半角英大文字として扱われます。

12.4　規則および注意事項

pdsdblod コマンド実行時の規則および注意事項について説明します。

12.4.1　規則

(1)　pdsdblod コマンドの実行条件

- pdsdblod コマンドは，HiRDB が稼働中のときだけ実行できます。
- pdsdblod コマンドは，システムマネジャがあるサーバマシンで実行してください。
- pdsdblod コマンドの実行時には，次のサーバが稼働している必要があります。
  - システムマネジャ
  - 1 つ以上のフロントエンドサーバ
  - ディクショナリサーバ
  - データロードまたはフォーマットライト対象の RD エリアを管理しているバックエンドサーバ
- RD エリアの状態が pdsdblod コマンドの実行条件を満たしていない場合，pdsdblod コマンドはエラーになります。この場合，「付録 I.1　RD エリアの状態によるユティリティの実行可否」を参照して，RD エリアの状態を変更してください。
- SDB ディレクトリ情報が常用常駐領域に常駐化されていないと，pdsdblod コマンドが実行できません。また，全ユニットの常用常駐領域中の SDB ディレクトリ情報が一致している必要があります。

(2)　コマンドの実行者

pdsdblod コマンドを実行する際，CONNECT 権限が必要になります。

(3)　コマンド実行前の作業

- 入力データファイルを準備し，処理対象の RD エリアがあるサーバマシンに格納してください。
- クライアント環境定義の PDUSER オペランドに pdsdblod コマンドの実行ユーザの認可識別子とパスワードを指定してください。

  このとき，OS ログインユーザの簡易認証機能は使用できません。
- pdsdblod 制御文を指定した制御文ファイルを準備し，システムマネジャがあるサーバマシンに格納してください。また，制御文ファイルを格納するディレクトリに対する実行権限，読み込み権限，および書き込み権限を pdsdblod コマンドの実行ユーザに付与してください。
- pdsdblod 制御文のオペランドに指定するディレクトリ（実行結果ファイルを格納するディレクトリなど）は事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に付与してください。

- PDDIR，PDCONFPATH，およびLANG環境変数が正しく設定されているかを確認してください。
- pdsdblodコマンドを実行する前に，pdsdbarc -aコマンドでSDBディレクトリ情報に関する情報を表示し，全ユニットの常用常駐領域のSDBディレクトリ情報が一致しているかどうかを確認してください。
- インデクス情報を作成するために，データロードまたはフォーマットライトの処理中に次に示すファイルが一時的に作成されます。
  - インデクス情報ファイル
  - ソート用ワークファイル

  これらのファイルは，インデクスを格納するRDエリアがあるサーバマシン上に作成されます。レコード件数に比例して上記のファイル容量も大きくなるため，大量のレコードをデータロードする場合は，ディスクの空き容量を確認するようにしてください。上記ファイルの容量見積もり式については，「3.6.1(1)　HiRDB/SDデータベース作成ユティリティ（pdsdblod）実行時のファイルの容量」を参照してください。

  なお，上記ファイルはデータロードまたはフォーマットライトが正常に終了すると，自動的に削除されます。

  また，性能面に大きな影響を及ぼすソート用バッファサイズを指定することを推奨します。pdsdblod制御文のidxload文のsortdirオペランドでソート用バッファサイズを指定します。

参考

インデクス情報ファイル，ソート用ワークファイルの出力先ディレクトリは，次に示す優先順位に従って決まります。1の指定がいちばん優先順位が高くなります。

1. pdsdblod制御文のidxload文で指定した出力先ディレクトリ
2. システム定義のpd_tmp_directoryオペランドに指定した出力先ディレクトリ
3. 環境変数TMPDIRに指定したディレクトリ
4. /tmpディレクトリ

(4)　推奨する運用方法

- pdsdblodコマンドを実行する前に，処理対象のRDエリアをpdholdコマンドで閉塞することを推奨します。

参考

pdsdblodコマンドの実行中は処理対象のRDエリアにアクセスできません。UAPやほかのコマンドがアクセスしようとした場合，排他待ち状態になり，排他待ち時間のタイムアウトまでUAPやコマンドにエラーが返りません。pdholdコマンドでRDエリアを閉塞しておくと，UAPやほかのコマンドがRDエリアにアクセスしたときにすぐにエラーが返ります。

- pdsdblod 制御文の environment 文の purge オペランドに yes を指定すると，既存の格納レコードが削除されてから，データロードまたはフォーマットライトが実行されます。なお，レコードが格納されていないときに purge オペランドに yes を指定してもエラーにはなりません。

(5) インデクスを再作成する際の pdsdblod コマンドの同時実行可否

次のどちらかの条件を満たす場合は，複数の pdsdblod コマンドを同時に実行してインデクスを再作成することができます。

- 処理対象の SDB データベースが異なる場合
- 処理対象のインデクスがある BES が異なる場合

なお，処理対象となるインデクスが同一 BES にある場合でも，処理対象となるインデクスのルートレコード型が異なるデータ格納用 RD エリアに格納されているときは，複数の pdsdblod コマンドを同時に実行してインデクスを再作成できます。

処理対象となるインデクスのルートレコード型が同じデータ格納用 RD エリアに格納されている場合は，pdsdblod コマンドを同時に実行してインデクスを再作成することはできません。

pdsdblod コマンドの同時実行可否の例を次に示します。この例では，同一 BES 内に，ルートレコード型を格納した RD エリア 1 と RD エリア 2 があるとします。このときの pdsdblod【1】と pdsdblod【2】が同時実行できるかどうかを説明しています。

実行するコマンド	再作成対象のインデクス	ルートレコード型を格納している RD エリア	pdsdblod コマンドの同時実行可否
pdsdblod【1】 pdsdblod【2】	インデクス 1 インデクス 1	RD エリア 1 RD エリア 1	×
pdsdblod【1】 pdsdblod【2】	インデクス 1 インデクス 2	RD エリア 1 RD エリア 1	×
pdsdblod【1】 pdsdblod【2】	インデクス 1 インデクス 1	RD エリア 1 RD エリア 2	○
pdsdblod【1】 pdsdblod【2】	インデクス 1 インデクス 2	RD エリア 1 RD エリア 2	○

（凡例）

○：pdsdblod コマンドを同時実行してインデクスを再作成できます。

×：pdsdblod コマンドを同時実行できません。

12.4.2 注意事項

- pdsdblod コマンドの実行中にシンクポイントダンプは取得されません。そのため，pdsdblod コマンドの実行中に HiRDB が異常終了すると，HiRDB の再開始に掛かる時間が長くなります。したがって，UAP やほかのコマンドの実行数が少ない時間帯に pdsdblod コマンドを実行してください。

- 標準出力または標準エラー出力へのメッセージ出力が抑止されている環境で pdsdblod コマンドを実行しないでください。pdsdblod コマンドの実行中，標準出力に経過メッセージが出力されます。また，エラーの発生時には，標準エラー出力にエラーメッセージが出力されます。そのため，標準出力または標準エラー出力へのメッセージ出力が抑止されている環境で pdsdblod コマンドを実行した場合，メッセージの出力待ちで pdsdblod コマンドが無応答状態になったり，pdsdblod コマンドが異常終了したりするおそれがあります。

  なお，標準出力および標準エラー出力に出力するメッセージの順番や数は，メッセージログファイルや syslogfile に出力されるメッセージの順番や数と一致しないことがあります。メッセージの順番や数を正確に把握したい場合は，メッセージログファイルまたは syslogfile に出力されたメッセージを確認してください。

- システム共通定義の pd_dec_sign_normalize オペランドに Y を指定している場合（DECIMAL 型の符号正規化機能を使用する場合），データロード時に PACKED DECIMAL FIXED 型の符号部が変換規則に従って変換されます。符号部の変換規則については，「12.4.3(1)　符号部に関する注意事項（符号部の変換規則）」を参照してください。

- インデクス一括作成モードでデータロードまたはフォーマットライトする場合，1 回の pdsdblod コマンドの実行でインデクス情報ファイルが 7,001 個以上同時にオープンされるとエラーになります。このとき，KFPA63003-E メッセージが出力されます。

  同時にオープンされるインデクス情報ファイル数は，SDB データベースに定義されたインデクス数と RD エリアの横分割数に比例します。例えば，2 つのインデクスが定義されていて，RD エリア 1～RD エリア 3 に横分割されている場合，同時にオープンされるインデクス情報ファイル数は，2（インデクス数）×3（RD エリアの横分割数※1）＝ 6 になります※2。

  インデクス情報ファイル数が 7,001 個以上になる場合は，load 文の area オペランドを指定して，RD エリア単位のデータロードまたはフォーマットライトを複数回実行するようにしてください。

  注※1

  サーバ間横分割されている場合は，バックエンドサーバ内での RD エリアの横分割数となります。

  注※2

  計算結果が 7,001 個以上の場合でも，実際に同時にオープンされるインデクス情報ファイル数が 7,000 個以下になることがあります。この場合，エラーにはなりません。

- pdsdblod コマンドをキャンセルする場合は，pdcancel コマンドを実行してください。pdsdblod コマンドのキャンセルに，OS の kill コマンドを使用しないでください。kill コマンドで pdsdblod コマンドをキャンセルすると，ユニットが異常終了するおそれがあります。

- OS がサポートしていない文字コードは使用できません。pdsdblod 制御文に ASCII コード以外の文字がある場合は，pdsdblod 制御文に記述する文字コード，LANG 環境変数に設定した文字コード，および pdsetup コマンドで指定した文字コードを合わせる必要があります。文字コードが合っていない場合は，pdsdblod コマンドがエラーになることがあります。

■pdsdblod コマンドと pdload コマンドの仕様差について（参考）

データロードを実行する次の 2 つのコマンドは，コマンドの実行時にサーバ名と RD エリア名を指定しますが，指定の組み合わせが同じであっても，コマンドの実行可否が異なることがあります。

- SDB データベースに対するデータロードを実行する pdsdblod コマンド
- 表に対するデータロードを実行する pdload コマンド

例えば，横分割している SDB データベースにデータロードする場合，pdsdblod コマンドの実行時にサーバ名（server オペランド）を指定して RD エリア名（area オペランド）を省略すると，エラーになります。一方，横分割している表にデータロードする場合，pdload コマンドの実行時にサーバ名（source 文）を指定して RD エリア名（source 文）を省略してもエラーになりません。各コマンドのオプション，制御文の説明を確認してからデータロードを実行するようにしてください。pdload コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

12.4.3 メインフレームで使用していたデータをデータロードする場合の注意事項

メインフレームで使用していた符号付きパック形式のデータをデータロードする場合の注意事項を説明します。

(1) 符号部に関する注意事項（符号部の変換規則）

符号付きパック形式（HiRDB/SD の PACKED DECIMAL FIXED 型）の符号部の仕様がメインフレームと HiRDB/SD で異なります。仕様差を次の表に示します。

表 12–8 メインフレームと HiRDB/SD の符号付きパック形式の符号部の仕様差

符号部	メインフレームの仕様	HiRDB/SD の仕様	データロード時に変換される値
X'A'	正の値を意味します。	−	X'C'
X'B'	負の値を意味します。	−	X'D'
X'C'	正の値を意味します。	正の値を意味します。	変換されません。
X'D'	負の値を意味します。	負の値を意味します。	変換されません。
X'E'	正の値を意味します。	−	X'C'
X'F'	正の値を意味します。	正の値を意味します。	X'C'

（凡例）

−：不正な値と見なし，エラーとします。

上記の表に示すように符号部の仕様に差があるため，データロード時に符号部が変換されます。ただし，符号部が変換されるのは，システム共通定義の pd_dec_sign_normalize オペランドに Y を指定した場合に限ります。そのため，データロードを実行する前に，pd_dec_sign_normalize オペランドに Y が指定されているか確認してください。

注意事項

pd_dec_sign_normalize オペランドに Y を指定しないでデータロードした場合，符号部が変換されない状態でデータベースに格納されます。この状態でデータを検索，更新した場合，検索結果が正しく返らなかったり，データベースが破壊されたりすることがあります。

(2) 精度に関する注意事項

精度（全体の桁数）が偶数の場合，PACKED DECIMAL FIXED 型のデータの先頭 4 ビットが B'0000'である必要があります。メインフレームで DECIMAL 型のデータの先頭 4 ビットが B'0000'でないデータを使用していた場合は，先頭 4 ビットを 0 クリアしてからデータロードしてください。

なお，精度が奇数の場合は，先頭 4 ビットの 0 クリアは必要ありません。

12.5　リターンコード

pdsdblod コマンドのリターンコードの意味および対処を次の表に示します。

表 12-9　pdsdblod コマンドのリターンコードの意味および対処

リターンコード	意味	対処
0	pdsdblod コマンドが正常終了しました。	なし。
4	データロードは完了しましたが，入力データに論理エラーがありました。	「12.8　入力データの論理エラーが発生したときの対処方法」を参照して対処してください。
8	pdsdblod 制御文のオペランドの指定に誤りがあるため，pdsdblod コマンドが実行できませんでした。	出力されたメッセージを参照して，pdsdblod 制御文のオペランドの指定を見直してください。
12	処理が続行できないエラーが発生し，pdsdblod コマンドが異常終了しました。	「12.9.1　pdsdblod コマンドが異常終了した場合」を参照して対処してください。

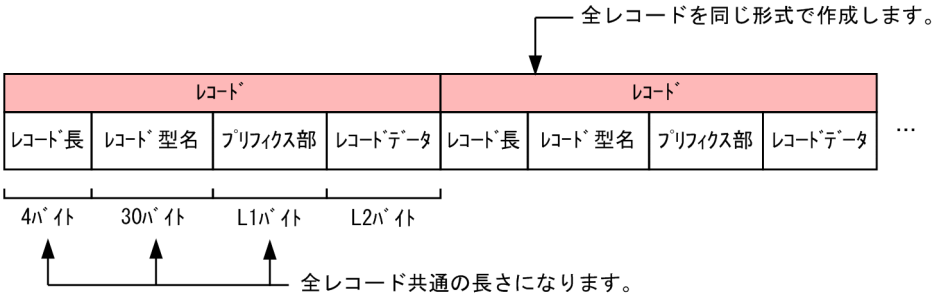
12.6　入力データファイルの形式

データロードをする前に，データベースに格納するレコードを格納したバイナリ形式の入力データファイルを準備しておく必要があります。ここでは，入力データファイルの形式について説明します。

12.6.1　レコードのデータ形式

入力データファイル中に格納するレコードのデータ形式を次の図に示します。この図に示すデータ形式に従って入力データを作成してください。

図 12-4　レコードのデータ形式



（凡例）

L1：プリフィクス部の長さで，0～32,767 バイトになります。

L2：SDB データベース定義で定義した全構成要素サイズの合計長になります。

注意事項

HiRDB/SD はエンディアンおよび文字コード変換を行わないため，入力データ中の数値データは OS のエンディアンおよび文字コードに合わせてください。

レコード内の各項目について説明します。

(1)　レコード長

次に示す計算式で求めたレコード長（単位：バイト）を格納してください。

レコード長＝ 4 ＋ 30 ＋ L1 ＋ L2

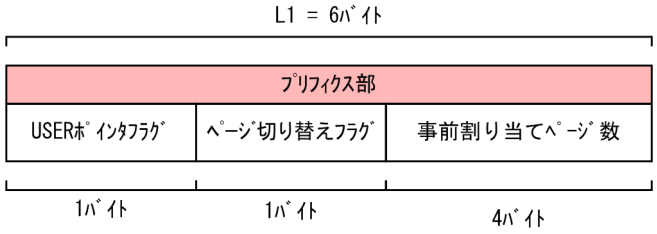
(2)　レコード型名

レコードのレコード型名を格納してください。左詰めで格納し，余った部分にはスペースを格納してください。

(3) プリフィクス部【4V FMB，4V AFM】

プリフィクス部には，親子集合型の情報を格納してください。ここには，USER ポインタフラグ，ページ切り替えフラグ，および事前割り当てページ数の情報を格納します。プリフィクス部のデータ形式の例を次の図に示します。

図 12–5　プリフィクス部のデータ形式の例



この例では，USER ポインタフラグ，ページ切り替えフラグ，事前割り当てページ数の順でプリフィクス部を構成していますが，指定順序に制約はありません。

また，USER ポインタフラグ，ページ切り替えフラグ，事前割り当てページ数のすべてを指定する必要はありません。必要な情報だけでプリフィクス部を構成しても問題ありません。USER ポインタフラグ，ページ切り替えフラグ，事前割り当てページ数のすべてが必要ない場合は，プリフィクス部を作成する必要はありません。

プリフィクス部を指定する場合の規則を次に示します。

- 各オペランドに指定した開始位置の情報を格納する領域の長さが重ならないように指定してください。
- load 文の prefix オペランドに指定されたプリフィックス長の範囲内で指定してください。

(a) USER ポインタフラグに設定する値【4V FMB】

USER ポインタフラグに設定する値を次の表に示します。

表 12–10　USER ポインタフラグに設定する値

項番	設定値	説明
1	C'U'	4V FMB の SDB データベースで USER ポインタを定義している場合に USER ポインタフラグを設定します。そのほかの場合に設定しても，設定は無効になります。 左記に示す値を子レコードに設定した場合，親レコードが USER ポインタを持つという意味になります。複数の子レコードに USER ポインタフラグを設定した場合，最後の子レコードに設定した USER ポインタフラグが有効になります。
2	上記以外	無視されます。

(b) ページ切り替えフラグに設定する値

ページ切り替えフラグに設定する値を次の表に示します。

表 12–11　ページ切り替えフラグに設定する値

項番	条件	設定値	説明
1	サブページ分割をしていない場合	C'C'	左記に示す値が設定されているレコードを格納する際，格納ページを切り替えて，空きページにそのレコードを格納します。このレコードを格納したページには，ほかのレコードも格納されます。
2		C'O'	左記に示す値が設定されているレコードを格納する際，格納ページを切り替えて，空きページにこのレコードを格納します。このレコードを格納したページには，ほかのレコードは格納されません。
3		上記以外	格納ページの切り替えは行われません。
4	サブページ分割をしている場合	C'C'	左記に示す値が設定されているレコードを格納する際，格納サブページを切り替えて未使用サブページにそのレコードを格納します。このレコードを格納したサブページには，ほかのレコードも格納されます。
5		C'O'	左記に示す値が設定されているレコードを格納する際，格納サブページを切り替えて，未使用サブページにこのレコードを格納します。このレコードを格納したサブページには，ほかのレコードは格納されません。
6		上記以外	格納サブページの切り替えは行われません。

ページ切り替えについては，「2.6.2(3) レコード格納時のページ切り替え【4V FMB，4V AFM】」を参照してください。

サブページ切り替えについては，「2.7.2(3)　レコード格納時のサブページ切り替え【4V FMB，4V AFM】」を参照してください。

(c)　事前割り当てページ数に設定する値【4V FMB】

事前割り当てページ数に設定する値を次の表に示します。

表 12–12　事前割り当てページ数に設定する値

項番	SDB データベース格納定義の STORAGE RECORD 句の SUBPAGE NUMBER の指定	指定値	事前ページ割り当て制御の内容
1	指定あり（サブページ）	0	事前割り当てサブページは割り当てられません。
2		1～2,147,483,647	指定値数分のサブページを事前割り当てサブページとして割り当てます。
3		上記以外	論理エラーを出力し，該当するルートレコードをスキップします。
4	指定なし（ページ）	0	事前割り当てページは割り当てられません。
5		1～2,147,483,647	指定値数分のページを事前割り当てページとして割り当てます。
6		上記以外	論理エラーを出力し，該当するルートレコードをスキップします。

(d) 留意事項

ルートレコード型の入力データ以外のレコードに，事前割り当てページ数を指定しても無視されます。事前割り当てページ数とページ切り替えフラグの指定値の関係を次の表に示します。

表 12-13 事前割り当てページ数とページ切り替えフラグの指定値の関係

項番	プリフィクス部の指定値			pdsdblod コマンドの動作
	事前割り当てページ数※1	ページ切り替えフラグ		
		ルートレコード	子レコード	
1	0	'c'	－	ページ切り替え※2 が有効になります。
2		'o'		
3		上記以外		ページ切り替え※2 および事前ページ割り当て※3 は適用されません。
4	1～2,147,483,647	'c'		論理エラーを出力し，該当するルートレコードおよびその子レコード群をスキップします。
5		'o'		
6		上記以外		事前ページ割り当て※3 が有効になります。
7	0	－	'c'	ページ切り替え※2 が有効になります。
8			'o'	
9			上記以外	ページ切り替え※2 および事前ページ割り当て※3 は適用されません。
10	1～2,147,483,647		'c'	論理エラーを出力し，該当するルートレコードおよびその子レコード群をスキップします。
11			'o'	
12			上記以外	事前ページ割り当て※3 が有効になります。

（凡例）

－：該当しません。

注※1

事前割り当てページ数の設定値が有効になるのはルートレコード型の入力データだけです。ルートレコード型以外の入力データに設定されていた場合は無視されます。

注※2

SDB データベース格納定義の STORAGE RECORD 句の SUBPAGE NUMBER オペランドでサブページ分割数が指定されている場合は，サブページ切り替えになります。

注※3

SDB データベース格納定義の STORAGE RECORD 句の SUBPAGE NUMBER オペランドでサブページ分割数が指定されている場合は，事前サブページ割り当てになります。

また，データロード時にレコードを格納するために確保したページ（またはサブページ）の数が，指定した事前割り当てページ数（または事前割り当てサブページ数）を超えた場合，該当するファミリを格納するために指定値を超えて確保したページ（またはサブページ）も，事前割り当てページ（事前割り当てサブページ）として扱われます。

(4) レコードデータ

レコードデータには，データベースに格納するデータを格納してください。SDB データベース定義で指定したレコード型と同じ構成要素を持つデータを格納します。レコードデータの例を次に示します。

SDB データベース定義の例（4V FMB の場合）

```
   :
RECORD レコード型名
 2 DBKEY_L2 XCHARACTER 1 TYPE K,L                    …科目
 2 DBKEY_L3 XCHARACTER 3 TYPE K,L                    …店番
 2 DBKEY_L4 XCHARACTER 5 TYPE K,L                    …顧客番号
 2 DBKEY_K2 INTEGER       TYPE K,P                   …上位の一連番号
 2 DBKEY_K3 INTEGER       TYPE K,P                   …上位の一連番号
 2 DBKEY    INTEGER       TYPE K,N                   …一連番号
 2 USERDATA CHARACTER 5  TYPE U,D                    …ユーザデータ
   :
```

レコードデータの例

科目	店番	顧客番号	上位の一連番号（0を設定）	上位の一連番号（0を設定）	一連番号（0を設定）	ユーザデータ
1	3	5	4	4	4	5

次に示す構成要素が定義されている場合は，領域を確保して 0 を設定してください。

- 自一連番号
- 上位の一連番号

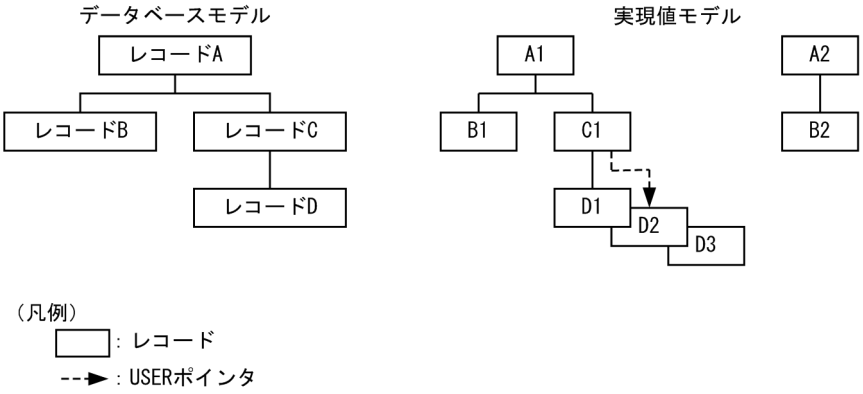
HiRDB/SD は，これらの構成要素の領域の値を参照しません。

12.6.2 レコードの格納順序

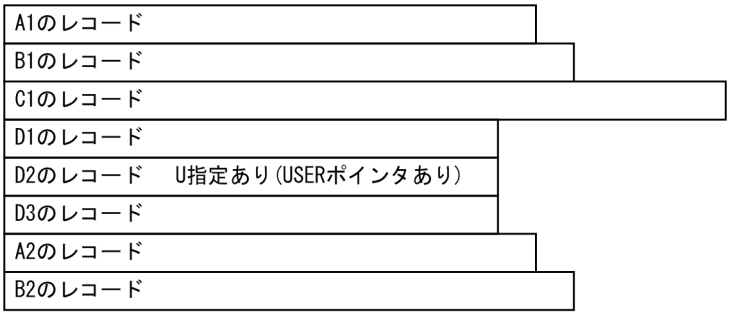
レコードの格納順序はデータベースの種類によって異なります。

(1) 4V FMB または SD FMB の SDB データベースの場合

次に示すデータベースモデルと実現値モデルを例にして，入力データファイルのレコードの格納順序を説明します。



この場合，入力データファイル中のレコード格納順序は次のようにしてください。



入力データファイル中のレコード格納順序に従って，データベースにレコードが格納されます。

規則および注意事項を次に示します。

- load 文の seqkeyck に yes を指定する場合は，ルートレコードのシーケンシャルインデクスのキー値が昇順になるように作成してください。
- 子レコードを親子集合に挿入する順序は，SDB データベース定義の ORDER 句の指定に関係なく，入力データファイル中のレコードの順番に格納します。ただし，SDB データベース定義の ORDER 句に SORTED DUPLICATES を指定した子レコード型のデータは，ユーザキーが KEY 句で指定した順序（ASCENDING は昇順，DESCENDING は降順）となるように作成してください。並び順が不正な場合，論理エラーになります。

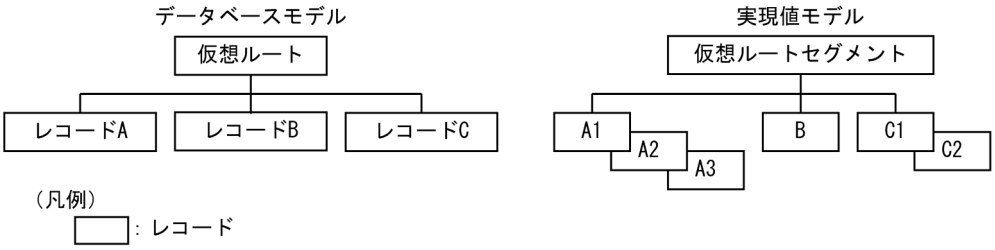
  また，SORTED DUPLICATES PROHIBITED を指定した場合は，キー値の重複時も論理エラーになります。
- 横分割された SDB データベースに二次インデクスが定義されている場合，レコードを格納する RD エリアと二次インデクスを格納する RD エリアが対応しているかどうかを，pdsdblod コマンドの実行時にチェックします。具体的には，レコードと二次インデクスの格納先が定義上の何番目の RD エリアなのかを求め，レコードと二次インデクスが定義上の同じ位置に格納されることをチェックします。レコードを格納する RD エリアと二次インデクスを格納する RD エリアが対応していない（異なる位置に格納される）場合は，論理エラーになります。

  なお，子レコードに定義した二次インデクスで，キー（SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素）を二次インデクスの RD エリア分割キーに指定した場合，ルートレコードの格納時に論理エラーになります。

チェックの例は，「17.4.6(4)　共通規則」を参照してください。

(2)　4V AFM の SDB データベースの場合

次に示すデータベースモデルと実現値モデルを例にして，入力データファイルのレコードの格納順序を説明します。



この場合，入力データファイル中のレコード格納順序は次のようにしてください。

入力データファイル中のレコード格納順序に従って，データベースにレコードが格納されます。

仮想ルートレコードに対する実現値のレコードは必要ありません。

子レコードを親子集合に挿入する順序は，SDB データベース定義の ORDER 句の指定に従います。ただし，SDB データベース定義の ORDER 句に SORTED DUPLICATES PROHIBITED を指定した子レコード型のデータは，ユーザキーが KEY 句で指定した順序（ASCENDING は昇順，DESCENDING は降順）となるように作成してください。並び順が不正な場合，論理エラーになります。

12.7　実行結果ファイルの出力形式

データロードまたはフォーマットライトの処理が完了すると，データロードまたはフォーマットライトの実行結果が実行結果ファイルに出力されます。実行結果ファイルに出力される情報について例を使って説明します。

実行結果ファイルの出力例

4V FMB の SDB データベースに対するデータロードで，インナレプリカ機能を使用している場合の出力例を次に示します。

なお，出力例中の番号は，［説明］の番号と対応しています。

```
pdsdblod (09-60) **** DB initialize load **** 2015-12-11 02:07:32 [1]     parallel
KFPB63501-E Schema record type is not found, [2]
           record=RECA, line=12, position=4
     :
KFPB63518-I Used page number more than specified page number,       [3]
            DBKEY(005): 464D423131                                  [4]
            Used page number:3                                      [5]
            Specified page number:2                                 [6]
     :
** initial load information list ** [7]
 server name : BES1  [8]
 schema name : FMB1  [9]
 generation  : 0     [10]
      no[11] area-name[12] schema-record-name[13]  storage-record-name[14] record-count[1
5]
       1    RDAREA01     FMB100                 FMB100                          53445
                         FMB111                 FMB111                          75886
                         FMB112                 FMB112                          61258
                         FMB113                 FMB113                          59471


** error record list **[16]
 no  error classification                                  error count
[17] [18]                                                  [19]
  1  Schema record type is not found                                5
  2  Schema record type is invalid                                  0
  3  Schema record type is out of sequence                          0
  4  Data type is incorrect                                         1
  5  Key value is invalid                                           1
  6  Duplicate sequential key                                       2
  7  Not sequential key order                                       0
  8  Sort key is incorrect                                          0
  9  Sequence number exceeds the maximum value                      0
 10  Key value is not specified for KEYDEF                          0
 11  Database key is incorrect                                      0
 12  Partitioning key is incorrect                                  0
 13  Page number is out of range                                    0
 14  Prefix combination is incorrect                                0
 15  Duplicate index key value                                      0
 16  Partitioning secondary index key is incorrect                  0
 17  Partitioning secondary index key is incorrect (child record)   0
              total error count = 9 [20]
```

```
** pdsdblod statistics information **                                    [21]
data delete                                                              [22]
 lock time (data delete):********************                            [23]
 data delete time:********************                                   [24]
 commit time (data delete):********************                         [25]
data load                                                                [26]
 lock time (data load):********************                             [27]
 start call time:********************                                   [28]
 index data extract time:********************                           [29]
 data file read time (sum):********************                         [30]
                     (max):********************                         [31]
                     (avg):********************                         [32]
                     (cnt):********************                         [33]
 data store time (sum):********************                             [34]
                 (max):********************                             [35]
                 (avg):********************                             [36]
                 (cnt):********************                             [37]
 finish call time:********************                                  [38]
 commit time (data load):********************                           [39]
 index data file sort time:********************                        [40]
 index load time:********************                                  [41]
 lock time (reflection conversion table load):********************      [42]
 midfile read time:(sum):********************                          [43]
                   (max):********************                          [44]
                   (avg):********************                          [45]
                   (cnt):********************                          [46]
 reflection conversion table load time (sum):********************      [47]
                                       (max):********************      [48]
                                       (avg):********************      [49]
                                       (cnt):********************      [50]
 commit time (reflection conversion table load):********************   [51]
```

[説明]

1. pdsdblod コマンドを実行した日時が表示されます。
2. 論理エラーに関するメッセージです。論理エラーに関するメッセージは，論理エラーが発生した数だけ表示されます。※1
3. 事前割り当てページ数または事前割り当てサブページ数を指定した場合に，指定したページ数を超えて割り当てられたときに出力されるインフォメーションメッセージです。ファミリ単位に，その数だけ表示されます。

   メッセージに続いて，[4]，[5]，[6]が出力されます。
4. ルートレコードのデータベースキー長と DBKEY が表示されます。16 進文字形式で表示されます。
5. ファミリを格納するために確保した事前割り当てページ数，または事前割り当てサブページ数が表示されます。
6. ユーザが指定した事前割り当てページ数，または事前割り当てサブページ数が表示されます。
7. 以降にデータロードの実行結果が表示されます。※2
8. 処理対象のバックエンドサーバ名が表示されます。
9. 処理対象の SDB データベース名が表示されます。

10. 処理対象の RD エリアの世代番号が表示されます。0 はオリジナル RD エリアを意味しています。インナレプリカ機能を使用している場合に表示されます。
11. 1 から始まる通番が表示されます。
12. レコードを格納した RD エリアの名称が表示されます。インナレプリカ機能を使用している場合は，レプリカ RD エリアを処理対象にしたときでもオリジナル RD エリア名が表示されます。
13. 格納先のレコード型名が表示されます。SDB データベース定義の RECORD 句に指定したレコード型名が表示されます。※3
14. 処理対象のレコード型名が表示されます。SDB データベース格納定義の STORAGE RECORD 句に指定した格納レコード名が表示されます。※3
15. データベースに格納されたレコード件数が表示されます。
16. 以降に論理エラーに関する情報が表示されます。論理エラーが発生した場合に表示されます。
17. 論理エラー番号が表示されます。
18. 論理エラーの内容が表示されます。詳細については，「12.8.3　論理エラーの詳細」を参照してください。
19. 論理エラーの種類ごとに，論理エラーの件数が表示されます。
20. 論理エラーの合計件数が表示されます。
21. pdsdblod コマンドのユーザサーバプロセスの実行時間に関する統計情報が表示されます。
22. 既存データ削除時の pdsdblod コマンドの実行時間に関する統計情報が表示されます。
23. 既存データ削除時の排他取得の実行時間がマイクロ秒単位で表示されます。
24. 既存データ削除の実行時間がマイクロ秒単位で表示されます。
25. 既存データ削除後の commit の実行時間がマイクロ秒単位で表示されます。
26. データ格納時の pdsdblod コマンドの実行時間に関する統計情報が表示されます。
27. データ格納時の排他取得の実行時間がマイクロ秒単位で表示されます。
28. システムが使用する内部情報が表示されます。
29. インデクス情報抽出の実行時間がマイクロ秒単位で表示されます。
30. 入力データファイルの読み込み実行時間の合計値がマイクロ秒単位で表示されます。
31. 入力データファイルの読み込み実行時間の最大値がマイクロ秒単位で表示されます。
32. 入力データファイルの読み込み実行時間の平均値がマイクロ秒単位で表示されます。
33. 入力データファイルの読み込み件数が表示されます。
34. データ格納実行時間の合計値がマイクロ秒単位で表示されます。
35. データ格納実行時間の最大値がマイクロ秒単位で表示されます。
36. データ格納実行時間平均値がマイクロ秒単位で表示されます。
37. データ格納の件数が表示されます。

38. システムが使用する内部情報が表示されます。

39. データ格納後の commit の実行時間がマイクロ秒単位で表示されます。

40. インデクス情報ファイルのソート実行時間がマイクロ秒単位で表示されます。

41. インデクス一括作成の実行時間がマイクロ秒単位で表示されます。

42. 追い付き反映キー対応表データ登録時の排他取得の実行時間がマイクロ秒単位で表示されます。

43. 中間ファイル読み込み実行時間の合計値がマイクロ秒単位で表示されます。

44. 中間ファイル読み込み実行時間の最大値がマイクロ秒単位で表示されます。

45. 中間ファイル読み込み実行時間の平均値がマイクロ秒単位で表示されます。

46. 中間ファイル読み込み件数が表示されます。

47. 追い付き反映キー対応表のデータ登録実行時間の合計値がマイクロ秒単位で表示されます。

48. 追い付き反映キー対応表のデータ登録実行時間の最大値がマイクロ秒単位で表示されます。

49. 追い付き反映キー対応表のデータ登録実行時間の平均値がマイクロ秒単位で表示されます。

50. 追い付き反映キー対応表のデータ登録の件数が表示されます。

51. 追い付き反映キー対応表のデータ登録後の commit の実行時間がマイクロ秒単位で表示されます。

注※1

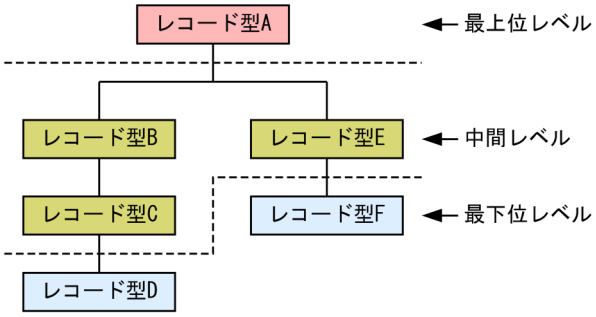
論理エラー番号とメッセージの対応は次のとおりになります。

- 論理エラー番号 1～14
  論理エラー番号 XX は KFPB635XX-E メッセージに対応しています。
  例えば，論理エラー番号 1 は KFPB63501-E メッセージに対応しています。
- 論理エラー番号 15
  KFPB63517-E メッセージに対応しています。
- 論理エラー番号 16
  KFPB63519-E メッセージに対応しています。
- 論理エラー番号 17
  KFPB63520-E メッセージに対応しています。

注※2

階層順（最上位レベル，中間レベル，最下位レベルの順）にレコード型の情報が表示されます。

（例）



この例の場合の表示順序は次のとおりになります。

1. レコード型 A の情報（最上位レベル）
2. レコード型 B の情報（中間レベルの上位）
3. レコード型 C の情報（中間レベルの下位）
4. レコード型 E の情報（中間レベルの下位）
5. レコード型 D の情報（最下位レベル）
6. レコード型 F の情報（最下位レベル）

- 中間レベルのレコード型が複数ある場合は，中間レベルの中で上位レベル，中間レベル，下位レベルの順に表示されます。
- 同一レベルのレコード型が複数ある場合は，pdsdbdef コマンドで定義した順に表示されます。
- SDB データベースの定義変更（pdsdbdef コマンドの*ALTER DICTIONARY 文および*ALTER DIRECTORY 文）でレコード型を追加した場合，追加したレコード型の情報は末尾に表示されます。

注※3

4V AFM の SDB データベースの場合，仮想ルートレコードのレコード型名は出力されません。

参考

pdsdblod コマンドが異常終了した場合，実行結果ファイルにすべての情報が出力されないことがあります。例えば，先頭のヘッダ行だけが表示されたり，障害発生時点までの情報だけが表示されたりすることがあります。

12.8　入力データの論理エラーが発生したときの対処方法

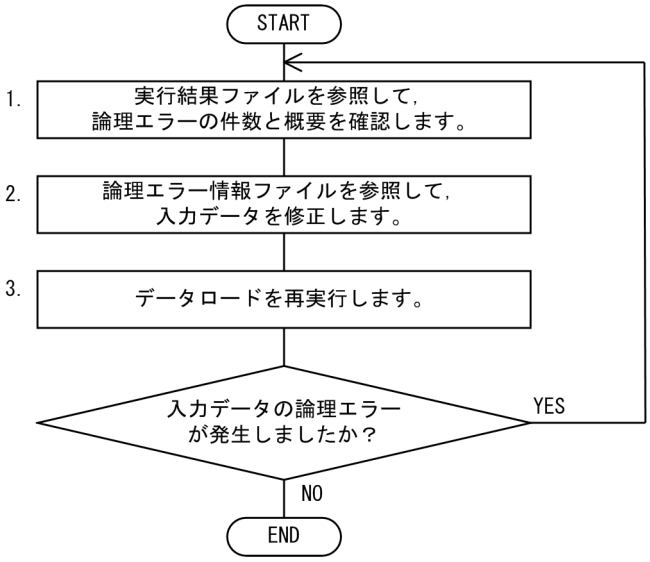
ここでは，入力データの論理エラーが発生したときの対処方法について説明します。

12.8.1　対処の流れ

データロードの際，正常なレコードだけがデータベースに格納され，論理エラーのレコードはデータベースに格納されません。論理エラーのレコードをデータベースに格納したい場合は，入力データを修正してからデータロードを再実行します。

入力データの論理エラーが発生したときの対処の流れを次の図に示します。なお，この手順は，論理エラー情報ファイルに論理エラー情報が出力されている（load 文の errdata オペランドが指定されている）ことが前提となっています。

図 12-6　入力データの論理エラーが発生したときの対処の流れ



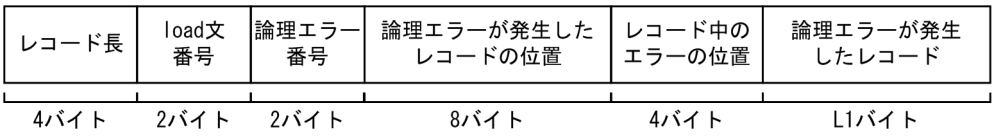
[説明]

1. 実行結果ファイルに論理エラーの件数と論理エラーの内容が出力されます。実行結果ファイルについては，「12.7　実行結果ファイルの出力形式」を参照してください。
2. 論理エラー情報ファイルに論理エラー情報が出力されます。この情報を参照して，入力データファイル中のデータを修正します。
3. 修正した入力データファイルを使用して，データロードを再実行します。

12.8.2　論理エラー情報ファイルの出力形式

論理エラー情報ファイルの出力形式を次の図に示します。

図 12-7　論理エラー情報ファイルの出力形式



（凡例）　L1：入力データのレコード長

[説明]

論理エラーの件数分，上記の形式で論理エラー情報がバイナリ形式で出力されます。

- レコード長

  論理エラー情報の 1 レコードのレコード長が出力されます。レコード長は，20 ＋ L1 バイトになります。

- load 文番号

  常に 1 が出力されます。

- 論理エラー番号

  論理エラーの原因を示す論理エラー番号が出力されます。論理エラー番号と論理エラーの内容については，「12.8.3　論理エラーの詳細」を参照してください。

- 論理エラーが発生したレコードの位置

  論理エラーが発生したレコードの位置が出力されます。例えば，5 が出力された場合，入力データファイル中の 5 レコード目に論理エラーが発生したことを意味しています。

- レコード中のエラーの位置

  論理エラー番号 1～14：レコード中の論理エラー発生個所の位置が出力されます。例えば，10 が出力された場合，レコード中の先頭から 10 バイト目に論理エラーが発生したことを意味しています。

  論理エラー番号 15：0 が出力されます。

  論理エラー番号 16～17：二次インデクスの RD エリア分割キーの位置が出力されます。

- 論理エラーが発生したレコード

  論理エラーが発生したレコードが出力されます。

参考

論理エラー情報ファイルの形式は入力データファイルの形式とは異なるため，論理エラー情報ファイルを入力データファイルとして使用することはできません。

12.8.3　論理エラーの詳細

論理エラー番号と論理エラーの内容を次の表に示します。

表 12–14　論理エラー番号と論理エラーの内容

論理エラー番号	論理エラーの内容
1	SDB データベース定義で定義されていないレコード型名が入力データファイル中にあります。
2	入力データファイル中のレコード型名に誤りがあります。4V AFM の SDB データベースの場合に考えられる原因を次に示します。 • レコード型名に仮想ルートレコードのレコード型名が設定されています。
3	入力データファイル中のレコードの並びが階層構造の順番になっていません。正しい階層構造の順番を次に示します。 1. 親レコード型のレコード 2. 親レコード型に関係づけられる子レコード型のレコード 3. 上位階層のレコード型のレコード 4. 上位階層のレコード型に関係づけられるレコード型のレコード 1 つの親レコード型に関係づけられる同じ階層の子レコード型が複数存在する場合，同じレコード型のレコードは連続させてください。
4	構成要素の値のデータ型が，対応する SDB データベース定義で指定したデータ型と合っていません。
5	入力データファイル中のキー値に論理エラーが発生しました。4V FMB または SD FMB の SDB データベースの場合に考えられる原因を次に示します。 • キー値が親レコード型のキー値と合っていません。
6	シーケンシャルインデクスのキー値が重複しています。
7	シーケンシャルインデクスが定義されているレコード型のインデクスキー値が，シーケンシャルに並んでいません，または同じ値が並んでいます。※
8	入力データファイル中のキー値に論理エラーが発生しました。考えられる原因を次に示します。 • SDB データベース定義に SORTED DUPLICATES FIRST または SORTED DUPLICATES LAST が指定されている場合，キー値の並び順に誤りがあります。 • SDB データベース定義に SORTED DUPLICATES PROHIBITED が指定されている場合，キー値の重複が発生しました。または，キー値の並び順に誤りがあります。※
9	SDB データベース定義の OCCURRENCE NUMBER 句に指定した一連番号の最大値を超えました。
10	SDB データベース格納定義の KEYDEF 句で指定されていないキー値があります。
11	キー値の並び順に誤りがあります。4V AFM の SDB データベースの場合に考えられる原因を次に示します。 • キー値の並び順が，SDB データベース格納定義の KEYDEF 句で指定した並び順と異なっています。※
12	横分割している SDB データベースに対して RD エリア単位のデータロードを実行した結果，格納対象外のデータ（ほかの RD エリアに格納するデータ）がありました。
13	入力データファイル中のデータベース固有領域に指定されている，事前割り当てページ数の指定値が，指定可能な最大値を超えています。
14	入力データファイル中のデータベース固有領域に指定されている組み合わせで論理エラーが発生しました。考えられる原因を次に示します。

論理エラー番号	論理エラーの内容
	• 事前割り当てページ数が指定されたルートレコードに，ページ切り替えフラグが指定されています。または，事前割り当てページ数が指定されたファミリ内に，ページ切り替えフラグが指定されています。
15	レコードの格納時，二次インデクスキーに対してキー重複が発生しました。 この論理エラーは，SDB データベース種別が SD FMB のときだけ発生します。SDB データベース種別が 4V のときは発生しません。
16	RD エリアの分割時，レコードを格納する RD エリアと二次インデクスを格納する RD エリアが対応していません。 レコードと二次インデクスの格納先が，定義上の何番目の RD エリアなのかを求め，レコードと二次インデクスが定義上の同じ位置の RD エリアに格納されることをチェックします。
17	RD エリアの分割時，レコードを格納する RD エリアと二次インデクスを格納する RD エリアが対応していません。子レコードでキー（SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素）を二次インデクスの RD エリア分割キーに指定した場合，ルートレコードの格納時に論理エラー情報が出力されます。 二次インデクスの RD エリア分割キーに対応するルートレコードの構成要素の値を基に，二次インデクスを格納する RD エリアを求めてチェックします。 チェックの例は，「17.4.6(4)　共通規則」を参照してください。

注※

キー値のデータ型が PACKED DECIMAL FIXED の場合，DECIMAL 型の符号正規化機能による符号部の変換規則に従って変換されたキー値の並び順でチェックされます。符号部の変換規則については，「12.4.3(1)　符号部に関する注意事項（符号部の変換規則）」を参照してください。

12.9　トラブルシューティング

ここでは，pdsdblod コマンドが異常終了したときの対処方法について説明します。

12.9.1　pdsdblod コマンドが異常終了した場合

(1)　pdsdblod コマンドが異常終了した場合の対処方法

pdsdblod コマンドが異常終了した場合の対処方法について「5.22　障害が発生したときの対処方法(pdsdblod コマンドの異常終了時)」を参照してください。

(2)　注意事項

- pdsdblod コマンドが異常終了した場合，インデクスを一括作成する際に作成されるインデクス情報ファイルが自動的に削除されないことがあります。この場合，エラーの原因を対策したあとにインデクス情報ファイルを削除してください。インデクス情報ファイルは，次に示すどれかのディレクトリ下に作成されます（優先順位は 1 の指定がいちばん上位です）。

  1. idxload 文の workdir オペランドに指定したディレクトリ
  2. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
  3. 環境変数 TMPDIR に指定したディレクトリ
  4. /tmp ディレクトリ

  インデクス情報ファイルの名称は次のようになります。

  - SDBLOD-IDX-HiRDB 識別子-インデクス名-インデクスが格納されている RD エリアの RD エリア ID※

  注※

  インナレプリカ機能を使用している場合は，pdsdblod コマンドの処理対象 RD エリアの RD エリア ID になります。

- 更新可能なオンライン再編成の実行時に pdsdblod コマンドが異常終了した場合，追い付き反映キー対応表へのデータ登録処理が正常に完了したかどうかを確認してください。次の場合は，追い付き反映キー対応表へのデータ登録処理が正常に完了していません。

  - KFPB63043-I メッセージ（追い付き反映キー対応表へのデータ登録処理が完了した旨のメッセージ）が出力されていない場合
  - KFPB63043-I メッセージがリターンコード 0 以外で出力された場合

  この場合，データロードを再実行する前に，追い付き反映キー対応表の初期化が必要になります。また，追い付き反映キー対応表を初期化した場合，BES 内の更新可能なオンライン再編成の対象 RD エリアに対して，データロードを再実行する必要があります。例えば，BES1 内で SDB データベースを RDAREA01 と RDAREA02 に横分割している場合，RDAREA01 と RDAREA02 にそれぞれ

pdsdblod コマンドを実行してデータロードを行います。この場合，RDAREA01 のデータロードは正常終了し，RDAREA02 のデータロードが異常終了して，追い付き反映キー対応表を初期化したときは，RDAREA01 と RDAREA02 の両方に対してデータロードを再実行する必要があります。

12.9.2 pdsdblod コマンドが無応答状態になった場合

pdsdblod コマンドが無応答状態になった場合，pdcancel コマンドで pdsdblod コマンドをキャンセルしてください。手順を次に示します。

手順 1

1. pdls -d prc コマンドを実行します。コマンドの実行結果は，リダイレクションでファイルに出力してください。

```
pdls -d prc>出力先ファイル名
```

2. 1.の実行結果を参照して，バックエンドサーバのプロセス ID を確認します。

```
HOSTNAME : host01(135225)
STATUS PID   UID    GID    SVID   TIME  PROGRAM   C-PID  C-GRP
   :   2924   :      :      :      :    pdsdblod    :      :
```

PROGRAM 欄に pdsdblod と表示されている行を検索してください。この行の PID 欄に表示されているプロセス ID（この例では 2924）が，pdsdblod コマンド実行中のバックエンドサーバのプロセス ID です。

pdsdblod が表示されていない場合は，手順 2 の方法でプロセス ID を確認してください。

3. pdcancel コマンドで，pdsdblod コマンド実行中のバックエンドサーバをキャンセルします。

```
pdcancel -X ユニット識別子 -i プロセスID -d
```

ユニット識別子には，pdsdblod コマンドの処理対象 RD エリアがあるユニットのユニット識別子を指定してください。

プロセス ID には，2.で確認したプロセス ID（この例では 2924）を指定してください。

手順 2

1. pdls -d rpc コマンドを実行します。コマンドの実行結果は，リダイレクションでファイルに出力してください。

```
pdls -d rpc -a>出力先ファイル名
```

2. 1.の実行結果を参照して，ユティリティサーバ（pdsdblodm）のプロセス ID を確認します。

```
Server Information :
  PID     SVID     STATUS …   CLTPID
  5444    0msdldN    :           :
```

SVID 欄に 0msdldN（末尾の N は 16 進の数字）と表示されている行を検索してください。この行の PID 欄に表示されているプロセス ID（この例では 5444）が，ユティリティサーバ（pdsdblodm）のプロセス ID です。

なお，0msdldN が複数表示されている場合は，CLTPID 欄（サーバのプロセスと接続しているクライアントのプロセス ID）を確認してください。無応答状態になった pdsdblod コマンドのプロセス ID が CLTPID 欄に表示されている行が該当する行になります。pdsdblod コマンドのプロセス ID は，OS の ps コマンドで確認できます。

3. pdcancel コマンドで，ユティリティサーバ（pdsdblodm）をキャンセルします。

```
pdcancel -X ユニット識別子 -i プロセスID -d
```

ユニット識別子には，pdsdblod コマンドの処理対象 RD エリアがあるユニットのユニット識別子を指定してください。

プロセス ID には，2.で確認したプロセス ID（この例では 5444）を指定してください。

なお，無応答状態の原因調査のために，pdsdblod コマンドのキャンセル後，pdinfoget コマンドで障害情報を取得してください。

注意事項

- ログレスモードの pdsdblod コマンドをキャンセルした場合，処理対象 RD エリアが障害閉塞します。そのため，pdsdblod コマンドを再実行する場合は，障害閉塞を解除する必要があります。障害閉塞を解除する方法については，「5.22　障害が発生したときの対処方法（pdsdblod コマンドの異常終了時）」を参照してください。
- pdsdblod コマンドをキャンセルする際，OS の kill コマンドを使用しないでください。kill コマンドで pdsdblod コマンドをキャンセルすると，ユニットが異常終了するおそれがあります。
- 大量のレコードをデータロードまたはフォーマットライトしているときに，pdsdblod コマンドをキャンセルすると，トランザクションのロールバック処理に時間が掛かります。このロールバック処理が完了するまで pdsdblod コマンドを再実行できません（コマンドの実行はできますが，排他待ち状態になります）。ロールバック処理が完了したかどうかは，pdls -d trn -a コマンドで確認できます。また，ロールバック処理が完了したときに，KFPB63451-E メッセージが出力されます。

12.9.3　RD エリアの容量不足が発生した場合

pdsdblod コマンドの実行中に処理対象の RD エリアの容量不足が発生すると，pdsdblod コマンドが異常終了します。この場合，「12.9.1　pdsdblod コマンドが異常終了した場合」の手順に従って，pdsdblod コマンドを再実行してください。

なお，pdsdblod コマンドを再実行する前に，容量不足が発生した RD エリアを拡張する必要があります。RD エリアの拡張方法については，マニュアル「HiRDB システム運用ガイド」の「RD エリアの容量を大きくする方法（RD エリアの拡張）」を参照してください。

12.9.4　タイムアウトが発生した場合

environment 文の exectime オペランドに指定した pdsdblod コマンドの実行時間の上限を超えた場合，pdsdblod コマンドが強制終了します。この場合，メッセージを確認してタイムアウトの原因を対処してください。そのあとに，「12.9.1　pdsdblod コマンドが異常終了した場合」の手順に従って，pdsdblod コマンドを再実行してください。

参考

exectime オペランドを省略した場合，システム定義の pd_utl_exec_time オペランド，または pd_cmd_exec_time オペランドに指定したタイムアウト時間が適用されます。

12.10　使用例

ここでは，pdsdblod コマンドの使用例について説明します。

なお，pdsdblod コマンドの使用例については，pdsdbrog コマンドの「13.10　使用例」でも説明しています。そちらもあわせてお読みください。

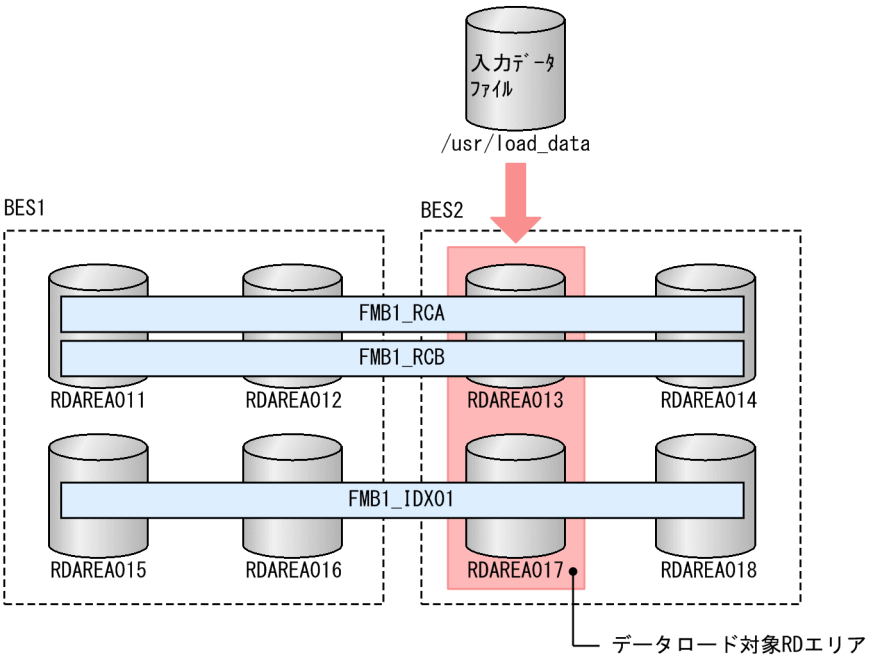
12.10.1　初期データロードの例（4V FMB の場合）

例題

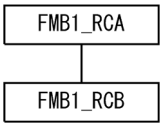
横分割されている SDB データベースに対して，RD エリア単位の初期データロードを実行します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V FMB です。
- インデクス FMB1_IDX01 が定義されています。

- 初期データロードの対象 RD エリアは RDAREA013 および RDAREA017 です。
- ページ切り替えフラグを設定します。

(2) 初期データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = FMB1              ...1
  logmode  = no                ...2
  purge    = yes               ...3
  exectime = 0                 ...4
  recnomsg = 1                 ...5
  divermsg = put               ...6
load
  type     = tree              ...7
  record   = FMB1_RCA          ...8
  server   = BES2              ...9
  area     = RDAREA013         ...10
  data     = /usr/load_data    ...11
  storinf  = /usr/load_result  ...12
  errdata  = /usr/err_data     ...13
  prefix   = 1                 ...14
  idxmode  = create            ...15
dbinf
  pagecflg = (0)               ...16
```

[説明]

1. SDB データベース名を指定します。
2. ログレスモードで初期データロードを実行します。
3. 既存レコードを削除して初期データロードを実行します。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. データロードが 1 万件完了するごとに経過メッセージを出力します。
6. ほかの RD エリアに格納するべきレコードは論理エラーとします。
7. 4V FMB の SDB データベースの場合は tree を指定します。
8. ルートレコードのレコード型名を指定します。
9. 初期データロード対象の RD エリアがあるバックエンドサーバを指定します。
10. 初期データロード対象の RD エリアを指定します。
11. 入力データファイル名を指定します。
12. 実行結果ファイル名を指定します。

13. 論理エラー情報ファイル名を指定します。

14. プリフィクス部の長さを指定します。

15. インデクス一括作成モードでインデクスを作成します。

16. ページ切り替えフラグの開始位置を指定します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB1
    DBTYPE 4V FMB
    RECORD FMB1_RCA
      2 DBKEY
       3  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  USERDA0             CHARACTER 30   TYPE  U,D
    RECORD FMB1_RCB
      2  KEYDATA1            XCHARACTER 1   TYPE  K,A
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDB0             CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB1_RCA
        ORDER LAST
      MEMBER FMB1_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SDBOPTION
      FUNCTION
        REFER     USE
        ADD       USE
        UPDATE    USE
        ERASE     USE
        ALLERASE  NOUSE
        DBLODUTL  USE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB1 FOR FMB1
  DBTYPE 4V
  STORAGE RECORD FMB1_RCA
    SEQUENTIAL FMB1_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN ((RDAREA015),(RDAREA016),(RDAREA017),(RDAREA018))
    WITHIN ((RDAREA011) KEYDATA1=(X'F3'),
            (RDAREA012) KEYDATA1=(X'F4'),
            (RDAREA013) KEYDATA1=(X'F5'),
            (RDAREA014) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB1_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA011),(RDAREA012),(RDAREA013),(RDAREA014))
  SET SETAB
    OWNER POINTER FOR FIRST LAST USER MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SDBOPTION
  END STORAGE SCHEMA
```

```
*ENTRY DIRECTORY
  DBSCHM FMB1
```

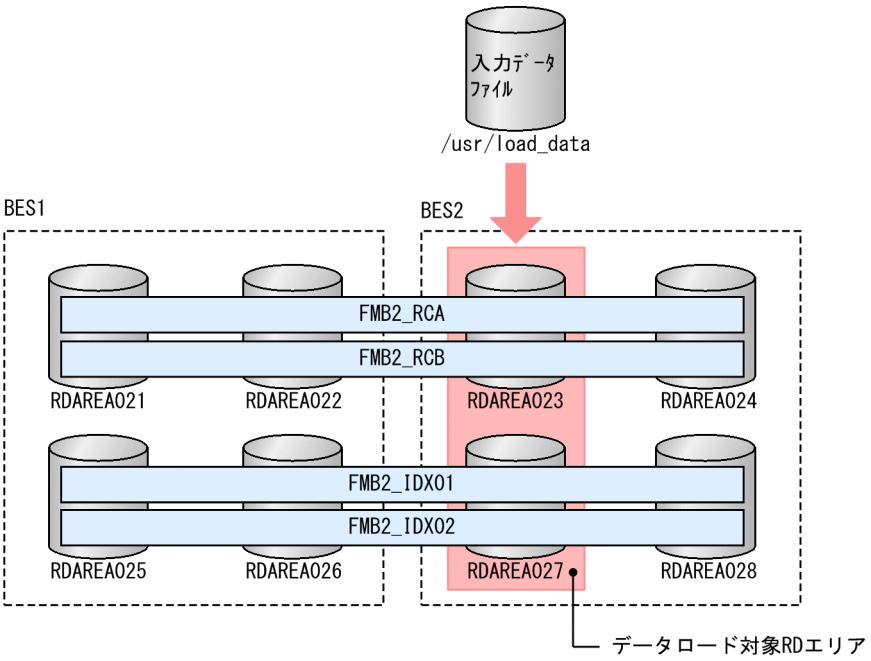
12.10.2　初期データロードの例（4V FMB の SDB データベースに二次インデクスを定義している場合）

例題

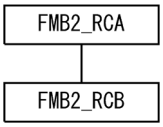
横分割されている SDB データベースに対して，RD エリア単位の初期データロードを実行します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V FMB です。
- インデクス FMB2_IDX01 および FMB2_IDX02 が定義されています。
- 初期データロードの対象 RD エリアは RDAREA023 および RDAREA027 です。
- ページ切り替えフラグを設定します。

(2) 初期データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema    = FMB2             ...1
  logmode   = no               ...2
  purge     = yes              ...3
  exectime  = 0                ...4
  recnomsg  = 1                ...5
  divermsg  = put              ...6
load
  type      = tree             ...7
  record    = FMB2_RCA         ...8
  server    = BES2             ...9
  area      = RDAREA023        ...10
  data      = /usr/load_data   ...11
  storinf   = /usr/load_result ...12
  errdata   = /usr/err_data    ...13
  prefix    = 1                ...14
  idxmode   = create           ...15
dbinf
  pagecflg  = (0)              ...16
```

[説明]

1. SDB データベース名を指定します。
2. ログレスモードで初期データロードを実行します。
3. 既存レコードを削除して初期データロードを実行します。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. 初期データロードが 1 万件完了するごとに経過メッセージを出力します。
6. ほかの RD エリアに格納するべきレコードは論理エラーとします。
7. 4V FMB の SDB データベースの場合は tree を指定します。
8. ルートレコードのレコード型名を指定します。
9. 初期データロード対象の RD エリアがあるバックエンドサーバを指定します。
10. 初期データロード対象の RD エリアを指定します。
11. 入力データファイル名を指定します。
12. 実行結果ファイル名を指定します。
13. 論理エラー情報ファイル名を指定します。
14. プリフィクス部の長さを指定します。

15. インデクス一括作成モードでインデクスを作成します。

16. ページ切り替えフラグの開始位置を指定します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB2
    DBTYPE 4V FMB
    RECORD FMB2_RCA
      2 DBKEY
       3  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  USERDA0             CHARACTER 30   TYPE  U,D
    RECORD FMB2_RCB
      2  KEYDATA1            XCHARACTER 1   TYPE  K,A
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDB0             CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB2_RCA
        ORDER LAST
      MEMBER FMB2_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SDBOPTION
      FUNCTION
        REFER     USE
        ADD       USE
        UPDATE    USE
        ERASE     USE
        ALLERASE  NOUSE
        DBLODUTL  USE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB2 FOR FMB2
  DBTYPE 4V
  STORAGE RECORD FMB2_RCA
    SEQUENTIAL FMB2_IDX01 FOR RECORD
      ORDER KEY ASCENDING KEYDATA1
      WITHIN ((RDAREA025),(RDAREA026),(RDAREA027),(RDAREA028))
    WITHIN ((RDAREA021) KEYDATA1=(X'F3'),
            (RDAREA022) KEYDATA1=(X'F4'),
            (RDAREA023) KEYDATA1=(X'F5'),
            (RDAREA024) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB2_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA021),(RDAREA022),(RDAREA023),(RDAREA024))
    SECONDARY INDEX FMB1_IDX02
      ORDER KEY ASCENDING KEYDATA1,DBKEY
      WITHIN ((RDAREA025),(RDAREA026),(RDAREA027),(RDAREA028))
  SET SETAB
    OWNER POINTER FOR FIRST LAST USER MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SDBOPTION
  END STORAGE SCHEMA
```

```
*ENTRY DIRECTORY
  DBSCHM FMB2
```

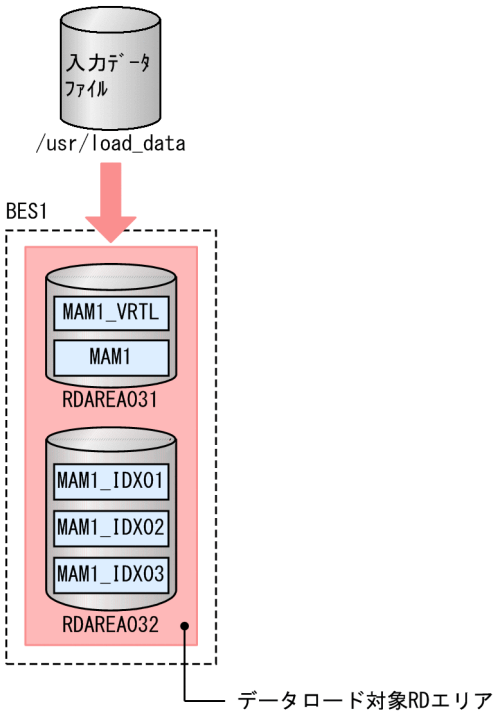
12.10.3　初期データロードの例（4V MAM の場合）

例題

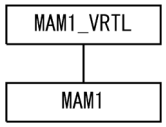
SDB データベースの初期データロードを実行します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V MAM です。
- インデクス MAM1_IDX01～MAM1_IDX03 が定義されています。

(2) 初期データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = MAM1              ...1
  logmode  = no                ...2
  purge    = yes               ...3
  exectime = 0                 ...4
  recnomsg = 1                 ...5
load
  type     = single            ...6
  record   = MAM1_VRTL         ...7
  server   = BES1              ...8
  data     = /usr/load_data    ...9
  storinf  = /usr/load_result  ...10
  errdata  = /usr/err_data     ...11
  prefix   = 0                 ...12
  idxmode  = create            ...13
```

[説明]

1. SDB データベース名を指定します。
2. ログレスモードで初期データロードを実行します。
3. 既存レコードを削除して初期データロードを実行します。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. 初期データロードが 1 万件完了するごとに経過メッセージを出力します。
6. 4V AFM の SDB データベースの場合は single を指定します。
7. 仮想ルートレコードのレコード型名を指定します。
8. 初期データロード対象の RD エリアがあるバックエンドサーバを指定します。
9. 入力データファイル名を指定します。
10. 実行結果ファイル名を指定します。
11. 論理エラー情報ファイル名を指定します。
12. プリフィクス部の長さを指定します。
13. インデクス一括作成モードでインデクスを作成します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA MAM1
```

```
   DBTYPE 4V  MAM
   RECORD MAM1_VRTL
    02 DBKEY01 CHARACTER 4         TYPE D,L
    02 DBKEY02 CHARACTER 4         TYPE K,L
   RECORD MAM1
    02 DBKEY01     CHARACTER 4     TYPE D,L
    02 DBKEY
     03 DBKEY02    CHARACTER 4     TYPE K,L
     03 DBKEYKN    INTEGER         TYPE K,N
    02 SORTKEY1 XCHARACTER 6       TYPE U,K,X'0000'
    02 USERFLD1 XCHARACTER 3       TYPE U,D,X'0000'
    02 USERFLD2 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD3 XCHARACTER 3       TYPE U,D,X'0000'
    02 USERFLD4 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD5 XCHARACTER 2       TYPE U,D,X'0000'
    02 USERFLD6 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD7 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD8 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD9 XCHARACTER 5       TYPE U,D,X'0000'
    02 USRDATA1 XCHARACTER 416     TYPE U,D,X'0000'
    FUNCTION
     REFER       USE
     ADD         USE
     UPDATE      USE
     ERASE     NOUSE
     ALLERASE  NOUSE
    RECORDID X'0001'
   SET SETM1
    OWNER MAM1_VRTL
     ORDER SORTED DUPLICATES PROHIBITED
    MEMBER MAM1
    INSERTION AUTOMATIC
    RETENTION FIXED
     KEY  ASCENDING  SORTKEY1
    SETOPTION
     OCCURRENCE NUMBER 65535
   REUSE NO
   SDBOPTION
     FUNCTION
      DBLODUTL    USE
      FORMAT    NOUSE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA  MAM1  FOR  MAM1
   DBTYPE 4V
   STORAGE RECORD MAM1_VRTL
    SEQUENTIAL  MAM1_IDX01  FOR RECORD
     ORDER KEY ASCENDING  DBKEY01,DBKEY02
     WITHIN RDAREA032
    WITHIN RDAREA031
   STORAGE RECORD MAM1
    CLUSTERED SETM1
     WITHIN RDAREA031
     POINTER AREA SIZE 0
     SECONDARY INDEX MAM1_IDX02
       ORDER KEY ASCENDING  DBKEY01,DBKEY02,DBKEYKN
       WITHIN RDAREA032
     SECONDARY INDEX MAM1_IDX03
```

```
        ORDER KEY ASCENDING  DBKEY01,DBKEY02,SORTKEY1
        WITHIN RDAREA032
    SET SETM1
     OWNER POINTER FOR FIRST LAST MEMBER
     MEMBER POINTER FOR NEXT PRIOR TENANT
    SDBOPTION
      KEYDEF DBKEY01
       DATA 'MAM1'
      KEYDEF DBKEY02
       DATA 'AAAA'
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM MAM1
```

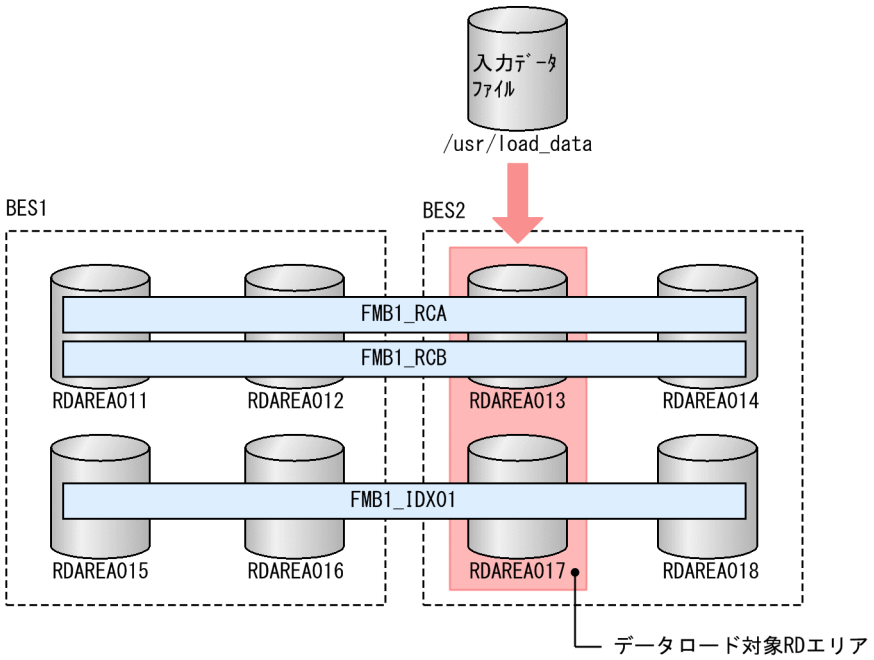
12.10.4　初期データロードの例（SD FMB の場合）

例題

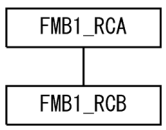
横分割されている SDB データベースに対して，RD エリア単位の初期データロードを実行します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は SD FMB です。
- インデクス FMB1_IDX01 が定義されています。
- 初期データロードの対象 RD エリアは RDAREA013 および RDAREA017 です。

(2) 初期データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = FMB1              ...1
  logmode  = no                ...2
  purge    = yes               ...3
  exectime = 0                 ...4
  recnomsg = 1                 ...5
  divermsg = put               ...6
load
  type     = tree              ...7
  record   = FMB1_RCA          ...8
  server   = BES2              ...9
  area     = RDAREA013         ...10
  data     = /usr/load_data    ...11
  storinf  = /usr/load_result  ...12
  errdata  = /usr/err_data     ...13
  prefix   = 0                 ...14
  idxmode  = create            ...15
```

[説明]

1. SDB データベース名を指定します。
2. ログレスモードで初期データロードを実行します。
3. 既存レコードを削除して初期データロードを実行します。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. データロードが 1 万件完了するごとに経過メッセージを出力します。
6. ほかの RD エリアに格納するべきレコードは論理エラーとします。
7. SD FMB の SDB データベースの場合は tree を指定します。
8. ルートレコードのレコード型名を指定します。
9. 初期データロード対象の RD エリアがあるバックエンドサーバを指定します。
10. 初期データロード対象の RD エリアを指定します。
11. 入力データファイル名を指定します。

12. 実行結果ファイル名を指定します。

13. 論理エラー情報ファイル名を指定します。

14. プリフィクス部の長さに 0 を指定します。

15. インデクス一括作成モードでインデクスを作成します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB1
    DBTYPE SD FMB
    RECORD FMB1_RCA
      2 KEYDATA
       3  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  USERDA0             CHARACTER 30   TYPE  U,D
    RECORD FMB1_RCB
      2  KEYDATA1            XCHARACTER 1   TYPE  K,A
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDB0             CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB1_RCA
        ORDER LAST
      MEMBER FMB1_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB1 FOR FMB1
  DBTYPE SD
  STORAGE RECORD FMB1_RCA
    SEQUENTIAL FMB1_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN ((RDAREA015),(RDAREA016),(RDAREA017),(RDAREA018))
    WITHIN ((RDAREA011) KEYDATA1=(X'F3'),
            (RDAREA012) KEYDATA1=(X'F4'),
            (RDAREA013) KEYDATA1=(X'F5'),
            (RDAREA014) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB1_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA011),(RDAREA012),(RDAREA013),(RDAREA014))
  SET SETAB
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM FMB1
```

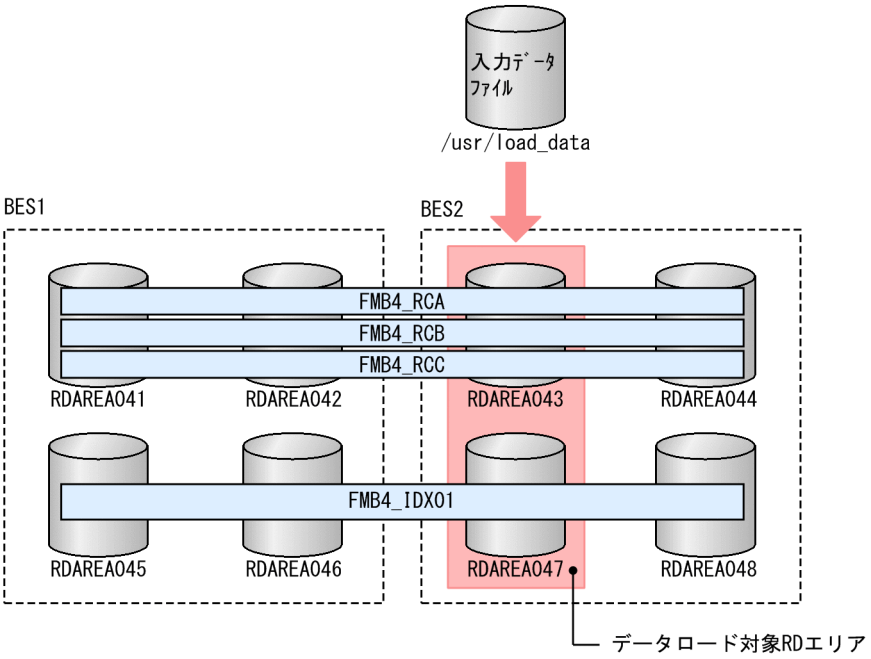
12.10.5 追加データロードの例（4V FMB の場合）

例題

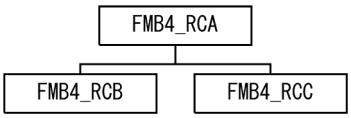
横分割されている SDB データベースに対して，RD エリア単位の追加データロードを実行します。

(1) 構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V FMB です。
- インデクス FMB4_IDX01 が定義されています。
- 追加データロードの対象 RD エリアは RDAREA043 および RDAREA047 です。
- 事前割り当てページ数を設定します。

(2) 追加データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = FMB4              ...1
  logmode  = no                ...2
  purge    = no                ...3
```

```
  exectime = 0                ...4
 load
  type     = tree             ...5
  record   = FMB4_RCA         ...6
  server   = BES2             ...7
  area     = RDAREA043        ...8
  data     = /usr/load_data   ...9
  storinf  = /usr/load_result ...10
  errdata  = /usr/err_data    ...11
  prefix   = 4                ...12
 dbinf
  pagenum  = (0)              ...13
```

[説明]

1. SDB データベース名を指定します。
2. ログレスモードで追加データロードを実行します。
3. 既存レコードを削除しないで追加データロードを実行します。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. 4V FMB の SDB データベースの場合は tree を指定します。
6. ルートレコードのレコード型名を指定します。
7. 追加データロード対象の RD エリアがあるバックエンドサーバを指定します。
8. 追加データロード対象の RD エリアを指定します。
9. 入力データファイル名を指定します。
10. 実行結果ファイル名を指定します。
11. 論理エラー情報ファイル名を指定します。
12. プリフィクス部の長さを指定します。
13. 事前割り当てページ数の開始位置を指定します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB4
    DBTYPE 4V FMB
    RECORD FMB4_RCA
      2 DBKEY
       3  KEYDATA1          XCHARACTER 1   TYPE  K,A
      2  USERDA0            CHARACTER 30   TYPE  U,D
    RECORD FMB4_RCB
      2  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  DBKEY              INTEGER        TYPE  K,N
      2  USERDB0            CHARACTER 20   TYPE  U,D
    RECORD FMB4_RCC
      2  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  DBKEY              INTEGER        TYPE  K,N
      2  USERDC0            CHARACTER 20   TYPE  U,D
```

```
    SET SETAB
      OWNER FMB4_RCA
        ORDER LAST
      MEMBER FMB4_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SET SETAC
      OWNER FMB4_RCA
        ORDER LAST
      MEMBER FMB4_RCC
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SDBOPTION
      FUNCTION
        REFER     USE
        ADD       USE
        UPDATE    USE
        ERASE     USE
        ALLERASE  NOUSE
        DBLODUTL  USE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB4 FOR FMB4
  DBTYPE 4V
  STORAGE RECORD FMB4_RCA
    SEQUENTIAL FMB4_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN ((RDAREA045),(RDAREA046),(RDAREA047),(RDAREA048))
    WITHIN ((RDAREA041) KEYDATA1=(X'F3'),
            (RDAREA042) KEYDATA1=(X'F4'),
            (RDAREA043) KEYDATA1=(X'F5'),
            (RDAREA044) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB4_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA041),(RDAREA042),(RDAREA043),(RDAREA044))
  STORAGE RECORD FMB4_RCC
    CLUSTERED SETAC
      WITHIN ((RDAREA041),(RDAREA042),(RDAREA043),(RDAREA044))
  SET SETAB
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SET SETAC
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SDBOPTION
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM FMB4
```

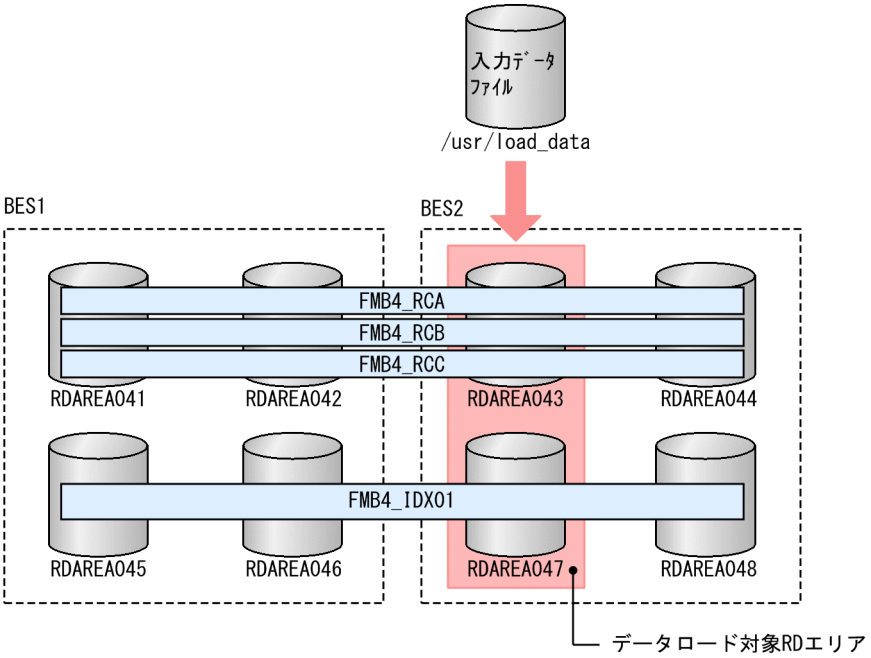
12.10.6　追加データロードの例（SD FMB の場合）

例題

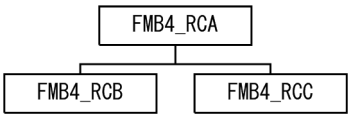
横分割されている SDB データベースに対して，RD エリア単位の追加データロードを実行します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は SD FMB です。
- インデクス FMB4_IDX01 が定義されています。
- 追加データロードの対象 RD エリアは RDAREA043 および RDAREA047 です。

(2)　追加データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = FMB4              ...1
  logmode  = no                ...2
  purge    = no                ...3
  exectime = 0                 ...4
load
  type     = tree              ...5
  record   = FMB4_RCA          ...6
  server   = BES2              ...7
  area     = RDAREA043         ...8
  data     = /usr/load_data    ...9
  storinf  = /usr/load_result  ...10
  errdata  = /usr/err_data     ...11
  prefix   = 0                 ...12
```

[説明]

1. SDB データベース名を指定します。
2. ログレスモードで追加データロードを実行します。
3. 既存レコードを削除しないで追加データロードを実行します。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. SD FMB の SDB データベースの場合は tree を指定します。
6. ルートレコードのレコード型名を指定します。
7. 追加データロード対象の RD エリアがあるバックエンドサーバを指定します。
8. 追加データロード対象の RD エリアを指定します。
9. 入力データファイル名を指定します。
10. 実行結果ファイル名を指定します。
11. 論理エラー情報ファイル名を指定します。
12. プリフィクス部の長さに 0 を指定します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB4
    DBTYPE SD FMB
    RECORD FMB4_RCA
      2 KEYDATA
       3  KEYDATA1          XCHARACTER 1   TYPE  K,A
      2  USERDA0            CHARACTER 30   TYPE  U,D
    RECORD FMB4_RCB
      2  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  DBKEY              INTEGER        TYPE  K,N
      2  USERDB0            CHARACTER 20   TYPE  U,D
    RECORD FMB4_RCC
      2  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  DBKEY              INTEGER        TYPE  K,N
      2  USERDC0            CHARACTER 20   TYPE  U,D
```

```
    SET SETAB
      OWNER FMB4_RCA
        ORDER LAST
      MEMBER FMB4_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SET SETAC
      OWNER FMB4_RCA
        ORDER LAST
      MEMBER FMB4_RCC
        INSERTION AUTOMATIC
        RETENTION FIXED
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB4 FOR FMB4
  DBTYPE SD
  STORAGE RECORD FMB4_RCA
    SEQUENTIAL FMB4_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN ((RDAREA045),(RDAREA046),(RDAREA047),(RDAREA048))
    WITHIN ((RDAREA041) KEYDATA1=(X'F3'),
            (RDAREA042) KEYDATA1=(X'F4'),
            (RDAREA043) KEYDATA1=(X'F5'),
            (RDAREA044) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB4_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA041),(RDAREA042),(RDAREA043),(RDAREA044))
  STORAGE RECORD FMB4_RCC
    CLUSTERED SETAC
      WITHIN ((RDAREA041),(RDAREA042),(RDAREA043),(RDAREA044))
  SET SETAB
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SET SETAC
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM FMB4
```

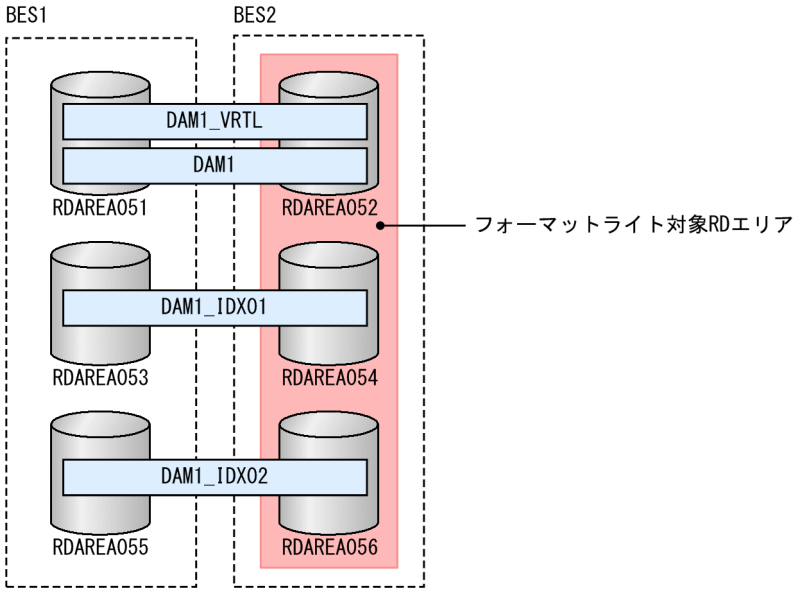
12.10.7　フォーマットライトの例（4V DAM の場合）

例題

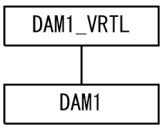
横分割されている SDB データベースに対して，バックエンドサーバ単位のフォーマットライトを実行します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V DAM です。
- インデクス DAM1_IDX01～DAM1_IDX02 が定義されています。
- フォーマットライトの対象 RD エリアは BES2 の RD エリアです。

(2) フォーマットライトの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = DAM1              ...1
  logmode  = no                ...2
  purge    = yes               ...3
  exectime = 0                 ...4
  recnomsg = 1                 ...5
load
  write    = use               ...6
  record   = DAM1_VRTL         ...7
  server   = BES2              ...8
  storinf  = /usr/load_result  ...9
  idxmode  = create            ...10
```

[説明]

1. SDB データベース名を指定します。
2. ログレスモードでフォーマットライトを実行します。
3. 既存レコードを削除してフォーマットライトを実行します。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. フォーマットライトが 1 万件完了するごとに経過メッセージを出力します。
6. フォーマットライトを実行する指定です。
7. 仮想ルートレコードのレコード型名を指定します。
8. フォーマットライト対象の RD エリアがあるバックエンドサーバを指定します。
9. 実行結果ファイル名を指定します。
10. インデクス一括作成モードでインデクスを作成します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA DAM1
    DBTYPE 4V DAM
    RECORD DAM1_VRTL
      2 KEYDBMEI          CHARACTER 4    TYPE  D,L
      2 KEYHIDUKE         XCHARACTER 1   TYPE  K,A
    RECORD DAM1
      2 KEYDBMEI          CHARACTER 4    TYPE  D,L
      2 DBKEY
       3 KEYHIDUKE        XCHARACTER 1   TYPE  K,A
       3 KEYNUMBER        INTEGER        TYPE  K,N
      2 USERDB0           CHARACTER 20   TYPE  U,D
    SET SETD1
      OWNER DAM1_VRTL
        ORDER LAST
      MEMBER DAM1
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 10
  SDBOPTION
    FUNCTION
      DBLODUTL  USE
      FORMAT    USE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA DAM1 FOR DAM1
    DBTYPE 4V
  STORAGE RECORD DAM1_VRTL
    SEQUENTIAL DAM1_IDX01 FOR RECORD
      ORDER KEY ASCENDING KEYDBMEI,KEYHIDUKE
     WITHIN  ((RDAREA053),(RDAREA054))
    WITHIN ((RDAREA051) KEYHIDUKE=(X'C1'),
            (RDAREA052) KEYHIDUKE=(X'C2'))
```

```
  STORAGE RECORD DAM1
    CLUSTERED SETD1
     WITHIN ((RDAREA051),(RDAREA052))
  SECONDARY INDEX DAM1_IDX02
  ORDER KEY ASCENDING
      KEYDBMEI,KEYHIDUKE, KEYNUMBER
      WITHIN  ((RDAREA055),(RDAREA056))
  SET SETD1
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SDBOPTION
      KEYDEF KEYDBMEI
        DATA 'DAM1'
      KEYDEF KEYHIDUKE
        DATA X'C1'
        DATA X'C2'
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM DAM1
```

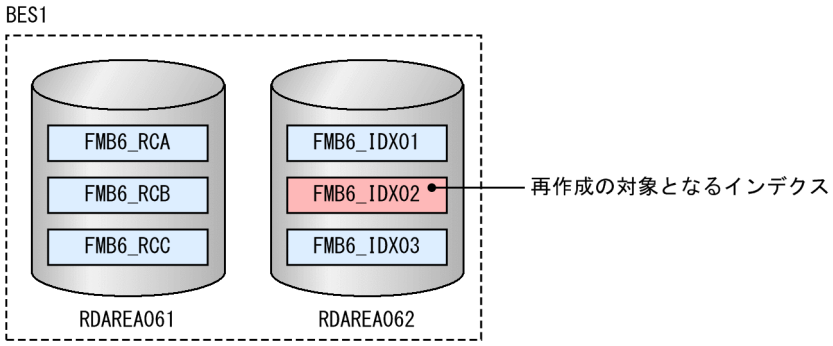
12.10.8　インデクスの再作成の例（4V FMB の場合）

例題

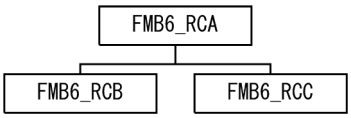
誤ってインデクス格納用 RD エリアを初期化してしまったため，インデクス格納用 RD エリア単位に，インデクスを再作成します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V FMB です。

- FMB6_RCA には，シーケンシャルインデクス FMB6_IDX01 が定義されています。
- FMB6_RCB には，二次インデクス FMB6_IDX02 が定義されています。
- FMB6_RCC には，二次インデクス FMB6_IDX03 が定義されています。
- インデクスの再作成の対象となるインデクスは，FMB6_IDX02 です。
- FMB6_IDX02 が格納されている RD エリアは RDAREA062 です。
- インデクス情報を抽出するデータが格納されている RD エリアは RDAREA061 です。

(2) インデクス再作成の実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema    = FMB6              ...1
  logmode   = all               ...2
index
  idxremode = recrt             ...3
  idxarea   = RDAREA062         ...4
  idxname   = FMB6_IDX02        ...5
```

[説明]

1. SDB データベース名を指定します。
2. ログ取得モードで実行します。
3. インデクス再作成モードで実行します。
4. インデクスの再作成対象となる RD エリアを指定します。
5. インデクスの再作成対象となる RD エリアに格納されているインデクス名を指定します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB6
    DBTYPE 4V FMB
    RECORD FMB6_RCA
      2 DBKEY
       3  KEYDATA1           XCHARACTER 1   TYPE  K,L
      2  USERDA0             CHARACTER 30   TYPE  U,D
    RECORD FMB6_RCB
      2  KEYDATA1            XCHARACTER 1   TYPE  K,L
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDB0             CHARACTER 20   TYPE  U,D
    RECORD FMB6_RCC
      2  KEYDATA1            XCHARACTER 1   TYPE  K,L
```

```
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDC0             CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB6_RCA
        ORDER LAST
      MEMBER FMB6_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SET SETAC
      OWNER FMB6_RCA
        ORDER LAST
      MEMBER FMB6_RCC
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SDBOPTION
      FUNCTION
        REFER     USE
        ADD       USE
        UPDATE    USE
        ERASE     USE
        ALLERASE  NOUSE
        DBLODUTL  USE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB6 FOR FMB6
  DBTYPE 4V
  STORAGE RECORD FMB6_RCA
    SEQUENTIAL FMB6_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN (RDAREA062)
    WITHIN (RDAREA061)
  STORAGE RECORD FMB6_RCB
    CLUSTERED SETAB
      WITHIN (RDAREA061)
    SECONDARY INDEX FMB6_IDX02
      ORDER KEY ASCENDING  KEYDATA1,DBKEY
     WITHIN RDAREA062
  STORAGE RECORD FMB6_RCC
    CLUSTERED SETAC
      WITHIN (RDAREA061)
    SECONDARY INDEX FMB6_IDX03
      ORDER KEY ASCENDING  KEYDATA1,DBKEY
     WITHIN RDAREA062
  SET SETAB
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SET SETAC
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SDBOPTION
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM FMB6
```

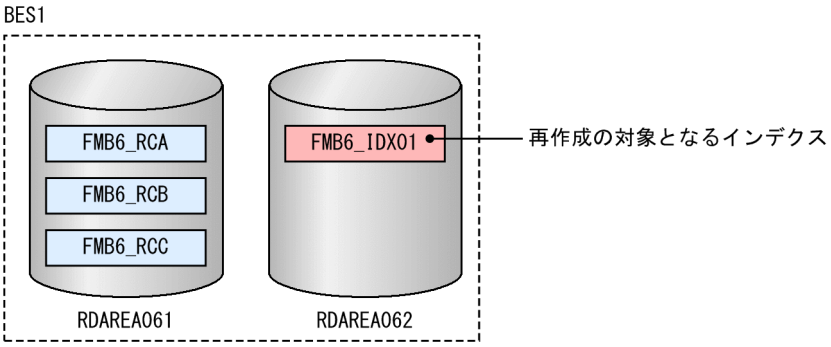
12.10.9　インデクスの再作成の例（SD FMB の場合）

例題

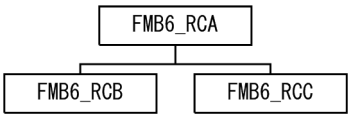
誤ってインデクス格納用 RD エリアを初期化してしまったため，インデクス格納用 RD エリア単位に，インデクスを再作成します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は SD FMB です。
- FMB6_RCA には，シーケンシャルインデクス FMB6_IDX01 が定義されています。
- インデクスの再作成の対象となるインデクスは，FMB6_IDX01 です。
- FMB6_IDX01 が格納されている RD エリアは RDAREA062 です。
- インデクス情報を抽出するデータが格納されている RD エリアは RDAREA061 です。

(2)　インデクス再作成の実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema    = FMB6              ...1
  logmode   = all               ...2
```

```
 index
   idxremode = recrt            ...3
   idxarea   = RDAREA062        ...4
   idxname   = FMB6_IDX01       ...5
```

[説明]

1. SDB データベース名を指定します。
2. ログ取得モードで実行します。
3. インデクス再作成モードで実行します。
4. インデクスの再作成対象となる RD エリアを指定します。
5. インデクスの再作成対象となる RD エリアに格納されているインデクス名を指定します。

(3) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB6
    DBTYPE SD FMB
    RECORD FMB6_RCA
      2 KEYDATA
       3  KEYDATA1           XCHARACTER 1   TYPE  K,L
      2  USERDA0             CHARACTER 30   TYPE  U,D
    RECORD FMB6_RCB
      2  KEYDATA1            XCHARACTER 1   TYPE  K,L
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDB0             CHARACTER 20   TYPE  U,D
    RECORD FMB6_RCC
      2  KEYDATA1            XCHARACTER 1   TYPE  K,L
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDC0             CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB6_RCA
        ORDER LAST
      MEMBER FMB6_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
    SET SETAC
      OWNER FMB6_RCA
        ORDER LAST
      MEMBER FMB6_RCC
        INSERTION AUTOMATIC
        RETENTION FIXED
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB6 FOR FMB6
  DBTYPE SD
  STORAGE RECORD FMB6_RCA
    SEQUENTIAL FMB6_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN (RDAREA062)
    WITHIN (RDAREA061)
  STORAGE RECORD FMB6_RCB
    CLUSTERED SETAB
```

```
      WITHIN (RDAREA061)
  STORAGE RECORD FMB6_RCC
    CLUSTERED SETAC
      WITHIN (RDAREA061)
  SET SETAB
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SET SETAC
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM FMB6
```

13 HiRDB/SD データベース再編成ユティリティ (pdsdbrog)

この章では，HiRDB/SD データベース再編成ユティリティの機能と使い方について説明します。

なお，この章では，HiRDB/SD データベース再編成ユティリティを pdsdbrog コマンドと表記しています。

13.1　機能

pdsdbrog コマンドには次に示す機能があります。

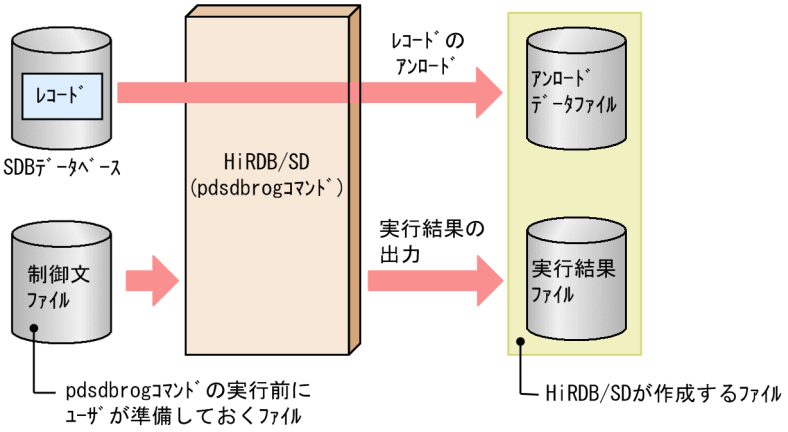
- レコードのアンロード

この機能について説明します。

13.1.1　レコードのアンロード

データベースに格納されているレコードをアンロードデータファイルに出力できます。この機能をレコードのアンロードといいます。レコードのアンロードの概要を次の図に示します。

図 13–1　レコードのアンロードの概要



(1)　アンロードの際に準備するファイル

アンロードを実行する前に，次に示すファイルを準備しておきます。

- 制御文ファイル

  pdsdbrog コマンドの動作条件を指定した pdsdbrog 制御文を格納したファイルです。制御文ファイルは通常ファイル上に作成してください。pdsdbrog 制御文については，「13.3　pdsdbrog 制御文」を参照してください。

(2)　アンロードの際に作成されるファイル

アンロードの際，HiRDB/SD は次に示すファイルを作成します。これらのファイルは通常ファイル上に作成されます。

- アンロードデータファイル

  データベース中のレコードを格納したファイルです。アンロードデータファイルの出力形式については，「13.6　アンロードデータファイルの出力形式」を参照してください。

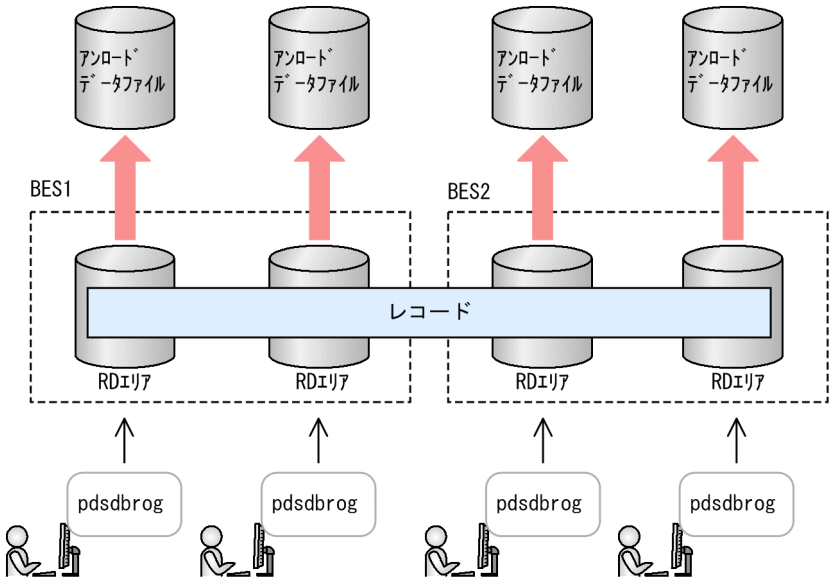
- アンロードの実行結果ファイル

  pdsdbrog コマンドの実行結果を出力するファイルです。実行結果ファイルに出力される情報については，「13.7　アンロードの実行結果ファイルの出力形式」を参照してください。

(3)　SDB データベースが横分割されている場合のアンロード

SDB データベースが複数の RD エリアに横分割されている場合，RD エリア単位にアンロードを実行します。例えば，4 つの RD エリアに SDB データベースが横分割されている場合，アンロードを 4 回実行します。SDB データベースが横分割されている場合のアンロードを次の図に示します。

図 13–2　SDB データベースが横分割されている場合のアンロード



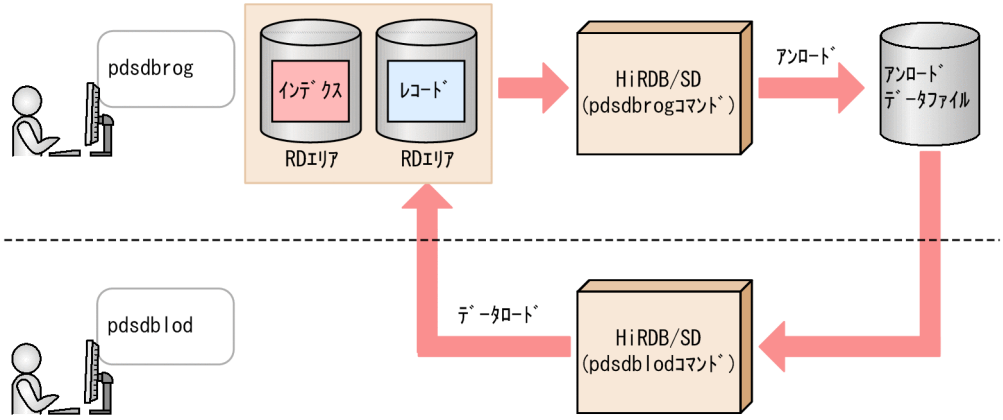
(4)　アンロードの主な利用方法

アンロードの主な利用方法を次に示します。

(a)　SDB データベースを再編成するときに利用する

レコードの追加や削除を繰り返すと，レコードの配置が乱れてレコードのアクセス効率や格納効率が低下します。このような場合，pdsdbrog コマンドでレコードをいったんアンロードしてアンロードデータファイルを作成します。そのアンロードデータファイルを入力データファイルにして pdsdblod コマンドでデータロードを行います。レコードを再格納することによって，レコードの配置が適正化され，レコードのアクセス効率や格納効率が改善されます。この一連の処理を SDB データベースの再編成といいます。SDB データベースの再編成の概要を次の図に示します。

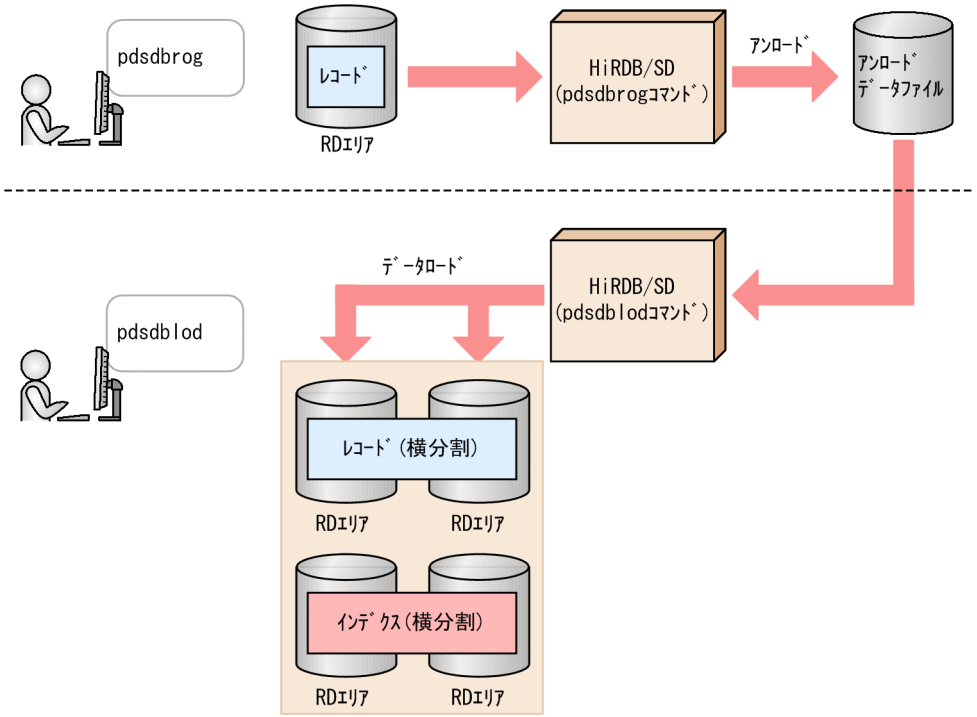
図 13-3　SDB データベースの再編成の概要



(b)　既存の SDB データベースを横分割するときに利用する

既存の SDB データベースを横分割する場合，pdsdbrog コマンドでレコードをいったんアンロードしてアンロードデータファイルを作成します。レコード型の定義を変更したあとに，アンロードデータファイルを入力データファイルにして pdsdblod コマンドでデータロードを行います。既存の SDB データベースを横分割する場合の流れを次の図に示します。

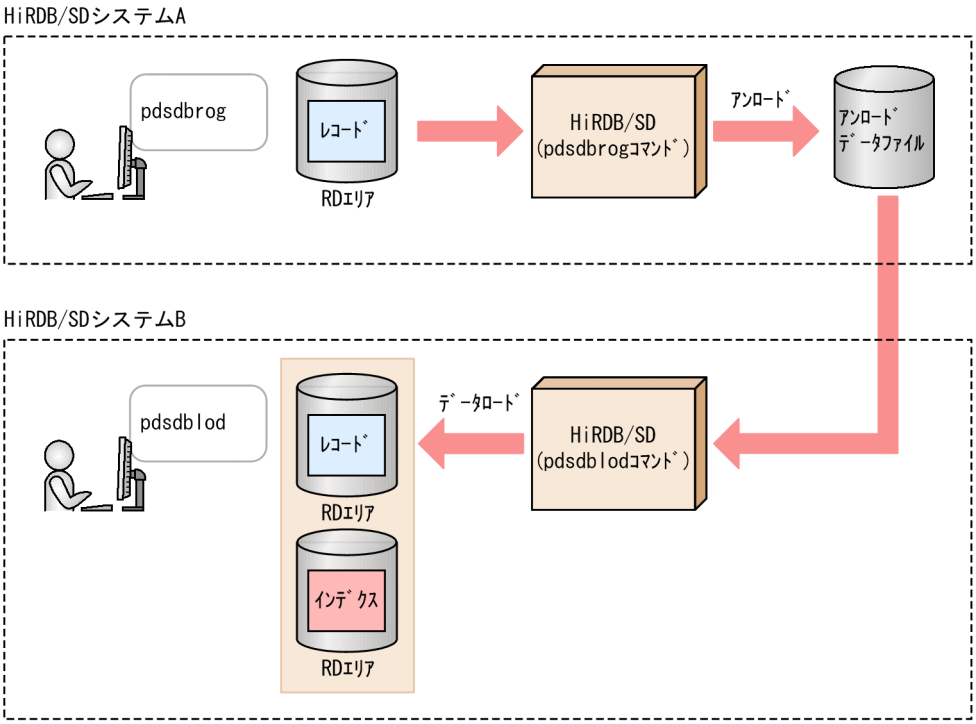
図 13-4　既存の SDB データベースを横分割する場合の流れ



(c)　ほかの HiRDB システムに SDB データベースを移行するときに利用する

ほかの HiRDB システムに SDB データベースを移行する場合，pdsdbrog コマンドでレコードをいったんアンロードしてアンロードデータファイルを作成します。そのアンロードデータファイルを入力データファイルにして pdsdblod コマンドでほかの HiRDB システムのデータベースにデータロードを行います。ほかの HiRDB システムに SDB データベースを移行する場合の流れを次の図に示します。

図 13-5　ほかの HiRDB システムに SDB データベースを移行する場合の流れ



なお，ほかの HiRDB システムに SDB データベースを移行する場合は，次に示す条件をすべて満たす必要があります。

- 移行元と移行先で，バイトオーダ，文字コード，HiRDB のバージョン，および HiRDB のアドレッシングモードが同じであること
- 移行元と移行先の SDB データベース定義および SDB データベース格納定義が同じであること※

  注※　次に示す条件を除きます。
  - RD エリア名および RD エリア数（シーケンシャルインデクス用，格納レコード用，二次インデクス用）
  - SDB データベースの横分割条件

上記の条件を満たしていない場合，移行先の HiRDB システムで SDB データベースを正しく格納できません。ただし，移行先の HiRDB システムでデータロードする際に，論理エラーのチェックが行われるため，論理エラーと判定されたレコードは格納されません。

参考

HiRDB システム内でデータを移行することはできません。

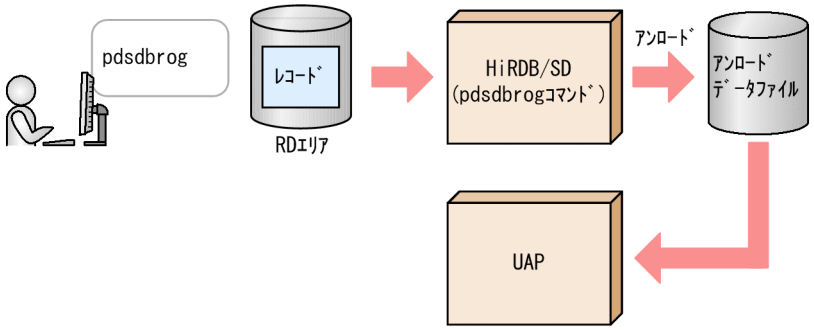
(d)　アンロードデータファイルを UAP の入力情報として利用する

アンロードデータファイルのプリフィックス部に，データベースキー値，キーのサイズ，およびユーザデータサイズなどの情報を出力できます。プリフィックス部に出力されるこれらの情報を TMS-4V/SP 情報といいます。

アンロードデータファイルを UAP の入力情報として利用する場合などは，TMS-4V/SP 情報を使用してください。

アンロードデータファイルを UAP の入力情報として利用する場合の流れを次の図に示します。

図 13-6　アンロードデータファイルを UAP の入力情報として利用する場合の流れ



注

ここで説明している運用方法は，TMS-4V/SP から移行した UAP を使用することを想定した運用方法です。

アンロードデータファイルのプリフィックス部に TMS-4V/SP 情報を出力するかどうかは，unload 文の unldkind オペランドで指定します。

また，4V DAM の SDB データベースの場合は，afmtype オペランドで出力形式を選択できます。

13.2　コマンドの形式

```
pdsdbrog 制御文ファイル名
```

制御文ファイル名：

pdsdbrog 制御文を指定した制御文ファイルの名称を絶対パスまたは相対パスで指定します。pdsdbrog 制御文については，次の「13.3　pdsdbrog 制御文」で説明します。

13.3　pdsdbrog 制御文

pdsdbrog 制御文には，pdsdbrog コマンドの動作条件を指定します。pdsdbrog 制御文は，environment 文，unload 文，および dbinf 文で構成されています。ここでは，各文の指定形式と各文に指定するオペランドについて説明します。

注意事項

- pdsdbrog 制御文は environment 文，unload 文，dbinf 文の順に指定してください。また，environment 文と unload 文は必ず指定してください。
- pdsdbrog 制御文はテキストファイルに作成します。テキストファイルを保存する際，pdsetup コマンド実行時に指定した文字コード種別で保存してください。
- 環境変数 LANG またはクライアント環境定義の PDLANG オペランドで，使用する文字コードに UTF-8 を指定した場合，pdsdbrog 制御文を記述したファイルには BOM を持つファイルが指定できます（BOM を読み飛ばします）。
- pdsdbrog 制御文中にコメントは記述できません。

13.3.1　environment 文

environment 文には，pdsdbrog コマンドの基本的な動作条件を指定します。

指定形式

```
environment
      schema = SDBデータベース名
     〔exectime = pdsdbrogコマンドの実行時間の上限〕
     〔recnomsg = 処理経過メッセージの出力間隔〕
     〔generation = 世代番号〕
```

●schema ＝ SDB データベース名

～＜識別子＞((1～30 バイト))

アンロードを実行する SDB データベースの名称を指定します。SDB データベース定義の SCHEMA 句に指定した SDB データベース名を指定します。

SDB データベース名を引用符（''）で囲んだ場合，英大文字と英小文字が区別されます。引用符（''）で囲まない場合，すべて英大文字として扱われます。

●exectime ＝ pdsdbrog コマンドの実行時間の上限

～＜符号なし整数＞((0～35,791,394))（単位：分）

pdsdbrog コマンドの実行時間の上限を分単位で指定します。このオペランドに指定した時間を超えても pdsdbrog コマンドの処理が完了しない場合，pdsdbrog コマンドを強制終了します。このとき，無応答障害の原因調査のための障害情報が$PDDIR/spool/save ディレクトリ下に出力されます。出力される障害情報を次に示します。

- コアファイル
- .deb ファイル
- コマンド名<障害情報の出力日時>プロセス ID.txt ファイル※

注※

このファイルには，次に示すコマンドの実行結果が出力されます。

・pdls -d lck

・pdls -d rpc -a

なお，このオペランドは無応答障害の検知を目的としています。したがって，pdsdbrog コマンドの予想実行時間に対して，十分に余裕を持たせた時間を指定してください。例えば，pdsdbrog コマンドの予想実行時間が 7 分程度の場合，このオペランドには 20 分ぐらいの時間を指定してください。

アンロードを夜間バッチとして実行する場合などにこのオペランドの指定を推奨します。このオペランドを指定しておくと，通信障害（一時的な障害を含む）やディスク障害などが発生して，pdsdbrog コマンドが無応答状態になった場合，pdsdbrog コマンドを自動的に異常終了させることができます。これによって，pdsdbrog コマンドの無応答状態が，UAP やほかのコマンドの実行に影響が及ぶのを防ぐことができます。

《留意事項》

- このオペランドに 0 を指定した場合は，pdsdbrog コマンドの実行時間の上限は設定されません。
- このオペランドを省略した場合，システム定義の pd_utl_exec_time オペランド，または pd_cmd_exec_time オペランドに指定した時間が，pdsdbrog コマンドの実行時間の上限になります。pd_utl_exec_time オペランド，および pd_cmd_exec_time オペランドについては，マニュアル「HiRDB システム定義」を参照してください。
- オペランドの優先順位は高い順に次のとおりです（優先順位は 1 の指定がいちばん上位です）。

優先順位	指定個所	オペランド名
1	pdsdbrog 制御文の environment 文	exectime
2	システム定義	pd_utl_exec_time
3	システム定義	pd_cmd_exec_time

●recnomsg = 処理経過メッセージの出力間隔

～<符号なし整数>((1～1,000))《10》（単位：1 万件）

アンロードの処理経過メッセージの出力間隔を，1 万件（レコード）単位で指定します。例えば，このオペランドに 5 を指定すると，アンロードが 5 万件（レコード）終わるごとに経過メッセージ（KFPB63014-I）が出力されます。

●generation = 世代番号

～<符号なし整数>((0～10))

処理対象の RD エリアの世代番号を指定します。このオペランドは，インナレプリカ機能使用時に指定します。

なお，SD FMB の SDB データベースを使用している場合，このオペランドは省略してください。

0：オリジナル RD エリアを処理対象とする場合に指定します。

1～10：指定した世代のレプリカ RD エリアを処理対象とする場合に指定します。

このオペランドを省略した場合，カレント RD エリアが処理対象になります。カレント RD エリア以外の RD エリアを処理対象とする場合に，このオペランドを指定してください。

《運用上の注意事項》

- レプリカ RD エリアに対して RD エリア単位のアンロードを実行する場合，unload 文の area オペランドにはオリジナル RD エリア名を指定してください。このオペランドには処理対象の RD エリアの世代番号を指定してください。
- pdsdbrog コマンドでレコードをアンロードし，pdsdbload コマンドでデータロードする場合，アンロードとデータロードの実行時に同じ世代番号を指定してください。異なる世代番号を指定してもチェックアウトされないため，指定された世代番号の RD エリア（アンロードを実行した RD エリアとは異なる RD エリア）にデータロードされます。

> **参考**
>
> このオペランドに指定した世代番号の RD エリアがあるかどうかをチェックします。チェックの結果，該当する RD エリアがない場合，KFPB63305-E メッセージを出力してリターンコード 8 で pdsdbrog コマンドが終了します。RD エリア間のレプリカステータスが不一致の場合，KFPB63306-E メッセージを出力してリターンコード 8 で pdsdbrog コマンドが終了します。

13.3.2 unload 文

unload 文には，アンロード処理に関する情報を指定します。

指定形式

```
unload
     record = レコード型名
    〔server = バックエンドサーバ名〕
    〔area = アンロード対象のRDエリア名〕
     unldfile = アンロードデータファイル名
    〔unldinf = アンロードの実行結果ファイル名〕
```

```
    prefix = プリフィクス部の長さ
    unldkind =  {lod | lod_4v}
   〔afmtype =  {type1 | type2} 〕
```

●record = レコード型名

～<識別子>((1～30 バイト))

アンロードを実行するレコードのレコード型名を指定します。SDB データベース定義の RECORD 句に指定したレコード型名を指定します。

4V FMB または SD FMB の SDB データベースの場合は，ルートレコードのレコード型名を指定してください。4V AFM の SDB データベースの場合は，仮想ルートレコードのレコード型名を指定してください。

●server = バックエンドサーバ名

～<識別子>((1～8 バイト))

アンロード対象のレコードを格納している RD エリアを管理しているバックエンドサーバの名称を指定します。

英大文字と英小文字の指定が区別されます。

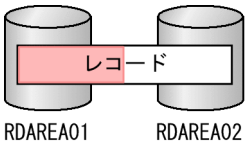
このオペランドの指定は省略することができます。

●area = アンロード対象の RD エリア名

～<識別子>((1～30 バイト))

SDB データベースが横分割されている場合，アンロード対象の RD エリアを指定します。SDB データベースが横分割されている場合，RD エリア単位にアンロードを実行する必要があるため，このオペランドを必ず指定してください。

例えば，次に示すように SDB データベースが横分割されている場合，このオペランドに RDAREA01 を指定すると，RDAREA01 に格納されているレコードがアンロード対象になります。



《インナレプリカ機能使用時の注意事項》

レプリカ RD エリアに対してアンロードを実行する場合，このオペランドにはオリジナル RD エリア名を指定してください。そして，environment 文の generation オペランドに，処理対象 RD エリアの世代番号を指定してください。

●unldfile = アンロードデータファイル名

～<パス名>((1～1,023 バイト))

アンロードデータファイル名を絶対パスで指定します。アンロードデータファイルは，アンロード対象の SDB データベースがあるサーバマシンに作成されます。そのサーバマシンのパス名を指定します。

《注意事項》

- このオペランドに指定するファイルのディレクトリは事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与してください。ディレクトリがない場合，または権限がない場合は，pdsdbrog コマンドがエラーになります。
- 不正なアンロードデータファイル名を指定した場合，リターンコード 12 で pdsdbrog コマンドが異常終了することがあります。
- このオペランドに指定したファイルがすでに存在する場合は一度削除してから再作成されます。削除処理はファイルのオープン前に実行するため，パス名が不正な場合などは削除処理（システムコール：unlink 関数）がエラーとなり，pdsdbrog コマンドが異常終了することがあります。

> **参考**
>
> アンロードデータファイルの入出力時に使用されるバッファ長は，システム定義の pd_utl_file_buff_size オペランドの値が適用されます。

●unldinf = アンロードの実行結果ファイル名

～<パス名>((1～1,023 バイト))

アンロードが完了すると，アンロードの実行結果が実行結果ファイルに出力されます。このオペランドには，実行結果ファイルの名称を絶対パスで指定します。実行結果ファイルは，処理対象 RD エリアがあるサーバマシンに作成されます。

このオペランドを省略した場合，次に示すどれかのディレクトリ下に実行結果ファイルが作成されます（優先順位は 1 の指定がいちばん上位です）。

1. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
2. 環境変数 TMPDIR に指定したディレクトリ
3. /tmp ディレクトリ

実行結果ファイルの名称は SDBROG-xxxxxxxxx になります。xxxxxxxxx は，ファイル作成時間，プロセス ID などが含まれた文字列になります。作成されたファイルの名称は KFPB63032-I メッセージで確認できます。

《注意事項》

- このオペランドに指定するファイルのディレクトリは事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に対して付与してください。ディレクトリがない場合，または権限がない場合は，pdsdbrog コマンドがエラーになります。
- このオペランドに指定したファイルがすでに存在する場合，そのファイルは上書きされます。

実行結果ファイルに出力される情報については，「13.7　アンロードの実行結果ファイルの出力形式」を参照してください。

●prefix = プリフィクス部の長さ

～<符号なし整数>((0～32,767))（単位：バイト）

アンロードデータファイルに出力するレコードのプリフィクス部の長さをバイト単位で指定します。次に示すどれかに該当する場合には，このオペランドに 1 以上の値を指定してください。

- アンロードデータファイルを入力データファイルとしてデータロードするときに，USER ポインタフラグを引き継ぐ場合※1
- アンロードデータファイルを入力データファイルとしてデータロードするときに，事前割り当てページ数を引き継ぐ場合※2
- アンロードデータファイルを入力データファイルとしてデータロードするときに，ページ切り替えフラグを設定する場合※3

注※1

pdsdblod によるデータロード，またはレコードの格納（STORE）によるデータ格納時に設定した USER ポインタフラグを，データベースの再編成後の 4V FMB の SDB データベースに引き継ぐ場合は，dbinf 文の userpflg オペランドを必ず指定してください。

dbinf 文の userpflg オペランドの指定を省略すると USER ポインタフラグを設定しない状態でリロードされます。

注※2

pdsdblod によるデータロード，またはレコードの格納（STORE）によるデータ格納時に設定した事前割り当てページ数を，データベースの再編成後の 4V FMB の SDB データベースに引き継ぐ場合は，dbinf 文の pagenum オペランドを必ず指定してください。

dbinf 文の pagenum オペランドの指定を省略すると事前割り当てページを確保しない状態でリロードされます。

注※3

ページ切り替えフラグはアンロードの対象になりません。アンロードデータファイル中のプリフィクス部にページ切り替えフラグを設定する領域が用意されるだけです。

このため，データロード時にページ切り替えによる格納配置制御をする場合は当該領域を編集する必要があります。

対処方法の詳細については，「13.8　ページ切り替えフラグの扱い【4V FMB，4V AFM】」を参照してください。

アンロードデータファイルを入力データファイルとしてデータロードする場合，このオペランドで指定するプリフィクス長と，pdsdblod 制御文の load 文の prefix オペランドで指定するプリフィクス長を同じにする必要があります。プリフィクス長が異なると，データロードができません。

ページ切り替えフラグの設定の詳細については，「13.8　ページ切り替えフラグの扱い【4V FMB，4V AFM】」を参照してください。

《注意事項》

- プリフィクス部が必要ない場合は 0 を指定してください。
- unldkind オペランドに lod_4v を指定した場合は，このオペランドに 94 を指定してください。
- プリフィクス部については，「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」を参照してください。ここで説明している L1 の値をこのオペランドに指定します。

参考

このオペランドに 1 以上を指定した場合，アンロードデータファイル中のプリフィクス部は 0（ゼロ）で初期化されます。

●unldkind = {lod | lod_4v}

アンロードデータファイルの種類を指定します。このオペランドの指定によって，アンロードデータファイルのプリフィクス部に出力される情報が決まります。アンロードデータファイルの使用目的に従って，このオペランドを指定してください。

lod：

アンロードデータファイルをデータロードの入力データファイルとして使用する場合（UAP の入力情報として使用しない場合）に指定します。

lod_4v：

アンロードデータファイルを UAP の入力情報として使用する場合に指定します。この場合，プリフィクス部には TMS-4V/SP 情報が出力されます。出力される TMS-4V/SP 情報については，「13.6.3　プリフィクス部の出力形式（unldkind オペランドに lod_4v を指定した場合）【4V FMB，4V AFM】」を参照してください。

なお，4V DAM の SDB データベースをアンロードする場合は，TMS-4V/SP 情報の出力形式を，このオペランドの次に説明する afmtype オペランドで指定できます。

また，作成されたアンロードデータファイルは，データロードの入力データファイルとしても使用できます。

注意事項

SD FMB の SDB データベースの場合，lod_4v は指定できません。

《注意事項》

このオペランドに lod_4v を指定する場合の注意事項を次に示します。

- prefix オペランドには，必ず 94 を指定してください。
- dbinf 文の userpflg オペランドを指定する場合は，必ず 77 を指定してください。
- dbinf 文の pagenum オペランドは指定できません。
- 4V FMB の SDB データベースのレコードをアンロードする場合，次に示す制限があります。

  ・ルートレコード型で指定したデータベースキー（SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素の合計長）が 36 バイト以下であること

  ・レコード型の RD エリア分割キーが 4 バイト以下であること

  ・レコード型名が 8 バイト以下であること

  上記の制限値を超えた場合，pdsdbrog コマンドがエラーになります。
- 4V AFM の SDB データベースのレコードをアンロードする場合，次に示す制限があります。

  ・SDB データベース名が 4 バイト以下であること（5 バイト以降は切り捨てられます）

・afmtype オペランドに type2 を指定した場合，レコード型名が 8 バイト以下であること（ただし，仮想ルートレコード型名を除く）

・データベースキーが 32 バイト以下であること※

注※　この制限値を超えた場合，pdsdbrog コマンドがエラーになります。

●afmtype = {type1 | type2}

4V DAM の SDB データベースのレコードをアンロードする場合に，アンロードデータファイルのプリフィクス部に出力する TMS-4V/SP 情報の出力形式を指定します。出力形式の詳細については，「13.6.3　プリフィクス部の出力形式（unldkind オペランドに lod_4v を指定した場合）【4V FMB, 4V AFM】」を参照してください。

type1：

TMS-4V/SP 情報に SDB データベース名を出力する形式を選択する場合に指定します。

type2：

TMS-4V/SP 情報にレコード型名を出力する形式を選択する場合に指定します。

《前提条件》

- このオペランドは，4V DAM の SDB データベースのレコードをアンロードする場合に指定します。4V FMB または SD FMB の SDB データベースをアンロードする場合は，このオペランドを指定できません。
- unldkind オペランドに lod_4v を指定している必要があります。

《注意事項》

- 4V FMB または SD FMB の SDB データベースをアンロードする場合にこのオペランドを指定すると，pdsdbrog コマンドがエラーになります。
- 4V MAM，4V TAM，または 4V SAM の SDB データベースをアンロードする場合にこのオペランドに type2 を指定すると，pdsdbrog コマンドがエラーになります。4V MAM，4V TAM，および 4V SAM の SDB データベースの場合は，TMS-4V/SP 情報は type1 の形式で出力されます。

参考

TMS-4V/SP で type2 に相当するデータベースとは，次に示す 2 つの条件を満たすデータベースです。

- 動的割り当て型の 4V DAM の SDB データベースである。
- セグメントレベルキーの定義がない。

13.3.3　dbinf 文【4V FMB】

アンロードデータファイルに USER ポインタフラグを格納する場合に dbinf 文を指定します。

4V AFM または SD FMB の SDB データベースの場合，dbinf 文は指定できません。

指定形式

```
dbinf

     〔userpflg = （USERポインタフラグの格納位置）〕
     〔pagenum = （事前割り当てページ数の格納位置）〕
```

●userpflg = （USER ポインタフラグの格納位置）

～<符号なし整数>((0～32,766))

USER ポインタフラグの格納位置を指定します。プリフィクス部の先頭を 0 とし，USER ポインタフラグの格納位置を指定してください。

子レコードに対して USER ポインタで親子集合の接続がある場合，このオペランドに指定した位置に C'U'が格納されます。そのほかの場合は，unldkind オペランドの指定値によって次の値が格納されます。

- unldkind オペランドに lod を指定している場合：0x00
- unldkind オペランドに lod_4v を指定している場合：半角空白

なお，このオペランドに指定する格納位置は，次に示す条件を満たす必要があります。

- 格納位置＋ 1≦プリフィクス部の長さ
- pagenum オペランドで指定した格納位置と重ならないこと

《注意事項》

- pdsdblod コマンドでデータロードをする際，pdsdblod 制御文の dbinf 文の userpflg オペランドには，このオペランドで指定した位置を指定してください。このオペランドを省略した場合は，pdsdblod 制御文の dbinf 文の userpflg オペランドも省略してください。
- unload 文の unldkind オペランドに lod_4v を指定した場合にこのオペランドを指定するときは，このオペランドに 77 を指定してください。
- userpflg オペランドを省略して pdsdbrog コマンドを実行した場合，USER ポインタを持つ子レコードをアンロードすると，pdsdbrog コマンドは警告メッセージを出力しますが，処理は続行してリターンコード 4 で終了します。このため，そのまま pdsdblod コマンドでデータロードをすると，USER ポインタによる親子集合の接続関係が初期化されます。

●pagenum = （事前割り当てページ数の格納位置）

～<符号なし整数>((0～32,763))

事前割り当てページ数の格納位置を指定します。プリフィクス部の先頭を 0 として，事前割り当てページ数の格納位置を指定してください。指定された位置から 4 バイトの領域へ事前割り当てページ数を設定します。

アンロードしたルートレコードに事前割り当てページ数の設定がある場合，ページ数を格納します。事前割り当てページ数の設定がないレコードの場合は 0 を設定します。

なお，このオペランドに指定する格納位置は，次に示す条件を満たす必要があります。

- 格納位置＋ 4≦プリフィクス部の長さ
- userpflg オペランドで指定した格納位置と重ならないこと

《注意事項》

- pdsdblod コマンドでデータロードをする際，pdsdblod 制御文の dbinf 文の pagenum オペランドには，このオペランドで指定した位置を指定してください。このオペランドを省略した場合は，pdsdblod 制御文の dbinf 文の pagenum オペランドも省略してください。
- pagenum オペランドを省略して pdsdbrog コマンドを実行した場合，事前ページ割り当て機能を適用しているファミリのルートレコードをアンロードすると，pdsdbrog コマンドは警告メッセージを出力しますが，処理は続行してリターンコード 4 で終了します。このため，そのままデータロードをすると，事前ページ割り当て機能が適用されません。

13.3.4 pdsdbrog 制御文の各オペランドの指定可否

pdsdbrog 制御文の各オペランドの指定可否を次の表に示します。

表 13–1 pdsdbrog 制御文の各オペランドの指定可否

項番	オペランド名		データベースの種類			備考
			4V FMB	4V AFM	SD FMB	
1	environment 文	schema	○	○	○	environment 文は必ず指定してください。
2		exectime	△	△	△	
3		recnomsg	△	△	△	
4		generation	△	△	△	
5	unload 文	record	○	○	○	unload 文は必ず指定してください。
6		server	△	△	△	
7		area	△	△	△	
8		unldfile	○	○	○	
9		unldinf	△	△	△	
10		prefix	○	○	○	
11		unldkind	○	○	○	
12		afmtype	×	△	×	
13	dbinf 文	userpflg	△	×	×	アンロードデータファイルに USER ポインタフラグを格納する場合に dbinf 文を指定してください。
14		pagenum	△	×	×	事前ページ割り当て機能を適用した SDB データベースの場合は dbinf 文を指定してください。

（凡例）

○：必ず指定するオペランドです。

△：指定を検討するオペランドです。

×：指定不要なオペランドです。

13.3.5　pdsdbrog 制御文の記述規則

pdsdbrog 制御文の記述規則は，pdsdblod 制御文の記述規則と同じです。pdsdblod 制御文の記述規則については，「12.3.8　pdsdblod 制御文の記述規則」を参照してください。

13.4　規則および注意事項

pdsdbrog コマンド実行時の規則および注意事項について説明します。

13.4.1　規則

(1)　pdsdbrog コマンドの実行条件

- pdsdbrog コマンドは，HiRDB が稼働中のときだけ実行できます。
- pdsdbrog コマンドは，システムマネジャがあるサーバマシンで実行してください。
- pdsdbrog コマンドの実行時には，次のサーバが稼働している必要があります。
  - システムマネジャ
  - 1 つ以上のフロントエンドサーバ
  - ディクショナリサーバ
  - アンロード対象の RD エリアを管理しているバックエンドサーバ
- RD エリアの状態が pdsdbrog コマンドの実行条件を満たしていない場合，pdsdbrog コマンドはエラーになります。この場合，「付録 I.1　RD エリアの状態によるユティリティの実行可否」を参照して，RD エリアの状態を変更してください。
- SDB ディレクトリ情報が常用常駐領域に常駐化されていないと，pdsdbrog コマンドが実行できません。また，全ユニットの常用常駐領域中の SDB ディレクトリ情報が一致している必要があります。

(2)　コマンドの実行者

pdsdbrog コマンドを実行する際，CONNECT 権限が必要になります。

(3)　コマンド実行前の作業

- クライアント環境定義の PDUSER オペランドに pdsdbrog コマンドの実行ユーザの認可識別子とパスワードを指定してください。

  このとき，OS ログインユーザの簡易認証機能は使用できません。
- pdsdbrog 制御文を指定した制御文ファイルを準備し，システムマネジャがあるサーバマシンに格納してください。また，制御文ファイルを格納するディレクトリに対する実行権限，読み込み権限，および書き込み権限を pdsdbrog コマンドの実行ユーザに付与してください。
- pdsdbrog 制御文のオペランドに指定するディレクトリ（実行結果ファイルを格納するディレクトリなど）は事前に作成しておいてください。また，作成したディレクトリの実行権限，読み込み権限，および書き込み権限を HiRDB 管理者に付与してください。
- PDDIR，PDCONFPATH，および LANG 環境変数が正しく設定されているかを確認してください。

- pdsdbrog コマンドを実行する前に，pdsdbarc -a コマンドで SDB ディレクトリ情報に関する情報を表示し，全ユニットの常用常駐領域の SDB ディレクトリ情報が一致しているかどうかを確認してください。

(4) 推奨する運用方法

pdsdbrog コマンドを実行する前に，処理対象の RD エリアを pdhold コマンドで閉塞することを推奨します。

参考

pdsdbrog コマンドの実行中は処理対象の RD エリアにアクセスできません。UAP やほかのコマンドがアクセスしようとした場合，排他待ち状態になり，排他待ち時間のタイムアウトまで UAP やコマンドにエラーが返りません。pdhold コマンドで RD エリアを閉塞しておくと，UAP やほかのコマンドが RD エリアにアクセスしたときにすぐにエラーが返ります。

13.4.2 注意事項

(1) 実行環境に関する注意事項

1. 標準出力または標準エラー出力へのメッセージ出力が抑止されている環境で pdsdbrog コマンドを実行しないでください。pdsdbrog コマンドの実行中は，標準出力に経過メッセージが出力されます。また，エラーの発生時には，標準エラー出力にエラーメッセージが出力されます。そのため，標準出力または標準エラー出力へのメッセージ出力が抑止されている環境で pdsdbrog コマンドを実行した場合，メッセージの出力待ちで pdsdbrog コマンドが無応答状態になったり，pdsdbrog コマンドが異常終了したりするおそれがあります。

   なお，標準出力および標準エラー出力に出力するメッセージの順番や数は，メッセージログファイルや syslogfile に出力されるメッセージの順番や数と一致しないことがあります。メッセージの順番や数を正確に把握したい場合は，メッセージログファイルまたは syslogfile に出力されたメッセージを確認してください。

2. pdsdbrog コマンドをキャンセルする場合は，pdcancel コマンドを実行してください。pdsdbrog コマンドのキャンセルに，OS の kill コマンドを使用しないでください。kill コマンドで pdsdbrog コマンドをキャンセルすると，ユニットが異常終了するおそれがあります。

3. OS がサポートしていない文字コードは使用できません。pdsdbrog 制御文に ASCII コード以外の文字がある場合は，pdsdbrog 制御文に記述する文字コード，LANG 環境変数に設定した文字コード，および pdsetup コマンドで指定した文字コードを合わせる必要があります。文字コードが合っていない場合は，pdsdbrog コマンドがエラーになることがあります。

4. アンロードの際は，PACKED DECIMAL FIXED 型の符号部は変換されません。PACKED DECIMAL FIXED 型の符号部はデータロードの際に変換されます。

(2) 系切り替え機能を使用している場合の注意事項

アンロードデータファイルは通常ファイルとして作成されるため，アンロードの実行中に系切り替えが発生すると OS の I/O バッファが無効になり，アンロードデータファイルの内容が保証されなくなります。そのため，アンロードの実行中に系切り替えが発生した場合は，系が切り替わったあとにアンロードを再実行してください。

(3) pdsdbrog コマンドと pdrorg コマンドとの仕様差について（参考）

アンロードを実行する次の 2 つのコマンドは，コマンドの実行時に BES 名と RD エリア名を指定しますが，指定の組み合わせが同じであっても，コマンドの実行可否が異なることがあります。

- SDB データベースのレコードをアンロードする pdsdbrog コマンド
- 表のデータをアンロードする pdrorg コマンド

例えば，横分割している SDB データベースのレコードをアンロードする場合，pdsdbrog コマンドの実行時にサーバ名（server オペランド）を指定して RD エリア名（area オペランド）を省略すると，エラーになります。

一方，横分割している表のデータをアンロードする場合，pdrorg コマンドの実行時にサーバ名（unload 文）を指定して RD エリア名（-r オプション）を省略してもエラーになりません。各コマンドのオプション，制御文の説明を確認してからデータロードを実行するようにしてください。pdrorg コマンドについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

13.5　リターンコード

pdsdbrog コマンドのリターンコードの意味および対処を次の表に示します。

表 13–2　pdsdbrog コマンドのリターンコードの意味および対処

リターンコード	意味	対処
0	pdsdbrog コマンドが正常終了しました。	なし。
4	pdsdbrog コマンドの処理が完了しました。ただし，警告レベルのエラーが発生しました。	出力されたメッセージを参照して，メッセージの対処に従ってください。
8	pdsdbrog 制御文のオペランドの指定に誤りがあるため，pdsdbrog コマンドが実行できませんでした。	出力されたメッセージを参照して，pdsdbrog 制御文のオペランドの指定を見直してください。
12	処理が続行できないエラーが発生し，pdsdbrog コマンドが異常終了しました。	「13.9.1　pdsdbrog コマンドが異常終了した場合」を参照して対処してください。

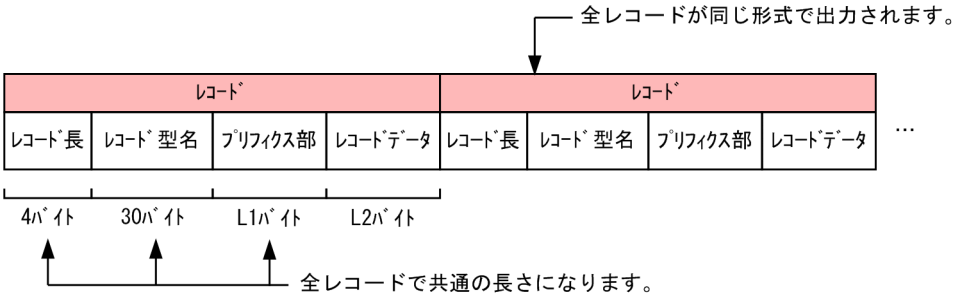
13.6　アンロードデータファイルの出力形式

アンロードデータファイルに出力されるレコードは，データロードの入力データと同じ形式で出力されます。ここでは，アンロードデータファイルに出力されるレコードのデータ形式とレコードの格納順序について説明します。

13.6.1　レコードのデータ形式

アンロードデータファイルに出力されるレコードのデータ形式を次の図に示します。

図 13–7　アンロードデータファイルに出力されるレコードのデータ形式



（凡例）

L1：プリフィクス部の長さで，0～32,767 バイトになります。

L2：SDB データベース定義で定義した全構成要素サイズの合計長になります。

レコード内の各項目（レコード長，レコード型名，プリフィクス部，およびレコードデータ）については，「12.6.1　レコードのデータ形式」を参照してください。

プリフィクス部の格納規則

- ページ切り替えフラグはアンロード対象になりません。
- USER ポインタフラグは，pdsdbrog 制御文の dbinf 文の userpflg オペランドで指定した位置に格納されます。設定値は C'U'です。

unload 文の unldkind オペランドに lod_4v を指定した場合

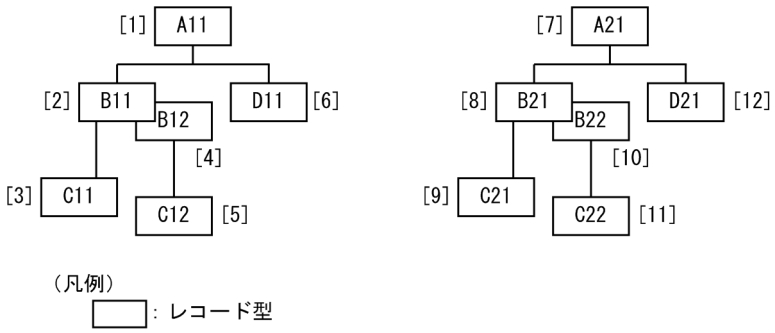
unload 文の unldkind オペランドに lod_4v を指定した場合，プリフィクス部に TMS-4V/SP 情報が出力されます。出力形式の詳細については，「13.6.3　プリフィクス部の出力形式（unldkind オペランドに lod_4v を指定した場合）【4V FMB，4V AFM】」を参照してください。

13.6.2　レコードの格納順序

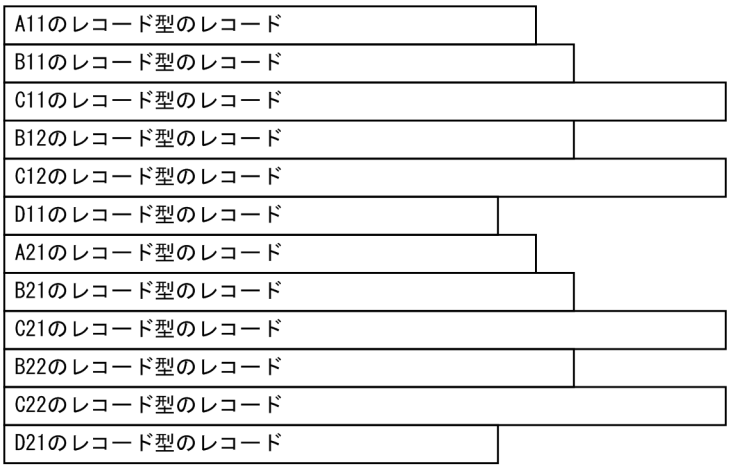
アンロードデータファイルのレコード格納順序はデータベースの種類によって異なります。

(1)　4V FMB または SD FMB の SDB データベースの場合

次に示す実現値モデルを例にして，アンロードデータファイルのレコード格納順序を説明します。



この場合，アンロードデータファイルのレコード格納順序は次のようになります。



レコード格納順序の規則

階層構造間の格納順序の規則を次に示します。

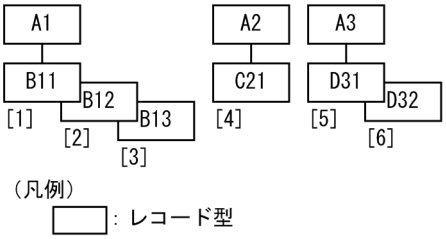
- 階層構造間の格納順序は，ルートレコード型に対するシーケンシャルインデクス順となります。

同一階層構造内の格納順序の規則を次に示します。

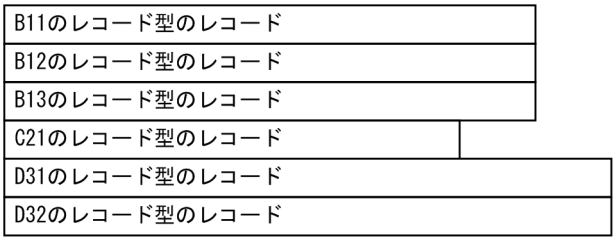
- レコード型の上位レベルから下位レベルに親子集合に従って格納されます。
- 同一レベルのレコード型が複数ある場合，SDB データベース定義の SET 句の定義順に親子集合に従って格納されます。
- 同一レコード型のデータが複数ある場合，NEXT ポインタの方向の順に格納されます。このデータに下位レベルのレコード型のデータがある場合は，下位レベルのデータが先に格納されます。

(2)　4V AFM の SDB データベースの場合

次に示す実現値モデルを例にして，アンロードデータファイルのレコード格納順序を説明します。



この場合，アンロードデータファイルのレコード格納順序は次のようになります。



レコード格納順序の規則

- 仮想ルートレコード型に対するシーケンシャルインデクス順に格納されます。子レコード型のレコードが複数ある場合は，NEXT ポインタの方向の順に格納されます。
- 仮想ルートレコード型のレコードはアンロードデータファイルに格納されません。

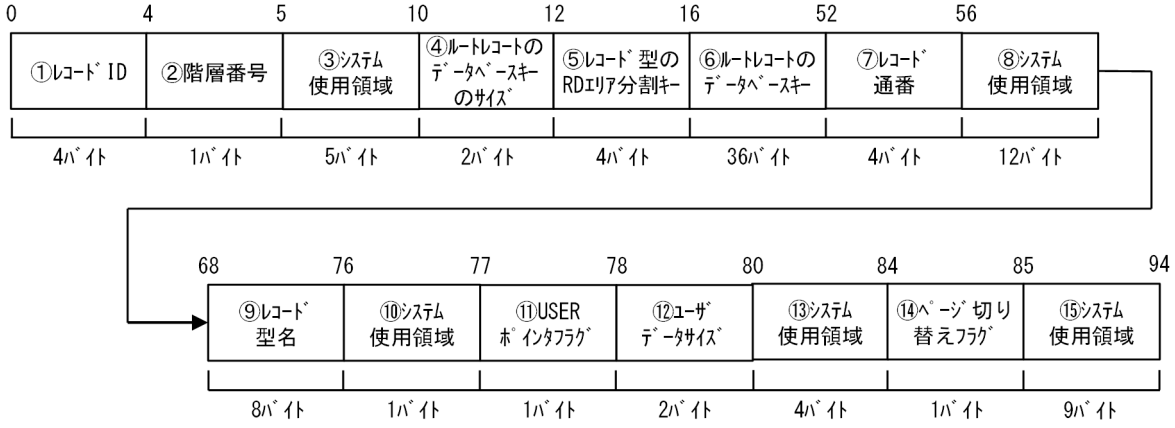
13.6.3 プリフィクス部の出力形式（unldkind オペランドに lod_4v を指定した場合）【4V FMB，4V AFM】

unload 文の unldkind オペランドに lod_4v を指定した場合の，プリフィクス部のデータ形式について説明します。

(1) 4V FMB の SDB データベースの場合

プリフィクス部のデータ形式（4V FMB の SDB データベースの場合）を次の図に示します。

図 13–8 プリフィクス部のデータ形式（4V FMB の SDB データベースの場合）



［説明］

図中の丸付き数字が次に示す表の項番に対応しています。

項番	項目	属性	長さ（バイト）	内容
1	レコード ID	char	4	レコード種別を示す情報が格納されます。 • ルートレコード型の場合："SEGR" • 子レコード型の場合："SEG "
2	階層番号	unsigned char	1	レコードの階層を示す番号が格納されます。 • ルートレコード型の場合：1 • 子レコード型の場合：2～15
3	システム使用領域	char	5	（X'00....00'）が格納されます。
4	ルートレコードのデータベースキーのサイズ	short	2	ルートレコード型の構成要素のうち，次の構成要素の長さの合計が格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素
5	レコード型の RD エリア分割キー	char	4	ルートレコード型の構成要素のうち，次に示す 2 つの条件を満たす構成要素に対応するレコード実現値が格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素 • SDB データベース定義の TYPE 句のデータ種別 2 に A を指定した構成要素 左詰めで格納され，余りがある場合は X'00'が格納されます。 該当する構成要素がない場合は，X'00....00'が格納されます。
6	ルートレコードのデータベースキー	char	36	ルートレコードのデータベースキーのサイズの出力対象となった構成要素に対応するレコード実現値が連結されて格納されます。左詰めで格納され，余りがある場合は X'00'が格納されます。
7	レコード通番	unsigned int	4	レコード通番が格納されます。ルートレコードには 1 が格納され，アンロードデータファイルのレコード格納順序に従い，順次 2，3，…と格納されます。アンロードデータファイルのレコード格納順序については，「13.6.2(1)　4V FMB または SD FMB の SDB データベースの場合」を参照してください。 なお，レコード通番は，ルートレコードごとに 1 にリセットされます。 また，オーバーフローした場合（4,294,967,295 を超えた場合）は，0 に戻ります。
8	システム使用領域	char	12	（X'00....00'）が格納されます。
9	レコード型名	char	8	アンロード対象のレコードのレコード型名が格納されます。レコード型名は左詰めで格納され，余りがある場合は半角空白が格納されます。
10	システム使用領域	char	1	（X'00'）が格納されます。

項番	項目	属性	長さ（バイト）	内容
11	USER ポインタフラグ	char	1	次に示す条件をすべて満たしている場合は U が格納されます。満たしていない場合は半角空白が格納されます。 • dbinf 文の userpflg オペランドが指定されている。 • SDB データベース格納定義で USER ポインタが定義されている。 • 子レコードに対して USER ポインタで親子集合の接続をしている。 USER ポインタフラグに設定される値の意味については，「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」の「表 12-10　USER ポインタフラグに設定する値」を参照してください。
12	ユーザデータサイズ	short	2	アンロード対象のレコードの構成要素のうち，次の構成要素の長さの合計値が格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に U を指定した構成要素
13	システム使用領域	int	4	任意の番号が格納されます。
14	ページ切り替えフラグ	char	1	ページ切り替えフラグは格納されません。（X'00'）が格納されます。
15	システム使用領域	char	9	（X'00....00'）が格納されます。

(2)　4V AFM の SDB データベースの場合

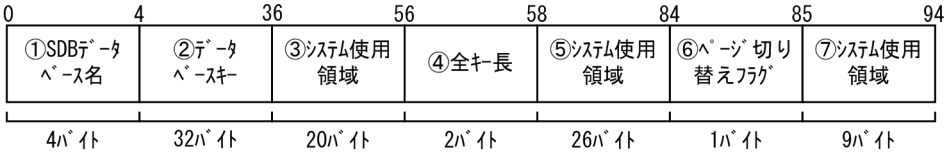
4V DAM の SDB データベースの場合，unload 文の afmtype オペランドの指定によって，プリフィクス部のデータ形式が決まります。

4V MAM，4V TAM，および 4V SAM の SDB データベースの場合，プリフィクス部は「(a)　afmtype オペランドに type1 を指定した場合」の形式になります。

(a)　afmtype オペランドに type1 を指定した場合

プリフィクス部のデータ形式（afmtype オペランドに type1 を指定した場合）を次の図に示します。

図 13–9　プリフィクス部のデータ形式（afmtype オペランドに type1 を指定した場合）



[説明]

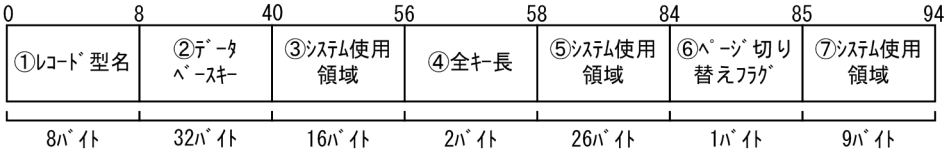
図中の丸付き数字が以下の表の項番に対応しています。

項番	項目	属性	長さ（バイト）	内容
1	SDB データベース名	char	4	仮想ルートレコードの構成要素のうち，次の構成要素に対応するレコード実現値が格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に D を指定した構成要素 SDB データベース名は左詰めで格納され，余りがある場合は半角空白が格納されます。また，SDB データベース名が 5 バイト以上の場合は，5 バイト目以降のデータを切り捨てます。
2	データベースキー	char	32	仮想ルートレコードの構成要素のうち，次の構成要素に対応するレコード実現値が連結されて格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素 左詰めで格納され，余りがある場合は X'00'が格納されます。 該当する構成要素がない場合は，X'00....00'が格納されます。
3	システム使用領域	char	20	（X'00....00'）が格納されます。
4	全キー長	short	2	アンロード対象のレコードの構成要素のうち，次の構成要素の長さの合計値が格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に D，K，または N を指定した構成要素
5	システム使用領域	char	26	（X'00....00'）が格納されます。
6	ページ切り替えフラグ	char	1	設定されません。（X'00'）が格納されます。
7	システム使用領域	char	9	（X'00....00'）が格納されます。

(b) afmtype オペランドに type2 を指定した場合

プリフィクス部のデータ形式（afmtype オペランドに type2 を指定した場合）を次の図に示します。

図 13–10 プリフィクス部のデータ形式（afmtype オペランドに type2 を指定した場合）



[説明]

図中の丸付き数字が以下の表の項番に対応しています。

項番	項目	属性	長さ（バイト）	内容
1	レコード型名	char	8	アンロード対象のレコードのレコード型名が格納されます。レコード型名は左詰めで格納され，余りがある場合は半角空白が格納されます。

項番	項目	属性	長さ（バイト）	内容
2	データベースキー	char	32	仮想ルートレコードの構成要素のうち，次の構成要素に対応するレコード実現値が連結されて格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素 左詰めで格納され，余りがある場合は X'00'が格納されます。 該当する構成要素がない場合は，X'00....00'が格納されます。
3	システム使用領域	char	16	（X'00....00'）が格納されます。
4	全キー長	short	2	アンロード対象のレコードの構成要素のうち，次の構成要素の長さの合計値が格納されます。 • SDB データベース定義の TYPE 句のデータ種別 1 に D，K，または N を指定した構成要素
5	システム使用領域	char	26	（X'00....00'）が格納されます。
6	ページ切り替えフラグ	char	1	設定されません。（X'00'）が格納されます。
7	システム使用領域	char	9	（X'00....00'）が格納されます。

13.7　アンロードの実行結果ファイルの出力形式

アンロード処理が完了すると，アンロードの実行結果が実行結果ファイルに出力されます。実行結果ファイルに出力される情報について例を使って説明します。

アンロードの実行結果ファイルの出力例

4V FMB の SDB データベースでインナレプリカ機能を使用している場合の出力例を次に示します。

```
pdsdbrog (09-60) **** DB reorganization **** 2015-07-10 18:01:49 [1]      parallel
** record unload information list **
 server name : BES1  [2]
 schema name : FMB1  [3]
 generation  : 0     [4]
 unload kind :lod    [5]
      no[6] area-name[7]  schema-record-name[8]  record-count[9]  method-information[10]
       1    RDAREA01      FMB100                        44512     INDEX:FMB1IDX1
                          FMB111                        62936     SET(N):FMB1F11
                          FMB112                        55879     SET(N):FMB1F12
                          FMB113                        55778     SET(N):FMB1F13
```

[説明]

1. pdsdbrog コマンドを実行した日時が表示されます。
2. 処理対象のバックエンドサーバ名が表示されます。
3. 処理対象の SDB データベース名が表示されます。
4. 処理対象の RD エリアの世代番号が表示されます。0 はオリジナル RD エリアを意味しています。インナレプリカ機能を使用している場合に表示されます。
5. 作成されたアンロードデータファイルの種類が表示されます。

   ・lod：データロードの入力データファイルに使用するアンロードデータファイル

   ・lod_4v:FMB：4V FMB の SDB データベースの TMS-4V/SP 情報がプリフィクス部に出力されたアンロードデータファイル

   ・lod_4v:AFM(type1)：4V AFM の SDB データベースの TMS-4V/SP 情報（SDB データベース名を出力）がプリフィクス部に出力されたアンロードデータファイル

   ・lod_4v:AFM(type2)：4V AFM の SDB データベースの TMS-4V/SP 情報（レコード型名を出力）がプリフィクス部に出力されたアンロードデータファイル
6. 1 から始まる通番が表示されます。
7. レコードをアンロードした RD エリアの名称が表示されます。インナレプリカ機能を使用している場合，レプリカ RD エリアを処理対象にしたときでもオリジナル RD エリア名が表示されます。
8. アンロードしたレコードのレコード型名が表示されます。※
9. アンロードしたレコード件数が表示されます。
10. レコードの検索方法が表示されます。

    ・INDEX:インデクス名

インデクスによるインデクス順検索

・SET(N):親子集合型名

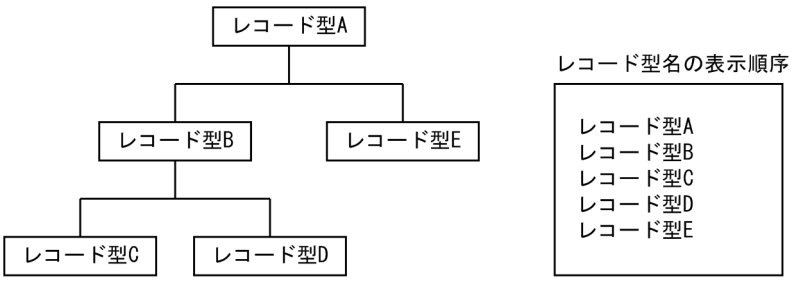
親子集合による NEXT ポインタ方向への検索（正方向検索）

注※

レコード型名の表示順序の規則を次に示します。

- レコード型の上位レベルから下位レベルに親子集合に従って表示されます。
- 同一レベルのレコード型が複数ある場合は，SDB データベース定義の SET 句の定義順に親子集合に従って表示されます。

（例）



なお，4V AFM の SDB データベースに対するアンロードの場合，仮想ルートレコードのレコード型名は出力されません。

参考

pdsdbrog コマンドが異常終了した場合，実行結果ファイルに一部の情報が出力されないことがあります。例えば，先頭のヘッダ行だけが表示されたり，障害発生時点までの情報だけが表示されたりすることがあります。

13.8　ページ切り替えフラグの扱い【4V FMB，4V AFM】

ページ切り替えフラグはアンロードの対象になりません。そのため，ページ切り替えフラグが必要な場合は，ページ切り替えフラグを設定する必要があります。

ページ切り替えフラグの設定の流れを次に示します。

1. pdsdbrog 制御文の prefix オペランドにプリフィクス長（1 以上）を指定して，アンロードを実行します。
2. アンロードデータファイルを直接修正して，ページ切り替えフラグを設定します。
3. pdsdblod 制御文に次の指定をしてデータロードを実行します。
   - prefix オペランドに，1 で指定したプリフィクス長と同じプリフィクス長を指定する※
   - pagecflg オペランドにページ切り替えフラグの開始位置を指定する

注※

アンロードデータファイルを入力データファイルとしてデータロードする場合，アンロード時に prefix オペランドに指定するプリフィクス長と，データロード時に prefix オペランドに指定するプリフィクス長を同じにする必要があります。プリフィクス長が異なる場合，データロードができません。

ページ切り替えフラグについては，「12.6.1(3)　プリフィクス部【4V FMB，4V AFM】」を参照してください。

13.9　トラブルシューティング

ここでは，pdsdbrog コマンドが異常終了したときの対処方法について説明します。

13.9.1　pdsdbrog コマンドが異常終了した場合

pdsdbrog コマンドが異常終了した場合，出力されたメッセージを確認して対処してください。そのあとに，pdsdbrog コマンドを再実行してください。

参考

- pdsdbrog コマンドが異常終了しても，アンロード対象のレコードに異常は発生しません。pdsdbrog コマンド実行前の状態のままです。
- pdsdbrog コマンドが異常終了した場合，未完成のアンロードデータファイルが自動的に削除されないことがあります。pdsdbrog コマンドを再実行する場合は，未完成のアンロードデータファイルが上書きされるため問題ありません。pdsdbrog コマンドを再実行しない場合は，未完成のアンロードデータファイルを削除してください。

13.9.2　pdsdbrog コマンドが無応答状態になった場合

pdsdbrog コマンドが無応答状態になった場合，pdcancel コマンドで pdsdbrog コマンドをキャンセルしてください。手順を次に示します。

手順 1

1. pdls -d prc コマンドを実行します。コマンドの実行結果は，リダイレクションでファイルに出力してください。

```
pdls -d prc>出力先ファイル名
```

2. 1.の実行結果を参照して，バックエンドサーバのプロセス ID を確認します。

```
HOSTNAME : host01(135225)
STATUS PID   UID    GID    SVID   TIME  PROGRAM   C-PID  C-GRP
   :   2924   :      :      :      :    pdsdbrog    :      :
```

PROGRAM 欄に pdsdbrog と表示されている行を検索してください。この行の PID 欄に表示されているプロセス ID（この例では 2924）が，pdsdbrog コマンド実行中のバックエンドサーバのプロセス ID です。

pdsdbrog が表示されていない場合は，手順 2 の方法でプロセス ID を確認してください。

3. pdcancel コマンドで，pdsdbrog コマンド実行中のバックエンドサーバをキャンセルします。

```
pdcancel -X ユニット識別子 -i プロセスID -d
```

ユニット識別子には，pdsdbrog コマンドの処理対象 RD エリアがあるユニットのユニット識別子を指定してください。

プロセス ID には，2.で確認したプロセス ID（この例では 2924）を指定してください。

手順 2

1. pdls -d rpc コマンドを実行します。コマンドの実行結果は，リダイレクションでファイルに出力してください。

```
pdls -d rpc -a>出力先ファイル名
```

2. 1.の実行結果を参照して，ユティリティサーバ（pdsdbrogm）のプロセス ID を確認します。

```
Server Information :
  PID     SVID     STATUS …   CLTPID
  5444    0msdrgN    :          :
```

SVID 欄に 0msdrgN（末尾の N は 16 進の数字）と表示されている行を検索してください。この行の PID 欄に表示されているプロセス ID（この例では 5444）が，ユティリティサーバ（pdsdbrogm）のプロセス ID です。

なお，0msdrgN が複数表示されている場合は，CLTPID 欄（サーバのプロセスと接続しているクライアントのプロセス ID）を確認してください。無応答状態になった pdsdbrog コマンドのプロセス ID が CLTPID 欄に表示されている行が該当する行になります。pdsdbrog コマンドのプロセス ID は，OS の ps コマンドで確認できます。

3. pdcancel コマンドで，ユティリティサーバ（pdsdbrogm）をキャンセルします。

```
pdcancel -X ユニット識別子 -i プロセスID -d
```

ユニット識別子には，pdsdbrog コマンドの処理対象 RD エリアがあるユニットのユニット識別子を指定してください。

プロセス ID には，2.で確認したプロセス ID（この例では 5444）を指定してください。

なお，無応答状態の原因調査のために，pdsdbrog コマンドのキャンセル後，pdinfoget コマンドで障害情報を取得してください。

注意事項

pdsdbrog コマンドをキャンセルする際，OS の kill コマンドを使用しないでください。kill コマンドで pdsdbrog コマンドをキャンセルすると，ユニットが異常終了するおそれがあります。

13.9.3 タイムアウトが発生した場合

environment 文の exectime オペランドに指定した pdsdbrog コマンドの実行時間の上限を超えた場合，pdsdbrog コマンドが強制終了します。この場合，メッセージを確認してタイムアウトの原因を対処してください。そのあとに，pdsdbrog コマンドを再実行してください。

参考

execime オペランドを省略した場合，システム定義の pd_utl_exec_time オペランド，または pd_cmd_exec_time オペランドに指定したタイムアウト時間が適用されます。

13.10　使用例

ここでは，pdsdbrog コマンドの使用例について説明します。

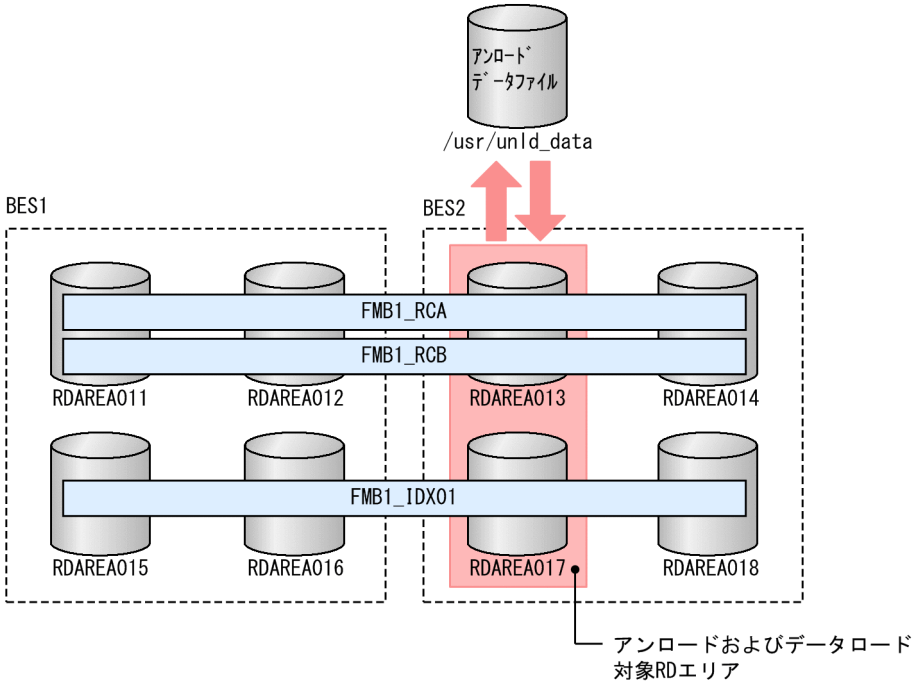
13.10.1　4V FMB の SDB データベースのアンロードおよびデータロード

例題

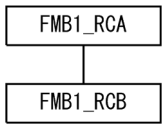
SDB データベースを再編成するために，横分割されている SDB データベースをアンロードします。そのあと，作成したアンロードデータファイルを入力データファイルにしてデータロードします。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V FMB です。
- インデクス FMB1_IDX01 が定義されています。

- アンロードおよびデータロードの対象は，RDAREA013 に格納されているレコードと，RDAREA017 に格納されているインデクスです。
- USER ポインタフラグもアンロード対象にします。

(2) アンロードの実行例

pdsdbrog コマンドの実行例

```
pdsdbrog /usr/control_rog_file
```

pdsdbrog 制御文（/usr/control_rog_file）の例

```
environment
  schema   = FMB1               ...1
  exectime = 0                  ...2
  recnomsg = 1                  ...3
unload
  record   = FMB1_RCA           ...4
  server   = BES2               ...5
  area     = RDAREA013          ...6
  unldfile = /usr/unld_data     ...7
  unldinf  = /usr/unld_result   ...8
  prefix   = 5                  ...9
  unldkind = lod                ...10
dbinf
  userpflg = (0)                ...11
  pagenum  = (1)                ...12
```

[説明]

1. SDB データベース名を指定します。
2. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
3. アンロードが 1 万件完了するごとに経過メッセージを出力します。
4. ルートレコードのレコード型名を指定します。
5. アンロード対象の RD エリアがあるバックエンドサーバを指定します。
6. アンロード対象の RD エリアを指定します。
7. アンロードデータファイル名を指定します。
8. 実行結果ファイル名を指定します。
9. プリフィクス部の長さを指定します。
10. データロードで使用できる入力データファイルの形式でアンロードデータファイルを出力します。
11. USER ポインタフラグの格納位置を指定します。
12. 事前割り当てページ数の格納位置を指定します。

(3) データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = FMB1              ...1
  logmode  = no                ...2
  purge    = yes               ...3
  exectime = 0                 ...4
  recnomsg = 1                 ...5
  divermsg = put               ...6
load
  type     = tree              ...7
  record   = FMB1_RCA          ...8
  server   = BES2              ...9
  area     = RDAREA013         ...10
  data     = /usr/unld_data    ...11
  storinf  = /usr/load_result  ...12
  errdata  = /usr/err_data     ...13
  prefix   = 5                 ...14
  idxmode  = create            ...15
dbinf
  userpflg = (0)               ...16
  pagenum  = (1)               ...17
```

[説明]

1. SDB データベース名を指定します。アンロード時に指定した SDB データベース名を指定します。
2. ログレスモードでデータロードを実行します。
3. 既存レコードを削除してデータロードを実行します。必ず yes を指定してください。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. データロードが 1 万件完了するごとに経過メッセージを出力します。
6. ほかの RD エリアのレコードは論理エラーとします。
7. 4V FMB の SDB データベースの場合は tree を指定します。
8. ルートレコードのレコード型名を指定します。アンロード時に指定したレコード型名を指定します。
9. データロード対象の RD エリアがあるバックエンドサーバを指定します。アンロード時に指定したバックエンドサーバ名を指定します。
10. データロード対象の RD エリアを指定します。アンロード時に指定した RD エリア名を指定します。
11. (2)で作成したアンロードデータファイルを入力データファイルとして指定します。
12. 実行結果ファイル名を指定します。
13. 論理エラー情報ファイル名を指定します。

14. プリフィクス部の長さを指定します。

15. インデクス一括作成モードでインデクスを作成します。

16. USER ポインタフラグの開始位置を指定します。アンロード時に指定した格納位置を指定します。

17. 事前割り当てページ数の開始位置を指定します。アンロード時に指定した格納位置を指定します。

(4) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB1
    DBTYPE 4V FMB
    RECORD FMB1_RCA
      2 DBKEY
       3  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  USERDA0             CHARACTER 30   TYPE  U,D
    RECORD FMB1_RCB
      2  KEYDATA1            XCHARACTER 1   TYPE  K,A
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDB0             CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB1_RCA
        ORDER LAST
      MEMBER FMB1_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SDBOPTION
      FUNCTION
        REFER     USE
        ADD       USE
        UPDATE    USE
        ERASE     USE
        ALLERASE  NOUSE
        DBLODUTL  USE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB1 FOR FMB1
  DBTYPE 4V
  STORAGE RECORD FMB1_RCA
    SEQUENTIAL FMB1_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN ((RDAREA015),(RDAREA016),(RDAREA017),(RDAREA018))
    WITHIN ((RDAREA011) KEYDATA1=(X'F3'),
            (RDAREA012) KEYDATA1=(X'F4'),
            (RDAREA013) KEYDATA1=(X'F5'),
            (RDAREA014) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB1_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA011),(RDAREA012),(RDAREA013),(RDAREA014))
  SET SETAB
    OWNER POINTER FOR FIRST LAST USER MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SDBOPTION
  END STORAGE SCHEMA
```

```
*ENTRY DIRECTORY
  DBSCHM FMB1
```

13.10.2　4V FMB の SDB データベースのアンロードおよびデータロード（インナレプリカ機能を使用したオンライン再編成の場合）

例題

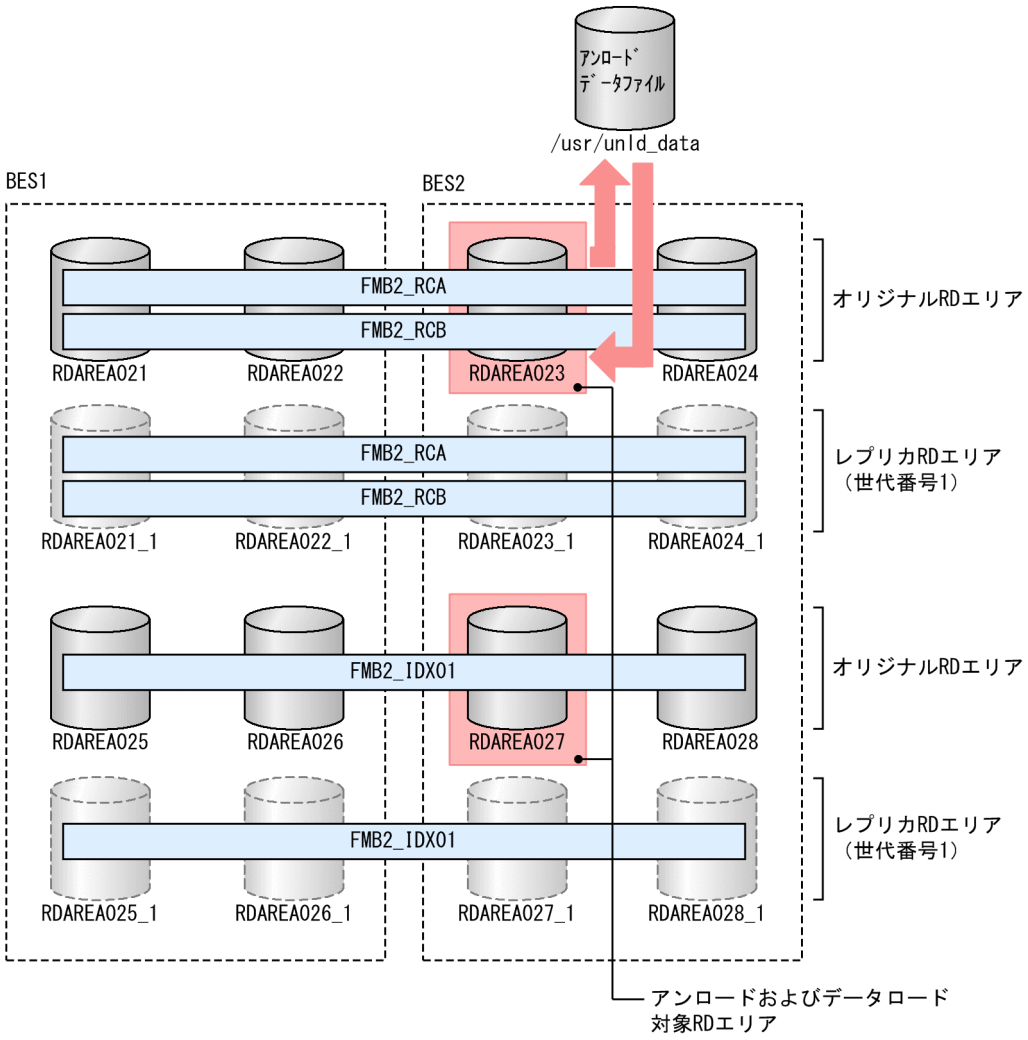
インナレプリカ機能を使用して，横分割されている SDB データベースのオリジナル RD エリア側をアンロードします。そのあと，作成したアンロードデータファイルを入力データファイルにしてオリジナル RD エリア側にデータロードします。

これによって，データベースを再編成します。

ここでは，オリジナル RD エリアの再編成中はレプリカ RD エリアでオンライン業務をする例を基に説明します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造

条件

- SDB データベース種別は 4V FMB です。
- インナレプリカ機能を使用しています。
- アンロードおよびデータロードの対象とするオリジナル RD エリアは RDAREA023 に格納されているレコードと，RD エリア RDAREA027 に格納されているインデクスです。
- オリジナル RD エリアは RDAREA023 と RDAREA027 です。それに対応するレプリカ RD エリアは RDAREA023_1 と RDAREA027_1 です。その世代番号は 1 です。
- インデクス FMB2_IDX01 が定義されています。
- USER ポインタフラグもアンロード対象にします。

(2) アンロードの実行例

pdsdbrog コマンドの実行例

```
pdsdbrog /usr/control_rog_file
```

pdsdbrog 制御文（/usr/control_rog_file）の例

```
environment
  schema     = FMB2             ...1
  exectime   = 0                ...2
  recnomsg   = 1                ...3
  generation = 0                ...4
unload
  record     = FMB2_RCA         ...5
  server     = BES2             ...6
  area       = RDAREA023        ...7
  unldfile   = /usr/unld_data   ...8
  unldinf    = /usr/unld_result ...9
  prefix     = 1                ...10
  unldkind   = lod              ...11
dbinf
  userpflg   = (0)              ...12
```

[説明]

1. SDB データベース名を指定します。
2. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
3. アンロードが 1 万件完了するごとに経過メッセージを出力します。
4. アンロード対象のオリジナル RD エリアの世代番号を指定します。
5. ルートレコードのレコード型名を指定します。
6. アンロード対象の RD エリアがあるバックエンドサーバを指定します。
7. アンロード対象の RD エリアを指定します。オリジナル RD エリア名を指定してください。
8. アンロードデータファイル名を指定します。
9. 実行結果ファイル名を指定します。
10. プリフィクス部の長さを指定します。
11. データロードで使用できる入力データファイルの形式でアンロードデータファイルを出力します。
12. USER ポインタフラグの格納位置を指定します。

(3) データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema     = FMB2             ...1
  logmode    = no               ...2
  purge      = yes              ...3
  exectime   = 0                ...4
  recnomsg   = 1                ...5
  divermsg   = put              ...6
  generation = 0                ...7
load
  type       = tree             ...8
  record     = FMB2_RCA         ...9
  server     = BES2             ...10
  area       = RDAREA023        ...11
  data       = /usr/unld_data   ...12
  storinf    = /usr/load_result ...13
  errdata    = /usr/err_data    ...14
  prefix     = 1                ...15
  idxmode    = create           ...16
dbinf
  userpflg   = (0)              ...17
oreload
  midfile    = /usr/mid_file    ...18
```

[説明]

1. SDB データベース名を指定します。アンロード時に指定した SDB データベース名を指定します。
2. ログレスモードでデータロードを実行します。
3. 既存レコードを削除してデータロードを実行します。必ず yes を指定してください。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. データロードが 1 万件完了するごとに経過メッセージを出力します。
6. ほかの RD エリアに格納するべきレコードは論理エラーとします。
7. データロード対象のオリジナル RD エリアの世代番号を指定します。
8. 4V FMB の SDB データベースの場合は tree を指定します。
9. ルートレコードのレコード型名を指定します。アンロード時に指定したレコード型名を指定します。
10. データロード対象の RD エリアがあるバックエンドサーバを指定します。アンロード時に指定したバックエンドサーバ名を指定します。
11. データロード対象の RD エリアを指定します。アンロード時に指定したオリジナル RD エリア名を指定します。
12. (2)で作成したアンロードデータファイルを入力データファイルとして指定します。
13. 実行結果ファイル名を指定します。
14. 論理エラー情報ファイル名を指定します。
15. プリフィクス部の長さを指定します。

16. インデクス一括作成モードでインデクスを作成します。

17. USER ポインタフラグの開始位置を指定します。アンロード時に指定した格納位置を指定します。

18. 追い付き反映キー対応表の中間ファイル名を指定します。

(4) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB2
    DBTYPE 4V FMB
    RECORD FMB2_RCA
      2 DBKEY
       3  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  USERDA0             CHARACTER 30   TYPE  U,D
    RECORD FMB2_RCB
      2  KEYDATA1            XCHARACTER 1   TYPE  K,A
      2  DBKEY               INTEGER        TYPE  K,N
      2  USERDB0             CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB2_RCA
        ORDER LAST
      MEMBER FMB2_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
      SETOPTION
        OCCURRENCE NUMBER 2147483647
    SDBOPTION
      FUNCTION
        REFER     USE
        ADD       USE
        UPDATE    USE
        ERASE     USE
        ALLERASE  NOUSE
        DBLODUTL  USE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB2 FOR FMB2
  DBTYPE 4V
  STORAGE RECORD FMB2_RCA
    SEQUENTIAL FMB2_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN ((RDAREA025),(RDAREA026),(RDAREA027),(RDAREA028))
    WITHIN ((RDAREA021) KEYDATA1=(X'F3'),
            (RDAREA022) KEYDATA1=(X'F4'),
            (RDAREA023) KEYDATA1=(X'F5'),
            (RDAREA024) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB2_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA021),(RDAREA022),(RDAREA023),(RDAREA024))
  SET SETAB
    OWNER POINTER FOR FIRST LAST USER MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  SDBOPTION
  END STORAGE SCHEMA
```

```
*ENTRY DIRECTORY
  DBSCHM FMB2
```

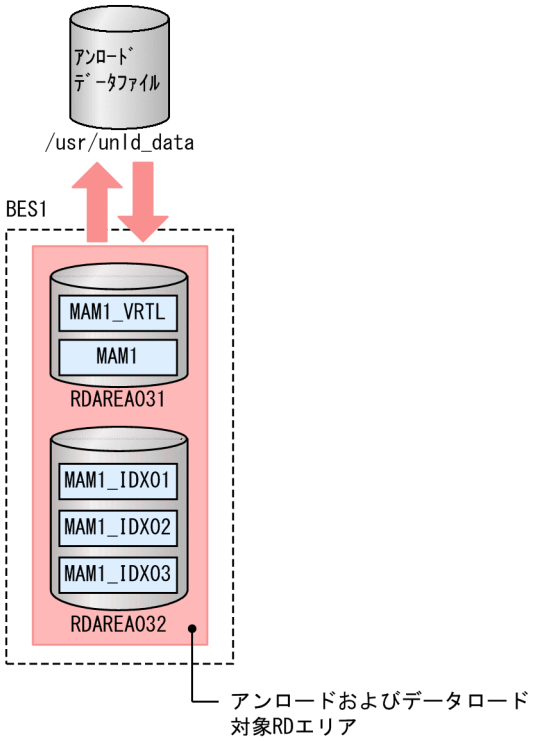
13.10.3　4V MAM の SDB データベースのアンロードおよびデータロード

例題

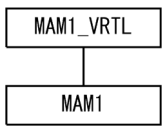
SDB データベースを再編成するために，SDB データベースをアンロードします。そのあと，作成したアンロードデータファイルを入力データファイルにしてデータロードします。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V MAM です。
- インデクス MAM1_IDX01～MAM1_IDX03 が定義されています。

(2) アンロードの実行例

pdsdbrog コマンドの実行例

```
pdsdbrog /usr/control_rog_file
```

pdsdbrog 制御文（/usr/control_rog_file）の例

```
environment
  schema   = MAM1              ...1
  exectime = 0                 ...2
  recnomsg = 1                 ...3
unload
  record   = MAM1_VRTL         ...4
  server   = BES1              ...5
  unldfile = /usr/unld_data    ...6
  unldinf  = /usr/unld_result  ...7
  prefix   = 0                 ...8
  unldkind = lod               ...9
```

[説明]

1. SDB データベース名を指定します。
2. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
3. アンロードが 1 万件完了するごとに経過メッセージを出力します。
4. 仮想ルートレコードのレコード型名を指定します。
5. アンロード対象の RD エリアがあるバックエンドサーバを指定します。
6. アンロードデータファイル名を指定します。
7. 実行結果ファイル名を指定します。
8. プリフィクス部の長さを指定します。
9. データロードで使用できる入力データファイルの形式でアンロードデータファイルを出力します。

(3) データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = MAM1              ...1
  logmode  = no                ...2
  purge    = yes               ...3
  exectime = 0                 ...4
  recnomsg = 1                 ...5
```

```
 load
   type     = single            ...6
   record   = MAM1_VRTL         ...7
   server   = BES1              ...8
   data     = /usr/unld_data    ...9
   storinf  = /usr/load_result  ...10
   errdata  = /usr/err_data     ...11
   prefix   = 0                 ...12
   idxmode  = create            ...13
```

[説明]

1. SDB データベース名を指定します。アンロード時に指定した SDB データベース名を指定します。
2. ログレスモードでデータロードを実行します。
3. 既存レコードを削除してデータロードを実行します。必ず yes を指定してください。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. データロードが 1 万件完了するごとに経過メッセージを出力します。
6. 4V AFM の SDB データベースの場合は single を指定します。
7. 仮想ルートレコードのレコード型名を指定します。アンロード時に指定したレコード型名を指定します。
8. データロード対象の RD エリアがあるバックエンドサーバを指定します。アンロード時に指定したバックエンドサーバ名を指定します。
9. (2)で作成したアンロードデータファイルを入力データファイルとして指定します。
10. 実行結果ファイル名を指定します。
11. 論理エラー情報ファイル名を指定します。
12. プリフィクス部の長さを指定します。
13. インデクス一括作成モードでインデクスを作成します。

(4) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA MAM1
   DBTYPE 4V  MAM
   RECORD MAM1_VRTL
    02 DBKEY01 CHARACTER 4         TYPE D,L
    02 DBKEY02 CHARACTER 4         TYPE K,L
   RECORD MAM1
    02 DBKEY01     CHARACTER 4     TYPE D,L
    02 DBKEY
     03 DBKEY02    CHARACTER 4     TYPE K,L
     03 DBKEYKN    INTEGER         TYPE K,N
    02 SORTKEY1 XCHARACTER 6       TYPE U,K,X'0000'
    02 USERFLD1 XCHARACTER 3       TYPE U,D,X'0000'
    02 USERFLD2 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD3 XCHARACTER 3       TYPE U,D,X'0000'
    02 USERFLD4 XCHARACTER 1       TYPE U,D,X'0000'
```

```
    02 USERFLD5 XCHARACTER 2       TYPE U,D,X'0000'
    02 USERFLD6 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD7 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD8 XCHARACTER 1       TYPE U,D,X'0000'
    02 USERFLD9 XCHARACTER 5       TYPE U,D,X'0000'
    02 USRDATA1 XCHARACTER 416     TYPE U,D,X'0000'
    FUNCTION
     REFER       USE
     ADD         USE
     UPDATE      USE
     ERASE     NOUSE
     ALLERASE  NOUSE
    RECORDID X'0001'
   SET SETM1
    OWNER MAM1_VRTL
     ORDER SORTED DUPLICATES PROHIBITED
    MEMBER MAM1
    INSERTION AUTOMATIC
    RETENTION FIXED
     KEY  ASCENDING  SORTKEY1
    SETOPTION
     OCCURRENCE NUMBER 65535
   REUSE NO
   SDBOPTION
     FUNCTION
      DBLODUTL    USE
      FORMAT    NOUSE
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA  MAM1  FOR  MAM1
   DBTYPE 4V
   STORAGE RECORD MAM1_VRTL
    SEQUENTIAL  MAM1_IDX01  FOR RECORD
     ORDER KEY ASCENDING  DBKEY01,DBKEY02
     WITHIN RDAREA032
    WITHIN RDAREA031
   STORAGE RECORD MAM1
    CLUSTERED SETM1
     WITHIN RDAREA031
     POINTER AREA SIZE 0
     SECONDARY INDEX MAM1_IDX02
       ORDER KEY ASCENDING  DBKEY01,DBKEY02,DBKEYKN
       WITHIN RDAREA032
     SECONDARY INDEX MAM1_IDX03
       ORDER KEY ASCENDING  DBKEY01,DBKEY02,SORTKEY1
       WITHIN RDAREA032
   SET SETM1
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
   SDBOPTION
     KEYDEF DBKEY01
      DATA 'MAM1'
     KEYDEF DBKEY02
      DATA 'AAAA'
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM MAM1
```

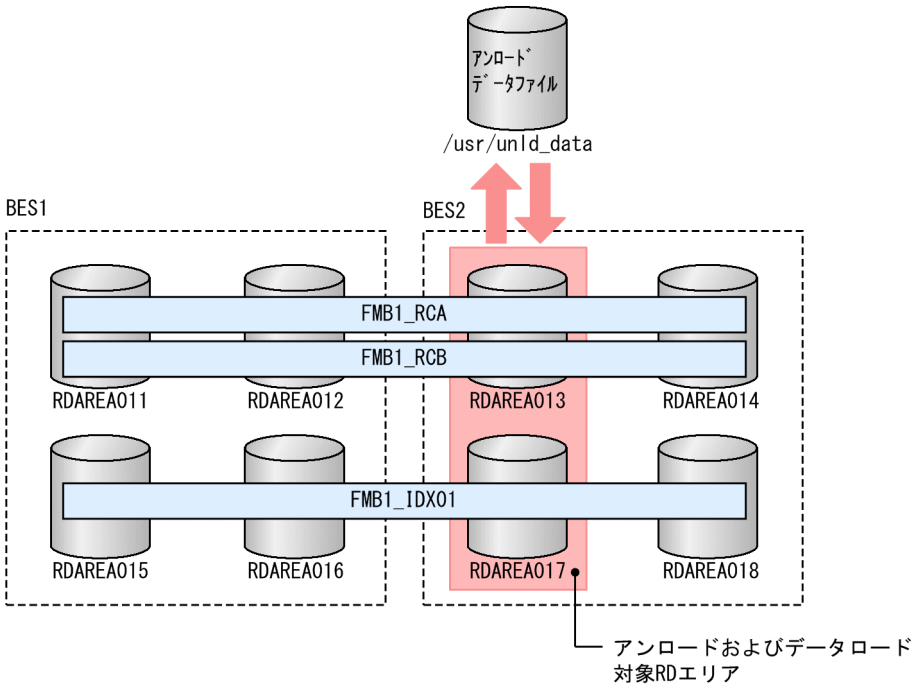
13.10.4　SD FMB の SDB データベースのアンロードおよびデータロード

例題

SDB データベースを再編成するために，横分割されている SDB データベースをアンロードします。そのあと，作成したアンロードデータファイルを入力データファイルにしてデータロードします。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造

条件

- SDB データベース種別は SD FMB です。
- インデクス FMB1_IDX01 が定義されています。
- アンロードおよびデータロードの対象は，RDAREA013 に格納されているレコードと，RDAREA017 に格納されているインデクスです。

(2)　アンロードの実行例

pdsdbrog コマンドの実行例

```
pdsdbrog /usr/control_rog_file
```

pdsdbrog 制御文（/usr/control_rog_file）の例

```
environment
  schema   = FMB1               ...1
  exectime = 0                  ...2
  recnomsg = 1                  ...3
unload
  record   = FMB1_RCA           ...4
  server   = BES2               ...5
  area     = RDAREA013          ...6
  unldfile = /usr/unld_data     ...7
  unldinf  = /usr/unld_result   ...8
  prefix   = 0                  ...9
  unldkind = lod                ...10
```

[説明]

1. SDB データベース名を指定します。
2. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
3. アンロードが 1 万件完了するごとに経過メッセージを出力します。
4. ルートレコードのレコード型名を指定します。
5. アンロード対象の RD エリアがあるバックエンドサーバを指定します。
6. アンロード対象の RD エリアを指定します。
7. アンロードデータファイル名を指定します。
8. 実行結果ファイル名を指定します。
9. プリフィクス部の長さに 0 を指定します。
10. データロードで使用できる入力データファイルの形式でアンロードデータファイルを出力します。

(3) データロードの実行例

pdsdblod コマンドの実行例

```
pdsdblod /usr/control_lod_file
```

pdsdblod 制御文（/usr/control_lod_file）の例

```
environment
  schema   = FMB1               ...1
  logmode  = no                 ...2
  purge    = yes                ...3
  exectime = 0                  ...4
  recnomsg = 1                  ...5
  divermsg = put                ...6
load
```

```
type     = tree               ...7
record   = FMB1_RCA           ...8
server   = BES2               ...9
area     = RDAREA013          ...10
data     = /usr/unld_data     ...11
storinf  = /usr/load_result   ...12
errdata  = /usr/err_data      ...13
prefix   = 0                  ...14
idxmode  = create             ...15
```

[説明]

1. SDB データベース名を指定します。アンロード時に指定した SDB データベース名を指定します。
2. ログレスモードでデータロードを実行します。
3. 既存レコードを削除してデータロードを実行します。必ず yes を指定してください。
4. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
5. データロードが 1 万件完了するごとに経過メッセージを出力します。
6. ほかの RD エリアのレコードは論理エラーとします。
7. SD FMB の SDB データベースの場合は tree を指定します。
8. ルートレコードのレコード型名を指定します。アンロード時に指定したレコード型名を指定します。
9. データロード対象の RD エリアがあるバックエンドサーバを指定します。アンロード時に指定したバックエンドサーバ名を指定します。
10. データロード対象の RD エリアを指定します。アンロード時に指定した RD エリア名を指定します。
11. (2)で作成したアンロードデータファイルを入力データファイルとして指定します。
12. 実行結果ファイル名を指定します。
13. 論理エラー情報ファイル名を指定します。
14. プリフィクス部の長さに 0 を指定します。
15. インデクス一括作成モードでインデクスを作成します。

(4) SDB データベースの定義

```
*ENTRY DICTIONARY
  SCHEMA FMB1
    DBTYPE SD FMB
    RECORD FMB1_RCA
      2 KEYDATA
       3  KEYDATA1          XCHARACTER 1   TYPE  K,A
      2  USERDA0            CHARACTER 30   TYPE  U,D
    RECORD FMB1_RCB
      2  KEYDATA1           XCHARACTER 1   TYPE  K,A
      2  DBKEY              INTEGER        TYPE  K,N
      2  USERDB0            CHARACTER 20   TYPE  U,D
    SET SETAB
      OWNER FMB1_RCA
```

```
        ORDER LAST
      MEMBER FMB1_RCB
        INSERTION AUTOMATIC
        RETENTION FIXED
  END SCHEMA
*ENTRY DICTIONARY
  STORAGE SCHEMA FMB1 FOR FMB1
  DBTYPE SD
  STORAGE RECORD FMB1_RCA
    SEQUENTIAL FMB1_IDX01 FOR RECORD
      ORDER KEY ASCENDING  KEYDATA1
      WITHIN ((RDAREA015),(RDAREA016),(RDAREA017),(RDAREA018))
    WITHIN ((RDAREA011) KEYDATA1=(X'F3'),
            (RDAREA012) KEYDATA1=(X'F4'),
            (RDAREA013) KEYDATA1=(X'F5'),
            (RDAREA014) KEYDATA1=(X'F6'))
  STORAGE RECORD FMB1_RCB
    CLUSTERED SETAB
      WITHIN ((RDAREA011),(RDAREA012),(RDAREA013),(RDAREA014))
  SET SETAB
    OWNER POINTER FOR FIRST LAST MEMBER
    MEMBER POINTER FOR NEXT PRIOR TENANT
  END STORAGE SCHEMA
*ENTRY DIRECTORY
  DBSCHM FMB1
```

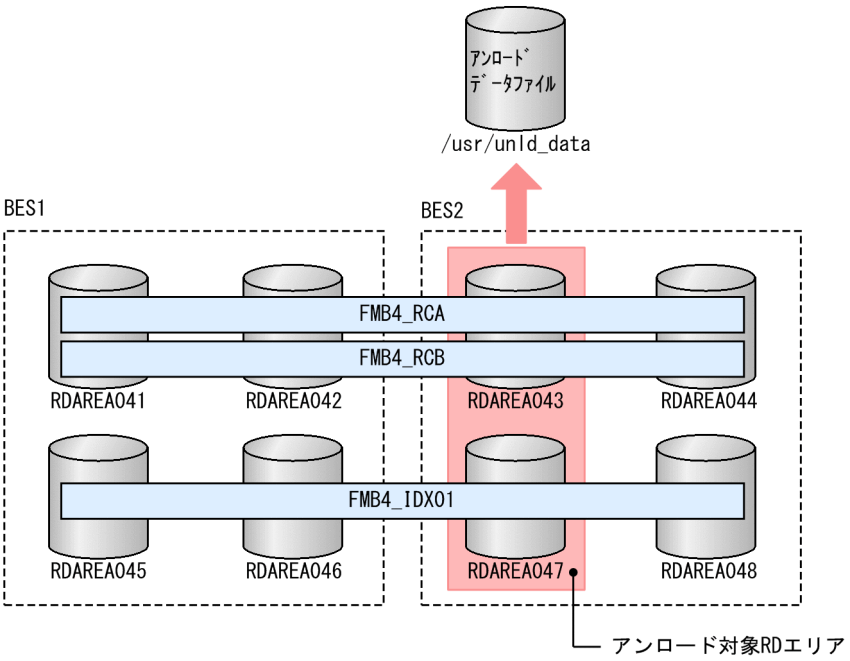
13.10.5　4V FMB の SDB データベースのアンロード（アンロードデータファイルを UAP の入力情報として使用する場合）

例題

横分割されている SDB データベースをアンロードします。作成したアンロードデータファイルは UAP の入力情報として使用します。

(1)　構成および条件

SDB データベースの構成



SDB データベースの構造

条件

- SDB データベース種別は 4V FMB です。
- インデクス FMB4_IDX01 が定義されています。
- アンロードの対象は，RDAREA043 に格納されているレコードです。
- USER ポインタフラグもアンロード対象にします。

(2) アンロードの実行例

pdsdbrog コマンドの実行例

```
pdsdbrog /usr/control_rog_file
```

pdsdbrog 制御文（/usr/control_rog_file）の例

```
environment
  schema   = FMB4              ...1
  exectime = 0                 ...2
  recnomsg = 1                 ...3
unload
  record   = FMB4_RCA          ...4
  server   = BES2              ...5
  area     = RDAREA043         ...6
```

```
  unldfile = /usr/unld_data     ...7
  unldinf  = /usr/unld_result   ...8
  prefix   = 94                 ...9
  unldkind = lod_4v             ...10
dbinf
  userpflg = (77)               ...11
```

[説明]

1. SDB データベース名を指定します。
2. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
3. アンロードが 1 万件完了するごとに経過メッセージを出力します。
4. ルートレコードのレコード型名を指定します。
5. アンロード対象の RD エリアがあるバックエンドサーバを指定します。
6. アンロード対象の RD エリアを指定します。
7. アンロードデータファイル名を指定します。
8. 実行結果ファイル名を指定します。
9. プリフィクス部の長さを指定します。必ず 94 を指定してください。
10. 作成したアンロードデータファイルを UAP の入力情報として使用するため，lod_4v を指定します。
11. USER ポインタフラグの格納位置を指定します。必ず 77 を指定してください。

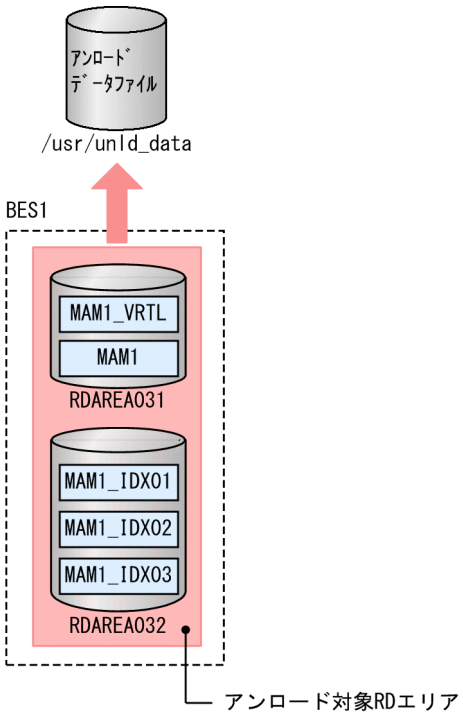
13.10.6 4V MAM の SDB データベースのアンロード（アンロードデータファイルを UAP の入力情報として使用する場合）

例題

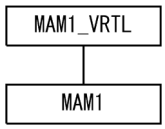
SDB データベースをアンロードします。作成したアンロードデータファイルは UAP の入力情報として使用します。

(1) 構成および条件

SDB データベースの構成



SDB データベースの構造



条件

- SDB データベース種別は 4V MAM です。
- インデクス MAM1_IDX01～MAM1_IDX03 が定義されています。
- アンロードの対象は，RDAREA031 に格納されているレコードです。

(2) アンロードの実行例

pdsdbrog コマンドの実行例

```
pdsdbrog /usr/control_rog_file
```

pdsdbrog 制御文（/usr/control_rog_file）の例

```
environment
  schema   = MAM1              ...1
  exectime = 0                 ...2
  recnomsg = 1                 ...3
unload
  record   = MAM1_VRTL         ...4
  server   = BES1              ...5
  unldfile = /usr/unld_data    ...6
```

```
unldinf  = /usr/unld_result  ...7
prefix   = 94                ...8
unldkind = lod_4v            ...9
```

[説明]

1. SDB データベース名を指定します。
2. コマンドの実行時間の上限を指定します。0 を指定した場合，上限は設定されません。
3. アンロードが 1 万件完了するごとに経過メッセージを出力します。
4. 仮想ルートレコードのレコード型名を指定します。
5. アンロード対象の RD エリアがあるバックエンドサーバを指定します。
6. アンロードデータファイル名を指定します。
7. 実行結果ファイル名を指定します。
8. プリフィクス部の長さを指定します。必ず 94 を指定してください。
9. 作成したアンロードデータファイルを UAP の入力情報として使用するため，lod_4v を指定します。

14 HiRDB/SD データベースアクセスユティリティ (pdsdbexe)

この章では，HiRDB/SD データベースアクセスユティリティの機能と使い方について説明します。

なお，この章では，HiRDB/SD データベースアクセスユティリティを pdsdbexe コマンドと表記しています。

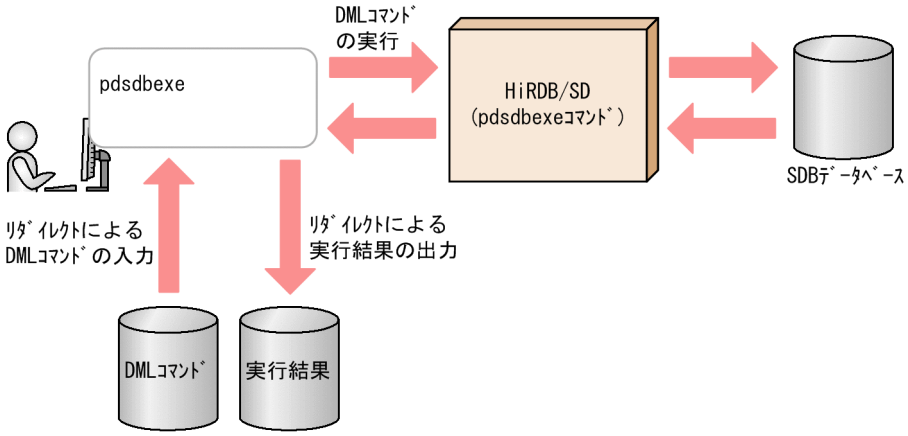
14.1　機能

pdsdbexe コマンドを使用すると，SDB データベースを操作する DML コマンドを標準入力から実行できます。そのため，UAP を作成しなくても，SDB データベースのレコードの検索や，レコード実現値の格納，更新，削除などができます。

また，リダイレクトによって，標準入力および標準出力をファイルに切り替えることができるため，ファイル中に記述した DML コマンドを実行し，その結果をファイルに出力することもできます。

pdsdbexe コマンドの概要を次の図に示します。

図 14-1　pdsdbexe コマンドの概要



参考

pdsdbexe コマンドは，UAP と同等の位置づけです。

また，TP1/FSP が起動していない環境でも実行できます。

SDB データベース種別が SD FMB の SDB データベースに対してアクセスする場合，SDB 用 UAP 環境定義が有効になります。SDB 用 UAP 環境定義については，「9.3　SDB 用 UAP 環境定義【SD FMB】」を参照してください。

pdsdbexe コマンドを実行すると，SDB データベースを操作する DML コマンドのほかに，SQL コマンドおよび pdsdbexe サブコマンドも実行できます。DML コマンド，SQL コマンド，および pdsdbexe サブコマンドの総称を pdsdbexe 操作コマンドといいます。pdsdbexe 操作コマンドの一覧を次の表に示します。

表 14-1　pdsdbexe 操作コマンドの一覧

項番	コマンドの種類	コマンド名	機能
1	DML コマンド	CLEAR	指定したデータベースキーに対応したレコード実現値を一括して削除します。
2		ERASE	指定したレコード実現値を削除します。

項番	コマンドの種類	コマンド名	機能
3		FETCH	1 つのレコード実現値を取得します。また，操作対象レコードの位置づけを行います。
4		FETCHDB_ALL	ルートレコード以下のすべてのレコード実現値を取得します。
5		FINISH	HiRDB/SD に対して個別終了要求を行い，SDB データベースに対する操作の終了処理を行います。
6		MODIFY	1 つのレコード実現値を更新します。
7		START	HiRDB/SD に対して個別開始要求を行い，SDB データベースに対する操作の開始処理を行います。
8		STORE	1 つのレコード実現値を格納します。
9	SQL コマンド	COMMIT	トランザクションをコミットして正常終了します。
10		CONNECT	HiRDB に接続します。
11		DISCONNECT	実行中のトランザクションのコミット後に，HiRDB からの切り離しを実行します。
12		ROLLBACK	トランザクションを取り消します。
13	pdsdbexe サブコマンド	#EXIT	HiRDB サーバからの切り離しを実行し，pdsdbexe コマンドを終了します。
14		#HELP	pdsdbexe 操作コマンドの指定形式と指定規則を出力します。
15		#USAGE	SDB データベースまたはレコードに対する DML コマンドのひな形を出力します。

SDB データベースに対して実行できる操作は，SDB データベース種別によって異なります。そのため，上記の表に示す DML コマンドには，実行できる SDB データベースの種別が限定されている DML コマンドがあります。SDB データベースに対して実行できる操作については，「2.5.1(1)　SDB データベースに対する操作（API)」，または「2.5.2(1)　SDB データベースに対する操作（DML)」を参照してください。

14.2　コマンドの形式

```
pdsdbexe　〔-u 認可識別子〔/パスワード〕〕
          〔-h ホスト名〕
          〔-n ポート番号〕
          〔-d　{4V｜SD}〕
```

●-u 認可識別子〔/パスワード〕

認可識別子：～<識別子>((1～30 文字))

パスワード：～<識別子>((1～30 文字))

HiRDB に接続する認可識別子とパスワードを指定します。

指定規則を次に示します。

- 認可識別子またはパスワードの文字列中に英小文字がある場合は，認可識別子およびパスワードを引用符（ " ）で囲んでください。また，-u オプションの指定値全体をアポストロフィ（ ' ）で囲んでください。

  （例）-u '"User01"/"Pswd01"'

  引用符で囲んで指定しないと，認可識別子およびパスワードの文字列はすべて英大文字と見なされます。例えば，「-u User01/Pswd01」と指定した場合，「-u USER01/PSWD01」と指定したと見なされます。

なお，このオプションを省略した場合（認可識別子およびパスワードの両方を省略した場合），pdsdbexe コマンドの開始時に認可識別子およびパスワードの入力要求があります。

また，認可識別子を指定してパスワードを省略した場合，pdsdbexe コマンドの開始時にパスワードの入力要求があります。

このときの認可識別子およびパスワードの指定形式については，「14.7.2(4)(a)　コマンド入力画面で CONNECT コマンドを実行する場合」を参照してください。

●-h ホスト名

～<識別子>((1～511 バイト))

接続先の HiRDB のホスト名を指定します。ホスト名の指定形式は，クライアント環境定義の PDHOST と同じです。PDHOST については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

このオプションを省略した場合，クライアント環境定義の PDHOST の指定値が適用されます。そのため，PDHOST の指定が省略されていると，pdsdbexe コマンドを実行した際にエラーとなります。

●-n ポート番号

～<符号なし整数>((5,001～65,535))

接続先の HiRDB のポート番号を指定します。

このオプションを省略した場合，クライアント環境定義の PDNAMEPORT の指定値が適用されます。そのため，PDNAMEPORT の指定が省略されていると，pdsdbexe コマンドを実行した際にエラーとなります。

PDNAMEPORT については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

●-d {4V | SD}

アクセス対象の SDB データベースの種別を指定します。

4V：

SDB データベース種別が 4V のデータベースにアクセスする場合に指定します。

SD：

SDB データベース種別が SD のデータベースにアクセスする場合に指定します。

このオプションを省略した場合，4V が仮定されます。

SDB データベース種別によって DML コマンドの形式が異なります。

-d オプションの指定値と DML コマンドの対応を次の表に示します。

表 14-2 -d オプションの指定値と DML コマンドの対応

項番	-d オプションの指定値	DML コマンドの参照先
1	4V	「14.4 DML コマンド【4V FMB，4V AFM】」を参照してください
2	SD	「14.5 DML コマンド【SD FMB】」を参照してください

■pdsdbexe コマンドの指定例

次に示す条件で pdsdbexe コマンドを実行する場合の指定例を示します。

- 認可識別子 User01，パスワード Pswd01 で HiRDB に接続します。
- 接続先の HiRDB のホスト名は host01 とします。
- 接続先の HiRDB のポート番号は 20000 とします。
- 操作対象の SDB データベースの種別は SD とします。

```
pdsdbexe -u '"User01"/"Pswd01"' -h host01 -n 20000 -d SD
```

14.3　pdsdbexe コマンドの実行手順

pdsdbexe コマンドの実行手順を例題形式で説明します。

なお，例題中に出てくる DML コマンド，SQL コマンド，および pdsdbexe サブコマンドについては，次の個所を参照してください。

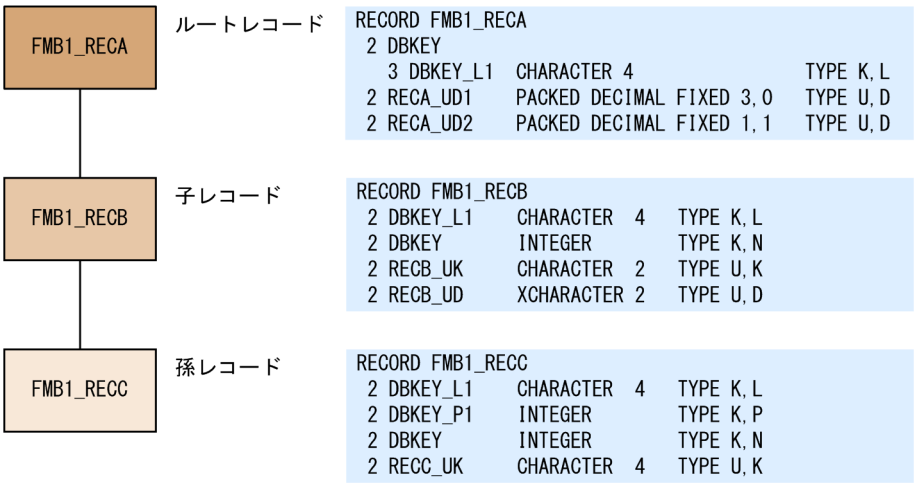
- 「14.4　DML コマンド【4V FMB，4V AFM】」
- 「14.5　DML コマンド【SD FMB】」
- 「14.6　DML コマンド共通の規則」
- 「14.7　SQL コマンド」
- 「14.8　pdsdbexe サブコマンド」

14.3.1　4V FMB の SDB データベースのレコードを更新する

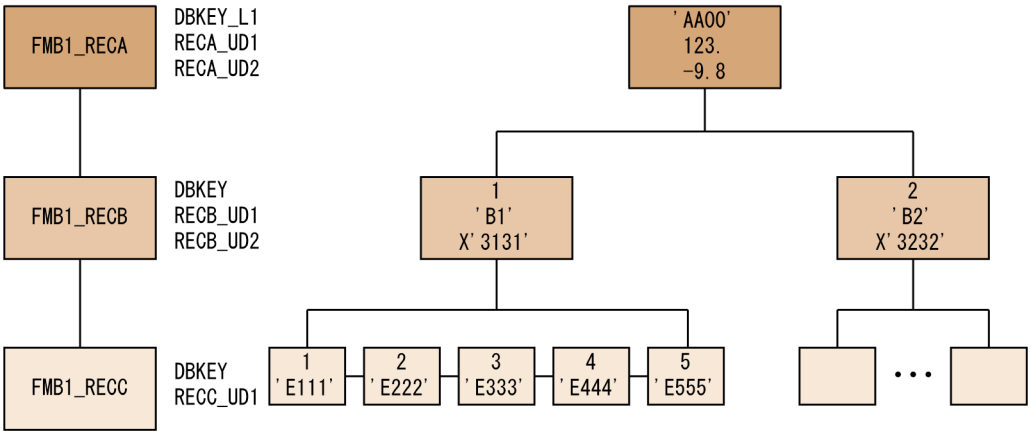
4V FMB の SDB データベースのレコードを更新します。

- SDB データベース名：FMB1
- 更新対象のレコード名：FMB1_RECB

■4V FMB の SDB データベースの階層構造



■格納されているレコード実現値



手順

1. pdsdbexe コマンドを実行する

```
pdsdbexe
```

-u オプションに認可識別子を指定した場合は，手順の 2.をスキップして，手順の 3.に進んでください。-u オプションに認可識別子およびパスワードを指定した場合は，手順の 2.～3.をスキップして，手順の 4.に進んでください。

2. 認可識別子の入力要求に応答する

```
pdsdbexe **** SDB Database Executor 09-60    ****  2015-12-14 16:49:44

user-id:"User01"     ←認可識別子を指定します。
```

HiRDB に接続する認可識別子を指定して，Enter キーを押してください。

- 認可識別子に英小文字がある場合は，引用符（ " ）で囲んでください。

3. パスワードの入力要求に応答する

```
pdsdbexe **** SDB Database Executor 09-60    ****  2015-12-14 16:49:44

user-id:"User01"
"User01"
password:_____←パスワードを指定します。
```

認可識別子のパスワードを指定して，Enter キーを押してください。

- パスワードに英小文字がある場合は，引用符（ " ）で囲んでください。
- パスワードに入力した文字は表示されません。

4. START コマンドを実行して個別開始する

```
START "FMB1"
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8

     ACSMODE UPDATE
```

```
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8
     ENQOPT EXCLUSIVE;
```

START コマンドの実行結果：

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2015-12-14 16:20:49.16
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 09-60
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME "FMB1"
  HANDLE NUMBER                   : 1
  ACSMODE                         : UPDATE
  ENQOPT                          : EXCLUSIVE
  AUTODEQ                         : NO
  RDAREAVAL                       : NO
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

5. FETCH コマンドを実行して，ルートレコードの検索を行う

```
FETCH "FMB1_RECA" FIRST;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY_L1"                  'AA00'
      2 "RECA_UD1"                  123.
      3 "RECA_UD2"                  -9.8
  ROWID                           : X'0700000019000000000000100'
  PAGE NUM                        : 100
KFPB64050-I FETCH command processing completed
```

6. FETCH コマンドを実行して，更新対象レコードへの位置づけを行う

```
FETCH "FMB1_RECB" LAST WHERE (DBKEY=2);
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                     2
      2 "RECB_UK"                   'B2'
      3 "RECB_UD"                   X'3232'
  FETCH SERIAL NUM                : 2
  ROWID                           : X'0700000019000000000000800'
  USER POINTER SERIAL NUM         : 0
  SECOND INDEX KIND               : NOUSE
  RECORD POINTER STATUS           : LAST
KFPB64050-I FETCH command processing completed
```

7. MODIFY コマンドを実行して，レコードを更新する

```
MODIFY "FMB1_RECB" SET "RECB_UD" = X'3333';
```

8. 更新対象のレコードが正しいかを確認する

```
MODIFY DATA
    1 "RECB_UD"
      <BEFORE> X'3232'
      <AFTER>  X'3333'
AGREE TO MODIFY ? [ENTER YES OR NO]
YES   ←上記の情報を確認後，YESを指定します。
```

MODIFY コマンドの実行結果：

```
MODIFY INFORMATION
  ROWID                          : X'0700000019000000000000800'
KFPB64050-I MODIFY command processing completed
```

9. FETCH コマンドを実行して，レコードが更新されたことを確認する

```
FETCH "FMB1_RECB" LAST WHERE (DBKEY=2);
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          2
      2 "RECB_UK"                        'B2'
      3 "RECB_UD"                        X'3333'
  FETCH SERIAL NUM               : 2
  ROWID                          : X'0700000019000000000000800'
  SECOND INDEX KIND              : NOUSE
  RECORD POINTER STATUS          : LAST
KFPB64050-I FETCH command processing completed
```

10. COMMIT コマンドを実行して，レコードの更新を確定する

```
COMMIT;
```

COMMIT コマンドの実行結果：

```
KFPB64050-I COMMIT command processing completed
```

11. #EXIT コマンドを実行して，pdsdbexe コマンドを終了する

```
#EXIT;
```

#EXIT コマンドの実行結果：

```
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

表示されるプロンプトについて

pdsdbexe コマンドを実行すると，次の表に示すどれかのプロンプトが表示されます。

表 14-3　入力画面で表示されるプロンプト

項番	プロンプトの表示形式	意味
1	user-id:	HiRDB に接続する認可識別子の入力待ち状態です。 pdsdbexe コマンドの実行後，または SQL コマンドの CONNECT コマンドの実行後に表示されます。ただし，pdsdbexe コマンドまたは CONNECT コマンドの実行時，認可識別子を指定した場合は，このプロンプトは表示されません。
2	password:	パスワードの入力待ち状態です。 pdsdbexe コマンドの実行後，または SQL コマンドの CONNECT コマンドの実行後に表示されます。ただし，pdsdbexe コマンドまたは CONNECT コマンドの実行時，パスワードを指定した場合は，このプロンプトは表示されません。 パスワードを入力しても，入力した文字は表示されません。
3	COMMAND ?	通常の入力待ち状態です。pdsdbexe 操作コマンドを指定できます。
4	NEXT ?	継続行の入力待ち状態です。 pdsdbexe 操作コマンドの終わりを示すセミコロン（;）が指定されない状態で Enter キーを押すと，その行は継続行（pdsdbexe 操作コマンドが途中である）と見なされます。このプロンプトが表示された場合，pdsdbexe 操作コマンドの続きを入力してください。

14.3.2　4V FMB の SDB データベースのレコードを削除する

4V FMB の SDB データベースのレコードを削除します。

- SDB データベース名：FMB1
- 削除対象のレコード名：FMB1_RECC

  4V FMB の SDB データベースの階層構造，および格納されているレコード実現値については，「14.3.1 4V FMB の SDB データベースのレコードを更新する」と同じです。

手順

1. pdsdbexe コマンドを実行する

```
pdsdbexe -u ’”User01”/”Pswd01”’
```

2. START コマンドを実行して個別開始する

```
START ”FMB1”
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8

     ACSMODE UPDATE
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8
     ENQOPT EXCLUSIVE;
```

START コマンドの実行結果：

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2015-01-15 14:58:41.91
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 09-60
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME "FMB1"
  HANDLE NUMBER                   : 1
  ACSMODE                         : UPDATE
  ENQOPT                          : EXCLUSIVE
  AUTODEQ                         : NO
  RDAREAVAL                       : NO
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

3. FETCH コマンドを実行して，ルートレコードの検索を行う

```
FETCH "FMB1_RECA" FIRST WHERE (DBKEY='AA00');
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY_L1"                    'AA00'
      2 "RECA_UD1"                    123.
      3 "RECA_UD2"                    -9.8
  ROWID                           : X'0700000019000000000000100'
  PAGE NUM                        : 100
KFPB64050-I FETCH command processing completed
```

4. FETCH コマンドを実行して，子レコードを検索する

```
FETCH "FMB1_RECB" FIRST WHERE (DBKEY=1);
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                       1
      2 "RECB_UK"                     'B1'
      3 "RECB_UD"                     X'3131'
  FETCH SERIAL NUM                : 1
  ROWID                           : X'0700000019000000000000200'
  USER POINTER SERIAL NUM         : 0
  SECOND INDEX KIND               : NOUSE
  RECORD POINTER STATUS           : FIRST
KFPB64050-I FETCH command processing completed
```

5. FETCH コマンドを実行して，削除対象レコードへの位置づけを行う

```
FETCH "FMB1_RECC" FIRST;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          1
      2 "RECC_UK"                        'E111'
  FETCH SERIAL NUM               : 1
  ROWID                          : X'0700000019000000000000300'
  SECOND INDEX KIND              : NOUSE
  RECORD POINTER STATUS          : FIRST
KFPB64050-I FETCH command processing completed
```

6. ERASE コマンドを実行して，レコードを削除する

```
ERASE "FMB1_RECC" TARGET OCCURRENCE;
```

7. 削除対象のレコードが正しいかを確認する

```
ERASE RECORD INFORMATION
  RECORD KIND                    : CHILD RECORD
  DBKEY                          : 1
AGREE TO ERASE DBKEY AND ALL ? [ENTER YES OR NO]
YES    ←上記の情報を確認後，YESを指定します。
```

ERASE コマンドの実行結果：

```
ERASE INFORMATION
KFPB64050-I ERASE command processing completed
```

8. FETCH コマンドを実行して，レコードが削除されたことを確認する

```
FETCH "FMB1_RECC" FIRST;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  SECOND INDEX KIND              : NOUSE
  FETCH TERMINATED               : NOUSE
KFPA12100-I No rows satisfying search condition
KFPB64050-I FETCH command processing completed
```

9. COMMIT コマンドを実行して，レコードの削除を確定する

```
COMMIT;
```

COMMIT コマンドの実行結果：

```
KFPB64050-I COMMIT command processing completed
```

10. #EXIT コマンドを実行して，pdsdbexe コマンドを終了する

```
#EXIT;
```

#EXIT コマンドの実行結果：

```
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

参考

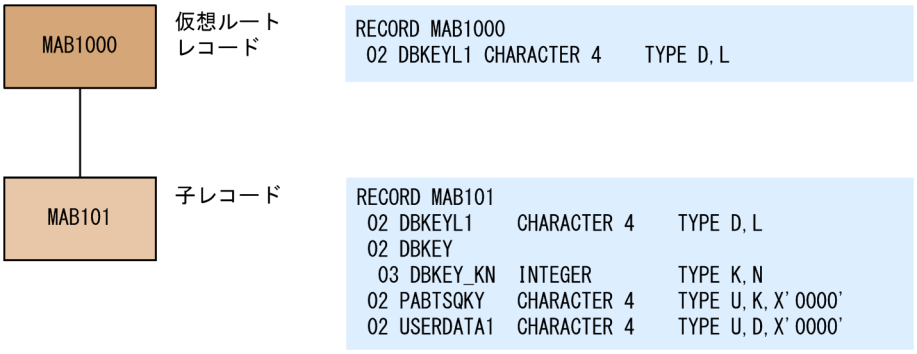
コマンド入力画面に表示されるプロンプトについては，「表 14-3　入力画面で表示されるプロンプト」を参照してください。

14.3.3　4V AFM の SDB データベースのレコードを更新する

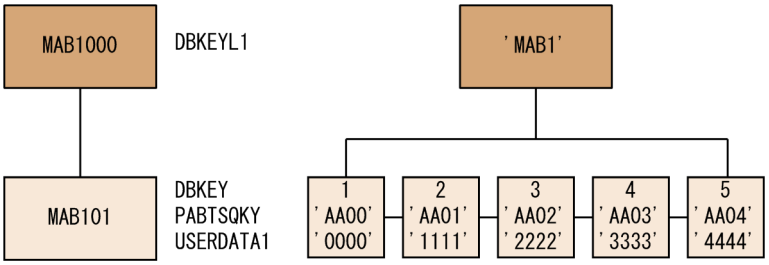
4V MAM の SDB データベースのレコードを更新します。

- SDB データベース種別：4V MAM
- SDB データベース名：MAB1
- 更新対象のレコード名：MAB101

■4V MAM の SDB データベースの階層構造



■格納されているレコード実現値



手順

1. pdsdbexe コマンドを実行する

```
pdsdbexe -u ’”User01”/”Pswd01”’
```

2. START コマンドを実行して個別開始する

```
START ”MAB1”
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8

     ACSMODE UPDATE
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8
     ENQOPT EXCLUSIVE;
```

START コマンドの実行結果：

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2015-12-14 16:20:49.16
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 09-60
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME ”MAB1”
  HANDLE NUMBER                   : 1
  ACSMODE                         : UPDATE
  ENQOPT                          : EXCLUSIVE
  AUTODEQ                         : NO
  RDAREAVAL                       : NO
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

3. FETCH コマンドを実行して，更新対象レコードへの位置づけを行う

```
FETCH ”MAB101” FIRST;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 ”DBKEY_KN”                  1
      2 ”PABTSQKY”                  ’AA00’
      3 ”USERDATA1”                 ’0000’
  FETCH SERIAL NUM                : 1
  ROWID                           : X’240000001A00000000000100’
  SECOND INDEX KIND               : NOUSE
  RECORD POINTER STATUS           : FIRST
KFPB64050-I FETCH command processing completed
```

4. MODIFY コマンドを実行して，レコードを更新する

```
MODIFY ”MAB101” SET ”USERDATA1” = ’MDFY’;
```

5. 更新対象のレコードが正しいかを確認する

```
MODIFY DATA
    1 "USERDATA1"
      <BEFORE> '0000'
      <AFTER>  'MDFY'
AGREE TO MODIFY ? [ENTER YES OR NO]
YES   ←上記の情報を確認後，YESを指定します。
```

MODIFY コマンドの実行結果：

```
MODIFY INFORMATION
  ROWID                          : X'240000001A00000000000100'
KFPB64050-I MODIFY command processing completed
```

6. FETCH コマンドを実行して，レコードが更新されたことを確認する

```
FETCH "MAB101" FIRST;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY_KN"                    1
      2 "PABTSQKY"                    'AA00'
      3 "USERDATA1"                   'MDFY'
  FETCH SERIAL NUM                : 1
  ROWID                           : X'240000001A00000000000100'
  SECOND INDEX KIND               : NOUSE
  RECORD POINTER STATUS           : FIRST
KFPB64050-I FETCH command processing completed
```

7. COMMIT コマンドを実行して，レコードの更新を確定する

```
COMMIT;
```

COMMIT コマンドの実行結果：

```
KFPB64050-I COMMIT command processing completed
```

8. #EXIT コマンドを実行して，pdsdbexe コマンドを終了する

```
#EXIT;
```

#EXIT コマンドの実行結果：

```
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

参考

コマンド入力画面に表示されるプロンプトについては，「表 14-3　入力画面で表示されるプロンプト」を参照してください。

14.3.4　4V AFM の SDB データベースのレコードを削除する

4V MAM の SDB データベースのレコードを削除します。

- SDB データベース種別：4V MAM
- SDB データベース名：MAB1
- 削除対象のレコード名：MAB101

  4V MAM の SDB データベースの階層構造，および格納されているレコード実現値については，「14.3.3　4V AFM の SDB データベースのレコードを更新する」と同じです。

手順

1. pdsdbexe コマンドを実行する

```
pdsdbexe -u ’”User01”/”Pswd01”’
```

2. START コマンドを実行して個別開始する

```
START ”MAB1”
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8

     ACSMODE UPDATE
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8
     ENQOPT EXCLUSIVE;
```

START コマンドの実行結果：

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2015-12-14 16:20:49.16
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 09-60
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME ”MAB1”
  HANDLE NUMBER                   : 1
  ACSMODE                         : UPDATE
  ENQOPT                          : EXCLUSIVE
  AUTODEQ                         : NO
  RDAREAVAL                       : NO
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

3. FETCH コマンドを実行して，削除対象レコードへの位置づけを行う

```
FETCH ”MAB101” FIRST WHERE (DBKEY=3);
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY_KN"                    3
      2 "PABTSQKY"                    'AA02'
      3 "USERDATA1"                   '2222'
  FETCH SERIAL NUM              : 3
  ROWID                         : X'240000001A00000000000300'
  SECOND INDEX KIND             : SERIAL NUM INDEX
  RECORD POINTER STATUS         : NONE
KFPB64050-I FETCH command processing completed
```

4. ERASE コマンドを実行して，レコードを削除する

```
ERASE "MAB101" TARGET SEGMENT;
```

5. 削除対象のレコードが正しいかを確認する

```
ERASE RECORD INFORMATION
  RECORD KIND                   : CHILD RECORD
  DBKEY                         : 3
AGREE TO ERASE DBKEY AND FOLLOWERS ? [ENTER YES OR NO]
YES    ←上記の情報を確認後，YESを指定します。
```

ERASE コマンドの実行結果：

```
ERASE INFORMATION
KFPB64050-I ERASE command processing completed
```

6. FETCH コマンドを実行して，レコードが削除されたことを確認する

```
FETCH "MAB101" FIRST WHERE (DBKEY=3);
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  SECOND INDEX KIND             : SERIAL NUM INDEX
  FETCH TERMINATED              : NOUSE
KFPA12100-I No rows satisfying search condition
KFPB64050-I FETCH command processing completed
```

7. COMMIT コマンドを実行して，レコードの削除を確定する

```
COMMIT;
```

COMMIT コマンドの実行結果：

```
KFPB64050-I COMMIT command processing completed
```

8. #EXIT コマンドを実行して，pdsdbexe コマンドを終了する

```
#EXIT;
```

#EXIT コマンドの実行結果：

```
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

参考

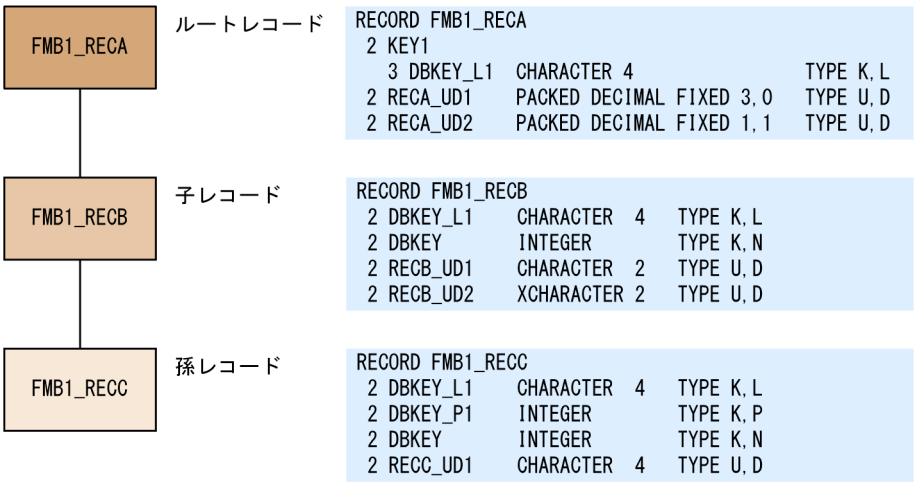
コマンド入力画面に表示されるプロンプトについては，「表 14-3　入力画面で表示されるプロンプト」を参照してください。

14.3.5　SD FMB の SDB データベースのレコードを検索する

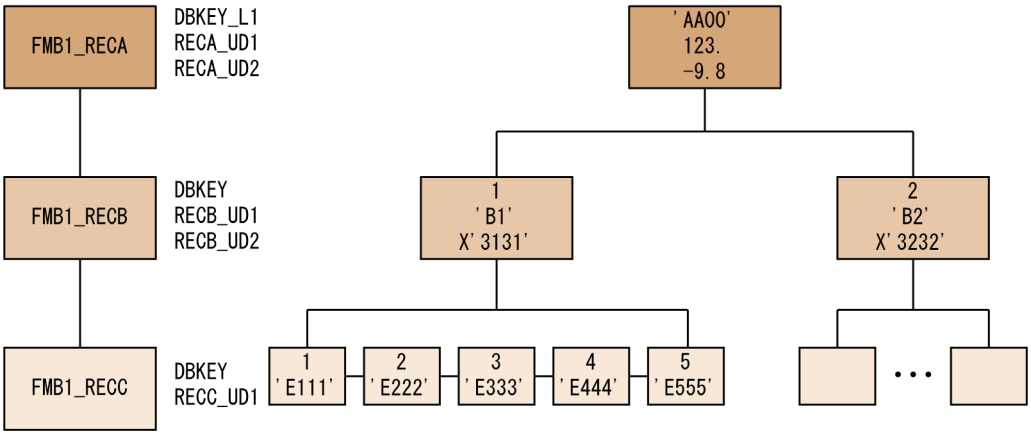
SD FMB の SDB データベースのレコードを検索します。

- SDB データベース名：FMB1
- 検索対象のレコード名：FMB1_RECA，FMB1_RECB

■SD FMB の SDB データベースの階層構造



■格納されているレコード実現値



手順

1. pdsdbexe コマンドを実行する

```
pdsdbexe -d SD
```

SD FMB の SDB データベースにアクセスする場合，-d オプションに SD を指定します。

-u オプションに認可識別子を指定した場合は，手順の 2.をスキップして，手順の 3.に進んでください。-u オプションに認可識別子およびパスワードを指定した場合は，手順の 2.～3.をスキップして，手順の 4.に進んでください。

2. 認可識別子の入力要求に応答する

```
pdsdbexe **** SDB Database Executor 10-01    ****  2018-12-14 16:49:44

user-id:"User01"     ←認可識別子を指定します。
```

HiRDB に接続する認可識別子を指定して，Enter キーを押してください。

- 認可識別子に英小文字がある場合は，引用符（ " ）で囲んでください。

3. パスワードの入力要求に応答する

```
pdsdbexe **** SDB Database Executor 10-01    ****  2018-12-14 16:49:44

user-id:"User01"
"User01"
password:_____←パスワードを指定します。
```

認可識別子のパスワードを指定して，Enter キーを押してください。

- パスワードに英小文字がある場合は，引用符（ " ）で囲んでください。
- パスワードに入力した文字は表示されません。

4. START コマンドを実行して個別開始する

```
START "FMB1";
```

START コマンドの実行結果：

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2018-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2018-12-14 16:20:49.16
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 10-01
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME "FMB1"
  HANDLE NUMBER                   : 1
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

5. FETCH コマンドを実行して，ルートレコードの検索を行う

```
FETCH FIRST "FMB1_RECA";
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY_L1"                    'AA00'
      2 "RECA_UD1"                    123.
      3 "RECA_UD2"                    -9.8
  ROWID                           : X'0700000019000000000000100'
KFPB64050-I FETCH command processing completed
```

6. FETCH コマンドを実行して，子レコードの検索を行う

```
FETCH FIRST "FMB1_RECB" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                       1
      2 "RECB_UD1"                    'B1'
      3 "RECB_UD2"                    X'3131'
  FETCH SERIAL NUM                : 1
  ROWID                           : X'0700000019000000000000800'
  RECORD POINTER STATUS           : FIRST
KFPB64050-I FETCH command processing completed
```

7. FETCH コマンドを実行して，子レコードのさらに子レコードの検索を行う

```
FETCH FIRST "FMB1_RECC" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                       1
      2 "RECC_UD1"                    'E111'
  FETCH SERIAL NUM                : 1
```

```
  ROWID                          : X'0700000019000000000000300'
  RECORD POINTER STATUS          : FIRST
KFPB64050-I FETCH command processing completed
```

8. #EXIT コマンドを実行して，pdsdbexe コマンドを終了する

```
#EXIT;
```

#EXIT コマンドの実行結果：

```
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

参考

コマンド入力画面に表示されるプロンプトについては，「表 14-3　入力画面で表示されるプロンプト」を参照してください。

14.3.6　SD FMB の SDB データベースのレコードを更新する

SD FMB の SDB データベースのレコードを更新します。

- SDB データベース名：FMB1
- 更新対象のレコード名：FMB1_RECB

SD FMB の SDB データベースの階層構造，および格納されているレコード実現値については，「14.3.5 SD FMB の SDB データベースのレコードを検索する」と同じです。

手順

1. pdsdbexe コマンドを実行する

```
pdsdbexe -d SD
```

SD FMB の SDB データベースにアクセスする場合，-d オプションに SD を指定します。

-u オプションに認可識別子を指定した場合は，手順の 2.をスキップして，手順の 3.に進んでください。-u オプションに認可識別子およびパスワードを指定した場合は，手順の 2.～3.をスキップして，手順の 4.に進んでください。

2. 認可識別子の入力要求に応答する

```
pdsdbexe **** SDB Database Executor 10-02    ****  2019-04-16 16:25:23

user-id:"User01"     ←認可識別子を指定します。
```

HiRDB に接続する認可識別子を指定して，Enter キーを押してください。

- 認可識別子に英小文字がある場合は，引用符（ " ）で囲んでください。

3. パスワードの入力要求に応答する

```
pdsdbexe **** SDB Database Executor 10-02    ****  2019-04-16 16:25:23

user-id:"User01"
"User01"
password:_____←パスワードを指定します。
```

認可識別子のパスワードを指定して，Enter キーを押してください。

- パスワードに英小文字がある場合は，引用符（ " ）で囲んでください。
- パスワードに入力した文字は表示されません。

4. START コマンドを実行して個別開始する

```
START "FMB1";
```

START コマンドの実行結果：

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2019-04-16 16:24:26.64
  SDB DIRECTORY UPDATE DATE       : 2019-04-16 16:26:33.61
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 10-02
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME "FMB1"
  HANDLE NUMBER                   : 1
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

5. FETCH コマンドを実行して，ルートレコードの検索を行う

```
FETCH FIRST "FMB1_RECA";
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
   FETCH DATA
       1 "DBKEY_L1"                    'AA00'
       2 "RECA_UD1"                    123.
       3 "RECA_UD2"                    -9.8
   ROWID                           : X'1C000000190000000000000100'
KFPB64050-I FETCH command processing completed
```

6. FETCH コマンドを実行して，子レコードの検索を行う

```
FETCH FIRST "FMB1_RECB" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          1
      2 "RECB_UD1"                       'B1'
      3 "RECB_UD2"                       X'3131'
  FETCH SERIAL NUM               : 1
  ROWID                          : X'1C000000190000000000200'
  RECORD POINTER STATUS          : FIRST
KFPB64050-I FETCH command processing completed
```

7. FETCH コマンドを実行して，更新対象レコードへの位置づけを行う

```
FETCH FOR UPDATE NEXT "FMB1_RECB" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          2
      2 "RECB_UD1"                       'B2'
      3 "RECB_UD2"                       X'3232'
  FETCH SERIAL NUM               : 2
  ROWID                          : X'1C000000190000000000800'
  RECORD POINTER STATUS          : LAST
KFPB64050-I FETCH command processing completed
```

8. MODIFY コマンドを実行して，レコードを更新する

```
MODIFY "FMB1_RECB" SET "RECB_UD2" = X'3333';
```

9. 更新対象のレコードが正しいかを確認する

```
MODIFY DATA
    1 "RECB_UD2"
      <BEFORE> X'3232'
      <AFTER>  X'3333'
AGREE TO MODIFY ? [ENTER YES OR NO]
YES    ←上記の情報を確認後，YESを指定します。
```

MODIFY コマンドの実行結果：

```
MODIFY INFORMATION
  ROWID                          : X'1C000000190000000000800'
KFPB64050-I MODIFY command processing completed
```

10. FETCH コマンドを実行して，子レコードの検索を行う

```
FETCH FIRST "FMB1_RECB" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          1
      2 "RECB_UD1"                       'B1'
```

```
      3 "RECB_UD2"                    X'3131'
  FETCH SERIAL NUM              : 1
  ROWID                         : X'1C00000019000000000000200'
  RECORD POINTER STATUS         : FIRST
KFPB64050-I FETCH command processing completed
```

11. FETCH コマンドを実行して，レコードが更新されたことを確認する

```
FETCH NEXT "FMB1_RECB" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                     2
      2 "RECB_UD1"                  'B2'
      3 "RECB_UD2"                  X'3333'
  FETCH SERIAL NUM              : 2
  ROWID                         : X'1C00000019000000000000800'
  RECORD POINTER STATUS         : LAST
KFPB64050-I FETCH command processing completed
```

12. COMMIT コマンドを実行して，レコードの更新を確定する

```
COMMIT;
```

COMMIT コマンドの実行結果：

```
KFPB64050-I COMMIT command processing completed
```

13. #EXIT コマンドを実行して，pdsdbexe コマンドを終了する

```
#EXIT;
```

#EXIT コマンドの実行結果：

```
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

参考

コマンド入力画面に表示されるプロンプトについては，「表 14-3　入力画面で表示されるプロンプト」を参照してください。

14.3.7　SD FMB の SDB データベースのレコードを削除する

SD FMB の SDB データベースのレコードを削除します。

- SDB データベース名：FMB1

- 削除対象のレコード名：FMB1_RECC

SD FMB の SDB データベースの階層構造，および格納されているレコード実現値については，「14.3.5 SD FMB の SDB データベースのレコードを検索する」と同じです。

手順

1. pdsdbexe コマンドを実行する

```
pdsdbexe -u '"User01"/"Pswd01"'  -d SD
```

SD FMB の SDB データベースにアクセスする場合，-d オプションに SD を指定します。

2. START コマンドを実行して個別開始する

```
START "FMB1";
```

START コマンドの実行結果：

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2019-04-16 16:24:26.64
  SDB DIRECTORY UPDATE DATE       : 2019-04-16 16:48:35.13
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 10-02
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME "FMB1"
  HANDLE NUMBER                   : 1
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

3. FETCH コマンドを実行して，ルートレコードの検索を行う

```
FETCH FIRST "FMB1_RECA";
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY_L1"                  'AA00'
      2 "RECA_UD1"                  123.
      3 "RECA_UD2"                  -9.8
  ROWID                           : X'1C000000190000000000000100'
KFPB64050-I FETCH command processing completed
```

4. FETCH コマンドを実行して，子レコードを検索する

```
FETCH FIRST "FMB1_RECB" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          1
      2 "RECB_UD1"                       'B1'
      3 "RECB_UD2"                       X'3131'
  FETCH SERIAL NUM               : 1
  ROWID                          : X'1C000000190000000000200'
  RECORD POINTER STATUS          : FIRST
KFPB64050-I FETCH command processing completed
```

5. FETCH コマンドを実行して，削除対象レコードへの位置づけを行う

```
FETCH FOR UPDATE FIRST "FMB1_RECC" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          1
      2 "RECC_UD1"                       'E111'
  FETCH SERIAL NUM               : 1
  ROWID                          : X'1C000000190000000000300'
  RECORD POINTER STATUS          : FIRST
KFPB64050-I FETCH command processing completed
```

6. ERASE コマンドを実行して，レコードを削除する

```
ERASE "FMB1_RECC" ALL;
```

7. 削除対象のレコードが正しいかを確認する

```
ERASE RECORD INFORMATION
  RECORD KIND                    : CHILD RECORD
  DBKEY                          : 1
AGREE TO ERASE DBKEY AND FOLLOWERS ? [ENTER YES OR NO]
YES    ←上記の情報を確認後，YESを指定します。
```

ERASE コマンドの実行結果：

```
KFPB64050-I ERASE command processing completed
```

8. FETCH コマンドを実行して，レコードが削除されたことを確認する

```
FETCH FIRST "FMB1_RECC" WITHIN OWNER;
```

FETCH コマンドの実行結果：

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          2
      2 "RECC_UD1"                       'E222'
  FETCH SERIAL NUM               : 2
  ROWID                          : X'1C000000190000000000400'
  RECORD POINTER STATUS          : FIRST
KFPB64050-I FETCH command processing completed
```

9. COMMIT コマンドを実行して，レコードの削除を確定する

```
COMMIT;
```

COMMIT コマンドの実行結果：

```
KFPB64050-I COMMIT command processing completed
```

10. #EXIT コマンドを実行して，pdsdbexe コマンドを終了する

```
#EXIT;
```

#EXIT コマンドの実行結果：

```
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

参考

コマンド入力画面に表示されるプロンプトについては，「表 14-3　入力画面で表示されるプロンプト」を参照してください。

14.3.8　ファイルに記述した DML コマンドを実行する場合

4V FMB の SDB データベースのルートレコードに，レコード実現値を格納します。入力ファイル（infile.txt）に記述した DML コマンドを実行し，実行結果を出力ファイル（outfile.txt）に出力します。

- SDB データベース名：FMB1
- 操作対象レコード名：FMB1_RECA

1. DML コマンドを入力ファイル（infile.txt）に記述する

```
START "FMB1" ACSMODE UPDATE ENQOPT EXCLUSIVE;
STORE "FMB1_RECA" SET "DBKEY_L1" = 'KEY1',
                      "RECA_UD1" = 123.,
                      "RECA_UD2" = -9.8
                  PAGESWITCH OCCUPY
                  ;
YES     ←YESを指定します。※
COMMIT;
#EXIT;
```

注※　STORE コマンドの実行を許可する YES を指定する必要があります。

2. pdsdbexe コマンドを実行する

```
pdsdbexe -u '"User01"/"Pswd01"' <infile.txt >outfile.txt
```

3. 実行結果（outfile.txt）を確認する

```
pdsdbexe **** SDB Database Executor 09-60    ****  2015-12-14 16:49:44

----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
CONNECT INFORMATION
  SERVER NAME "testfes1"
  PROCESS ID                      : 12319
  CONNECT NUMBER                  : 12
KFPB64050-I CONNECT command processing completed
COMMAND ?     +----2----+----3----+----4----+----5----+----6----+----7----+----8

START "FMB1" ACSMODE UPDATE ENQOPT EXCLUSIVE;
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2015-12-14 16:20:49.16
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 09-60
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME "FMB1"
  HANDLE NUMBER                   : 1
  ACSMODE                         : UPDATE
  ENQOPT                          : EXCLUSIVE
  AUTODEQ                         : NO
  RDAREAVAL                       : NO
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
COMMAND ?     +----2----+----3----+----4----+----5----+----6----+----7----+----8

STORE "FMB1_RECA" SET "DBKEY_L1" = 'KEY1',
NEXT ?        +----2----+----3----+----4----+----5----+----6----+----7----+----8

                    "RECA_UD1" = 123.,
NEXT ?        +----2----+----3----+----4----+----5----+----6----+----7----+----8

                    "RECA_UD2" = -9.8
NEXT ?        +----2----+----3----+----4----+----5----+----6----+----7----+----8

                PAGESWITCH OCCUPY
NEXT ?        +----2----+----3----+----4----+----5----+----6----+----7----+----8

                ;
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
STORE RECORD INFORMATION
  RECORD KIND                     : ROOT RECORD
AGREE TO STORE ? [ENTER YES OR NO]
YES
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
STORE INFORMATION
  STORE SERIAL NUM                : 0
  ROWID                           : X'070000001B00000000000100'
  SECOND INDEX KIND               : NOUSE
  RECORD POINTER STATUS           : NONE
```

```
KFPB64050-I STORE command processing completed
COMMAND ?     +----2----+----3----+----4----+----5----+----6----+----7----+----8

COMMIT;
KFPB64050-I COMMIT command processing completed
COMMAND ?     +----2----+----3----+----4----+----5----+----6----+----7----+----8

#EXIT;
KFPB64050-I DISCONNECT command processing completed
KFPB64050-I #EXIT command processing completed

KFPB64000-I The pdsdbexe terminated, return code = 0
```

■入力ファイルの記述規則

1. 入力ファイル中に記述する文字の文字コードは，次の表に示すようにクライアント環境定義の PDLANG で指定している文字コードと同じにしてください。PDLANG については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

表 14-4　入力ファイルの文字コード

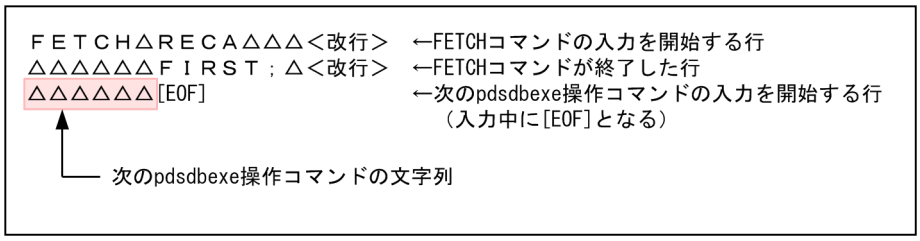
PDLANG の指定値	入力ファイルの文字コード	備考
SJIS	Shift-JIS	なし。
UTF8	Unicode（UTF-8）	BOM は付加できません。

参考

PDLANG で指定している文字コードは，pdsetup コマンドの-c オプションで指定した文字コードと同じです。

2. 改行コードは，X'0A'（LF）または，X'0D0A'（CRLF）としてください。
3. #EXIT コマンドの指定は省略できます。
4. EOF 制御文字がある場合，#EXIT コマンドの指定と同じ処理が行われます。
5. #EXIT コマンドを省略した場合，pdsdbexe 操作コマンドが終了した行の改行後から EOF 制御文字までの間に，次の pdsdbexe 操作コマンドの開始となる入力があると，エラーとなります。そのため，ファイルの末尾には#EXIT コマンドを指定することを推奨します。

   エラーとなる例を次に示します。



（凡例）△：空白

■出力ファイルへの出力規則

1. 出力ファイル中の文字コードは，次の表に示すようにクライアント環境定義の PDLANG で指定している文字コードと同じになります。PDLANG については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

表 14–5　出力ファイルの文字コード

PDLANG の指定値	出力ファイルの文字コード	備考
SJIS	Shift-JIS	なし。
UTF8	Unicode（UTF-8）	BOM は付加されません。

2. 改行コードは，X'0A'（LF）で出力されます。ただし，エラーが発生した場合，標準出力に X'0D0A'（CRLF）が出力されることがあります。

14.4　DML コマンド【4V FMB，4V AFM】

SDB データベース種別が 4V FMB または 4V AFM の SDB データベースを操作する場合に使用する DML コマンドの機能と入力形式について説明します。

4V FMB または 4V AFM の SDB データベースを操作する場合，pdsdbexe コマンド起動時の-d オプションを省略するか-d オプションに 4V を指定します。

14.4.1　CLEAR（レコードの一括削除）

(1)　機能

指定したレコードのデータベースキーに対応したレコード実現値を一括して削除します。

(2)　指定形式

```
CLEAR レコード名
      WHERE（DBKEY = 条件値1）;
```

注意事項

最後にセミコロン（；）の指定が必要です。

(3)　オペランドの説明

●レコード名

～＜識別子＞((1～30 バイト))

レコード実現値を一括削除するレコード名を指定します。

レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

●WHERE (DBKEY = 条件値 1)

一括削除するレコードのデータベースキーを=条件で指定します。

条件値 1 ::=

{DBKEY に対応する値｜

(構成要素に対応する値〔，構成要素に対応する値〕...)}

DBKEY に対応する値として，DBKEY の構成要素全体に対して 1 つの値を指定してください。

DBKEY に対応する値の指定形式を次の表に示します。

表 14-6　DBKEY に対応する値の指定形式

項番	DBKEY のデータ型	条件値の形式※	指定例	備考
1	CHARACTER	文字列定数	DBKEY='AAABBB'	DBKEY を構成するすべての構成要素のデータ型が CHARACTER の場合
2		16 進文字列定数	DBKEY=X'414141424242'	
3	INTEGER	整数定数	DBKEY=100	DBKEY が一連番号の場合，または DBKEY の構成要素数が 1 つでデータ型が INTEGER の場合
4		16 進文字列定数	DBKEY=X'64000000'	
5	PACKED DECIMAL FIXED	10 進数定数	DBKEY=123.456	DBKEY の構成要素数が 1 つでデータ型が PACKED DECIMAL FIXED の場合
6		16 進文字列定数	DBKEY=X'0123456C'	
7	上記以外	16 進文字列定数	DBKEY=X'010241414164000000'	なし

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

DBKEY に対応する値のうち，一連番号に対応する部分には任意の値を指定してください。

構成要素に対応する値として，DBKEY の 1 つの構成要素に対して値を指定してください。指定できる値の個数と順序は DBKEY の構成要素と一致させてください。

構成要素に対応する値の指定形式を次の表に示します。

表 14-7　構成要素に対応する値の指定形式

項番	構成要素のデータ型	条件値の形式※	指定例
1	CHARACTER	文字列定数	'AAABBB'
2		16 進文字列定数	X'414141424242'
3	XCHARACTER	16 進文字列定数	X'01020304'
4	PACKED DECIMAL FIXED	10 進数定数	123.456
5		16 進文字列定数	X'123C'
6	INTEGER	整数定数	100
7		16 進文字列定数	X'64000000'

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

構成要素に対応する値のうち，一連番号に対応する部分には任意の値を指定してください。

(4) 使用例

CLEAR コマンドを実行してレコード実現値を一括削除する手順を次に示します。

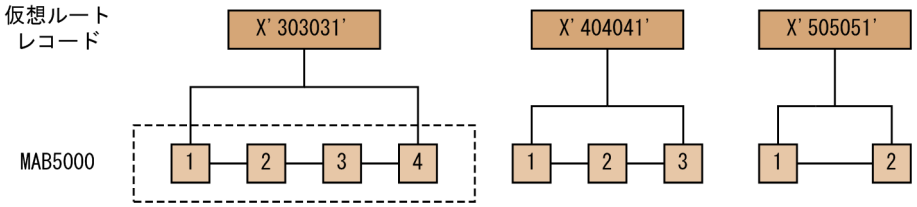
SDB データベース定義

操作する SDB データベース定義は次のとおりです。

- SDB データベース種別：4V DAM
- SDB データベース名：DAMX
- 削除対象のレコード名：MAB5000
- 削除対象のデータベースキー：X'303031'
- 削除対象のレコードの DBKEY：一連番号の構成要素を含む

レコード実現値

格納されているレコード実現値は次のとおりです。この例では，破線内のレコードを一括削除します。



手順

1. CLEAR コマンドを実行する

```
CLEAR "MAB5000" WHERE (DBKEY=(X'303031',0));
```

2. 一括削除対象のレコード名とデータベースキーが正しいかを確認する

```
CLEAR RECORD INFORMATION
  NAME "MAB5000"
  DBKEY                              : X'30303100000000'
AGREE TO CLEAR ? [ENTER YES OR NO]
YES      ←上記の情報を確認後，YESを指定します。
```

CLEAR コマンドを実行してよいかどうかの確認メッセージが表示されます。

一括削除対象のレコード名（NAME），および一括削除対象のデータベースキー（DBKEY）を確認してから，YES を指定してください。

> **参考**
>
> 一括削除対象のレコード名などに誤りがある場合は，NO を指定して CLEAR コマンドの実行を中止してください。

3. CLEAR コマンドの実行結果を確認する

```
KFPB64050-I CLEAR command processing completed
```

CLEAR コマンドの実行結果が表示されます。

(5) CLEAR コマンドの実行時に表示される情報

CLEAR コマンドの実行時に表示される情報について説明します。

(a) CLEAR コマンドの実行確認メッセージ

CLEAR コマンドを実行した際，次に示す確認メッセージが表示されます。

（例）

```
CLEAR RECORD INFORMATION
  NAME "MAB5000"                                    ...1
  DBKEY                          : X'30303100000000'    ...2
AGREE TO CLEAR ? [ENTER YES OR NO]                  ...3
```

[説明]

1. 一括削除対象のレコード名が表示されます。
2. 一括削除対象のレコードのデータベースキーが表示されます。
   データベースキーに一連番号が含まれる場合，一連番号の部分には X'00000000'が表示されます。
3. CLEAR コマンドを実行するかどうかの確認メッセージが表示されます。
   レコード名およびデータベースキーを確認（1.および 2.の情報を確認）し，問題がない場合は YES または yes を指定してください。
   CLEAR コマンドの実行を中止する場合は，NO または no を指定してください。

参考

- YES または yes 以外を指定した場合，NO が指定されたと見なされます。

(6) 留意事項

1. CLEAR コマンドは，個別開始している状態で実行してください。
2. 検索条件の条件値は，DBKEY または構成要素のデータ型とデータ属性が一致している必要があります。pdsdbexe コマンドは，パディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工を行いません。また，データ型によっては，次の条件を満たす必要があります。
   - データ型が CHARACTER または XCHARACTER の場合
     データ長が一致している必要があります。
   - データ型が PACKED DECIMAL FIXED または INTEGER の場合

条件値が，構成要素のデータ型で表現できる範囲の値である必要があります。また，条件値を 16 進文字列定数で指定した場合，条件値の長さは，構成要素の長さと同じである必要があります。

なお，データ型が PACKED DECIMAL FIXED の場合，条件値に指定した文字数で整数部桁数および小数部桁数が決まります。

（例）

01234.000 → 整数部桁数 5，小数部桁数 3

.12345 → 整数部桁数 0，小数部桁数 5

上記の例のようにゼロサプレスされません。条件値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。

3. CLEAR コマンドが正常終了したあとに#EXIT コマンドを実行する場合は，次のように pdsdbexe 操作コマンドを実行してください。
   - レコードの一括削除を実行したトランザクションを正常終了させたい場合は，COMMIT コマンドを実行してください。
   - レコードの一括削除を実行したトランザクションを取り消したい場合は，ROLLBACK コマンドを実行してください。

14.4.2 ERASE（レコードの削除）

(1) 機能

指定したレコード実現値を削除します。

(2) 指定形式

```
ERASE レコード名
      TARGET  {SEGMENT | OCCURRENCE} ;
```

注意事項

- 最後にセミコロン（；）の指定が必要です。
- オペランドの指定順序は変更できません。

(3) オペランドの説明

●レコード名

～<識別子>((1～30 バイト))

レコード実現値を削除するレコード名を指定します。4V AFM の SDB データベースの場合は，子レコード名を指定します。

レコード名に英小文字を使用している場合は，レコード名を引用符（ " ）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

●TARGET　{SEGMENT ｜ OCCURRENCE}

レコード実現値の削除範囲を指定します。

SEGMENT：

最下位レベルのレコードに位置づけしている場合は，位置づけしているレコードのレコード実現値だけを削除します。

最下位レベル以外のレコードに位置づけしている場合は，位置づけしているレコードのレコード実現値と，その下位レベルのレコードのレコード実現値を削除します。

OCCURRENCE：

最下位レベルのレコード型の全レコード実現値を削除します。

4V FMB の SDB データベースの最下位レベルのレコード型に対してだけ OCCURRENCE を指定できます。

なお，4V FMB の SDB データベースで，かつルートレコードの場合は，最下位のレコードであっても OCCURRENCE を指定できません。

(4)　使用例

ERASE コマンドを実行してレコード実現値を削除する手順を次に示します。

手順

1. ERASE コマンドを実行する

```
ERASE ”FMB101” TARGET SEGMENT;
```

2. 削除対象のレコードが正しいかを確認する

```
ERASE RECORD INFORMATION
  RECORD KIND                    : CHILD RECORD
  DBKEY                          : 1
AGREE TO ERASE DBKEY AND FOLLOWERS ? [ENTER YES OR NO]
YES     ←上記の情報を確認後，YESを指定します。
```

ERASE コマンドを実行してよいかどうかの確認メッセージが表示されます。

削除対象のレコード種別（RECORD KIND），および削除対象レコードの DBKEY を確認してから，YES を指定してください。

参考

削除対象のレコード種別などに誤りがある場合は，NO を指定して ERASE コマンドの実行を中止してください。

3. ERASE コマンドの実行結果を確認する

```
ERASE INFORMATION
  USER POINTER                    : OPERATED
KFPB64050-I ERASE command processing completed
```

ERASE コマンドの実行結果が表示されます。

(5) ERASE コマンドの実行時に表示される情報

ERASE コマンドの実行時に表示される情報について説明します。

(a) ERASE コマンドの実行確認メッセージ

ERASE コマンドを実行した際，次に示す確認メッセージが表示されます。

（例）

```
ERASE RECORD INFORMATION
  RECORD KIND                     : CHILD RECORD         ...1
  DBKEY                           : 1                    ...2
AGREE TO ERASE DBKEY AND FOLLOWERS ? [ENTER YES OR NO]   ...3
```

[説明]

1. 削除対象のレコード種別が表示されます。
   ROOT RECORD：削除対象レコードがルートレコードであることを意味しています。
   CHILD RECORD：削除対象レコードが子レコードであることを意味しています。
2. 削除対象レコードの DBKEY（位置づけている DBKEY）が表示されます。
   削除対象レコードがルートレコードの場合は，ルートレコードの DBKEY が表示されます。
   削除対象レコードがルートレコード以外の場合は，指定したレコードの一連番号が表示されます。
3. ERASE コマンドを実行するかどうかの確認メッセージが表示されます。
   レコード種別および DBKEY を確認（1.および 2.の情報を確認）し，問題がない場合は YES または yes を指定してください。
   ERASE コマンドの実行を中止する場合は，NO または no を指定してください。

参考

- YES または yes 以外を指定した場合，NO が指定されたと見なされます。
- TARGET の指定値が SEGMENT の場合は，次の確認メッセージが表示されます。
  `AGREE TO ERASE DBKEY AND FOLLOWERS ? [ENTER YES OR NO]`
  TARGET の指定値が OCCURRENCE の場合は，次の確認メッセージが表示されます。
  `AGREE TO ERASE DBKEY AND ALL ? [ENTER YES OR NO]`

(b) ERASE コマンドの実行結果

ERASE コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
ERASE INFORMATION
  USER POINTER : OPERATED    ...1
```

[説明]

1. USER ポインタを操作したかどうかが表示されます。
   USER ポインタを操作した場合は，OPERATED が表示されます。USER ポインタを操作していない場合は，この項目は表示されません。

(6) 留意事項

1. ERASE コマンドは，個別開始している状態で実行してください。
2. 削除対象のレコードへの位置づけをしておいてください。位置づけをしていない場合は，エラーになります。
3. ERASE コマンドが正常終了したあとに#EXIT コマンドを実行する場合は，次のように pdsdbexe 操作コマンドを実行してください。
   - レコードの削除を実行したトランザクションを正常終了させたい場合は，COMMIT コマンドを実行してください。
   - レコードの削除を実行したトランザクションを取り消したい場合は，ROLLBACK コマンドを実行してください。

14.4.3 FETCH（レコードの検索）

(1) 機能

1 つのレコード実現値を取得します。また，操作対象レコードの位置づけを行います。

(2) 指定形式

```
FETCH レコード名
      {FIRST | LAST | NEXT | PRIOR}
      〔WHERE  {キーの条件  〔AND キー以外の条件〕 | キー以外の条件} 〕
      〔TERMOPT  {YES | NO} 〕
      〔SECIDXSUP  {YES | NO} 〕 ;
```

注意事項

- 最後にセミコロン（；）の指定が必要です。
- オペランドの指定順序は変更できません。

(3) オペランドの説明

●レコード名

～<識別子>((1～30 バイト))

検索対象のレコード名を指定します。

4V AFM の SDB データベースの場合は，子レコード名を指定します。

レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

● {FIRST ｜ LAST ｜ NEXT ｜ PRIOR}

検索で使用するポインタを指定します。

FIRST：FIRST ポインタから検索します。

LAST：LAST ポインタから検索します。

NEXT：NEXT ポインタから検索します。

PRIOR：PRIOR ポインタから検索します。

ポインタの詳細については，「2.5.4(2) 検索の基点となるレコードと検索の方向」，および「2.5.4(3) レコードの検索範囲」を参照してください。

4V TAM の SDB データベースのレコードの検索で NEXT を指定した場合，実行結果は保証できません。4V TAM の SDB データベースのレコードの検索では FIRST を指定してください。

●WHERE {キーの条件 〔AND キー以外の条件〕 ｜キー以外の条件}

検索条件を指定します。

キーの条件とキー以外の条件の両方を指定する場合は，それぞれを（ ）で囲み論理積（AND）で結合します。

キーの条件::＝

(DBKEY {=｜<｜<=｜>｜>=} 条件値 1

〔AND DBKEY {=｜<｜<=｜>｜>=} 条件値 1〕)

条件値 1 ::＝

{DBKEY に対応する値｜

(構成要素に対応する値〔，構成要素に対応する値〕...)}

検索条件を指定します。

DBKEY に対応する値として，DBKEY の構成要素全体に対して 1 つの値を指定してください。

DBKEY に対応する値の指定形式を次の表に示します。

表 14-8　DBKEY に対応する値の指定形式

項番	DBKEY のデータ型	条件値の形式※	指定例	備考
1	CHARACTER	文字列定数	DBKEY='AAABBB'	DBKEY を構成するすべての構成要素のデータ型が CHARACTER の場合
2		16 進文字列定数	DBKEY=X'414141424242'	
3	INTEGER	整数定数	DBKEY=100	DBKEY が一連番号の場合，または DBKEY の構成要素数が 1 つでデータ型が INTEGER の場合
4		16 進文字列定数	DBKEY=X'64000000'	
5	PACKED DECIMAL FIXED	10 進数定数	DBKEY=123.456	DBKEY の構成要素数が 1 つでデータ型が PACKED DECIMAL FIXED の場合
6		16 進文字列定数	DBKEY=X'0123456C'	
7	上記以外	16 進文字列定数	DBKEY=X'010241414164000000'	なし

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

構成要素に対応する値として，DBKEY の 1 つの構成要素に対して値を指定してください。指定できる値の個数と順序は DBKEY の構成要素と一致させてください。

構成要素に対応する値の指定形式を次の表に示します。

表 14-9　構成要素に対応する値の指定形式

項番	構成要素のデータ型	条件値の形式※	指定例
1	CHARACTER	文字列定数	'AAABBB'
2		16 進文字列定数	X'414141424242'
3	XCHARACTER	16 進文字列定数	X'01020304'
4	PACKED DECIMAL FIXED	10 進数定数	123.456
5		16 進文字列定数	X'123C'
6	INTEGER	整数定数	100
7		16 進文字列定数	X'64000000'

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

キーの条件の指定規則を次に示します。

- キーの条件は 2 個まで指定できます。
- キーの条件を 2 個指定する場合は，キーの条件が範囲指定となるように指定してください。範囲指定かどうかは，キー（DBKEY の構成要素）全体を XCHARACTER（16 進データ）と見なして判定します。ただし，一連番号については，INTEGER で判定します。

- ユーザキーの定義がある 4V FMB の SDB データベースの子レコードに対して，DBKEY（一連番号）の条件を指定する場合は，比較記号に'='だけが指定できます。それ以外を指定した場合は，エラーとなります。
- 格納条件分割をしたレコードの場合，基点条件の条件値のレコード型の RD エリア分割キー値が，SDB データベース格納定義の WITHIN 句で指定されたレコード型の RD エリア分割キー値でないときは，エラーとなります。

キー以外の条件::＝

（構成要素名 {=｜<>｜^=｜!=｜<｜<=｜>｜>=} 条件値 2

〔{AND ｜ OR} 構成要素名 {=｜<>｜^=｜!=｜<｜<=｜>｜>=} 条件値 2〕…）

条件値 2::＝構成要素に対応する値

検索条件を指定します。

構成要素に対応する値の指定形式を次の表に示します。

表 14-10 構成要素に対応する値の指定形式

項番	構成要素のデータ型	条件値の形式※	指定例
1	CHARACTER	文字列定数	'AAABBB'
2		16 進文字列定数	X'414141424242'
3	XCHARACTER	16 進文字列定数	X'01020304'
4	PACKED DECIMAL FIXED	10 進数定数	123.456
5		16 進文字列定数	X'123C'
6	INTEGER	整数定数	100
7		16 進文字列定数	X'64000000'

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

キー以外の条件の指定規則を次に示します。

- キー以外の条件に，DBKEY を指定する条件は指定できません。
- 9 バイト以上の構成要素名は指定できません。そのため，条件式に指定する構成要素名は，8 バイト以下で定義しておいてください。
- キー以外の条件に指定できる構成要素は，基本項目の構成要素だけです。集団項目の構成要素は指定できません。
- キー以外の条件は 8 個まで指定できます。ただし，キーの条件も一緒に指定する場合は，キー以外の条件の指定数の上限は，「8－キーの条件数」になります。
- ユーザキーとユーザキー以外の条件は，論理積（AND）で結合します。
- ユーザキーを条件式に指定する場合，次のことに注意してください。

  ・条件の先頭から指定する必要があります。

・条件は 2 個まで指定できます。
・条件を 2 個指定する場合は，範囲指定になるように指定する必要があります。

- ユーザキーの条件を指定する場合，キーの条件は指定できません。
- ユーザキーの条件に，比較記号「<>」，「^=」，「!=」は指定できません。
- ユーザキーの条件に，論理和（OR）は指定できません。

●TERMOPT　{YES | NO}

検索条件に該当するレコードがなく，検索結果が NOT FOUND のときに，レコードの終端を検知したかどうかの判定を，HiRDB/SD が行うかどうかを指定します。

YES：

レコードの終端を検知したかどうかを判定します。

NO：

レコードの終端を検知したかどうかを判定しません。

●SECIDXSUP　{YES | NO}

二次インデクスを使用するかどうかを指定します。

YES：

二次インデクスが使用できる場合でもポインタ検索を行います。

NO：

二次インデクスが使用できる場合は，二次インデクスを使用して検索を行います。

二次インデクスについては，「2.3.10　二次インデクス【4V FMB，4V AFM】」を参照してください。

(4)　使用例

■4V FMB の SDB データベースのルートレコードの検索例

```
FETCH "FMB1000" FIRST;
```

次の実行結果が表示されます。

```
FETCH INFORMATION
  FETCH DATA
      1 "KEYDATA1"                    'AAAA'
      2 "KEYDATA2"                    'BBBB'
      3 "USERDA0"                     123.45
  ROWID                          : X'0700000019000000000000100'
  PAGE NUM                       : 100
KFPB64050-I FETCH command processing completed
```

■4V FMB の SDB データベースの子レコードの検索例

```
FETCH "FMB101" FIRST;
```

次の実行結果が表示されます。

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                          1
      2 "USERDB0"                        'bbbb'
  FETCH SERIAL NUM               : 1
  ROWID                          : X'0700000019000000000000200'
  USER POINTER SERIAL NUM        : 0
  SECOND INDEX KIND              : NOUSE
  RECORD POINTER STATUS          : FIRST_LAST
KFPB64050-I FETCH command processing completed
```

■4V AFM の SDB データベースの子レコードの検索例

```
FETCH "RECB" FIRST;
```

次の実行結果が表示されます。

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY_KN"                       1
      2 "USERDATA1"                      'AAAA'
  FETCH SERIAL NUM               : 1
  ROWID                          : X'400000001200000000000100'
  SECOND INDEX KIND              : NOUSE
  RECORD POINTER STATUS          : FIRST_LAST
KFPB64050-I FETCH command processing completed
```

(5) FETCH コマンドの実行時に表示される情報

FETCH コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
FETCH INFORMATION
  FETCH DATA
      1 "OCNUM"                  1                        ...1
      2 "USERDATA"               '00000001'               ...1
  FETCH SERIAL NUM           : 1                          ...2
  ROWID                      : X'2800000019000000000008000'  ...3
  USER POINTER SERIAL NUM    : 0                          ...4
  SECOND INDEX KIND          : NOUSE                      ...5
  RECORD POINTER STATUS      : FIRST                      ...6
  FETCH TERMINATED           : NOUSE                      ...7
  PAGE NUM                   : 100                        ...8
```

[説明]

上記のすべての情報が常に表示される訳ではありません。検索対象のレコードや，検索結果によって表示される項目が異なります。

1. 構成要素の通番，構成要素名，検索データが表示されます。

   検索データは次の形式で表示されます。

項番	構成要素のデータ型	検索データの表示形式	表示例
1	CHARACTER	文字列定数の形式で表示されます。	'aaabbb'

項番	構成要素のデータ型	検索データの表示形式	表示例
2	XCHARACTER	16 進文字列定数の形式で表示されます。	X'01020304'
3	PACKED DECIMAL FIXED	10 進数定数の形式で表示されます。	123.456
4	INTEGER	整数定数の形式で表示されます。	100

各定数の形式については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

2. 検索したレコードの一連番号が表示されます。
3. 検索したレコードの ROWID が表示されます。
4. USER ポインタのレコードの一連番号の情報が返却された場合，USER ポインタが示す一連番号が表示されます。一連番号の情報が返却されなかった場合は，この項目は表示されません。
   なお，一連番号の情報は，子レコードの最初の検索時だけ返却されます。
5. 二次インデクスの使用有無が表示されます。
   NOUSE：二次インデクスは使用されませんでした。
   SERIAL NUM INDEX：一連番号を指定した二次インデクスが使用されました。
   USER KEY INDEX：ユーザキーを指定した二次インデクスが使用されました。
6. レコード格納時のレコードポインタ情報が表示されます。
   FIRST：検索したレコードは，FIRST ポインタで示されています。
   LAST：検索したレコードは，LAST ポインタで示されています。
   FIRST_LAST：検索したレコードは，FIRST ポインタおよび LAST ポインタで示されています。
   NONE：検索したレコードは，FIRST ポインタおよび LAST ポインタのどちらにも示されていません。
7. 終端検知をしたかどうかが表示されます。
   YES：終端検知しました。
   NO：終端検知しませんでした。
   NOUSE：終端検知機能を使用しませんでした。
8. 検索したルートレコードが属するファミリの事前割り当てページ数または事前割り当てサブページ数が出力されます。

(6) 留意事項

1. FETCH コマンドは，個別開始している状態で実行してください。
2. 検索条件の条件値は，DBKEY または構成要素のデータ型とデータ属性が一致している必要があります。pdsdbexe コマンドは，パディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工を行いません。また，データ型によっては，次の条件を満たす必要があります。
   - データ型が CHARACTER または XCHARACTER の場合
     データ長が一致している必要があります。
   - データ型が PACKED DECIMAL FIXED または INTEGER の場合

条件値が，構成要素のデータ型で表現できる範囲の値である必要があります。また，条件値を 16 進文字列定数で指定した場合，条件値の長さは，構成要素の長さと同じである必要があります。

なお，データ型が PACKED DECIMAL FIXED の場合，条件値に指定した文字数で整数部桁数および小数部桁数が決まります。

（例）

01234.000 → 整数部桁数 5，小数部桁数 3

.12345 → 整数部桁数 0，小数部桁数 5

上記の例のようにゼロサプレスされません。条件値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。

14.4.4 FETCHDB_ALL（複数レコードの検索）

(1) 機能

ルートレコード以下のすべてのレコード実現値を取得します。

(2) 指定形式

```
FETCHDB_ALL
      {FIRST | NEXT | SKIP}
      〔BUFFER_SIZE データ格納エリアサイズ〕
      〔WHERE キーの条件〕
      〔UNTIL NOT FOUND〕
      〔TERMOPT {YES | NO} 〕;
```

注意事項

- 最後にセミコロン（；）の指定が必要です。
- オペランドの指定順序は変更できません。

(3) オペランドの説明

● {FIRST ｜ NEXT ｜ SKIP}

検索の動作を指定します。

FIRST：条件式で指定されたルートレコードから検索を開始します。

NEXT：前回，複数レコードの検索で検索したレコードの，次のレコードから検索を開始します。

SKIP：前回，複数レコードの検索で検索したファミリの，次のファミリ（次のルートレコード）から検索を開始します。

検索の動作の詳細については，「2.5.11(6)　複数レコードの検索例」を参照してください。

●BUFFER_SIZE データ格納エリアサイズ　～<符号なし整数>（1000～65535）《MAX(ファミリの最大レコードサイズ×2,1000)》（単位：バイト）

検索したファミリのレコード実現値を取得するエリアのサイズを指定します。

データ格納エリアのサイズについては，「2.5.11(5)(a)　データ格納エリアの領域長の指定」を参照してください。

●WHERE キーの条件

検索条件としてキーの条件を指定します。

キーの条件::=

(DBKEY　{=｜<｜<=｜>｜>=}　条件値 1

〔AND DBKEY　{=｜<｜<=｜>｜>=}　条件値 1〕)

条件値 1 ::=

{DBKEY に対応する値｜

(構成要素に対応する値〔，構成要素に対応する値〕...)}

検索条件を指定します。

DBKEY に対応する値として，DBKEY の構成要素全体に対して 1 つの値を指定してください。

DBKEY に対応する値の指定形式を次の表に示します。

表 14-11　DBKEY に対応する値の指定形式

項番	DBKEY のデータ型	条件値の形式※	指定例	備考
1	CHARACTER	文字列定数	DBKEY='AAABBB'	DBKEY を構成するすべての構成要素のデータ型が CHARACTER の場合
2		16 進文字列定数	DBKEY=X'414141424242'	
3	INTEGER	整数定数	DBKEY=100	DBKEY が一連番号の場合，または DBKEY の構成要素数が 1 つでデータ型が INTEGER の場合
4		16 進文字列定数	DBKEY=X'64000000'	
5	PACKED DECIMAL FIXED	10 進数定数	DBKEY=123.456	DBKEY の構成要素数が 1 つでデータ型が PACKED DECIMAL FIXED の場合
6		16 進文字列定数	DBKEY=X'0123456C'	
7	上記以外	16 進文字列定数	DBKEY=X'010241414164000000'	なし

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

構成要素に対応する値として，DBKEY の 1 つの構成要素に対して値を指定してください。指定できる値の個数と順序は DBKEY の構成要素と一致させてください。

構成要素に対応する値の指定形式を次の表に示します。

表 14-12　構成要素に対応する値の指定形式

項番	構成要素のデータ型	条件値の形式※	指定例
1	CHARACTER	文字列定数	'AAABBB'
2		16 進文字列定数	X'414141424242'
3	XCHARACTER	16 進文字列定数	X'01020304'
4	PACKED DECIMAL FIXED	10 進数定数	123.456
5		16 進文字列定数	X'123C'
6	INTEGER	整数定数	100
7		16 進文字列定数	X'64000000'

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

キーの条件の指定規則を次に示します。

- キーの条件は 2 個まで指定できます。
- キーの条件を 2 個指定する場合は，キーの条件が範囲指定となるように指定してください。
- 格納条件分割をしたレコードの場合，基点条件の条件値のレコード型の RD エリア分割キー値が，SDB データベース格納定義の WITHIN 句で指定されたレコード型の RD エリア分割キー値でないと，エラーになります。

●UNTIL NOT FOUND

複数レコードの検索を，検索結果が NOT FOUND になるまで繰り返し実行する場合に指定します。

●TERMOPT　{YES | NO}

検索条件に該当するレコードがなく，検索結果が NOT FOUND のときに，レコードの終端を検知したかどうかの判定を，HiRDB/SD が行うかどうかを指定します。

YES：

レコードの終端を検知したかどうかを判定します。

NO：

レコードの終端を検知したかどうかを判定しません。

(4)　使用例

■複数レコードの検索例

```
FETCHDB_ALL FIRST;
```

次の実行結果が表示されます。

```
FETCHDB_ALL INFORMATION
  SPECIFIED BUFFER_SIZE           :  1000
  FETCH DATA SIZE                 :   240
  FETCH RECORD NUM                :     2
  FAMILY RECORD FETCH             : COMPLETED
  PAGE NUM                        : 100
FETCHDB_ALL RECORD INFORMATION    :     1/    2
  FETCH RECORD "FMB1000"
      1 "KEYDATA1"                      X'01'
      2 "USERDA0"                       'aaaaaaaaaa'
FETCHDB_ALL RECORD INFORMATION    :     2/    2
  FETCH RECORD "FMB101"
      1 "KEYDATA1"                      X'01'
      2 "DBKEY"                         1
      3 "USERDB0"                       '1111111111'
  FETCH SERIAL NUM                : 1
  USER POINTER RECORD             : NO
FETCHDB_ALL FAMILY INFORMATION
  RECOMMENDATION BUFFER_SIZE      :   240
KFPB64050-I FETCHDB_ALL command processing completed
```

(5) FETCHDB_ALL コマンドの実行時に表示される情報

FETCH コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
FETCHDB_ALL INFORMATION
  SPECIFIED BUFFER_SIZE           :  1000                  ...1
  FETCH DATA SIZE                 :   240                  ...2
  FETCH RECORD NUM                :     2                  ...3
  FAMILY RECORD FETCH             : COMPLETED              ...4
  FETCH TERMINATED                : NOUSE                  ...5
  PAGE NUM                        : 100                    ...6
FETCHDB_ALL RECORD INFORMATION    :     1/    2            ...7
  FETCH RECORD "FMB1000"                                   ...8
      1 "KEYDATA1"                      X'01'              ...9
      2 "USERDA0"                       'aaaaaaaaaa'       ...9
FETCHDB_ALL RECORD INFORMATION    :     2/    2            ...7
  FETCH RECORD "FMB101"                                    ...8
      1 "KEYDATA1"                      X'01'              ...9
      2 "DBKEY"                         1                  ...9
      3 "USERDB0"                       '1111111111'       ...9
  FETCH SERIAL NUM                : 1                      ...10
  USER POINTER RECORD             : NO                     ...11
FETCHDB_ALL FAMILY INFORMATION
 RECOMMENDATION BUFFER_SIZE      :   240                   ...12
```

[説明]

上記のすべての情報が常に表示される訳ではありません。検索対象のレコードや，検索結果によって表示される項目が異なります。

7～11 の情報は，検索したレコード数分繰り返し表示されます。

1. 複数レコードの検索で使用したデータ格納エリアのサイズが表示されます。

2. データ格納エリアに格納したデータのサイズが表示されます。
3. データ格納エリアに格納したレコード実現値数が表示されます。
4. ファミリのレコード検索が完了したかどうかが表示されます。
   COMPLETED：ファミリのすべてのレコードを検索しました。
   CONTINUE：ファミリに検索していないレコードがあります。
5. 終端検知をしたかどうかが表示されます。
   YES：終端検知しました。
   NO：終端検知しませんでした。
   NOUSE：終端検知機能を使用しませんでした。
6. 検索したルートレコードが属するファミリの事前割り当てページ数，または事前割り当てサブページ数が出力されます。
7. 検索したレコードの通番と，レコードの総数が表示されます。
8. 検索したレコードのレコード名が表示されます。
9. 構成要素の通番，構成要素名，検索データが表示されます。
   検索データは次の形式で表示されます。

項番	構成要素のデータ型	検索データの表示形式	表示例
1	CHARACTER	文字列定数の形式で表示されます。	'aaabbb'
2	XCHARACTER	16 進文字列定数の形式で表示されます。	X'01020304'
3	PACKED DECIMAL FIXED	10 進数定数の形式で表示されます。	123.456
4	INTEGER	整数定数の形式で表示されます。	100

   各定数の形式については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。
10. 検索したレコードの一連番号が表示されます。
11. USER ポインタにポイントされているかどうかが表示されます。
    YES：USER ポインタにポイントされています。
    NO：USER ポインタにポイントされていません。
12. 最少回数でファミリのすべてのレコードを取得するために必要なデータ格納エリアのサイズが表示されます。この値を BUFFER_SIZE 指定値の目安としてください。

(6) 留意事項

1. FETCHDB_ALL コマンドは，FETCHDB_ALL 専用の個別開始（START コマンドの FETCHDB_ALL ONLY 指定）をしている状態で実行してください。

2. 検索条件の条件値は，DBKEY または構成要素のデータ型とデータ属性が一致している必要があります。pdsdbexe コマンドは，パディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工を行いません。また，データ型によっては，次の条件を満たす必要があります。

   - データ型が CHARACTER または XCHARACTER の場合

     データ長が一致している必要があります。

   - データ型が PACKED DECIMAL FIXED または INTEGER の場合

     条件値が，構成要素のデータ型で表現できる範囲の値である必要があります。また，条件値を 16 進文字列定数で指定した場合，条件値の長さは，構成要素の長さと同じである必要があります。

     なお，データ型が PACKED DECIMAL FIXED の場合，条件値に指定した文字数で整数部桁数および小数部桁数が決まります。

     （例）

     01234.000 → 整数部桁数 5，小数部桁数 3

     .12345 → 整数部桁数 0，小数部桁数 5

     上記の例のようにゼロサプレスされません。条件値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。

14.4.5 FINISH（個別終了）

(1) 機能

HiRDB/SD に対して個別終了要求を行い，SDB データベースに対する操作の終了処理を行います。

(2) 指定形式

```
FINISH;
```

注意事項

最後にセミコロン（ ; ）の指定が必要です。

(3) 留意事項

1. FINISH コマンドは，個別開始している状態で実行してください。
2. FINISH コマンドを実行しても，HiRDB からの切り離しは行われません。そのため，再度 START コマンドを実行すると，SDB データベースの操作を行うことができます。

14.4.6　MODIFY（レコードの更新）

(1)　機能

1 つのレコード実現値を更新します。

(2)　指定形式

```
MODIFY レコード名
　　〔SET 構成要素名=更新値〔,構成要素名=更新値〕…〕
　　〔USERPTROPT　{SET | NOSET | CLEAR} 〕;
```

注意事項

- 最後にセミコロン（ ; ）の指定が必要です。
- オペランドの指定順序は変更できません。

(3)　オペランドの説明

●レコード名

～＜識別子＞((1～30 バイト))

レコード実現値を更新するレコード名を指定します。

4V AFM の SDB データベースの場合は，子レコード名を指定します。

レコード名に英小文字を使用している場合は，レコード名を引用符（ " ）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

●SET 構成要素名=更新値〔,構成要素名=更新値〕…

構成要素名：～＜識別子＞((1～30 バイト))

更新対象の構成要素名と更新値を指定します。

このオペランドを省略した場合，USER ポインタの操作だけを行います。

指定規則を次に示します。

- 最下位レベルの構成要素名を指定してください。
- 構成要素名は，更新する構成要素の個数分指定できます。
- 同じ構成要素名は指定できません。
- 構成要素名に英小文字を使用している場合は，構成要素名を引用符（ " ）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- 構成要素のデータ型に対応する更新値を指定してください。更新値の指定形式を次の表に示します。

表 14-13　更新値の指定形式

項番	構成要素のデータ型	更新値の形式※	指定例
1	CHARACTER	文字列定数	'AAABBB'
2		16 進文字列定数	X'414141424242'
3	XCHARACTER	16 進文字列定数	X'01020304'
4	PACKED DECIMAL FIXED	10 進数定数	123.456
5		16 進文字列定数	X'123C'
6	INTEGER	整数定数	100
7		16 進文字列定数	X'64000000'

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

●USERPTROPT　{SET | NOSET | CLEAR}

ポインタオプションを指定します。

SET：更新するレコード実現値を USER ポインタに設定します。

NOSET：更新するレコード実現値を USER ポインタに設定しません。

CLEAR：USER ポインタの設定を解除します。

USER ポインタを持たないレコード型の場合は，このオペランドを省略してください。

CLEAR は，更新対象のレコードが USER ポインタの示すレコードの場合に指定してください。

このオペランドを省略した場合は，ポインタオプションを使用しません。

USER ポインタの設定，解除については，「2.5.8(1)　USER ポインタの設定・解除【4V FMB】」を参照してください。

(4)　使用例

MODIFY コマンドを実行してレコード実現値を更新する手順を次に示します。

手順

1. MODIFY コマンドを実行する

```
MODIFY "FMB1000" SET "USERDA0"=100.45;
```

2. 更新対象のレコードが正しいかを確認する

```
MODIFY DATA
    1 "USERDA0"
      <BEFORE> 999.45
      <AFTER>  100.45
AGREE TO MODIFY ? [ENTER YES OR NO]
YES    ←上記の情報を確認後，YESを指定します。
```

MODIFY コマンドを実行してよいかどうかの確認メッセージが表示されます。
構成要素名，および更新前後のレコード実現値を確認してから，YES を指定してください。

参考

構成要素名などに誤りがある場合は，NO を指定して MODIFY コマンドの実行を中止してください。

3. MODIFY コマンドの実行結果を確認する

```
MODIFY INFORMATION
  ROWID                          : X'0700000019000000000000100'
KFPB64050-I MODIFY command processing completed
```

MODIFY コマンドの実行結果が表示されます。

(5) MODIFY コマンドの実行時に表示される情報

MODIFY コマンドの実行時に表示される情報について説明します。

(a) MODIFY コマンドの実行確認メッセージ

MODIFY コマンドを実行した際，次に示す確認メッセージが表示されます。

（例）

```
MODIFY DATA
    1 "USERDA0"                        ...1
      <BEFORE> 999.45                  ...2
      <AFTER>  100.45                  ...3
AGREE TO MODIFY ? [ENTER YES OR NO]    ...4
```

[説明]

1. 構成要素の通番および構成要素名が表示されます。MODIFY コマンドの SET オペランドに指定した順番になります。
2. 更新前のレコード実現値が表示されます。※
3. 更新後のレコード実現値が表示されます。※
4. MODIFY コマンドを実行するかどうかの確認メッセージが表示されます。
   1.～3.の情報を確認し，問題がない場合は YES または yes を指定してください。
   MODIFY コマンドの実行を中止する場合は，NO または no を指定してください。

   参考

   - YES または yes 以外を指定した場合，NO が指定されたと見なされます。

- USER ポインタの更新だけを行う場合は，MODIFY コマンドの実行確認メッセージは表示されません。

注※

更新前後のレコード実現値は次の形式で表示されます。

項番	構成要素のデータ型	更新前後のデータの表示形式	表示例
1	CHARACTER	文字列定数の形式で表示されます。	'aaabbb'
2	XCHARACTER	16 進文字列定数の形式で表示されます。	X'01020304'
3	PACKED DECIMAL FIXED	10 進数定数の形式で表示されます。 ただし，更新後のレコード実現値が 10 進数定数の形式で表現できない場合は，16 進文字列定数の形式で表示されます。	123.456
4	INTEGER	整数定数の形式で表示されます。	100

各定数の形式については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

(b) MODIFY コマンドの実行結果

MODIFY コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
MODIFY INFORMATION
  USER POINTER            : OPERATED                      ...1
  USER POINTER SERIAL NUM : 10                            ...2
  ROWID                   : X'880000001900000000000100'   ...3
```

[説明]

1. USER ポインタを操作したかどうかが表示されます。

   USER ポインタを操作した場合は，OPERATED が表示されます。USER ポインタを操作していない場合は，この項目は表示されません。

2. USER ポインタを操作した場合，USER ポインタが示す一連番号が表示されます。USER ポインタを操作していない場合は，この項目は表示されません。

3. 更新したレコードの ROWID が表示されます。

(6) 留意事項

1. MODIFY コマンドは，個別開始している状態で実行してください。
2. 更新対象のレコードへの位置づけをしておいてください。位置づけをしていない場合は，エラーになります。
3. 構成要素のデータ型とデータ属性が一致している更新値を指定する必要があります。更新値のパディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工は行われません。また，データ型によっては，次の条件を満たす必要があります。

- データ型が CHARACTER または XCHARACTER の場合

  データ長が一致している必要があります。

- データ型が PACKED DECIMAL FIXED または INTEGER の場合

  更新値が，構成要素のデータ型で表現できる範囲の値である必要があります。また，更新値を 16 進文字列定数で指定した場合，更新値の長さは，構成要素の長さと同じである必要があります。

  なお，データ型が PACKED DECIMAL FIXED の場合，更新値に指定した文字数で整数部桁数および小数部桁数が決まります。

  （例）

  01234.000 → 整数部桁数 5，小数部桁数 3

  .12345 → 整数部桁数 0，小数部桁数 5

  上記の例のようにゼロサプレスされません。更新値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。

4. MODIFY コマンドが正常終了したあとに#EXIT コマンドを実行する場合は，次のように pdsdbexe 操作コマンドを実行してください。
   - レコードの更新を実行したトランザクションを正常終了させたい場合は，COMMIT コマンドを実行してください。
   - レコードの更新を実行したトランザクションを取り消したい場合は，ROLLBACK コマンドを実行してください。

14.4.7 START（個別開始）

(1) 機能

HiRDB/SD に対して個別開始要求を行い，SDB データベースに対する操作の開始処理を行います。

(2) 指定形式

```
START SDBデータベース名
     〔ACSMODE  {READ｜UPDATE}〕
     〔ENQOPT  {SHARE｜EXCLUSIVE｜OCCUPY｜NONE}〕
     〔AUTODEQ  {YES｜NO}〕
     〔ACSRDAREA（RDエリア名〔,RDエリア名〕…）〕
     〔RDAREAVAL  {YES｜NO}〕
     〔FETCHDB_ALL ONLY〕;
```

注意事項

- 最後にセミコロン（；）の指定が必要です。
- オペランドの指定順序は変更できません。

(3) オペランドの説明

●SDB データベース名

～<識別子>((1～30 バイト))

操作対象の SDB データベース名を指定します。

SDB データベース名に英小文字を使用している場合は，SDB データベース名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

●ACSMODE {READ | UPDATE}

この個別開始でのアクセスモードを指定します。

READ：参照モード

UPDATE：更新モード

レコード実現値の格納，更新，または削除を実行する場合は UPDATE を指定してください。

●ENQOPT {SHARE | EXCLUSIVE | OCCUPY | NONE}

この個別開始での排他オプションを指定します。

SHARE：共用モード

EXCLUSIVE：排他モード

OCCUPY：占有モード

NONE：無排他モード

NONE を指定して個別開始を実行した場合の注意事項については，「2.9.7 排他モード指定による無排他検索機能【4V FMB，4V AFM】」を参照してください。

アクセスモードと排他オプションの指定可能な組み合わせを次の表に示します。

表 14-14 アクセスモードと排他オプションの指定可能な組み合わせ

排他オプション	アクセスモード	
	参照モード	更新モード
共用モード	○	×
排他モード	○	○
占有モード	○	○
無排他モード	○	×

（凡例）

○：指定できます。

×：指定できません。

なお，個別開始時には排他は掛かりません。

●AUTODEQ　{YES ｜ NO}

排他自動解除オプションを指定します。

YES：排他自動解除機能を使用します。

NO：排他自動解除機能を使用しません。

排他自動解除機能については，「2.9.4　排他自動解除機能」を参照してください。

●ACSRDAREA (RD エリア名〔,RD エリア名〕…)

～＜識別子＞((1～30 バイト))

検索対象とする RD エリアを指定します。このオペランドを省略した場合，すべての RD エリアを指定したと仮定されます。

このオペランドの機能詳細については，次の RDAREAVAL オペランドの説明を参照してください。

次の指定をした場合は，エラーになります。

- RD エリア名を重複して指定した場合
- 検索対象の SDB データベースが格納されている RD エリア以外の RD エリアを指定した場合

4V MAM，4V TAM，または 4V SAM の SDB データベースの場合，このオペランドの指定は無効となります。

●RDAREAVAL　{YES ｜ NO}

レコードの検索時に基点条件指定がある場合でも，ACSRDAREA オペランドで指定した RD エリア指定を有効とするかどうかを指定します。この指定を RD エリア指定有効オプションといいます。

YES：

基点条件指定がある場合でも，ACSRDAREA オペランドで指定した RD エリア指定を有効とします。

NO：

基点条件指定がある場合は，ACSRDAREA オペランドで指定した RD エリア指定を有効としません。

このオペランドの指定値と，レコードの検索範囲の関係を次の表に示します。

表 14-15　RDAREAVAL オペランドの指定値と，レコードの検索範囲の関係

検索時の指定	個別開始時の指定			検索範囲※2	アクセス種別
基点条件※1	RDAREAVAL オペランドの指定値	ACSRDAREA オペランドの RD エリアの指定※5	基点となる RD エリア		
指定なし	－	指定あり	－	ACSRDAREA オペランドに指定された RD エリアを，格納値※3 の昇順に検索します。 検索範囲の例については，「■検索範囲の例（その 1）」を参照してください。	RD エリア指定検索※4

検索時の指定	個別開始時の指定			検索範囲※2	アクセス種別
基点条件※1	RDAREAVAL オペランドの指定値	ACSRDAREA オペランドの RD エリアの指定※5	基点となる RD エリア		
		指定なし	－	全 RD エリアを格納値の昇順に検索します。	
指定あり	YES	指定あり	含む	基点条件に指定された検索開始位置から，その RD エリアと ACSRDAREA オペランドに指定された RD エリアを格納値の昇順に検索します。	
			含まない	基点条件に指定された格納値を含むレコード実現値と，ACSRDAREA オペランドに指定された RD エリアを格納値の昇順に検索します。 検索範囲の例については，「■検索範囲の例（その 2)」を参照してください。	
		指定なし	－	基点条件に指定された検索開始位置から，全 RD エリアを格納値の昇順に検索します。	
	NO	－	－	基点条件で指定された検索開始位置から，その RD エリアだけを格納値の昇順に検索します。	基点検索

（凡例）

－：該当しません。

注※1

検索範囲の開始位置を指定するキーの条件のことを基点条件といいます。条件式の比較記号が，>，>=，および=の場合に，基点条件の指定ありとなります。例えば，「DBKEY>B」または「DBKEY=B」と指定した場合は，検索範囲の開始位置 B が指定されているため，基点条件の指定ありとなります。一方，「DBKEY<H」と指定した場合は，検索範囲の開始位置が指定されていないため，基点条件の指定なしとなります。

注※2

検索範囲の決定は，「表 K-1　4V FMB の SDB データベース検索時の検索範囲決定」，「表 K-2　4V AFM の SDB データベース検索時の検索範囲決定」，「表 K-3　SD FMB の SDB データベース検索時の検索範囲決定」に記載の契機で行われます。以降の NEXT および PRIOR 指定の検索は，そのときに決定した検索範囲を対象に行われます。

注※3

境界値分割をしている場合は，「格納値」を「境界値に指定された値の範囲」と読み替えてください。

注※4

シーケンシャルインデクスの先頭の構成要素以外をレコード型の RD エリア分割キーに指定したレコードに対して，ACSRDAREA オペランドを指定して RD エリア指定検索をする場合，次の条件をすべて満たしている必要があります。

・レコード型の RD エリア分割キーに指定した構成要素よりも前の全構成要素に対して KEYDEF 句を指定している。

・上記で指定した KEYDEF ごとの DATA で定義したキー値が 1 つだけである。

上記の条件をすべて満たしていない場合は，エラーになります。

注※5

4V AFM の SDB データベースで，ACSRDAREA オペランドの指定なしの場合は，この表で説明している検索範囲が，さらに次の範囲に限定されます。

・基点条件指定なしの場合

表中の検索範囲で，最初に指定と一致した仮想ルートレコードのレコード実現値の子レコードを検索範囲とします。

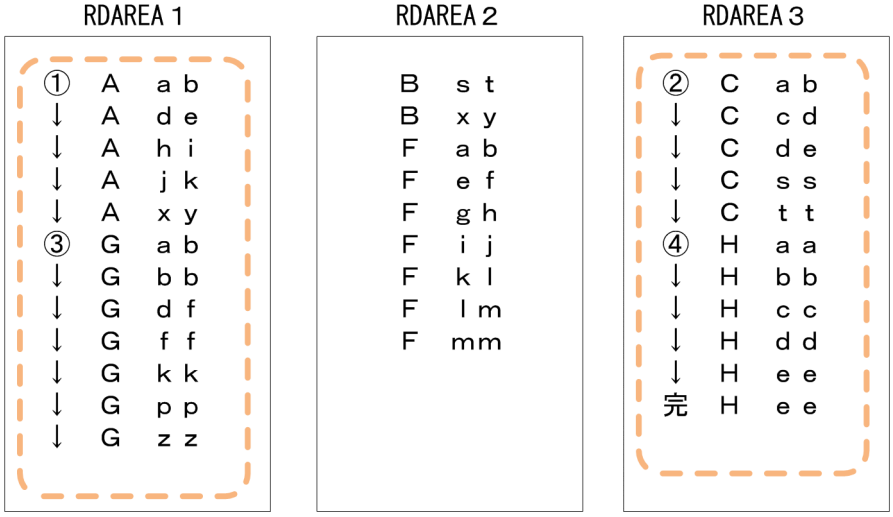
・基点条件指定ありの場合

表中の検索範囲で，基点条件で指定されたキー値に対応する仮想ルートレコードのレコード実現値の子レコードを検索範囲とします。

■検索範囲の例（その 1）

次の条件下で，レコードの検索をした場合，点線枠内が検索範囲となります。分割条件値が昇順となる A→C→G→H の順に検索します。

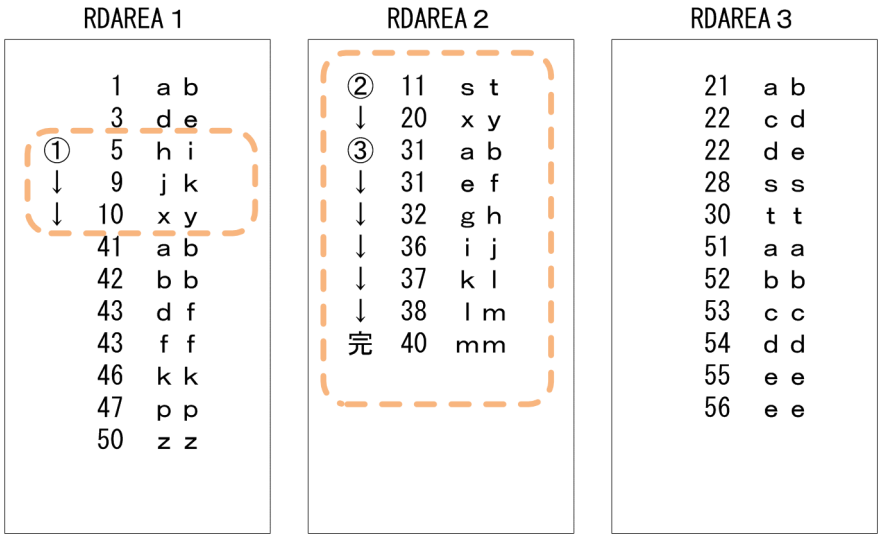
- 基点条件の指定なし
- ACSRDAREA オペランドに RDAREA1 および RDAREA3 を指定



■検索範囲の例（その 2）

次の条件下で，レコードの検索をした場合，点線枠内が検索範囲となります。分割条件値が，1～10 → 11～20 → 31～40 となるように検索します。

- 基点条件に「DBKEY≧5hi」を指定
- ACSRDAREA オペランドに RDAREA2 を指定



●FETCHDB_ALL ONLY

FETCHDB_ALL コマンドを実行する個別開始の場合に指定します。

このオペランドを指定して個別開始をした場合，FETCHDB_ALL コマンドおよび FINISH コマンド以外の DML コマンドは実行できません。また，このオペランドを省略した場合，FETCHDB_ALL コマンドを実行できません。

(4) 使用例

■4V FMB の SDB データベース（FMB1）の参照だけをする場合

```
START ”FMB1”;
```

次の実行結果が表示されます。

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2015-12-14 16:20:49.16
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 09-60
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME ”FMB1”
  HANDLE NUMBER                   : 1
  ACSMODE                         : READ
  ENQOPT                          : SHARE
  AUTODEQ                         : NO
  RDAREAVAL                       : NO
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

■4V MAM の SDB データベース（MAB1）のレコードを更新する場合

```
START ”MAB1”
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8
```

```
     ACSMODE UPDATE
NEXT ? +----2----+----3----+----4----+----5----+----6----+----7----+----8
     ENQOPT EXCLUSIVE;
```

次の実行結果が表示されます。

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2015-12-14 16:20:49.16
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 09-60
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME "MAB1"
  HANDLE NUMBER                   : 1
  ACSMODE                         : UPDATE
  ENQOPT                          : EXCLUSIVE
  AUTODEQ                         : NO
  RDAREAVAL                       : NO
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I START command processing completed
```

(5) START コマンドの実行時に表示される情報

START コマンドを実行した場合に表示される情報について説明します。

(a) SDB ディレクトリ情報ファイルの日時情報

START コマンドを実行した際（トランザクション内の初回の START コマンド実行時），次に示す SDB ディレクトリ情報ファイルの日時情報が表示されます。

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2015-12-14 16:20:46.67  ...1
  SDB DIRECTORY UPDATE DATE       : 2015-12-14 16:20:49.16  ...2
```

[説明]

1. 常用常駐領域中の SDB ディレクトリ情報の SDB 定義の最終更新日時（pdsdbdef コマンドの実行日時）が表示されます。
2. 常用常駐領域中の SDB ディレクトリ情報の最終更新日時が表示されます。

(b) HiRDB のバージョン情報

START コマンドを実行した際（トランザクション内の初回の START コマンド実行時），次に示す HiRDB のバージョン情報が表示されます。

```
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : vv-rr-xn
KFPB64050-I DINF(GET DATE) command processing completed
```

[説明]

下線部分に接続先の HiRDB のバージョンが表示されます。

(c) 個別開始に関する情報

START コマンドを実行した際，次に示す個別開始に関する情報が表示されます。

```
START INFORMATION
  SDB DATABASE NAME "FMB1"                ...1
  HANDLE NUMBER               : 1         ...2
  ACSMODE                     : READ      ...3
  ENQOPT                      : SHARE     ...4
  AUTODEQ                     : NO        ...5
  ACSRDAREA                   : "RDAREA1" ...6
                                "RDAREA2" ...6
  RDAREAVAL                   : NO        ...7
  FETCHDB_ALL ONLY                        ...8
```

[説明]

1. START コマンドに指定した SDB データベース名が表示されます。
2. ハンドル番号が表示されます。
3. ACSMODE オペランドの指定値，または省略時解釈値が表示されます。
4. ENQOPT オペランドの指定値，または省略時解釈値が表示されます。
5. AUTODEQ オペランドの指定値，または省略時解釈値が表示されます。
6. ACSRDAREA オペランドの指定値が表示されます。ACSRDAREA オペランドが省略されている場合は，この項目は表示されません。
7. RDAREAVAL オペランドの指定値，または省略時解釈値が表示されます。
8. FETCHDB_ALL ONLY オペランドが指定されている場合に表示されます。FETCHDB_ALL ONLY オペランドが省略されている場合は，この項目は表示されません。

(6) 留意事項

個別開始の実行後，個別終了を実行していない状態で，個別開始を実行することはできません。

14.4.8　STORE（レコードの格納）

(1)　機能

1 つのレコード実現値を格納します。

(2)　指定形式

```
STORE レコード名
      SET 構成要素名=格納値〔,構成要素名=格納値〕…
     〔DIRECTION  {NEXT｜PRIOR}〕
     〔USERPTROPT  {SET｜NOSET}〕
     〔PAGESWITCH  {CHANGE｜OCCUPY｜NOUSE}〕
     〔PCTFREEVAL  {YES｜NO}〕
     〔SECIDXSUP  {YES｜NO}〕
     〔PAGENUM  事前割り当てページ数〕;
```

注意事項

- 最後にセミコロン（；）の指定が必要です。
- オペランドの指定順序は変更できません。

(3)　オペランドの説明

●レコード名

～＜識別子＞((1～30 バイト))

レコード実現値を格納するレコード名を指定します。

4V AFM の SDB データベースの場合は，子レコード名を指定します。

レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

●SET 構成要素名=格納値〔,構成要素名=格納値〕…

構成要素名：～＜識別子＞((1～30 バイト))

格納するレコードの値を，レコードの定義で指定した構成要素の順に指定します。

指定規則を次に示します。

- 最下位レベルの構成要素（基本項目）に対する格納値を指定してください。
- 指定が必要な構成要素をすべて指定してください。

- 指定が必要な構成要素については，「付録 K.3　入力情報とキーの定義の関係【4V FMB，4V AFM】」を参照してください。「付録 K.3　入力情報とキーの定義の関係【4V FMB，4V AFM】」を参照する際，pdsdbexe コマンドの場合，「レコード型名指定あり」の条件が該当します。
- 指定が必要な構成要素については，#USAGE コマンドの実行結果で確認できます。#USAGE コマンドの実行結果に表示されたひな形の SET オペランドに出力されている構成要素を，出力されている順にすべて指定してください。#USAGE コマンドについては，「14.8.3　#USAGE（DML コマンドのひな形の出力）」を参照してください。
- 構成要素名は，指定が必要な構成要素の個数まで指定できます。
- 構成要素名に英小文字を使用している場合は，構成要素名を引用符（ " ）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- 格納値の指定形式を次の表に示します。

表 14-16　格納値の指定形式

項番	構成要素のデータ型	格納値の形式※	指定例
1	CHARACTER	文字列定数	'AAABBB'
2		16 進文字列定数	X'414141424242'
3	XCHARACTER	16 進文字列定数	X'01020304'
4	PACKED DECIMAL FIXED	10 進数定数	123.456
5		16 進文字列定数	X'123C'
6	INTEGER	整数定数	100
7		16 進文字列定数	X'64000000'

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

●DIRECTION　{NEXT ｜ PRIOR}

格納位置を検索する方向を指定します。

NEXT：

格納位置を NEXT 方向に検索します。

格納対象のレコード型に位置指示子が位置づいている場合は，位置づいているレコード実現値を基点にし，NEXT 方向に検索します。位置指示子が空値の場合は，親の FIRST ポインタを基点にし，NEXT 方向に検索します。

PRIOR：

格納位置を PRIOR 方向に検索します。

格納対象のレコード型に位置指示子が位置づいている場合は，位置づいているレコード実現値を基点にし，PRIOR 方向に検索します。位置指示子が空値の場合は，親の LAST ポインタを基点にし，PRIOR 方向に検索します。

SDB データベース定義の ORDER オペランドに SORTED DUPLICATES PROHIBITED を指定しているときにこのオペランドを省略した場合は，NEXT が仮定されます。

なお，次の場合は，このオペランドの指定は無視されます。

- 4V FMB の SDB データベースのルートレコードの格納時
- SDB データベース定義の ORDER オペランドに LAST を指定した SDB データベースの場合

●USERPTROPT　{SET | NOSET}

ポインタオプションを指定します。

SET：格納するレコード実現値を USER ポインタに設定します。

NOSET：格納するレコード実現値を USER ポインタに設定しません。

このオペランドを省略した場合は，ポインタオプションを使用しません。格納対象が USER ポインタを持たないレコード型の場合は，このオペランドを省略してください。

USER ポインタの設定については，「2.5.7(1)　USER ポインタの設定【4V FMB】」を参照してください。

●PAGESWITCH　{CHANGE | OCCUPY | NOUSE}

ページ切り替えオプションを指定します。

サブページ分割をしている場合は，ページをサブページに読み替えてください。

CHANGE：

レコード格納時にページを切り替えて，新規ページを確保します。

OCCUPY：

レコード格納時にページを切り替えて，新規ページを確保します。確保したページは，格納したレコードで占有します（ほかのレコードを格納しません）。

NOUSE：

ページ切り替えオプションを指定しません。

なお，一括削除後のレコードの格納時，内部でレコードの検索または更新を行う場合は，このオペランドの指定は無視されます。

●PCTFREEVAL　{YES | NO}

SDB データベース格納定義の PCTFREE オペランドの指定を有効にするかどうかを指定します。

サブページ分割をしている場合は，ページをサブページに読み替えてください。

YES：

SDB データベース格納定義の PCTFREE オペランドの指定を有効にします。レコード格納時，SDB データベース格納定義の PCTFREE オペランドに指定したページ内の未使用領域の比率を適用します。

NO：

SDB データベース格納定義の PCTFREE オペランドの指定を有効にしません。レコード格納時，SDB データベース格納定義の PCTFREE オペランドに指定したページ内の未使用領域の比率を適用しません。

なお，一括削除後のレコードの格納時，内部でレコードの検索または更新を行う場合は，このオペランドの指定は無視されます。

●SECIDXSUP　{YES | NO}

二次インデクスを使用するかどうかを指定します。

YES：二次インデクスが使用できる場合でもポインタ検索を行います。

NO：二次インデクスが使用できる場合は，二次インデクスを使用して検索を行います。

二次インデクスについては，「2.3.10　二次インデクス【4V FMB，4V AFM】」を参照してください。

●PAGENUM 事前割り当てページ数　～<符号なし整数>（0～2,147,483,647）

事前割り当てページ数または事前割り当てサブページ数を指定します。

事前ページ割り当て機能については，「2.6.2(4)　事前ページ割り当て機能【4V FMB】」または「2.7.2(4)　事前ページ割り当て機能【4V FMB】」を参照してください。

(4)　使用例

STORE コマンドを実行してレコード実現値を格納する手順を次に示します。

手順

1. STORE コマンドを実行する

```
STORE "FMB101" SET "USERDB0"='DATA' PCTFREEVAL YES;
```

2. 格納対象のレコードが正しいかを確認する

```
STORE RECORD INFORMATION
  RECORD KIND                    : CHILD RECORD
  DBKEY                          : X'4141414101020304'
AGREE TO STORE ? [ENTER YES OR NO]
YES      ←上記の情報を確認後，YESを指定します。
```

STORE コマンドを実行してよいかどうかの確認メッセージが表示されます。

格納対象のレコード種別（RECORD KIND），および格納対象レコードの DBKEY（DBKEY）を確認してから，YES を指定してください。

参考

格納対象のレコード種別などに誤りがある場合は，NO を指定して STORE コマンドの実行を中止してください。

3. STORE コマンドの実行結果を確認する

```
STORE INFORMATION
  STORE SERIAL NUM               : 1
  ROWID                          : X'0700000019000000000000200'
  SECOND INDEX KIND              : NOUSE
  RECORD POINTER STATUS          : FIRST_LAST
KFPB64050-I STORE command processing completed
```

STORE コマンドの実行結果が表示されます。

(5) STORE コマンドの実行時に表示される情報

STORE コマンドを実行した場合に表示される情報について説明します。

(a) STORE コマンドの実行確認メッセージ

STORE コマンドを実行した際，次に示す確認メッセージが表示されます。

```
STORE RECORD INFORMATION
  RECORD KIND              : CHILD RECORD          ...1
  DBKEY                    : X'4141414101020304'   ...2
AGREE TO STORE ? [ENTER YES OR NO]                  ...3
```

[説明]

1. 格納対象のレコード種別が表示されます。

   ROOT RECORD：格納対象レコードがルートレコードであることを意味しています。

   CHILD RECORD：格納対象レコードが子レコードであることを意味しています。

2. 格納対象レコードの親レコードの DBKEY が表示されます。

   格納対象レコードが子レコードの場合に，この項目が表示されます。

   親レコードがルートレコードの場合，親レコードの DBKEY が表示されます。親レコードがルートレコードではない場合，親レコードの一連番号が表示されます。

   4V AFM の SDB データベースの場合の留意事項を次に示します。

   ・DBKEY の構成要素が 1 つも指定されていない場合，この項目は表示されません。

   ・DBKEY のうち，指定されていない部分は 0x00 が指定されたと見なされます。

3. STORE コマンドを実行するかどうかの確認メッセージが表示されます。

   レコード種別および親レコードの DBKEY を確認（1.および 2.の情報を確認）し，問題がない場合は YES または yes を指定してください。

   STORE コマンドの実行を中止する場合は，NO または no を指定してください。

   参考

   YES または yes 以外を指定した場合，NO が指定されたと見なされます。

(b) STORE コマンドの実行結果

STORE コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
STORE INFORMATION
  STORE SERIAL NUM           : 20                                ...1
  ROWID                      : X'880000001900000000000100'       ...2
  USER POINTER               : OPERATED                          ...3
  USER POINTER SERIAL NUM    : 20                                ...4
```

```
SECOND INDEX KIND          : USER KEY INDEX              ...5
RECORD POINTER STATUS      : LAST                        ...6
```

[説明]

1. 格納したレコードの一連番号が表示されます。
2. 格納したレコードの ROWID が表示されます。
3. USER ポインタを操作したかどうかが表示されます。
   USER ポインタを操作した場合は，OPERATED が表示されます。USER ポインタを操作していない場合は，この項目は表示されません。
4. USER ポインタを操作した場合，USER ポインタが示す一連番号が表示されます。USER ポインタを操作していない場合は，この項目は表示されません。
5. 二次インデクスを使用したかどうかが表示されます。
   NOUSE：二次インデクスは使用されていません。
   SERIAL NUM INDEX：一連番号を指定した二次インデクスが使用されました。
   USER KEY INDEX：ユーザキーを指定した二次インデクスが使用されました。
6. レコード格納時のレコードポインタ情報が表示されます。
   FIRST：格納したレコードは，FIRST ポインタで示されています。
   LAST：格納したレコードは，LAST ポインタで示されています。
   FIRST LAST：格納したレコードは，FIRST ポインタおよび LAST ポインタで示されています。
   NONE：格納したレコードは，FIRST ポインタおよび LAST ポインタのどちらでも示されていません。

(6) 留意事項

1. STORE コマンドは，個別開始している状態で実行してください。
2. 子レコードにレコードを格納する場合，親レコードまでの位置づけをしておいてください。位置づけをしていない場合は，エラーになります。
3. 構成要素のデータ型とデータ属性が一致している格納値を指定する必要があります。格納値のパディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工は行われません。また，データ型によっては，次の条件を満たす必要があります。
   - データ型が CHARACTER または XCHARACTER の場合
     データ長が一致している必要があります。
   - データ型が PACKED DECIMAL FIXED または INTEGER の場合
     格納値が，構成要素のデータ型で表現できる範囲の値である必要があります。また，格納値を 16 進文字列定数で指定した場合，格納値の長さは，構成要素の長さと同じである必要があります。
     なお，データ型が PACKED DECIMAL FIXED の場合，格納値に指定した文字数で整数部桁数および小数部桁数が決まります。
     （例）
     01234.000 → 整数部桁数 5，小数部桁数 3

.12345 → 整数部桁数 0，小数部桁数 5

上記の例のようにゼロサプレスされません。格納値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。

4. STORE コマンドが正常終了したあとに#EXIT コマンドを実行する場合は，次のように pdsdbexe 操作コマンドを実行してください。

- レコードの格納を実行したトランザクションを正常終了させたい場合は，COMMIT コマンドを実行してください。
- レコードの格納を実行したトランザクションを取り消したい場合は，ROLLBACK コマンドを実行してください。

14.4.9 DML コマンド共通の規則

START コマンド以外の DML コマンドは，個別開始している状態で実行してください。

14.5　DML コマンド【SD FMB】

SDB データベース種別が SD FMB の SDB データベースを操作する場合に使用する DML コマンドの機能と入力形式について説明します。

SD FMB の SDB データベースを操作する場合，pdsdbexe コマンド起動時の-d オプションに SD を指定します。この場合，SDB 用 UAP 環境定義が有効になります。SDB 用 UAP 環境定義については，「9.3 SDB 用 UAP 環境定義【SD FMB】」を参照してください。

14.5.1　ERASE（レコードの削除）

(1)　機能

指定したレコード実現値を削除します。

(2)　指定形式

指定形式については，「17.4.1　ERASE 文」の「(2)　形式」の「DML コマンドで記述する ERASE 文」を参照してください。

注意事項

- 最後にセミコロン（；）の指定が必要です。
- オペランドの指定順序は変更できません。

(3)　オペランドの説明

オペランドの説明については，「17.4.1　ERASE 文」の「(3)　オペランドの説明」を参照してください。

(4)　使用例

ERASE コマンドを実行してレコード実現値を削除する手順を次に示します。

手順

1. ERASE コマンドを実行する

```
ERASE ”FMB101” ALL;
```

2. 削除対象のレコードが正しいかを確認する

```
ERASE RECORD INFORMATION
  RECORD KIND                    : CHILD RECORD
  DBKEY                          : 1
```

```
AGREE TO ERASE DBKEY AND FOLLOWERS ? [ENTER YES OR NO]
YES      ←上記の情報を確認後，YESを指定します。
```

ERASE コマンドを実行してよいかどうかの確認メッセージが表示されます。

削除対象のレコード種別（RECORD KIND），および削除対象レコードのデータベースキーを確認してから，YES を指定してください。

参考

削除対象のレコード種別などに誤りがある場合は，NO を指定して ERASE コマンドの実行を中止してください。

3. ERASE コマンドの実行結果を確認する

```
KFPB64050-I ERASE command processing completed
```

ERASE コマンドの実行結果が表示されます。

(5) ERASE コマンドの実行時に表示される情報

ERASE コマンドの実行時に表示される情報について説明します。

ERASE コマンドを実行した際，次に示す確認メッセージが表示されます。

（例）

```
ERASE RECORD INFORMATION
  RECORD KIND                    : CHILD RECORD         ...1
  DBKEY                          : 1                    ...2
AGREE TO ERASE DBKEY AND FOLLOWERS ? [ENTER YES OR NO]   ...3
```

［説明］

1. 削除対象のレコード種別が表示されます。
   ROOT RECORD：削除対象レコードがルートレコードであることを意味しています。
   CHILD RECORD：削除対象レコードが子レコードであることを意味しています。
2. 削除対象レコードのデータベースキー（位置づけているデータベースキー）が表示されます。
   削除対象レコードがルートレコードの場合は，ルートレコードのデータベースキーが表示されます。
   削除対象レコードがルートレコード以外の場合は，指定したレコードの一連番号が表示されます。
3. ERASE コマンドを実行するかどうかの確認メッセージが表示されます。
   レコード種別およびデータベースキーを確認（1.および 2.の情報を確認）し，問題がない場合は YES または yes を指定してください。
   ERASE コマンドの実行を中止する場合は，NO または no を指定してください。

参考

YES または yes 以外を指定した場合，NO が指定されたと見なされます。

(6) 留意事項

1. SDB 用 UAP 環境定義の subschema オペランドの-a オプションに update を指定してください。SDB 用 UAP 環境定義については，「9.3 SDB 用 UAP 環境定義【SD FMB】」を参照してください。
2. ERASE コマンドは，個別開始している状態で実行してください。
3. 削除対象のレコードへの位置づけをしておいてください。位置づけをしていない場合は，エラーになります。
4. ERASE コマンドが正常終了したあとに#EXIT コマンドを実行する場合は，次のように pdsdbexe 操作コマンドを実行してください。
   - レコードの削除を実行したトランザクションを正常終了させたい場合は，COMMIT コマンドを実行してください。
   - レコードの削除を実行したトランザクションを取り消したい場合は，ROLLBACK コマンドを実行してください。

そのほかの留意事項については，「17.4.1 ERASE 文」の「(4) 共通規則」を参照してください。

14.5.2 FETCH（レコードの検索）

(1) 機能

1 つのレコード実現値を取得します。また，操作対象レコードの位置づけを行います。

(2) 指定形式

指定形式については，「17.4.2 FETCH 文」の「(2) 形式」の「DML コマンドで記述する FETCH 文」を参照してください。

注意事項

- 最後にセミコロン（；）の指定が必要です。
- オペランドの指定順序は変更できません。

(3) オペランドの説明

オペランドの説明については，「17.4.2 FETCH 文」の「(3) オペランドの説明」を参照してください。

(4) 使用例

■SD FMB の SDB データベースのルートレコードの検索例

```
FETCH FIRST "FMB1000";
```

次の実行結果が表示されます。

```
FETCH INFORMATION
  FETCH DATA
      1 "KEYDATA1"                    'AAAA'
      2 "KEYDATA2"                    'BBBB'
      3 "USERDA0"                     123.45
  ROWID                          : X'0700000019000000000000100'
KFPB64050-I FETCH command processing completed
```

■SD FMB の SDB データベースの子レコードの検索例

```
FETCH FIRST "FMB101" WITHIN OWNER;
```

次の実行結果が表示されます。

```
FETCH INFORMATION
  FETCH DATA
      1 "DBKEY"                       1
      2 "USERDB0"                     'bbbb'
  FETCH SERIAL NUM               : 1
  ROWID                          : X'0700000019000000000000200'
  RECORD POINTER STATUS          : FIRST_LAST
KFPB64050-I FETCH command processing completed
```

(5) FETCH コマンドの実行時に表示される情報

FETCH コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
FETCH INFORMATION
  FETCH DATA
      1 "OCNUM"                   1                       ...1
      2 "USERDATA"                '00000001'              ...1
  FETCH SERIAL NUM           : 1                          ...2
  ROWID                      : X'2800000019000000000008000'  ...3
  RECORD POINTER STATUS      : FIRST                      ...4
```

[説明]

上記のすべての情報が常に表示される訳ではありません。検索対象のレコードや，検索結果によって表示される項目が異なります。

1. 構成要素の通番，構成要素名，検索データが表示されます。

   検索データは次の形式で表示されます。

項番	構成要素のデータ型	検索データの表示形式	表示例
1	CHARACTER	文字列定数の形式で表示されます。	'aaabbb'

項番	構成要素のデータ型	検索データの表示形式	表示例
2	XCHARACTER	16 進文字列定数の形式で表示されます。	X'01020304'
3	PACKED DECIMAL FIXED	10 進数定数の形式で表示されます。	123.456
4	INTEGER SMALLINT	整数定数の形式で表示されます。	100

各定数の形式については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

2. 検索したレコードの一連番号が表示されます。
3. 検索したレコードの ROWID が表示されます。
4. レコード格納時のレコードポインタ情報が表示されます。
   FIRST：検索したレコードは，FIRST ポインタで示されています。
   LAST：検索したレコードは，LAST ポインタで示されています。
   FIRST_LAST：検索したレコードは，FIRST ポインタおよび LAST ポインタで示されています。
   NONE：検索したレコードは，FIRST ポインタおよび LAST ポインタのどちらにも示されていません。

(6) 留意事項

1. FETCH コマンドに FOR UPDATE を指定する場合，SDB 用 UAP 環境定義の subschema オペランドの-a オプションに update を指定してください。SDB 用 UAP 環境定義については，「9.3 SDB 用 UAP 環境定義【SD FMB】」を参照してください。
2. FETCH コマンドは，個別開始している状態で実行してください。

そのほかの留意事項については，「17.4.2 FETCH 文」の「(4) 共通規則」を参照してください。

14.5.3 FINISH（個別終了）

(1) 機能

HiRDB/SD に対して個別終了要求を行い，SDB データベースに対する操作の終了処理を行います。

(2) 指定形式

```
FINISH;
```

注意事項

最後にセミコロン（;）の指定が必要です。

(3) 留意事項

1. SDB データベースの操作終了後，操作対象の SDB データベースに対して FINISH コマンドを実行する必要があります。
2. FINISH コマンドは，個別開始している状態で実行してください。
3. FINISH コマンドを実行しても，HiRDB からの切り離しは行われません。そのため，再度 START コマンドを実行すると，SDB データベースの操作を行うことができます。
4. FINISH コマンドを実行しないで，トランザクションを終了させるとシステムが自動的に個別終了を実行します。

14.5.4 MODIFY（レコードの更新）

(1) 機能

1 つのレコード実現値を更新します。

(2) 指定形式

指定形式については，「17.4.5 MODIFY 文」の「(2) 形式」の「DML コマンドで記述する MODIFY 文」を参照してください。

注意事項

- 最後にセミコロン（ ; ）の指定が必要です。
- オペランドの指定順序は変更できません。

(3) オペランドの説明

オペランドの説明については，「17.4.5 MODIFY 文」の「(3) オペランドの説明」を参照してください。

(4) 使用例

MODIFY コマンドを実行してレコード実現値を更新する手順を次に示します。

手順

1. MODIFY コマンドを実行する

```
MODIFY ”FMB1000” SET ”USERDA0”=100.45;
```

2. 更新対象のレコードが正しいかを確認する

```
MODIFY DATA
    1 "USERDA0"
      <BEFORE> 999.45
      <AFTER>  100.45
AGREE TO MODIFY ? [ENTER YES OR NO]
YES   ←上記の情報を確認後，YESを指定します。
```

MODIFY コマンドを実行してよいかどうかの確認メッセージが表示されます。

構成要素名，および更新前後のレコード実現値を確認してから，YES を指定してください。

参考

構成要素名などに誤りがある場合は，NO を指定して MODIFY コマンドの実行を中止してください。

3. MODIFY コマンドの実行結果を確認する

```
MODIFY INFORMATION
  ROWID                          : X'0700000019000000000000100'
KFPB64050-I MODIFY command processing completed
```

MODIFY コマンドの実行結果が表示されます。

(5) MODIFY コマンドの実行時に表示される情報

MODIFY コマンドの実行時に表示される情報について説明します。

(a) MODIFY コマンドの実行確認メッセージ

MODIFY コマンドを実行した際，次に示す確認メッセージが表示されます。

（例）

```
MODIFY DATA
    1 "USERDA0"                        ...1
      <BEFORE> 999.45                  ...2
      <AFTER>  100.45                  ...3
AGREE TO MODIFY ? [ENTER YES OR NO]    ...4
```

[説明]

1. 構成要素の通番および構成要素名が表示されます。MODIFY コマンドの SET オペランドに指定した順番になります。
2. 更新前のレコード実現値が表示されます。※
3. 更新後のレコード実現値が表示されます。※
4. MODIFY コマンドを実行するかどうかの確認メッセージが表示されます。

   1.～3.の情報を確認し，問題がない場合は YES または yes を指定してください。

MODIFY コマンドの実行を中止する場合は，NO または no を指定してください。

参考

YES または yes 以外を指定した場合，NO が指定されたと見なされます。

注※

更新前後のレコード実現値は次の形式で表示されます。

項番	構成要素のデータ型	更新前後のデータの表示形式	表示例
1	CHARACTER	文字列定数の形式で表示されます。	'aaabbb'
2	XCHARACTER	16 進文字列定数の形式で表示されます。	X'01020304'
3	PACKED DECIMAL FIXED	10 進数定数の形式で表示されます。 ただし，更新後のレコード実現値が 10 進数定数の形式で表現できない場合は，16 進文字列定数の形式で表示されます。	123.456
4	INTEGER SMALLINT	整数定数の形式で表示されます。	100

各定数の形式については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

(b) MODIFY コマンドの実行結果

MODIFY コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
MODIFY INFORMATION
  ROWID                    : X'880000001900000000000100'  ...1
```

[説明]

1. 更新したレコードの ROWID が表示されます。

(6) 留意事項

1. SDB 用 UAP 環境定義の subschema オペランドの-a オプションに update を指定してください。SDB 用 UAP 環境定義については，「9.3 SDB 用 UAP 環境定義【SD FMB】」を参照してください。
2. MODIFY コマンドは，個別開始している状態で実行してください。
3. 更新対象のレコードへの位置づけをしておいてください。位置づけをしていない場合は，エラーになります。
4. 構成要素のデータ型とデータ属性が一致している更新値を指定する必要があります。更新値のパディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工は行われません。また，データ型によっては，次の条件を満たす必要があります。
   - データ型が CHARACTER または XCHARACTER の場合

     データ長が一致している必要があります。

- データ型が PACKED DECIMAL FIXED，INTEGER または SMALLINT の場合

  更新値が，構成要素のデータ型で表現できる範囲の値である必要があります。また，更新値を 16 進文字列定数で指定した場合，更新値の長さは，構成要素の長さと同じである必要があります。

  なお，データ型が PACKED DECIMAL FIXED の場合，更新値に指定した文字数で整数部桁数および小数部桁数が決まります。

  （例）

  01234.000 → 整数部桁数 5，小数部桁数 3

  .12345 → 整数部桁数 0，小数部桁数 5

  上記の例のようにゼロサプレスされません。更新値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。

5. MODIFY コマンドが正常終了したあとに#EXIT コマンドを実行する場合は，次のように pdsdbexe 操作コマンドを実行してください。
   - レコードの更新を実行したトランザクションを正常終了させたい場合は，COMMIT コマンドを実行してください。
   - レコードの更新を実行したトランザクションを取り消したい場合は，ROLLBACK コマンドを実行してください。

そのほかの留意事項については，「17.4.5　MODIFY 文」の「(4)　共通規則」を参照してください。

14.5.5　START（個別開始）

(1)　機能

HiRDB/SD に対して個別開始要求を行い，SDB データベースに対する操作の開始処理を行います。

(2)　指定形式

```
START SDBデータベース名;
```

注意事項

最後にセミコロン（ ; ）の指定が必要です。

(3)　オペランドの説明

●SDB データベース名

～<識別子>((1～30 バイト))

操作対象の SDB データベース名を指定します。

SDB データベース名に英小文字を使用している場合は，SDB データベース名を引用符（ " ）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

(4) 使用例

■SD FMB の SDB データベース（FMB1）を操作する場合

```
START ”FMB1”;
```

次の実行結果が表示されます。

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2018-12-14 16:20:46.67
  SDB DIRECTORY UPDATE DATE       : 2018-12-14 16:20:49.16
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : 10-03
KFPB64050-I DINF(GET DATE) command processing completed
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
START INFORMATION
  SDB DATABASE NAME ”FMB1”
  HANDLE NUMBER                   : 1
KFPB64050-I DINF(GET SDB SUMMARY) command processing completed
KFPB64050-I DINF(GET SDB DETAIL) command processing completed
KFPB64050-I DINF(GET SDB DATABASE) command processing completed
KFPB64050-I START command processing completed
```

(5) START コマンドの実行時に表示される情報

START コマンドを実行した場合に表示される情報について説明します。

(a) SDB ディレクトリ情報ファイルの日時情報

START コマンドを実行した際（トランザクション内の初回の START コマンド実行時），次に示す SDB ディレクトリ情報ファイルの日時情報が表示されます。

```
DIRECTORY CHECK DATE INFORMATION
  PDSDBDEF EXEC DATE              : 2018-12-14 16:20:46.67  ...1
  SDB DIRECTORY UPDATE DATE       : 2018-12-14 16:20:49.16  ...2
```

[説明]

1. 常用常駐領域中の SDB ディレクトリ情報の SDB 定義の最終更新日時（pdsdbdef コマンドの実行日時）が表示されます。
2. 常用常駐領域中の SDB ディレクトリ情報の最終更新日時が表示されます。

(b) HiRDB のバージョン情報

START コマンドを実行した際（トランザクション内の初回の START コマンド実行時），次に示す HiRDB のバージョン情報が表示されます。

```
HiRDB SERVER VERSION INFORMATION
  SERVER VERSION                  : vv-rr-xn
KFPB64050-I DINF(GET DATE) command processing completed
```

[説明]

下線部分に接続先の HiRDB のバージョンが表示されます。

(c)　個別開始に関する情報

START コマンドを実行した際，次に示す個別開始に関する情報が表示されます。

```
START INFORMATION
  SDB DATABASE NAME ”FMB1”                ...1
  HANDLE NUMBER              : 1          ...2
```

[説明]

1. START コマンドに指定した SDB データベース名が表示されます。
2. ハンドル番号が表示されます。

(6)　留意事項

SDB データベースを操作する前に，操作対象の SDB データベースに対して START コマンドを実行する必要があります。

個別開始の実行後，個別終了を実行していない状態で，個別開始を実行することはできません。

指定した SDB データベースのアクセスに際して，SDB 用 UAP 環境定義が有効になります。SDB 用 UAP 環境定義については「9.3　SDB 用 UAP 環境定義【SD FMB】」を参照してください。

14.5.6　STORE（レコードの格納）

(1)　機能

1 つのレコード実現値を格納します。

(2)　指定形式

指定形式については，「17.4.6　STORE 文」の「(2)　形式」の「DML コマンドで記述する STORE 文」を参照してください。

注意事項

- 最後にセミコロン（；）の指定が必要です。

- オペランドの指定順序は変更できません。

(3) オペランドの説明

オペランドの説明については，「17.4.6 STORE 文」の「(3) オペランドの説明」を参照してください。

(4) 使用例

STORE コマンドを実行してレコード実現値を格納する手順を次に示します。

手順

1. STORE コマンドを実行する

```
STORE "FMB101" SET "USERDB0"='DATA';
```

2. 格納対象のレコードが正しいかを確認する

```
STORE RECORD INFORMATION
  RECORD KIND                    : CHILD RECORD
  DBKEY                          : X'4141414101020304'
AGREE TO STORE ? [ENTER YES OR NO]
YES     ←上記の情報を確認後，YESを指定します。
```

STORE コマンドを実行してよいかどうかの確認メッセージが表示されます。

格納対象のレコード種別（RECORD KIND），および格納対象レコードのデータベースキー（DBKEY）を確認してから，YES を指定してください。

参考

格納対象のレコード種別などに誤りがある場合は，NO を指定して STORE コマンドの実行を中止してください。

3. STORE コマンドの実行結果を確認する

```
STORE INFORMATION
  STORE SERIAL NUM               : 1
  ROWID                          : X'0700000019000000000000200'
  RECORD POINTER STATUS          : FIRST_LAST
KFPB64050-I STORE command processing completed
```

STORE コマンドの実行結果が表示されます。

(5) STORE コマンドの実行時に表示される情報

STORE コマンドを実行した場合に表示される情報について説明します。

(a) STORE コマンドの実行確認メッセージ

STORE コマンドを実行した際，次に示す確認メッセージが表示されます。

```
STORE RECORD INFORMATION
  RECORD KIND              : CHILD RECORD          ...1
  DBKEY                    : X'4141414101020304'   ...2
AGREE TO STORE ? [ENTER YES OR NO]                  ...3
```

[説明]

1. 格納対象のレコード種別が表示されます。

   ROOT RECORD：格納対象レコードがルートレコードであることを意味しています。

   CHILD RECORD：格納対象レコードが子レコードであることを意味しています。

2. 格納対象レコードの親レコードのデータベースキーが表示されます。

   格納対象レコードが子レコードの場合に，この項目が表示されます。

   親レコードがルートレコードの場合，親レコードのデータベースキーが表示されます。親レコードがルートレコードではない場合，親レコードの一連番号が表示されます。

3. STORE コマンドを実行するかどうかの確認メッセージが表示されます。

   レコード種別および親レコードのデータベースキーを確認（1.および 2.の情報を確認）し，問題がない場合は YES または yes を指定してください。

   STORE コマンドの実行を中止する場合は，NO または no を指定してください。

   参考

   YES または yes 以外を指定した場合，NO が指定されたと見なされます。

(b) STORE コマンドの実行結果

STORE コマンドの実行が終了した際，次に示す実行結果が表示されます。

```
STORE INFORMATION
  STORE SERIAL NUM         : 20                              ...1
  ROWID                    : X'880000001900000000000100'     ...2
  RECORD POINTER STATUS    : LAST                            ...3
```

[説明]

1. 格納したレコードの一連番号が表示されます。
2. 格納したレコードの ROWID が表示されます。
3. レコード格納時のレコードポインタ情報が表示されます。

   FIRST：格納したレコードは，FIRST ポインタで示されています。

   LAST：格納したレコードは，LAST ポインタで示されています。

   FIRST LAST：格納したレコードは，FIRST ポインタおよび LAST ポインタで示されています。

NONE：格納したレコードは，FIRST ポインタおよび LAST ポインタのどちらでも示されていません。

(6) 留意事項

1. SDB 用 UAP 環境定義の subschema オペランドの-a オプションに update を指定してください。SDB 用 UAP 環境定義については，「9.3 SDB 用 UAP 環境定義【SD FMB】」を参照してください。
2. STORE コマンドは，個別開始している状態で実行してください。
3. 子レコードにレコードを格納する場合，親レコードまでの位置づけをしておいてください。位置づけをしていない場合は，エラーになります。
4. 構成要素のデータ型とデータ属性が一致している格納値を指定する必要があります。格納値のパディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工は行われません。また，データ型によっては，次の条件を満たす必要があります。
   - データ型が CHARACTER または XCHARACTER の場合

     データ長が一致している必要があります。
   - データ型が PACKED DECIMAL FIXED，INTEGER または SMALLINT の場合

     格納値が，構成要素のデータ型で表現できる範囲の値である必要があります。また，格納値を 16 進文字列定数で指定した場合，格納値の長さは，構成要素の長さと同じである必要があります。

     なお，データ型が PACKED DECIMAL FIXED の場合，格納値に指定した文字数で整数部桁数および小数部桁数が決まります。

     （例）

     01234.000 → 整数部桁数 5，小数部桁数 3

     .12345 → 整数部桁数 0，小数部桁数 5

     上記の例のようにゼロサプレスされません。格納値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。
5. STORE コマンドが正常終了したあとに#EXIT コマンドを実行する場合は，次のように pdsdbexe 操作コマンドを実行してください。
   - レコードの格納を実行したトランザクションを正常終了させたい場合は，COMMIT コマンドを実行してください。
   - レコードの格納を実行したトランザクションを取り消したい場合は，ROLLBACK コマンドを実行してください。

そのほかの留意事項については，「17.4.6 STORE 文」の「(4) 共通規則」を参照してください。

14.6　DML コマンド共通の規則

START コマンド以外の DML コマンドは，個別開始している状態で実行してください。

14.7　SQL コマンド

SQL コマンドの機能と入力形式について説明します。

14.7.1　COMMIT（トランザクションの正常終了）

(1)　機能

現在のトランザクションを正常終了させ，同期点を設定し 1 コミットメント単位を生成します。そのトランザクションが更新したデータベースの内容を有効にします。

(2)　指定形式

```
COMMIT;
```

注意事項

最後にセミコロン（ ; ）の指定が必要です。

(3)　留意事項

個別開始を実行した状態で COMMIT コマンドを実行すると，システムが個別終了を実行した後でトランザクションを正常終了します。

14.7.2　CONNECT（HiRDB への接続）

(1)　機能

HiRDB に接続します。

DISCONNECT コマンドを実行して HiRDB からの切り離しをしたあとに，HiRDB に再度接続する際に CONNECT コマンドを実行します。

(2)　指定形式

```
CONNECT 〔認可識別子〔/パスワード〕 〔ホスト名/ポート番号〕〕;
```

注意事項

最後にセミコロン（；）の指定が必要です。

(3) オペランドの説明

●認可識別子

～<識別子>((1～30 文字))

HiRDB に接続する認可識別子を指定します。

認可識別子に英小文字を使用している場合は，認可識別子を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

なお，認可識別子およびパスワードの指定を省略した場合，CONNECT コマンドの実行後，認可識別子およびパスワードの入力要求があります。

●パスワード

～<識別子>((1～30 文字))

認可識別子のパスワードを指定します。

パスワードに英小文字を使用している場合は，パスワードを引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

なお，認可識別子を指定してパスワードを省略した場合，CONNECT コマンドの実行後，パスワードの入力要求があります。

●ホスト名

～<識別子>((1～511 バイト))

接続先の HiRDB のホスト名を指定します。ホスト名の指定形式は，クライアント環境定義の PDHOST と同じです。PDHOST については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

次の条件に該当する文字列をホスト名に指定する場合は，ホスト名を引用符（"）で囲んで指定してください。

- 英小文字を指定する場合
- 31 バイト以上の文字列を指定する場合
- 英字，数字，下線（_）以外を指定する場合
- 先頭に英文字以外を指定する場合

なお，ホスト名およびポート番号の指定を省略した場合，pdsdbexe コマンドの引数に指定したホスト名とポート番号が適用されます。pdsdbexe コマンドの引数でもホスト名とポート番号を省略した場合は，クライアント環境定義の PDHOST と PDNAMEPORT の指定値が適用されます。

PDHOST および PDNAMEPORT については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

●ポート番号

～<符号なし整数>((5,001～65,535))

接続先の HiRDB のポート番号を指定します。

(4) 使用例

CONNECT コマンドを実行して，HiRDB に接続する手順を次に示します。

(a) コマンド入力画面で CONNECT コマンドを実行する場合

手順

1. CONNECT コマンドを実行する

```
CONNECT;
```

2. 認可識別子とパスワードを指定する

```
user-id:"User01"  ←認可識別子を指定して，［Enter］キーを押します。
"User01"
password:_______  ←パスワードを指定して，［Enter］キーを押します。
```

- 入力したパスワードは表示されません。
- 認可識別子またはパスワードに英小文字がある場合は，引用符（ " ）で囲んでください。
- 認可識別子およびパスワードは，入力行の先頭から指定してください。
- パスワードの指定は省略できません。

> **参考**
>
> - 1.の手順で，認可識別子およびパスワードを指定した場合は，上記の認可識別子およびパスワードの入力要求は表示されません。
> - 1.の手順で，認可識別子だけを指定した場合は，パスワードの入力要求だけが表示されます。

3. HiRDB に接続したことを確認する

```
CONNECT INFORMATION
  SERVER NAME "fes01"
  PROCESS ID                    : 31615
  CONNECT NUMBER                : 5
KFPB64050-I CONNECT command processing completed
```

HiRDB に接続すると，上記のメッセージが表示されます。

CONNECT INFORMATION には，次の情報が表示されます。

- SERVER NAME：接続したサーバ名が表示されます。
- PROCESS ID：接続したサーバプロセスの ID が表示されます。

- CONNECT NUMBER：コネクト通番が表示されます。

■ホスト名およびポート番号を指定する場合

ホスト名およびポート番号を指定する場合の CONNECT コマンドの指定例を次に示します。

（例）

認可識別子 User01，パスワード Pswd01 で HiRDB に接続します。接続先 HiRDB のホスト名は host1 で，ポート番号は 20000 です。

```
CONNECT ”User01”/”Pswd01” ”host1”/20000;
```

注意事項

英小文字を指定する場合は，引用符（"）で囲んでください。囲まないと，英小文字は英大文字と見なされます。

(b)　ファイルに記述した CONNECT コマンドをリダイレクトで実行する場合

ファイルに記述した CONNECT コマンドをリダイレクトで実行する場合，次のどちらかの形式で CONNECT コマンドを記述してください。

（例）

認可識別子 User01，パスワード Pswd01 で HiRDB に接続します。

- その 1

```
CONNECT ”User01”/”Pswd01”;
```

- その 2

```
CONNECT;
”User01”  ←認可識別子を記述します（最後に改行が必要）。
”Pswd01”  ←パスワードを記述します（最後に改行が必要）。
```

認可識別子またはパスワードに英小文字がある場合は，引用符（ " ）で囲んでください。

その 2 の形式で記述する場合，認可識別子およびパスワードは，入力行の先頭から記述してください。

14.7.3　DISCONNECT（HiRDB からの切り離し）

(1)　機能

実行中のトランザクションのコミット後に，HiRDB からの切り離しを実行します。

(2)　指定形式

```
DISCONNECT;
```

注意事項

最後にセミコロン（；）の指定が必要です。

(3) 留意事項

1. DISCONNECT コマンドを実行した場合，HiRDB からの切り離しは実行されますが，pdsdbexe コマンドは終了しません。pdsdbexe コマンドを終了するには，#EXIT コマンドを実行する必要があります。
2. DISCONNECT コマンドの実行後，CONNECT コマンドを実行すると，再度 HiRDB に接続できます。
3. DISCONNECT コマンドを実行した場合，実行中のトランザクションのコミット後に，HiRDB からの切り離しを実行します。そのため，トランザクションを取り消したい場合は，DISCONNECT コマンドを実行する前に ROLLBACK コマンドを実行してください。

14.7.4 ROLLBACK（トランザクションの取り消し）

(1) 機能

トランザクションを取り消して，そのトランザクション内でのデータベースの更新を無効にします。

(2) 指定形式

```
ROLLBACK;
```

注意事項

最後にセミコロン（；）の指定が必要です。

(3) 留意事項

個別開始を実行した状態で ROLLBACK コマンドを実行すると，システムが個別終了を実行した後でトランザクションを取り消します。

14.8　pdsdbexe サブコマンド

pdsdbexe サブコマンドの機能と入力形式について説明します。

14.8.1　#EXIT（pdsdbexe コマンドの終了）

(1)　機能

HiRDB サーバからの切り離しを実行し，pdsdbexe コマンドを終了します。

(2)　指定形式

```
#EXIT;
```

注意事項

最後にセミコロン（ ; ）の指定が必要です。

(3)　留意事項

1. 実行中のトランザクションを正常終了させたい場合は，#EXIT コマンドを実行する前に，必ず COMMIT コマンドを実行してください。
2. 実行中のトランザクションを取り消したい場合は，#EXIT コマンドを実行する前に，必ず ROLLBACK コマンドを実行してください。

14.8.2　#HELP（ヘルプ情報の出力）

(1)　機能

pdsdbexe 操作コマンドの指定形式と指定規則を出力します。

(2)　指定形式

(a)　pdsdbexe コマンド起動時の-d オプションに 4V を指定した場合

```
#HELP  {CLEAR | ERASE | FETCH | FETCHDB_ALL | FINISH | MODIFY | START | STORE
       | COMMIT | CONNECT | DISCONNECT | ROLLBACK
       | #EXIT | #HELP | #USAGE}
```

```
      〔 {SUMMARY | ALL} 〕 ;
```

(b)　pdsdbexe コマンド起動時の-d オプションに SD を指定した場合

```
#HELP  {ERASE | FETCH | FINISH | MODIFY | START | STORE
       | COMMIT | CONNECT | DISCONNECT | ROLLBACK
       | #EXIT | #HELP | #USAGE}
      〔 {SUMMARY | ALL} 〕 ;
```

注意事項

最後にセミコロン（ ; ）の指定が必要です。

(3)　オペランドの説明

●4V の場合

{CLEAR | ERASE | FETCH | FETCHDB_ALL | FINISH | MODIFY | START | STORE
| COMMIT | CONNECT | DISCONNECT | ROLLBACK
| #EXIT | #HELP | #USAGE}

●SD の場合

{ERASE | FETCH | FINISH | MODIFY | START | STORE
| COMMIT | CONNECT | DISCONNECT | ROLLBACK
| #EXIT | #HELP | #USAGE}

指定形式と指定規則を出力する pdsdbexe 操作コマンドを指定します。

CLEAR：CLEAR コマンドの指定形式と指定規則を出力します。

ERASE：ERASE コマンドの指定形式と指定規則を出力します。

FETCH：FETCH コマンドの指定形式と指定規則を出力します。

FETCHDB_ALL：FETCHDB_ALL コマンドの指定形式と指定規則を出力します。

FINISH：FINISH コマンドの指定形式と指定規則を出力します。

MODIFY：MODIFY コマンドの指定形式と指定規則を出力します。

START：START コマンドの指定形式と指定規則を出力します。

STORE：STORE コマンドの指定形式と指定規則を出力します。

COMMIT：COMMIT コマンドの指定形式と指定規則を出力します。

CONNECT：CONNECT コマンドの指定形式と指定規則を出力します。

DISCONNECT：DISCONNECT コマンドの指定形式と指定規則を出力します。

ROLLBACK：ROLLBACK コマンドの指定形式と指定規則を出力します。

#EXIT：#EXIT コマンドの指定形式と指定規則を出力します。

#HELP：#HELP コマンドの指定形式と指定規則を出力します。

#USAGE：#USAGE コマンドの指定形式と指定規則を出力します。

- {SUMMARY | ALL}

  出力する情報を指定します。

  SUMMARY：pdsdbexe 操作コマンドの指定形式だけを出力します。

  ALL：pdsdbexe 操作コマンドの指定形式および指定規則を出力します。

(4) 使用例

START コマンドの指定形式および指定規則を出力します。

```
#HELP START ALL;
```

次の実行結果が表示されます。

```
START "SDB_DATABASE_NAME"
       [ACSMODE   { READ  | UPDATE }]
       [ENQOPT    { SHARE | EXCLUSIVE | OCCUPY | NONE }]
       [AUTODEQ   { YES   | NO }]
       [ACSRDAREA ("RDAREA_NAME"[,"RDAREA_NAME"]...)]
       [RDAREAVAL { YES   | NO }]
       [FETCHDB_ALL ONLY]
      ;

機能
   HiRDB/SDに対して個別開始要求を行い,
   SDBデータベースに対する操作の開始処理を行う。
説明
(a)"SDB_DATABASE_NAME":SDBデータベース名
   操作対象のSDBデータベース名を指定する
(b)ACSMODE
   アクセスモードを指定する
   ・READ:参照モード
   ・UPDATE:更新モード
(c)ENQOPT
   排他オプションを指定する
   ・SHARE:共用モード
   ・EXCLUSIVE:排他モード
   ・OCCUPY:占有モード
   ・NONE:無排他モード
(d)AUTODEQ
   排他自動解除オプションを指定する
   ・YES:排他自動解除機能を使用する
   ・NO:排他自動解除機能を使用しない
(e)ACSRDAREA
   "RDAREA_NAME":RDエリア名
   検索対象とするRDエリアを指定する
   本オペランドを省略した場合，すべてのRDエリアを指定したと仮定される
(f)RDAREAVAL
```

```
   RDエリア指定有効オプションを指定する
    ・YES:基点条件指定がある場合でも，RDエリア指定を有効とする
    ・NO:基点条件指定がある場合には，RDエリア指定を有効としない
(g)FETCHDB_ALL ONLY
   FETCHDB_ALLコマンド専用の個別開始とする場合に指定する

KFPB64050-I #HELP command processing completed
```

参考

出力情報の文字コードは，クライアント環境定義の PDLANG で指定した文字コードになります。

14.8.3 #USAGE（DML コマンドのひな形の出力）

(1) 機能

SDB データベースまたはレコードに対する DML コマンドのひな形を出力します。出力されたひな形を基にして，DML コマンドを作成できます。

(2) 指定形式

(a) pdsdbexe コマンド起動時の-d オプションに 4V を指定した場合

```
#USAGE  {CLEAR | ERASE | FETCH | FETCHDB_ALL | FINISH | MODIFY | START | STORE}
      〔 {SDBデータベース名 | レコード名} 〕
      〔 {DBKEY | USERKEY} 〕 ;
```

(b) pdsdbexe コマンド起動時の-d オプションに SD を指定した場合

```
#USAGE  {ERASE | FETCH | FINISH | MODIFY | START | STORE}
      {SDBデータベース名 | レコード名}
```

注意事項

最後にセミコロン（ ; ）の指定が必要です。

(3) オペランドの説明

●4V の場合

{CLEAR | ERASE | FETCH | FETCHDB_ALL | FINISH | MODIFY | START | STORE}

●SD の場合

{ERASE ｜ FETCH ｜ FINISH ｜ MODIFY ｜ START ｜ STORE}

ひな形を出力する DML コマンドを指定します。

CLEAR：CLEAR コマンドのひな形を出力します。

ERASE：ERASE コマンドのひな形を出力します。

FETCH：FETCH コマンドのひな形を出力します。

FETCHDB_ALL：FETCHDB_ALL コマンドのひな形を出力します。

FINISH：FINISH コマンドのひな形を出力します。

MODIFY：MODIFY コマンドのひな形を出力します。

START：START コマンドのひな形を出力します。

STORE：STORE コマンドのひな形を出力します。

#USAGE コマンドを実行できるタイミングを次に示します。

- START コマンドのひな形を出力する場合

  個別開始前または個別終了後に，#USAGE コマンドを実行できます。個別開始から個別終了（COMMIT コマンド，ROLLBACK コマンド，DISCONNECT コマンドを含む）までの間は，#USAGE コマンドを実行できません。

- CLEAR コマンド，ERASE コマンド，FETCH コマンド，FETCHDB_ALL コマンド，FINISH コマンド，MODIFY コマンド，または STORE コマンドのひな形を出力する場合

  個別開始から個別終了（COMMIT コマンド，ROLLBACK コマンド，DISCONNECT コマンドを含む）までの間に限り，#USAGE コマンドを実行できます。個別開始前または個別終了後には，#USAGE コマンドを実行できません。

● {SDB データベース名｜レコード名}

～<識別子>((1～30 バイト))

DML コマンドの操作対象となる SDB データベース名またはレコード名を指定します。

START または FINISH を指定した場合は，SDB データベース名を指定します。

CLEAR，ERASE，FETCH，MODIFY，または STORE を指定した場合は，レコード名を指定します。

SDB データベース名またはレコード名に英小文字を使用している場合は，SDB データベース名またはレコード名を引用符（ " ）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

● {DBKEY ｜ USERKEY}

出力する条件の種別を指定します。このオペランドは，FETCH を指定した場合に指定できます。

DBKEY：DBKEY の条件を出力します。

USERKEY：ユーザキーの条件を出力します。ユーザキーの定義がある場合に指定できます。

(4) 使用例

■CLEAR コマンドのひな形を出力する場合

```
#USAGE CLEAR ”MAM1000”;
```

次の実行結果が表示されます。

```
CLEAR ”MAM1000”
      WHERE (DBKEY = X’********’)
;
```

■ERASE コマンドのひな形を出力する場合（-d オプションに 4V を指定した場合）

```
#USAGE ERASE ”FMB1000”;
```

次の実行結果が表示されます。

```
ERASE ”FMB1000”
      TARGET { SEGMENT | OCCURRENCE }
;
```

■ERASE コマンドのひな形を出力する場合（-d オプションに SD を指定した場合）

```
#USAGE ERASE ”FMB1000”;
```

次の実行結果が表示されます。

```
ERASE ”FMB1000”
      ALL
;
```

■FETCH コマンドのひな形を出力する場合（-d オプションに 4V を指定した場合）

```
#USAGE FETCH ”FMB1000”;
```

次の実行結果が表示されます。

```
FETCH ”FMB1000”
      { FIRST | LAST | NEXT | PRIOR }
      WHERE
        (DBKEY = ’********’)
      TERMOPT   { YES | NO }
      SECIDXSUP { YES | NO }
;
```

■FETCH コマンドのひな形を出力する場合（ルートレコード，-d オプションに SD を指定した場合）

```
#USAGE FETCH ”FMB1000”;
```

次の実行結果が表示されます。

```
FETCH FOR UPDATE
      { FIRST | NEXT }
      ”FMB1000”
      WHERE
```

```
        ("KEY1" = '********')
;
```

■FETCH コマンドのひな形を出力する場合（子レコード，-d オプションに SD を指定した場合）

```
#USAGE FETCH "FMB1001";
```

次の実行結果が表示されます。

```
FETCH FOR UPDATE
      { FIRST | LAST | NEXT | PRIOR }
      "FMB1001"
      WITHIN OWNER
;
```

■FETCHDB_ALL コマンドのひな形を出力する場合

```
#USAGE FETCHDB_ALL;
```

次の実行結果が表示されます。

```
FETCHDB_ALL
      { FIRST | NEXT | SKIP }
      BUFFER_SIZE *****
      WHERE (DBKEY <= X'********' AND
             DBKEY >= X'********')
      UNTIL NOT FOUND
      TERMOPT   { YES | NO }
;
```

■MODIFY コマンドのひな形を出力する場合（-d オプションに 4V を指定した場合）

```
#USAGE MODIFY "FMB1000";
```

次の実行結果が表示されます。

```
MODIFY "FMB1000"
       SET "USERDA0" = ***.**
       USERPTROPT { SET | NOSET | CLEAR }
;
```

■MODIFY コマンドのひな形を出力する場合（-d オプションに SD を指定した場合）

```
#USAGE MODIFY "FMB1000";
```

次の実行結果が表示されます。

```
MODIFY "FMB1000"
       SET "USERDA0" = ***.**
;
```

■START コマンドのひな形を出力する場合

```
#USAGE START "FMB1";
```

次の実行結果が表示されます。

```
START "FMB1"
      ACSMODE   { READ  | UPDATE }
      ENQOPT    { SHARE | EXCLUSIVE | OCCUPY | NONE }
      AUTODEQ   { YES   | NO }
      ACSRDAREA ("*****","*****")
      RDAREAVAL { YES   | NO }
      FETCHDB_ALL ONLY
;
```

■STORE コマンドのひな形を出力する場合（-d オプションに 4V を指定した場合）

```
#USAGE STORE "FMB1000";
```

次の実行結果が表示されます。

```
STORE "FMB1000"
      SET "KEYDATA1" = '****'
        , "KEYDATA2" = '****'
        , "USERDA0" = ***.**
      DIRECTION  { NEXT | PRIOR }
      USERPTROPT { SET | NOSET }
      PAGESWITCH { CHANGE | OCCUPY | NOUSE }
      PCTFREEVAL { YES | NO }
      SECIDXSUP  { YES | NO }
      PAGENUM    **********
;
```

■STORE コマンドのひな形を出力する場合（-d オプションに SD を指定した場合）

```
#USAGE STORE "FMB1000";
```

次の実行結果が表示されます。

```
STORE "FMB1000"
      SET "KEYDATA1" = '****'
        , "KEYDATA2" = '****'
        , "USERDA0" = ***.**
;
```

14.9　規則および留意事項

pdsdbexe コマンド実行時の規則および留意事項について説明します。

14.9.1　規則

(1)　pdsdbexe コマンドの実行条件

- pdsdbexe コマンドは，HiRDB が稼働中のときだけ実行できます。
- pdsdbexe コマンドは，HiRDB に接続可能なマシンで実行してください。

参考

接続先の HiRDB のバージョンと pdsdbexe コマンドのバージョンが異なっていても pdsdbexe コマンドを実行できます。DML コマンドは，接続先の HiRDB のバージョンの仕様に従って実行されます。そのため，接続先の HiRDB のバージョンが古い場合，未サポートの機能が原因でエラーになることがあります。

接続先の HiRDB のバージョンは，個別開始時の HiRDB のバージョン情報に出力されます。
pdsdbexe コマンドのバージョンは，pdsdbexe コマンドの開始メッセージに出力されます。

(2)　コマンドの実行者

pdsdbexe コマンドを実行する際，CONNECT 権限が必要になります。

(3)　コマンド実行前の作業

- pdsdbexe コマンド実行時のカレントディレクトリに，pdsdbexe コマンドの実行者の書き込み権限があることを確認してください。
- クライアント環境定義の PDLANG に，pdsetup コマンドで指定した文字コードが指定されていることを確認してください。PDLANG については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

(4)　pdsdbexe 操作コマンドの指定規則

1. pdsdbexe 操作コマンドの終わりには，セミコロン（;）を記述してください。セミコロン以降は，コメントとして扱われます。セミコロンがない場合は，継続行（pdsdbexe 操作コマンドが途中である）として扱われます。例を次に示します。

   （例）

```
START "FMB1"                        ←継続行
      ACSRDAREA("RDAREA1",          ←継続行
                "RDAREA2");         ←DMLコマンドの終了
```

参考

継続行は，改行記号が削除されたあとに連結されます。

2. 改行コードには，LF（X'0A'）または，CRLF（X'0D0A'）を指定できます。pdsdbexe 操作コマンド中の任意の位置に改行コードを指定できますが，改行コードを区切り文字として使用することはできません。

   pdsdbexe 操作コマンド中に指定された改行コードは，pdsdbexe 操作コマンドの実行時にすべて削除されます。そのため，文字列定数の途中に改行コードを指定しても問題ありません（長い文字列定数を複数行に分けて記述できます）。

3. pdsdbexe 操作コマンド中のキーワードとキーワードの間や，キーワードと名前の間などには，区切り文字を指定する必要があります。区切り文字には，空白（X'20'）またはタブ（X'09'）を指定できます。

   例えば，「MODIFY RECB」と指定した場合，MODIFY と RECB の間に区切り文字を指定する必要があります。

   区切り文字を指定する必要がある個所については，マニュアル「HiRDB SQL リファレンス」の「区切り文字の挿入」を参照してください。

参考

キーワードとは，コマンド中に指定するコマンド名や，オペランド名などのことで，例えば次の文字列が該当します。

- START，FETCH，#EXIT などのコマンド名
- ACSMODE などのオペランド名と，READ，UPDATE などのオペランドの指定値

名前とは，コマンド中に指定する SDB データベース名などのことで，レコード名，構成要素名，RD エリア名，認可識別子，パスワードなど，識別子として指定する項目が該当します。

4. 認可識別子などの名前に英小文字を使用する場合は，名前を引用符（ " ）で囲んでください。引用符で囲まないと，英小文字は英大文字に変換されます。また，アポストロフィ（ ' ）または引用符で囲まれた部分は変換されません。

5. 名前に予約語を使用する場合は，名前を引用符（ " ）で囲んでください。囲まないと pdsdbexe 操作コマンドがエラーになることがあるため，名前は引用符で囲んで指定することを推奨します。予約語については，「付録 O.1　DML の予約語」を参照してください。

6. 次の個所にコメントを記述できます。/*と*/で囲まれた部分がコメントになります。

   - セミコロンから改行までの間
   - pdsdbexe 操作コマンド中ではなく，pdsdbexe 操作コマンドの文字列を含まない行

   （例 1）

```
START "FMB1"
       ACSRDAREA("RDAREA1",
                 "RDAREA2");      /* 個別開始 */<改行>
```

（例 2）

```
/* 個別開始 */
START "FMB1"
       ACSRDAREA("RDAREA1",
                 "RDAREA2");
/* FMB1へのアクセス */    /* RDエリア名RDAREA1,RDAREA2 */
```

7. pdsdbexe 操作コマンド中にコメントは記述できません。

（指定できない例）

```
START "FMB1"
       ACSRDAREA("RDAREA1",  /* 個別開始 */
       /* FMB1へのアクセス */
                 "RDAREA2");
```

8. コメントを指定した行には，pdsdbexe 操作コマンドを指定できません。

（指定できない例）

```
/* 個別開始 */    START "FMB1"
       ACSRDAREA("RDAREA1",
                 "RDAREA2");
```

9. コメントを複数行に分けて記述することはできません。

（指定できない例）

```
/* 個別開始
   FMB1へのアクセス */
 START "FMB1"
       ACSRDAREA("RDAREA1",
                 "RDAREA2");
```

10. 1 行の長さは，改行コードを含んで 65,530 バイトを超えてはいけません。

11. pdsdbexe 操作コマンドの長さは，2,000,000 バイトを超えてはいけません（pdsdbexe 操作コマンド中に指定した区切り文字も含む長さです）。

12. X'00'～X'1F'のコードを指定した場合，空白に変換されます。

13. pdsdbexe 操作コマンドで指定する文字の文字コードは，pdsetup コマンドで指定した文字コードと同じである必要があります。

14.9.2 留意事項

- pdsdbexe コマンドは，UAP と同様にクライアント環境定義が有効となります。ただし，クライアント環境定義の PDAUTOCONNECT は有効となりません。OFF が仮定された状態で動作します。

PDAUTOCONNECT については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

- pdsdbexe コマンドは，HiRDB で暗黙的ロールバックが発生しないエラーが起きたことを検知した場合，内部的にロールバックを実行し，トランザクションを取り消します。
- pdsdbexe コマンドは，回復不要 FES に接続して SDB データベースを操作できません。システム内に回復不要 FES ユニットがある場合は，クライアント環境定義の PDFESHOST および PDSERVICEGRP を指定して，回復不要 FES 以外のフロントエンドサーバに接続してください。PDFESHOST および PDSERVICEGRP については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

14.10　リターンコード

pdsdbexe コマンドのリターンコードの意味および対処を次の表に示します。

表 14–17　pdsdbexe コマンドのリターンコードの意味および対処

リターンコード	意味	対処
0	pdsdbexe コマンドが正常終了しました。	なし。
4	pdsdbexe コマンドの処理が完了しました。ただし，警告レベルのエラーが発生しました。	出力されたメッセージを参照して，メッセージの対処に従ってください。
8	HiRDB または pdsdbexe コマンドでエラーを検知しました。	
12	処理が続行できないエラーが発生しました。	

14.11　トラブルシューティング

ここでは，pdsdbexe コマンドの実行中に障害が発生したときの対処方法について説明します。

14.11.1　障害が発生したときの対処方法

pdsdbexe コマンドの実行中に障害が発生した場合，出力されたメッセージを確認して対処してください。

(1)　トラブルシュート機能

SDB データベースの操作で障害が発生した場合，HiRDB クライアントの次のトラブルシュート機能を使用して障害要因を調査できます。

- SQL トレース機能
- クライアントエラーログ機能
- 拡張 SQL エラー情報出力機能
- UAP 統計レポート機能

上記のトラブルシュート機能の詳細については，「16.5　UAP の障害対策」およびマニュアル「HiRDB UAP 開発ガイド」の「トラブルシュート」を参照してください。

また，pdsdbexe コマンドが異常終了した場合，クライアント環境定義の PDCLTPATH に指定したディレクトリ下に簡易ダンプが出力されます。ファイル名は次のとおりです（出力メッセージに簡易ダンプファイル名が表示されます）。

- ファイル名：

  pdsdbexe.abort.YYYYMMDDhhmmss.プロセスID

  YYYYMMDDhhmmss：出力日時

PDCLTPATH を省略した場合，または PDCLTPATH に指定したディレクトリに書き込みができなかった場合は，カレントディレクトリに簡易ダンプが出力されます。

なお，出力された簡易ダンプファイルを保守員に渡すことがあるため，簡易ダンプファイルは退避しておいてください。

(2)　UAP 処理時間監視機能との関係

pdsdbexe コマンドには，UAP 処理時間監視機能が適用されます。pdsdbexe コマンドが無応答状態などになった場合，クライアント環境定義の PDSWAITTIME に指定した最大待ち時間を超えると，実行中のトランザクションをロールバックし，HiRDB から自動的に切り離されます。

参考

次のケースでトランザクションが未終了の場合も，UAP 処理時間監視機能が適用されます。

- OS の kill コマンドでプロセスを中断した場合
- [Ctr]＋[c]で処理をキャンセルした場合

PDSWAITTIME については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

(3) 自動再接続機能との関係

クライアント環境定義の PDAUTORECONNECT に YES を指定している場合，pdsdbexe コマンドには自動再接続機能が適用されます。PDAUTORECONNECT については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。自動再接続機能については，マニュアル「HiRDB UAP 開発ガイド」の「自動再接続機能」を参照してください。

15 DML プリプロセサ（pdsdbcbl）【SD FMB】

この章では，DML プリプロセサ（pdsdbcbl）の機能と使い方について説明します。

15.1 機能

DML が記述されている COBOL の UAP ソースプログラムのプリプロセスを，DML プリプロセサ（pdsdbcbl）で実行します。プリプロセスを実行すると，UAP のソースプログラム中の DML を，COBOL のコンパイラでコンパイルできる高級言語の命令に置換したポストソースが出力されます。

DML プリプロセサ（pdsdbcbl）の機能，コマンドの指定方法については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編)」の「DML プリプロセサ（pdsdbcbl)」を参照してください。

16 HiRDB クライアントの環境設定

この章では，HiRDB クライアントのインストールや UAP の作成と実行に必要な環境定義の方法などについて説明します。

16.1　概要

4V FMB または 4V AFM の SDB データベースを操作するには，TP1/FSP が提供するライブラリ関数を使用します。SDB データベースにアクセスする UAP を開発するときの作業の流れ，UAP の特長，および SDB データベースとのインタフェースなどについては，次のマニュアルを参照してください。

- 「TP1/Financial Service Platform 使用の手引」
- 「TP1/Financial Service Platform プログラム作成の手引」
- 「TP1/Server Base Enterprise Option　プログラム作成の手引」

SD FMB の SDB データベースを操作するには，DML を使用します。DML については，「17.　DML リファレンス【SD FMB】」を参照してください。SDB データベースにアクセスする UAP を開発するときの作業の流れ，および SDB データベースとのインタフェースなどについては，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」を参照してください。

SDB データベースに対する操作については，「2.5　SDB データベースの操作」を参照してください。

16.2　UAP の設計

ここでは，性能向上，操作性向上に関する UAP の設計について説明します。

UAP を設計する上で必要な，UAP の記述言語，インタフェース領域などについては，マニュアル「TP1/Financial Service Platform 使用の手引」，マニュアル「TP1/Financial Service Platform プログラム作成の手引」，およびマニュアル「TP1/Server Base Enterprise Option　プログラム作成の手引」を参照してください。

なお，ここでは，「4V FMB または 4V AFM の SDB データベースを操作する UAP の設計」を対象として説明します。「SD FMB の SDB データベースを操作する UAP の設計」については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」を参照してください。

16.2.1　性能向上，操作性向上に関する UAP の設計【4V FMB，4V AFM】

(1)　SDB データベースを操作する API のエラーの判定と処置

SDB データベースを操作する API のエラーの判定と処置については，マニュアル「HiRDB UAP 開発ガイド」の「SQL のエラーの判定と処置」を参照してください。なお，「SQL」は「SDB データベースを操作する API」に読み替えてください。

(2)　ブロック転送機能

構造型 DB 機能を使用する HiRDB では，サポートしていません。

(3)　複数接続機能

複数接続機能については，マニュアル「HiRDB UAP 開発ガイド」の「複数接続機能」を参照してください。なお，「SQL」は「SDB データベースを操作する API」に読み替えてください。

(4)　自動再接続機能

自動再接続機能については，マニュアル「HiRDB UAP 開発ガイド」の「自動再接続機能」を参照してください。

(5)　文字コード変換機能

文字コード変換機能については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の一覧」にある「PDCLTCNVMODE」を参照してください。

構造型 DB 機能を使用する HiRDB では，変換対象となる文字列はエラーメッセージだけです。

(6) マルチスレッド対応

マルチスレッド対応については，マニュアル「HiRDB UAP 開発ガイド」の「複数接続機能」を参照してください。なお，「SQL」は「SDB データベースを操作する API」に読み替えてください。

16.3 HiRDB クライアントの環境設定

ここでは，クライアント環境定義について説明します。そのほかの HiRDB クライアントの環境設定については，次に示すマニュアルを参照してください。

HiRDB クライアントの種類

HiRDB クライアントには次の 2 つの種類があります。

- HiRDB Structured Data Access Facility/Run Time
- HiRDB Structured Data Access Facility/Developer's Kit

HiRDB Structured Data Access Facility/Run Time は，UAP の実行だけができます。

HiRDB Structured Data Access Facility/Developer's Kit は，SD FMB の SDB データベースを操作する UAP のプリプロセス，コンパイル，リンケージと，UAP の実行ができます。

HiRDB クライアントのインストール

HiRDB クライアントのインストールについては，マニュアル「HiRDB UAP 開発ガイド」の「HiRDB クライアントのインストール」を参照してください。

HiRDB クライアントの適用 OS

- Red Hat Enterprise Linux Server 7（64-bit x86_64）
- Red Hat Enterprise Linux Server 8（64-bit x86_64）
- Red Hat Enterprise Linux Server 9（64-bit x86_64）

HiRDB クライアントのディレクトリおよびファイル構成

HiRDB クライアントのディレクトリおよびファイル構成については，マニュアル「HiRDB UAP 開発ガイド」の「HiRDB クライアントのディレクトリおよびファイル構成」を参照してください。

HiRDB Structured Data Access Facility/Developer's Kit をインストールしたときに自動的に作成される HiRDB/SD 固有のファイルとディレクトリを次の表に示します。

表 16-1　HiRDB Structured Data Access Facility/Developer's Kit をインストールしたときに自動的に作成される HiRDB/SD 固有のファイルとディレクトリ

名称	ディレクトリ名	ファイル名
ヘッダファイル	/HiRDB/include	PDSDBHND.CBL
コマンド，ユティリティ	/HiRDB/client/utl	pdsdbcbl
		pdsdbprp

注

下線の部分は，HiRDB のインストールディレクトリを示します。

HiRDB が提供するライブラリは，複数接続用（マルチスレッド対応），複数接続 XA 用（動的接続または静的接続のマルチスレッド対応）のファイルだけです。

hosts ファイルの設定

hosts ファイルの設定については，マニュアル「HiRDB UAP 開発ガイド」の「hosts ファイルの設定」を参照してください。

16.3.1　クライアント環境定義（環境変数の設定）

UAP を実行するためには，HiRDB クライアントごとにクライアント環境定義を設定しておく必要があります。

(1)　クライアント環境定義の設定形式

クライアント環境定義の設定形式については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定形式」を参照してください。

(2)　OLTP 下の X/Open に従った API を使用した UAP をクライアントとする場合の指定方法

構造型 DB 機能を使用する HiRDB では，OpenTP1 下の UAP だけをサポートします。

OpenTP1 下の UAP については，マニュアル「HiRDB UAP 開発ガイド」の「OLTP 下の X/Open に従った API を使用した UAP をクライアントとする場合の指定方法」の次の項目を参照してください。

- 「OpenTP1 下の UAP をクライアントとする場合」
- 「TP1/EE 下の UAP をクライアントとする場合（UNIX 版限定）」

(3)　クライアント環境定義の一覧

クライアント環境定義の一覧を次の表に示します。クライアント環境定義の各オペランドの説明については，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

参考

DML を記述した UAP のプリプロセス時に参照する環境変数 PDCBLLIB については，次のマニュアルを参照してください。

- マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「DML プリプロセサ（pdsdbcbl コマンド）」の「環境変数の設定」

表中の「番号」欄は，マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」での，各クライアント環境定義のオペランドの番号と対応しています。

なお，クライアント環境定義のオペランドのうち，SDB データベースを操作する API または DML の実行時に有効とならないオペランドがあります。「有効/無効」欄が×のオペランドは，SDB データベースを操作する API または DML の実行時に有効となりません。

表 16-2　クライアント環境定義の一覧

番号	環境変数名	有効/無効	環境変数の分類
1	PDHOST	○	システム構成※1
2	PDNAMEPORT	○	
3	PDFESHOST	○	
4	PDSERVICEGRP	○	
5	PDSRVTYPE	○	
6	PDSERVICEPORT	○	
7	PDFESGRP	○	
8	PDCLTRCVPORT	○	
9	PDCLTRCVADDR	○	
10	PDTMID	○	OLTP 下の X/Open に従った API を使用するクライアント
11	PDXAMODE	○	
12	PDTXACANUM	○	
13	PDXARCVWTIME	○	
14	PDXATRCFILEMODE	○	
15	PDXAAUTORECONNECT	○	
16	HiRDB_PDHOST	○	
17	HiRDB_PDNAMEPORT	○	
18	HiRDB_PDTMID	○	
19	HiRDB_PDXAMODE	○	
20	PDUSER	○	ユーザ実行環境
21	PDCLTAPNAME	○	
22	PDCLTLANG	○	
23	PDLANG	○	
24	PDDBLOG	○	
25	PDEXWARN	△※4	
26	PDSUBSTRLEN	×	
27	PDCLTCNVMODE	△※5	

番号	環境変数名	有効/無効	環境変数の分類
28	PDCLTGAIJIDLL	×	
29	PDCLTGAIJIFUNC	×	
30	PDCLTCNVUOCLIB	×	
31	PDCLTCNVUOCFUNC	×	
32	PDCLTCNVBYTERATIO	×	
33	PDCLTGRP	○	
34	PDAUTORECONNECT	○	
35	PDRCCOUNT	○	
36	PDRCINTERVAL	○	
37	PDUAPENVFILE	○	
38	PDDBBUFLRU	○	
39	PDHATRNQUEUING	○	
40	PDCLTBINDLOOPBACKADDR	×	
41	PDEXTDECCHECK	×	
42	PDRCTIMING	○	
43	PDAUTHTYPE	○	
44	PDDEFAULTOPTION	○	
45	PDASTHOST	×	UAP からのコマンド実行
46	PDASTPORT	×	
47	PDSYSTEMID	×	
48	PDASTUSER	×	
49	PDCMDWAITTIME	×	
50	PDCMDTRACE	×	
51	PDIPC	○	プロセス間メモリ通信機能
52	PDSENDMEMSIZE	○	
53	PDRECVMEMSIZE	○	
54	PDCWAITTIME	○	システム監視
55	PDSWAITTIME	○	
56	PDSWATCHTIME	○	
57	PDCWAITTIMEWRNPNT	○	
58	PDKALVL	○	

番号	環境変数名	有効/無効	環境変数の分類
59	PDKATIME	○	
60	PDTIMEDOUTRETRY	○	
61	PDCONREFRCOUNT	○	
62	PDCONREFRINTERVAL	○	
63	PDNBLOCKWAITTIME	○	
64	PDCONNECTWAITTIME	○	
65	PDCLTPATH	○	トラブルシュート
66	PDSQLTRACE	○	
67	PDUAPERLOG	○	
68	PDERRSKIPCODE	○	
69	PDPRMTRC	×	
70	PDPRMTRCSIZE	×	
71	PDTRCMODE	○	
72	PDUAPREPLVL	○	
73	PDREPPATH	○	
74	PDTRCPATH	○	
75	PDSQLTRCOPENMODE	○	
76	PDSQLTEXTSIZE	×	
77	PDSQLEXECTIME	○	
78	PDRCTRACE	○	
79	PDWRTLNPATH	×	
80	PDWRTLNFILSZ	×	
81	PDWRTLNCOMSZ	×	
82	PDUAPEXERLOGUSE	○	
83	PDUAPEXERLOGPRMSZ	×	
84	PDDNDPTRACE	×	
85	PDSQLTRCFMT	○	
86	PDVWOPTMODE	×	アクセスパス表示ユティリティ用アクセスパス情報ファイル
87	PDTAAPINFPATH	×	HiRDB SQL Tuning Advisor用アクセスパス情報ファイル
88	PDTAAPINFMODE	×	

番号	環境変数名	有効/無効	環境変数の分類
89	PDTAAPINFSIZE	×	
90	PDSTJTRNOUT	○	UAP に関する統計情報の出力単位
91	PDLOCKLIMIT	○	排他制御
92	PDDLKPRIO	○	
93	PDLOCKSKIP	×	
94	PDFORUPDATEEXLOCK	×	
95	PDISLLVL	×	SQL 関連
96	PDSQLOPTLVL	×	
97	PDADDITIONALOPTLVL	×	
98	PDHASHTBLSIZE	×	
99	PDDFLNVAL	×	
100	PDAGGR	×	
101	PDCMMTBFDDL	×	
102	PDPRPCRCLS	×	
103	PDAUTOCONNECT	△※2	
104	PDDDLDEAPRPEXE	×	
105	PDDDLDEAPRP	×	
106	PDLCKWAITTIME	○	
107	PDCURSORLVL	×	
108	PDDELRSVWDFILE	×	
109	PDCALCMDWAITTIME	×	
110	PDSTANDARDSQLSTATE	△※3	
111	PDBLKF	×	ブロック転送機能
112	PDBINARYBLKF	×	
113	PDBLKBUFFSIZE	×	
114	PDBINDRETRYCOUNT	○	HiRDB の通信処理
115	PDBINDRETRYINTERVAL	○	
116	PDDBACCS	○	インナレプリカ機能
117	PDDBORGUAP	×	更新可能なオンライン再編成
118	PDSPACELVL	×	データの空白変換

番号	環境変数名	有効/無効	環境変数の分類
119	PDCLTRDNODE	×	XDM/RD E2 接続機能
120	PDTP1SERVICE	×	
121	PDRDCLTCODE	×	
122	PDCNSTRNTNAME	×	参照制約および検査制約
123	PDTMPTBLRDAREA	×	一時表
124	PDBESCONHOLD	○	バックエンドサーバ接続保持機能
125	PDBESCONHTI	○	
126	PDODBSTATCACHE	×	ODBC 関数
127	PDODBESCAPE	×	
128	PDGDATAOPT	×	
129	PDODBLOCATOR	×	
130	PDODBSPLITSIZE	×	
131	PDODBCWRNSKIP	×	
132	PDJETCOMPATIBLE	×	
133	PDODBGINFOSUPPRESS	×	
134	PDODBSTANDARDARGSIZE	×	
135	PDODBSTANDARDSQLSTATE	×	
136	PDODBSTANDARDDESCCOL	×	
137	PDODBSTANDARDGTYPEINFO	×	
138	PDPLGIXMK	×	プラグイン
139	PDPLUGINNSUB	×	
140	PDPLGPFSZ	×	
141	PDPLGPFSZEXP	×	
142	PDHSICOPTIONS	×	HSIC 限定
143	PDJDBFILEDIR	×	JDBC ドライバ
144	PDJDBFILEOUTNUM	×	
145	PDJDBONMEMNUM	×	
146	PDJDBTRACELEVEL	×	
147	PDJDBFILESIZE	×	
148	PDDNDPCOMPATIBLE	×	ADO.NET

（凡例）

○：SDB データベースを操作する API または DML の実行時に有効となります。

△：SDB データベースを操作する API または DML の実行時に有効となりますが，注意事項があります。

×：SDB データベースを操作する API または DML の実行時に有効となりません。

注※1

システム構成に関する環境変数には，HiRDB サーバと接続するときに必要な情報を指定します。HiRDB サーバとの接続形態によっては，環境変数が指定できないことがあります。

HiRDB サーバとの接続形態については，マニュアル「HiRDB UAP 開発ガイド」の「HiRDB サーバと接続するための環境変数と接続形態との関係」を参照してください。

注※2

pdsdbexe コマンドに対しては，PDAUTOCONNECT の指定が適用されません。pdsdbexe コマンドは，OFF が仮定された状態で動作します。

注※3

システム定義との関連については，マニュアル「HiRDB UAP 開発ガイド」を参照してください。SQLSTATE についてはマニュアル「HiRDB メッセージ」を参照してください。

注※4

一度の要求で複数のレコードに対してアクセスする要求の場合は，最後のレコードに対するアクセスで発生した警告つきのリターンコードだけ受け取ります。複数のレコードに対してアクセスする要求については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」の次の項目を参照してください。

・「SDB ハンドラ制御（ee_sdh_～）(C 言語)」の「ee_sdh_accs」

・「SDB ハンドラ制御（CBLEESDH)」の「CBLEESDH('ACCS ')」

注※5

SDB データベースを操作する API または DML の実行時は省略または NOUSE を指定してください。指定した場合，API 実行時に変換対象となる文字列はエラーメッセージだけです。

(4) SDB データベースを操作する API または DML の実行時にだけ指定できるクライアント環境定義

SDB データベースを操作する API または DML の実行時には，「表 16-2　クライアント環境定義の一覧」で説明しているクライアント環境定義のオペランド以外に，次の表に示すオペランドが指定できます。

表 16-3　SDB データベースを操作する API または DML の実行時にだけ指定できるクライアント環境定義の一覧

番号	環境変数名	環境変数の分類
1	PDUAPEXERLOGDMLSZ	トラブルシュート
2	PDUAPEXERLOGDMLDATA	
3	PDSDBPRMTRC	
4	PDSDBTRCCNDSIZE	
5	PDSDBTRCELMSIZE	
6	PDSDBTRCKEYSIZE	
7	PDSDBTRCDATASIZE	

番号	環境変数名	環境変数の分類
8	PDSDBUAPFILE	ユーザ実行環境
9	PDSDBUAPDIR	

●PDUAPEXERLOGDMLSZ=DML 履歴用バッファのサイズ

～〈符号なし整数〉((0，6～65,535))《80》（単位：キロバイト）

SDB データベースを操作する API または DML の実行時に確保する DML 履歴用バッファのサイズをキロバイト単位で指定します。このオペランドに 0 を指定した場合は，DML 履歴用バッファを確保しません。

DML 履歴用バッファについては「16.5.4(2)　DML 履歴用バッファ」を参照してください。

《見積もり方法》

指定値は，次の計算式から見積もってください。

指定値＝↑{(2.5 ＋ ukey_size)×req_num}↑（単位：キロバイト）

ukey_size：操作するレコードのユーザキーのサイズ（単位：キロバイト）

req_num：同一トランザクション内での SDB データベースを操作する API の要求回数，または同一トランザクション内での DML の要求回数

《留意事項》

- pdsdbexe コマンドを使ってアクセスする場合，見積もり方法の計算式で求めた値に，以下の計算式で求めた値を加算してください。

  ↑3.6×アクセスする SDB データベースの数↑（単位：キロバイト）

- 見積もり方法の指定値は，平均的な使用を想定した場合の目安です。SDB データベースを操作する API または DML の実行時の要求内容によっては，DML 履歴用バッファのサイズが不足することがあります。不足した場合，見積もり値より大きな値を指定してください。
- 一度の要求で複数のレコードに対してアクセスする要求の場合は，個々のアクセスごとに 1 回の要求として計算してください。複数のレコードに対してアクセスする要求については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」の「SDB ハンドラ制御（ee_sdh_～）（C 言語）」の「ee_sdh_accs」または「SDB ハンドラ制御（CBLEESDH）」の「CBLEESDH('ACCS ')」を参照してください。
- DML 履歴用バッファに記録される情報が多くなるに従って，拡張 SQL エラー情報に出力されるエリアダンプの出力量が増加します。その結果，拡張 SQL エラー情報の出力に掛かる時間が増加します。SQL 文，SDB データベースを操作する API または DML でエラーが発生した際の処理時間が，要求元のタイマ監視などで問題となる場合は，PDUAPEXERLOGDMLSZ の指定値を見直してください。
- DML 履歴用バッファは，HiRDB サーバとの接続後，最初に SDB データベースを操作する API または DML が正常終了した際に確保されます。そのため，SQL しか要求しないで DML 履歴用バッファを使用しない場合に，明示的に 0 を指定する必要はありません。

●PDUAPEXERLOGDMLDATA= {YES | NO}

拡張 SQL エラー情報の出力時，DMLINF の出力項目のうち STORDATA をエラーログファイルに出力するかどうかを指定します。

YES：

DMLINF の出力項目 STORDATA をエラーログファイルに出力します。

NO：

DMLINF の出力項目 STORDATA をエラーログファイルに出力しません。

出力される拡張 SQL エラー情報の詳細については，「16.5.4　拡張 SQL エラー情報出力機能」を参照してください。

●PDSDBPRMTRC= {YES | NO}

SQL トレースに DML のパラメタ情報および実行結果の情報を出力するかどうかを指定します。

YES：

DML のパラメタ情報および実行結果の情報を SQL トレースに出力します。

NO：

DML のパラメタ情報および実行結果の情報を SQL トレースに出力しません。

SQL トレースに出力される DML のパラメタ情報および実行結果の情報の詳細については，「16.5.2 SQL トレース機能」を参照してください。

YES を指定した場合は，出力する情報量が増えます。そのため，SQL トレースファイルのサイズ（PDSQLTRACE の指定値）を大きくする必要があります。また，必要に応じて，出力するエリア情報の最大データ長（PDSDBTRCCNDSIZE，PDSDBTRCELMSIZE，PDSDBTRCKEYSIZE，PDSDBTRCDATASIZE）を指定して，出力する情報量を抑える対処をしてください。

●PDSDBTRCCNDSIZE=SQL トレースに出力する条件の値ブロックの最大データ長

～〈符号なし整数〉((0～2,000,000))《2,000,000》（単位：バイト）

SQL トレースに出力するキーとキー以外の条件の値ブロックの最大データ長を指定します。

このオペランドは，PDSDBPRMTRC オペランドに YES を指定した場合に有効になります。

このオペランドに 0 を指定した場合は，キーとキー以外の条件の値ブロックの情報を SQL トレースに出力しません。

このオペランドの指定を省略するか，または 0 以外を指定した場合は，キーとキー以外の条件の値ブロックに加えて，キーとキー以外の条件の条件ブロックの情報も SQL トレースに出力します。キーとキー以外の条件の値ブロックのデータ長が指定の最大データ長を超えたときは，指定の最大データ長までの情報を出力し，最大データ長を超える部分は出力しません。

- 条件式形式オプションに'*FDN'を指定した場合

  キーの条件の条件ブロックおよび値ブロックの形式については，「図 7-13　キーの条件の条件ブロックの形式（条件式形式オプションに'*FDN'を指定した場合）」，「図 7-15　キーの条件の値ブロックの形式（条件式形式オプションに'*FDN'を指定した場合）」を参照してください。

キー以外の条件の条件ブロックおよび値ブロックの形式については，「図 7-14　キー以外の条件の条件ブロックの形式（条件式形式オプションに'*FDN'を指定した場合）」，「図 7-16　キー以外の条件の値ブロック形式（条件式形式オプションに'*FDN'を指定した場合）」を参照してください。

- 条件式形式オプションに'*EFN'を指定した場合

  キーの条件の条件ブロックおよび値ブロックの形式については，「図 7-17　キーの条件の条件ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）」，「図 7-19　キーの条件の値ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）」を参照してください。

  キー以外の条件の条件ブロックおよび値ブロックの形式については，「図 7-18　キー以外の条件の条件ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）」，「図 7-20　キー以外の条件の値ブロックの形式（条件式形式オプションに'*EFN'を指定した場合）」を参照してください。

●PDSDBTRCELMSIZE=SQL トレースに出力する構成要素指定エリアの最大データ長

～〈符号なし整数〉((0～2,000,000))《2,000,000》（単位：バイト）

SQL トレースに出力する構成要素指定エリアの最大データ長を指定します。

このオペランドは，PDSDBPRMTRC オペランドに YES を指定した場合に有効になります。

このオペランドに 0 を指定した場合は，構成要素指定エリアの情報を SQL トレースに出力しません。

このオペランドの指定を省略するか，または 0 以外を指定した場合は，構成要素指定エリアの情報を SQL トレースに出力します。構成要素指定エリアのデータ長が指定の最大データ長を超えたときは，指定の最大データ長までの情報を出力し，最大データ長を超える部分は出力しません。

構成要素指定エリアの形式については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

●PDSDBTRCKEYSIZE=SQL トレースに出力するキー報告エリアのキー情報の最大データ長

～〈符号なし整数〉((0～2,000,000))《2,000,000》（単位：バイト）

SQL トレースに出力するキー報告エリアのキー情報の最大データ長を指定します。

このオペランドは，PDSDBPRMTRC オペランドに YES を指定した場合に有効になります。

このオペランドに 0 を指定した場合は，キー報告エリアのキー情報を SQL トレースに出力しません。

このオペランドの指定を省略するか，または 0 以外を指定した場合は，キー報告エリアのキー情報を SQL トレースに出力します。キー報告エリアのキー情報のデータ長が指定の最大データ長を超えたときは，指定の最大データ長までの情報を出力し，最大データ長を超える部分は出力しません。

SDB データベース種別が 4V FMB および 4V AFM の場合のキー報告エリアのキー情報（キー部）の形式については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」を参照してください。

SDB データベース種別が SD FMB の場合のキー報告エリアのキー情報（キー部）の形式を次の図に示します。

図 16-1　キー報告エリアのキー情報（キー部）の形式（SD FMB の SDB データベースの場合）

サイズ部																キー部				
K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	予備	ルートキー ※2	※1	※1	…	※1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	K1	K2	K3		Kn

（凡例）K1～K15：各階層レベルのキーサイズ
予備：予備エリア
注※1　一連番号
注※2　ルートレコードのデータベースキー

●PDSDBTRCDATASIZE=SQL トレースに出力するデータ格納領域のレコード情報の最大データ長
～〈符号なし整数〉((0～2,000,000))《2,000,000》（単位：バイト）

SQL トレースに出力するデータ格納領域のレコード情報の最大データ長を指定します。

このオペランドは，PDSDBPRMTRC オペランドに YES を指定した場合に有効になります。

このオペランドに 0 を指定した場合は，データ格納領域のレコード情報を SQL トレースに出力しません。

このオペランドの指定を省略するか，または 0 以外を指定した場合は，データ格納領域のレコード情報を SQL トレースに出力します。データ格納領域のレコード情報のデータ長が指定の最大データ長を超えたときは，指定の最大データ長までの情報を出力し，最大データ長を超える部分は出力しません。

●PDSDBUAPFILE=SDB 用 UAP 環境定義ファイル名
～＜識別子＞((最大 8 文字))

このオペランドは，SDB 用 UAP 環境定義に関するオペランドです。

UAP が SD FMB の SDB データベースにアクセスする際に，使用する SDB 用 UAP 環境定義ファイルの名称を指定します。

UAP を個別の実行環境で動作させる場合（実行する UAP に対して SDB 用 UAP 環境定義を適用する場合），このオペランドを指定します。この指定をすることで，UAP ごとに実行環境を切り替える運用ができます。

SDB 用 UAP 環境定義については，次の個所を参照してください。

- 「4.9　SDB 用 UAP 環境定義ファイルの準備【SD FMB】」
- 「9.3　SDB 用 UAP 環境定義【SD FMB】」

このオペランドの指定例を次に示します。

（例）

```
PDSDBUAPFILE=uapenv01
```

《留意事項》

- このオペランドに指定した SDB 用 UAP 環境定義の指定内容に誤りがある場合は，UAP が HiRDB サーバに CONNECT したときにエラーになります。
- SDB 用 UAP 環境定義ファイル中に SDB 用 UAP 環境定義の指定がない場合，このオペランドの指定は無効になります。そのため，SDB 用 UAP 環境定義ファイルがないものとして動作します（エラーにはなりません）。

●PDSDBUAPDIR=SDB 用 UAP 環境定義ファイルの格納ディレクトリ名

～＜識別子＞((最大 8 文字))

このオペランドは，SDB 用 UAP 環境定義に関するオペランドです。

$PDCONFPATH/pdsdbuapenv 下に SDB 用 UAP 環境定義ファイルを格納するディレクトリを作成する場合に，このオペランドを指定します。

$PDCONFPATH/pdsdbuapenv 下に格納する SDB 用 UAP 環境定義ファイルの数が多くなると，HiRDB/SD がファイルサーチする際の処理時間が長くなります。そのため，SDB 用 UAP 環境定義ファイルの数が多くなる場合は，業務システムなどの単位で，$PDCONFPATH/pdsdbuapenv 下にさらにディレクトリを作成することを推奨します。

このオペランドの指定例を次に示します。

（例）

```
PDSDBUAPDIR=system01
```

上記の例のように指定した場合，$PDCONFPATH/pdsdbuapenv/system01 下に SDB 用 UAP 環境定義ファイルを格納する必要があります。

PDSDBUAPFILE オペランドおよび PDSDBUAPDIR オペランドの指定有無によって，適用される SDB 用 UAP 環境定義ファイルが次のように決まります。

- PDSDBUAPFILE オペランドだけを指定した場合（PDSDBUAPDIR オペランドを指定しない場合）

  （例）

  ```
  PDSDBUAPFILE=uapenv01
  ```

  $PDCONFPATH/pdsdbuapenv 下に格納されている SDB 用 UAP 環境定義ファイル（uapenv01）が適用されます。

- PDSDBUAPFILE オペランドおよび PDSDBUAPDIR オペランドを指定した場合

  （例）

  ```
  PDSDBUAPFILE=uapenv01
  PDSDBUAPDIR=system01
  ```

  $PDCONFPATH/pdsdbuapenv/system01 下に格納されている SDB 用 UAP 環境定義ファイル（uapenv01）が適用されます。

16.4　UAP 実行前の準備

ここでは，4V FMB または 4V AFM の SDB データベースを操作する UAP を実行する前の準備作業について説明します。UAP の実行手順，プリプロセスについては，次に示すマニュアルを参照してください。

- UAP の実行手順
  詳細については，マニュアル「HiRDB UAP 開発ガイド」の「UAP の実行手順」を参照してください。
- プリプロセス
  詳細については，マニュアル「HiRDB UAP 開発ガイド」の「プリプロセス」を参照してください。

SD FMB の SDB データベースを操作する UAP を実行する前の準備作業（UAP の実行手順，プリプロセス）については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の実行前準備(UAP のプリプロセス，コンパイル，リンケージ)」を参照してください。

16.4.1　コンパイルとリンケージ

4V FMB または 4V AFM の SDB データベースを操作する UAP のコンパイルおよびリンケージをするときは，HiRDB が提供する次の表に示す共用ライブラリファイルを指定します。

表 16–4　コンパイルおよびリンケージをするときに指定する共用ライブラリファイル

項番	条件	共用ライブラリファイル名	備考
1	OLTP 下でない場合	libzcltk64.so	複数接続機能に対応しています。
2	OLTP 下の場合	libzcltyk64.so	トランザクション登録方式は，動的登録または静的登録となります。

16.5　UAP の障害対策

ここでは，UAP 実行時の履歴やエラー情報を取得するトラブルシュート，および UAP の障害の種別と回復方法について説明します。

16.5.1　トラブルシュート

UAP に障害が発生した場合に，トラブルシュート機能を利用して障害要因を調査できます。トラブルシュート機能には次のものがあります。

- SQL トレース機能
- クライアントエラーログ機能
- 拡張 SQL エラー情報出力機能
- UAP 統計レポート機能

16.5.2　SQL トレース機能

実行した UAP の SQL トレース情報を SQL トレースファイルに取得します。SDB データベースを操作する API または DML の実行時にエラーが発生した場合，SQL トレース情報を参照すると，エラーの原因となる HiRDB の情報を特定できます。

SQL トレースファイルは，取得した情報で満杯になると，最も古い情報から順次新しい情報に書き替えられます。SQL トレース機能の詳細については，マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース機能」を参照してください。

(1)　SQL トレース情報の見方

SDB データベースを操作する API または DML の実行時に出力される SQL トレース情報の出力例を次の図に示します。

図 16-2　SQL トレース情報の出力例（クライアント環境定義 PDSQLTRCFMT に 1 を指定した場合）

```
** UAP TRACE (CLT:09-66(Jan 15 2018) SVR:09-66   US) LINX64(UNIX_64K)**

USER APPLICATION PROGRAM FILE NAME : SQLTRACE001_1
USERID : PT1
UAP START TIME : 2018/01/22 14:12:20
UAP ENVIRONMENT :
  LANG(ja_JP.SJIS)
  USER(PT1)
  HOST(sdb-bs2500-3)
  NAMEPORT(22453)
  FESHOST()
  SVCGRP()  SVCPORT()  SRVTYPE(WS)
  SWAIT(600)  CWAIT(0)  SWATCH(-1)
  BLKF(1)  RDABLKF(-1)  LCKLMT(0) ISLLVL(-1) DBLOG(ALL) DFLNVAL(NOUSE)
  AGGR(1024)  DLKPRIO(64)  EXWARN(NO)  VWOPTMODE(0)
  LOCKSKIP(NO)  CLTGRP(1)  DSQLOBJCACHE()  PLGIXMK(NO)
  CLTRCVPORT(0) CLTRCVADDR() PLGPFSZ(8192)
  PLGPFSZEXP(8192) SPACELVL(-1) STJTRNOUT(YES)
  OPTLVL()
  ADDITIONALOPTLVL()
  UAPREPLVL(s) REPPATH()
  TRCPATH()
  IPC() SENDMEMSIZE(0) RECVMEMSIZE(0)
  HASHTBLSIZE(-1) CMMTBFDDL() PRPCRCLS() DBACCS()
  SQLTRCOPENMODE() AUTOCONNECT() CWAITTIMEWRNPNT(-1) TCPCONOPT(1)
  WRTLNFILSZ(-1) WRTLNCOMSZ(1024)
  WRTLNPATH() UAPENVFILE()
  TP1SERVICE(NO) AUTORECONNECT(NO) RCCOUNT(0) RCINTERVAL(0)
  KALVL(2) KATIME(600) CLTCNVMODE(NOUSE)
  PRMTRC(NO) PRMTRCSIZE(256) BESCONHOLD() BESCONHTI(-1)
  BLKBUFFSIZE(10) BINARYBLKF(NO) FORUPDATEEXLOCK(YES) DBORGUAP(NO)
  CNSTRNTNAME() SQLTEXTSIZE(4096) RCTRACE(-1)
  FESGRP()
  NBLOCKWAITTIME(0) CONNECTWAITTIME(300) DBBUFLRU(YES)
  UAPEXERLOGUSE() UAPEXERLOGPRMSZ(-1) HJHASHINGMODE()
  DDLDEAPRP(NO) DELRSVWDFILE() HATRNQUEUING()
  ODBSPLITSIZE(0) NODELAYACK(YES) CURSORLVL(0)
  TAAPINFPATH() TAAPINFMODE(0) TAAPINFSIZE(409600)
  JETCOMPATIBLE(NO) SUBSTRLEN() BLKFUPD(YES) ARYERRPOS(YES)
  CALCMDWAITTIME(0) BLKFERRBREAK(YES) XAAUTORECONNECT(NO)
  CLTBINDLOOPBACKADDR(NO)
  STANDARDSQLSTATE() LCKWAITTIME(-1) DDLDEAPRPEXE(NO)
  ODBGINFOSUPPRESS(NO) ODBSTANDARDARGSIZE(NO)
  ODBSTANDARDSQLSTATE(NO) ODBSTANDARDDESCCOL(NO)
  TMPTBLRDAREA()
  EXTDECCHECK(NO)
  DNDPCOMPATIBLE(NO)
  RCTIMING()
  ODBSTANDARDGTYPEINFO(NO)
  HSICOPTIONS()
  DEFAULTOPTION(RECOM)
  TCPREDUCETIMEWAITPORT(YES) TCPREDUCECONCLOSERCVTIME(1000)
  TRNSUSPENDWATCHTIME(-1)
```

（続く）

（続き）

```
    CLTGAIJIMODE() CLTGAIJIFUNC() CLTGAIJIDLL()
    AUTHTYPE(PA)
    UAPEXERLOGDMLSZ(80) UAPEXERLOGDMLDATA(NO)
    SDBUAPFILE() SDBUAPDIR()
  CONNECTION STATUS :
    CURHOST(sdb-bs2500-3)
    CURPORT(35744) SVRNAME(testfes1)
    CNCTNO(8) SVRPID(3038) CLTPID(17018) CLTTID(-82057408) CLTCNCTHDL(0x1bf4210)

CNCT  CLPID CLTID NO     OP   SEC  SQL  SQL   START-TIME   END-TIME     OP   EXEC-TIME
NO                       CODE NO   CODE WARN                          TION
----- ----- ----- ------ ---- ---- ---- ----- ------------ ------------ ---- ----------
    8 17018 -82057      1 CNCT    0    0 -0000 14:12:20.384 14:12:20.386 0000        1995
    8 17018 -82057      7 EXSD    1    0 -0000 14:12:20.387 14:12:20.387 0000          72
     *DML* (*4V ) HNDLNO:          1 FNCCD:S REQCD:STRT DBNAME:FMBT2
           REQNUM:1
           RECNAME:
00:00:00.000072 00:00:00.000029  43 0 0 0 0 0 0
    8 17018 -82057      8 EXSD    1    0 -0000 14:12:20.388 14:12:20.389 0000         661
     *DML* (*4V ) HNDLNO:          1 FNCCD:A REQCD:FTCH DBNAME:FMBT2
           REQNUM:3
           RECNAME:FMBT2000
           RECNAME:FMBT202C001
           RECNAME:FMBT202C002
00:00:00.000661 00:00:00.000508  153 0 0 0 0 0 0
    8 17018 -82057      9 EXSD    1    0 -0000 14:12:20.393 14:12:20.393 0000         206
     *DML* (*4V ) HNDLNO:          1 FNCCD:E REQCD:FNSH DBNAME:FMBT2
           REQNUM:1
           RECNAME:
00:00:00.000206 00:00:00.000126  80 0 0 0 0 0 0
    8 17018 -82057     10 EXSD    1    0 -0000 14:12:20.394 14:12:20.394 0000         343
     *DML* (*4V ) HNDLNO:          2 FNCCD:A REQCD:FTCH DBNAME:FMBT2
           REQNUM:1   REQ+:STRT/FNSH
           RECNAME:FMBT2000
00:00:00.000343 00:00:00.000229  114 0 0 0 0 0 0
    8 17018 -82057     11 EXSD    1    0 -0000 14:12:20.396 14:12:20.396 0000         353
     *DML* (*4V ) HNDLNO:          3 FNCCD:A REQCD:FTCH DBNAME:FMBT2
           REQNUM:1   REQ+:STRT(0)
           RECNAME:FMBT2000
00:00:00.000353 00:00:00.000246  107 0 0 0 0 0 0
    8 17018 -82057     12 EXSD    1    0 -0000 14:12:20.398 14:12:20.398 0000         242
     *DML* (*4V ) HNDLNO:          3 FNCCD:E REQCD:FNSH DBNAME:FMBT2
           REQNUM:1
           RECNAME:
00:00:00.000242 00:00:00.000133  109 0 0 0 0 0 0
    8 17018 -82057     13 EXSD    1-66129-0000 14:12:20.399 14:12:20.400 0000         303
     *DML* (*4V ) HNDLNO:          0 FNCCD:A REQCD:FTCH DBNAME:FMBT2
           REQNUM:1   REQ+:STRT/FNSH
                      REQ+:STRT(0)
           RECNAME:FMBT2000
00:00:00.000303 00:00:00.000100  203 0 0 0 0 0 0
    8 17018 -82057     14 CMIT    0    0 -0000 14:12:20.404 14:12:20.405 0000        1130
    8 17018 -82057     15 DISC    0    0 -0000 14:12:20.406 14:12:20.406 0000         153
```

図 16-3　SQL トレース情報の出力例（クライアント環境定義 PDSQLTRCFMT に 2 を指定した場合）

```
** UAP TRACE TYPE2 (CLT:09-66(Jan 15 2018) SVR:09-66   US) LINX64(UNIX_64K)**

USER APPLICATION PROGRAM FILE NAME : SQLTRACE001_2
USERID : PT1
UAP START TIME : 2018/01/22 14:12:20
UAP ENVIRONMENT :
  LANG(ja_JP.SJIS)
  USER(PT1)
  HOST(sdb-bs2500-3)
  NAMEPORT(22453)
  FESHOST()
  SVCGRP()  SVCPORT()  SRVTYPE(WS)
  SWAIT(600)  CWAIT(0)  SWATCH(-1)
  BLKF(1)  RDABLKF(-1)  LCKLMT(0) ISLLVL(-1) DBLOG(ALL) DFLNVAL(NOUSE)
  AGGR(1024)  DLKPRIO(64)  EXWARN(NO)  VWOPTMODE(0)
  LOCKSKIP(NO)  CLTGRP(1)  DSQLOBJCACHE()  PLGIXMK(NO)
  CLTRCVPORT(0) CLTRCVADDR() PLGPFSZ(8192)
  PLGPFSZEXP(8192) SPACELVL(-1) STJTRNOUT(YES)
  OPTLVL()
  ADDITIONALOPTLVL()
  UAPREPLVL(s) REPPATH()
  TRCPATH()
  IPC() SENDMEMSIZE(0) RECVMEMSIZE(0)
  HASHTBLSIZE(-1) CMMTBFDDL() PRPCRCLS() DBACCS()
  SQLTRCOPENMODE() AUTOCONNECT() CWAITTIMEWRNPNT(-1) TCPCONOPT(1)
  WRTLNFILSZ(-1) WRTLNCOMSZ(1024)
  WRTLNPATH() UAPENVFILE()
  TP1SERVICE(NO) AUTORECONNECT(NO) RCCOUNT(0) RCINTERVAL(0)
  KALVL(2) KATIME(600) CLTCNVMODE(NOUSE)
  PRMTRC(NO) PRMTRCSIZE(256) BESCONHOLD() BESCONHTI(-1)
  BLKBUFFSIZE(10) BINARYBLKF(NO) FORUPDATEEXLOCK(YES) DBORGUAP(NO)
  CNSTRNTNAME() SQLTEXTSIZE(4096) RCTRACE(-1)
  FESGRP()
  NBLOCKWAITTIME(0) CONNECTWAITTIME(300) DBBUFLRU(YES)
  UAPEXERLOGUSE() UAPEXERLOGPRMSZ(-1) HJHASHINGMODE()
  DDLDEAPRP(NO) DELRSVWDFILE() HATRNQUEUING()
  ODBSPLITSIZE(0) NODELAYACK(YES) CURSORLVL(0)
  TAAPINFPATH() TAAPINFMODE(0) TAAPINFSIZE(409600)
  JETCOMPATIBLE(NO) SUBSTRLEN() BLKFUPD(YES) ARYERRPOS(YES)
  CALCMDWAITTIME(0) BLKFERRBREAK(YES) XAAUTORECONNECT(NO)
  CLTBINDLOOPBACKADDR(NO)
  STANDARDSQLSTATE() LCKWAITTIME(-1) DDLDEAPRPEXE(NO)
  ODBGINFOSUPPRESS(NO) ODBSTANDARDARGSIZE(NO)
  ODBSTANDARDSQLSTATE(NO) ODBSTANDARDDESCCOL(NO)
  TMPTBLRDAREA()
  EXTDECCHECK(NO)
  DNDPCOMPATIBLE(NO)
  RCTIMING()
  ODBSTANDARDGTYPEINFO(NO)
  HSICOPTIONS()
  DEFAULTOPTION(RECOM)
```

（続く）

（続き）

```
    TCPREDUCETIMEWAITPORT(YES) TCPREDUCECONCLOSERCVTIME(1000)
    TRNSUSPENDWATCHTIME(-1)
    CLTGAIJIMODE() CLTGAIJIFUNC() CLTGAIJIDLL()
    AUTHTYPE(PA)
    UAPEXERLOGDMLSZ(80) UAPEXERLOGDMLDATA(NO)
    SDBUAPFILE() SDBUAPDIR()
  HIRDBINIPATH :
  CONNECTION STATUS :
    CURHOST(sdb-bs2500-3)
    CURPORT(35744) SVRNAME(testfes1)
    CNCTNO(10) SVRPID(3038) CLTPID(17073) CLTTID(-783657152) CLTCNCTHDL(0x220c210)
  SOCKET STATUS :
    2018/01/22 14:12:20.589  CLT(10.197.12.113  :41134) -> RDM(10.197.12.113  :22453)
                             CLT()                 <- RDM()
    2018/01/22 14:12:20.589  CLT(10.197.12.113  :41136) -> NDM(10.197.12.113  :22453)
                             CLT()                 <- NDM()
    2018/01/22 14:12:20.589  CLT(10.197.12.113  :50668) -> SCD(10.197.12.113  :35744)
    2018/01/22 14:12:20.590  CLT(10.197.12.113  :55033) <- SRV(10.197.12.113  :59201)
```

（続く）

（続き）

```
CNCT  CLPID      CLTID      NO      OP   SEC  SQL      SQL   START-TIME              END-TIME                HJCMP  HJSCH
NO                                  CODE NO   CODE     WARN                                                COUNT  COUNT
----- ---------- ---------- ------ ---- ---- -------- ----- ----------------------- ----------------------- ------ ------
   10      17073 -783657152      1 CNCT    0        0 -0000 2018/01/22 14:12:20.588 2018/01/22 14:12:20      0      0
    *DML* (*4V ) HNDLNO:         1 FNCCD:S REQCD:STRT DBNAME:FMBT2
          REQNUM:1
          RECNAME:
   10      17073 -783657152      8 EXSD    1        0 -0000 2018/01/22 14:12:20.592 2018/01/22 14:12:20      0      0
    *DML* (*4V ) HNDLNO:         1 FNCCD:A REQCD:FTCH DBNAME:FMBT2
          REQNUM:3
          RECNAME:FMBT2000
          RECNAME:FMBT202C001
          RECNAME:FMBT202C002
   10      17073 -783657152      9 EXSD    1        0 -0000 2018/01/22 14:12:20.596 2018/01/22 14:12:20.     0      0
    *DML* (*4V ) HNDLNO:         1 FNCCD:E REQCD:FNSH DBNAME:FMBT2                                    (中略)
          REQNUM:1
          RECNAME:
   10      17073 -783657152     10 EXSD    1        0 -0000 2018/01/22 14:12:20.598 2018/01/22 14:12:20.5    0      0
    *DML* (*4V ) HNDLNO:         2 FNCCD:A REQCD:FTCH DBNAME:FMBT2
          REQNUM:1   REQ+:STRT/FNSH
          RECNAME:FMBT2000
   10      17073 -783657152     11 EXSD    1        0 -0000 2018/01/22 14:12:20.600 2018/01/22 14:12:20.6    0
    *DML* (*4V ) HNDLNO:         3 FNCCD:A REQCD:FTCH DBNAME:FMBT2
          REQNUM:1   REQ+:STRT(0)
          RECNAME:FMBT2000
   10      17073 -783657152     12 EXSD    1        0 -0000 2018/01/22 14:12:20.602 2018/01/22 14:12:20.6    0
    *DML* (*4V ) HNDLNO:         3 FNCCD:E REQCD:FNSH DBNAME:FMBT2
          REQNUM:1
          RECNAME:
   10      17073 -783657152     13 EXSD    1   -66129 -0000 2018/01/22 14:12:20.603 2018/01/22 14:12:20.6    0
    *DML* (*4V ) HNDLNO:         0 FNCCD:A REQCD:FTCH DBNAME:FMBT2
          REQNUM:1   REQ+:STRT/FNSH
                     REQ+:STRT(0)
          RECNAME:FMBT2000
   10      17073 -783657152     14 CMIT    0        0 -0000 2018/01/22 14:12:20.608 2018/01/22 14:12:20.6    0000
```

図 16-4　SQL トレース情報の出力例（クライアント環境定義 PDSDBPRMTRC に YES を指定した場合）

```
** UAP TRACE TYPE2 (CLT:10-01(Feb 12 2019) SVR:10-01   US) LINX64(UNIX_64K)**

USER APPLICATION PROGRAM FILE NAME : Unknown
USERID : PT1
UAP START TIME : 2019/02/12 16:34:42
UAP ENVIRONMENT :
  LANG(ja_JP.SJIS)
  USER()
  HOST(10.197.12.113)
  NAMEPORT(22461)
  FESHOST()
  SVCGRP()  SVCPORT()  SRVTYPE(WS)
  SWAIT(600)  CWAIT(0)  SWATCH(-1)
  BLKF(1)  RDABLKF(-1)  LCKLMT(0) ISLLVL(-1) DBLOG(ALL) DFLNVAL(NOUSE)
  AGGR(1024)  DLKPRIO(64)  EXWARN(NO)  VWOPTMODE(0)
  LOCKSKIP(NO)  CLTGRP(1)  DSQLOBJCACHE()  PLGIXMK(NO)
  CLTRCVPORT(0) CLTRCVADDR() PLGPFSZ(8192)
  PLGPFSZEXP(8192) SPACELVL(-1) STJTRNOUT(YES)
  OPTLVL()
  ADDITIONALOPTLVL()
  UAPREPLVL(s) REPPATH()
  TRCPATH()
  IPC() SENDMEMSIZE(0) RECVMEMSIZE(0)
  HASHTBLSIZE(-1) CMMTBFDDL() PRPCRCLS() DBACCS()
  SQLTRCOPENMODE() AUTOCONNECT() CWAITTIMEWRNPNT(-1) TCPCONOPT(1)
  WRTLNFILSZ(-1) WRTLNCOMSZ(1024)
  WRTLNPATH() UAPENVFILE()
  TP1SERVICE(NO) AUTORECONNECT(NO) RCCOUNT(0) RCINTERVAL(0)
  KALVL(2) KATIME(600) CLTCNVMODE(NOUSE)
  PRMTRC(NO) PRMTRCSIZE(256) BESCONHOLD() BESCONHTI(-1)
  BLKBUFFSIZE(10) BINARYBLKF(NO) FORUPDATEEXLOCK(YES) DBORGUAP(NO)
  CNSTRNTNAME() SQLTEXTSIZE(4096) RCTRACE(-1)
  FESGRP()
  NBLOCKWAITTIME(0) CONNECTWAITTIME(300) DBBUFLRU(YES)
  UAPEXERLOGUSE() UAPEXERLOGPRMSZ(-1) HJHASHINGMODE()
  DDLDEAPRP(NO) DELRSVWDFILE() HATRNQUEUING()
  ODBSPLITSIZE(0) NODELAYACK(YES) CURSORLVL(0)
  TAAPINFPATH() TAAPINFMODE(0) TAAPINFSIZE(409600)
  JETCOMPATIBLE(NO) SUBSTRLEN() BLKFUPD(YES) ARYERRPOS(YES)
  CALCMDWAITTIME(0) BLKFERRBREAK(YES) XAAUTORECONNECT(NO)
  CLTBINDLOOPBACKADDR(NO)
  STANDARDSQLSTATE() LCKWAITTIME(-1) DDLDEAPRPEXE(NO)
  ODBGINFOSUPPRESS(NO) ODBSTANDARDARGSIZE(NO)
  ODBSTANDARDSQLSTATE(NO) ODBSTANDARDDESCCOL(NO)
  TMPTBLRDAREA()
  EXTDECCHECK(NO)
  DNDPCOMPATIBLE(NO)
  RCTIMING()
  ODBSTANDARDGTYPEINFO(NO)
  HSICOPTIONS()
  DEFAULTOPTION(RECOM)
  TCPREDUCETIMEWAITPORT(YES) TCPREDUCECONCLOSERCVTIME(1000)
  TRNSUSPENDWATCHTIME(-1)
  CLTGAIJIMODE() CLTGAIJIFUNC() CLTGAIJIDLL()
  AUTHTYPE(PA)
  CLTCNVBYTERATIO() CLTCNVUOCFUNC()
```

（続く）

（続き）

```
   CLTCNVUOCLIB()
   TIMEDOUTRETRY(2) BINDRETRYCOUNT(10) BINDRETRYINTERVAL(0)
   ODBCWRNSKIP(NO) ODBLOCATOR(NO) ODBESCAPE(1)
   ODBSTATCACHE(0) GDATAOPT(NO) DNDPTRACE()
   CLTRDNODE() RDCLTCODE() UAPERLOG(65536)
   ERRSKIPCODE()
   TRCMODE(ERR) XATRCFILEMODE(SEPARATE)
   RETRYCOUNT(10) RETRYINTERVAL(700)
   CONREFRCOUNT(9) CONREFRINTERVAL(700)
   UAPEXERLOGDMLSZ(80) UAPEXERLOGDMLDATA(NO)
   SDBUAPDIR() SDBUAPFILE()
   SDBPRMTRC(YES) SDBTRCCNDSIZE(2000000) SDBTRCELMSIZE(2000000)
   SDBTRCKEYSIZE(2000000) SDBTRCDATASIZE(2000000)
 HIRDBINIPATH :
 CONNECTION STATUS :
   CURHOST(sdb-bs2500-3)
   CURPORT(52352) SVRNAME(sds01)
   CNCTNO(11) SVRPID(7763) CLTPID(7968) CLTTID(-1251788992) CLTCNCTHDL(0x1b03010)
 SOCKET STATUS :
   2019/02/12 16:34:42.992  CLT(10.197.12.113  :50918) -> RDM(10.197.12.113  :22461)
                            CLT()                  <- RDM()
   2019/02/12 16:34:42.992  CLT(10.197.12.113  :50920) -> NDM(10.197.12.113  :22461)
                            CLT()                  <- NDM()
   2019/02/12 16:34:42.992  CLT(10.197.12.113  :55820) -> SCD(10.197.12.113  :52352)
   2019/02/12 16:34:42.992  CLT(10.197.12.113  :50148) <- SRV(10.197.12.113  :59950)

CNCT  CLPID      CLTID      NO     OP   SEC  SQL      SQL   START-TIME              END-TIME                PREPROCESS FILE                PREPROCESS   HJCMP  HJSCH
NO                                 CODE NO   CODE     WARN                                                                              TIME         COUNT  COUNT
---- --------- ---------- ------ ---- ---- -------- ----- ----------------------- ----------------------- ------------------------------ ---------- ------ ------
  11       7968 -1251788992      1 CNCT    0        0 -0000 2019/02/12 16:34:42.991 2019/02/12 1        *                              *
  11       7968 -1251788992      2 EXSD    1        0 -0000 2019/02/12 16:34:42.993 2019/02/12 16:       *                              *            0      0
    *DML* (*4V ) HNDLNO:          1 FNCCD:S REQCD:STRT DBNAME:FMBT2
          REQNUM:1
          IV=  512,AM=R,EQ=S                                                                  (中略)                                                 (中略)
          RECNAME:
  11       7968 -1251788992      3 EXSD    1        0 -0000 2019/02/12 16:34:42.993 2019/02/12 16:       *                              *            0      0
    *DML* (*4V ) HNDLNO:          1 FNCCD:A REQCD:FTCH DBNAME:FMBT2
          REQNUM:3
          IV=  512,AM=R,EQ=S
          RECNAME:FMBT2000
          CS=R,AC=F,DS=D,TO=Y,CF=*FDN,CK=Y,KA=Y
          ROWID=0x2b0000001c00000000000100
          CONK=44 42 4b 45 59 20 20 20 45 51 5f                *DBKEY   EQ_     *
               03 30 30 30 30 31 32 33 34 35 36 37 38          *.000012345678   *
          KEY =03 30 30 30 30 31 32 33 34 35 36 37 38          *.000012345678   *
          DATA=41 41 42 42 43 43 43 44 45 45 45 46 47 47 48 49 *AABBCCCDEEEFGGHI*
               49 4a 4a 4b                                     *IJJK            *
          RECNAME:FMBT202C001
          CS=P,AC=F,DS=D,TO=Y,CF=*FDN,CK=Y,CD=Y,KA=Y
          NO=        29
          ROWID=0x2b0000001c00000000001e00
          CONK=44 42 4b 45 59 20 20 20 47 54 5f                *DBKEY   GT_     *
               0a 00 00 00                                     *....            *
          COND=55 44 41 54 41 31 20 20 45 51 5f                *UDATA1  EQ_     *
               65                                              *e               *
```

（続く）

（続き）

```
        KEY =03 30 30 30 30 31 32 33 34 35 36 37 38 1d 00 00   *.000012345678...*
             00                                                *.              *
        DATA=65 4c 4c 4c 4d 4d 4e 4e 4e 4f 4f 50 50 50 51 52   *eLLLMMNNNOOPPPQR*
             52 53 53 53 54 54 54 54 54 55 55 55 56             *RSSSTTTTTTUUUV  *
        RECNAME:FMBT202C002
        CS=P,AC=L,DS=D,TO=Y,KA=Y
        PI=80
        NO=         53
        ROWID=0x2b0000001d00000000002200
        KEY =03 30 30 30 30 31 32 33 34 35 36 37 38 35 00 00   *.0000123456785..*
             00                                                *.              *
        DATA=57 57 57 58 58 59 59 59 5a 5a 30 30 30 31 32 32   *WWWXXYYYZZ000122*
             33 33 33 34 35                                    *33345          *
11       7968 -1251788992      4 EXSD    1        0 -0000 2019/02/12 16:34:42.994 2019/02/12 16:34:42.994     *                *      0      0
  *DML* (*4V ) HNDLNO:          1 FNCCD:E REQCD:FNSH DBNAME:FMBT2
        REQNUM:1
        IV=  512
        RECNAME:
11       7968 -1251788992      5 EXSD    1        0 -0000 2019/02/12 16:34:42.994 2019/02/12 16:34:42.994     *                *      0      0
  *DML* (*4V ) HNDLNO:          2 FNCCD:S REQCD:STRT DBNAME:MAM6
        REQNUM:1
        IV=  512,AM=U,EQ=E                                                                                  (中略)                (中略)
        RECNAME:
11       7968 -1251788992      6 EXSD    1        0 -0000 2019/02/12 16:34:42.994 2019/02/12 16:34:42.994     *                *      0      0
  *DML* (*4V ) HNDLNO:          2 FNCCD:A REQCD:FTCH DBNAME:MAM6
        REQNUM:1
        IV=  512,AM=U,EQ=E
        RECNAME:MAM601
        CS=P,AC=F,DS=E,TO=Y,CF=*FDN,CD=Y,KA=Y
        2U=02
        NO=         5
        ROWID=0x2f0000001a00000000000500
        COND=55 4b 45 59 20 20 20 20 45 51 5f                  *UKEY    EQ_     *
             31 32 33 34 35 36                                 *123456          *
        ELM =2a 4e 46 44 03 00 00 00 05 00 00 00 00 00 00 00   **NFD............*
             07 00 00 00 00 00 00 00 0d 00 00 00 00 00 00 00   *................*
             2e                                                *                *
        KEY =31 32 33 34 35 36                                 *123456          *
        DATA=42 42 42 44 44 44 4a 4a 4a 4a 4a                  *BBBDDDJJJJJ     *
11       7968 -1251788992      7 EXSD    1        0 -0000 2019/02/12 16:34:42.994 2019/02/12 16:34:42.995     *                *      0      0
  *DML* (*4V ) HNDLNO:          2 FNCCD:A REQCD:MODF DBNAME:MAM6
        REQNUM:1
        IV=  512,AM=U,EQ=E
        RECNAME:MAM601
        DS=E
        ROWID=0x2f0000001a00000000000500
        DATA=41 42 43 44 45 46 47 48 49 4a 4b                  *ABCDEFGHIJK     *
11       7968 -1251788992      8 EXSD    1        0 -0000 2019/02/12 16:34:42.995 2019/02/12 16:34:42.995     *                *      0      0
  *DML* (*4V ) HNDLNO:          2 FNCCD:E REQCD:FNSH DBNAME:MAM6
        REQNUM:1
        IV=  512
        RECNAME:
```

[説明]

SQL トレース情報に出力される各項目については，マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース情報の見方」を参照してください。

ただし，上記の出力例で，枠で囲まれた部分の項目については，以下で説明します。サブページ分割をしている場合は，ページをサブページに読み替えてください。

出力内容	説明
UAPEXERLOGDMLSZ	クライアント環境定義の PDUAPEXERLOGDMLSZ オペランドの指定値
UAPEXERLOGDMLDATA	クライアント環境定義の PDUAPEXERLOGDMLDATA オペランドの指定値
SDBUAPFILE	クライアント環境定義の PDSDBUAPFILE オペランドの指定値
SDBUAPDIR	クライアント環境定義の PDSDBUAPDIR オペランドの指定値
SDBPRMTRC	クライアント環境定義の PDSDBPRMTRC オペランドの指定値
SDBTRCCNDSIZE	クライアント環境定義の PDSDBTRCCNDSIZE オペランドの指定値
SDBTRCELMSIZE	クライアント環境定義の PDSDBTRCELMSIZE オペランドの指定値

出力内容	説明
SDBTRCKEYSIZE	クライアント環境定義の PDSDBTRCKEYSIZE オペランドの指定値
SDBTRCDATASIZE	クライアント環境定義の PDSDBTRCDATASIZE オペランドの指定値
OPCODE	オペレーションコード SDB データベースを操作する API または DML の場合，EXSD が表示されます。また，TP1/FSP および HiRDB/SD データベースアクセスユティリティ（pdsdbexe）が内部的に SDB データベース定義情報を取得する場合にも EXSD が表示されます。 EXSD 以外のオペレーションコードについては，マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース情報の見方」を参照してください。
DML	ヘッダ情報
(*4V△) (*SD△)	要求元の識別記号 括弧で囲まれた 4 バイトで表示されます（△は半角空白を示します）。X'00'が含まれる場合は，スペースに置換されます。
HNDLNO	ハンドル番号
FNCCD	機能コード 1 バイトで表示されます。X'00'が含まれる場合は，スペースに置換されます。
REQCD	要求コード 4 バイトで表示されます。X'00'が含まれる場合は，スペースに置換されます。 STRT：個別開始 FNSH：個別終了 FTCH：レコードの検索（FETCH），または位置指示子の位置づけ（FIND） FTCA：複数レコードの検索（FETCHDB ALL） STOR：レコードの格納（STORE） MODF：レコードの更新（MODIFY） ERAS：レコードの削除（ERASE） CLAR：レコードの一括削除 LCAP：構成情報取得 DINF：SDB データベース定義情報取得
DBNAME	SDB データベース名 SDB データベース名が，指定したサイズ分（最大 30 バイト）表示されます。 SDB データベース名に X'00'が含まれる場合は，スペースに置換されます。
REQNUM	同時要求の個数
IV	システムが使用
AM	アクセスモード 1 バイトで表示されます。指定がないときは表示されません。また，SDB データベース種別が SD FMB の場合，DML でエラーが発生したときにも表示されません。 R：参照モード U：更新モード
EQ	排他オプション

出力内容	説明
	1 バイトで表示されます。指定がないときは表示されません。また，SDB データベース種別が SD FMB の場合，DML でエラーが発生したときにも表示されません。 S：共用モード E：排他モード O：占有モード N：無排他モード
AD	排他自動解除オプション 1 バイトで表示されます。指定がないときは表示されません。また，SDB データベース種別が SD FMB の場合，DML でエラーが発生したときにも表示されません。 Y：排他自動解除機能を使用する
RA	RD エリア指定種別 1 バイトで表示されます。指定がないときは表示されません。また，SDB データベース種別が SD FMB の場合，DML でエラーが発生したときにも表示されません。 Y：検索対象とする RD エリアを指定する
RV	RD エリア指定有効オプション 1 バイトで表示されます。指定がない場合は表示されません。 Y：基点条件指定がある場合でも，RD エリア指定を有効とする
LD	システムが使用
RE	レコード格納用 RD エリアの排他制御モード 2 バイトで表示されます。指定がないときは表示されません。また，SDB データベース種別が SD FMB の場合，DML でエラーが発生したときにも表示されません。 SR：意図共用モード PR：共用モード SU：意図排他モード PU：共用意図排他モード EX：排他モード
RECNAME	レコード型名 レコード型名が，指定したサイズ分（最大 30 バイト）表示されます。 レコード型名に X'00'が含まれる場合は，スペースに置換されます。なお，HiRDB サーバで実行された要求分だけ出力されます。
CS	位置指示子種別 1 バイトで表示されます。指定がない場合は表示されません。 R：レコード位置指示子 P：親子集合位置指示子
AC	指示コード 1 バイトで表示されます。指定がない場合は表示されません。 • FETCH または FIND の場合 F：FIRST ポインタを使用する検索 L：LAST ポインタを使用する検索 U：USER ポインタを使用する検索

出力内容	説明
	N：NEXT ポインタを使用する検索 P：PRIOR ポインタを使用する検索 R：ROWID 指定検索 W：親子集合型の位置指示子の親レコード検索 • ERASE の場合 O：最下位レベルのレコード型について全レコード削除 S：削除目的のレコードとその下位レベルのレコード削除 • FETCHDB ALL の場合 F：検索範囲の最小キーのルートレコードから検索を開始 N：前回検索したレコードの次のレコードから検索を開始 S：前回の検索したファミリの検索状態に関係なく，次のルートレコードから検索を開始
SC	検索コード 1 バイトで表示されます。指定がない場合は表示されません。 N：USER ポインタから NEXT 方向への検索 P：USER ポインタから PRIOR 方向への検索
PO	ポインタオプション 1 バイトで表示されます。指定がない場合は表示されません。 U：USER ポインタにする C：USER ポインタを解除する N：USER ポインタにしない
DS	データ格納形式オプション 1 バイトで表示されます。指定がない場合は表示されません。 D：ユーザキー以外のキーの構成要素を除いて，データ格納エリアに格納する E：構成要素指定エリアで指定した構成要素だけをデータ格納する K：キーの構成要素を除かないでデータ格納エリアに格納する N：データ格納エリアへの格納をしない
DI	システムが使用
RC	システムが使用
TO	レコードの終端を検知したかどうかの判定 1 バイトで表示されます。指定がない場合は表示されません。 Y：レコードの終端を検知したかどうかの判定を行う
PS	ページ切り替えオプション 1 バイトで表示されます。指定がない場合は表示されません。 C：レコード格納時にページを切り替え，未使用ページを確保する O：レコード格納時にページを切り替え，未使用ページを確保する（確保したページは格納したレコードで占有する）
PF	PCTFREE 有効化オプション 1 バイトで表示されます。指定がない場合は表示されません。

出力内容	説明
	Y：レコード格納時に，SDB 格納データベース定義で指定した PCTFREE のページ内未使用領域比率を有効にする N：レコード格納時に，SDB 格納データベース定義で指定した PCTFREE のページ内未使用領域比率を有効にしない
2S	二次インデクスの使用可否 1 バイトで表示されます。指定がない場合は表示されません。 Y：二次インデクスが使用できる場合でもポインタ検索を行う
CF	条件式形式オプション 4 バイトで表示されます。指定がない場合は表示されません。 *FDN：条件式の指定あり *EFN：条件式の指定あり
CK	キーの条件の条件ブロックエリアの指定 1 バイトで表示されます。指定がない場合は表示されません。 Y：キーの条件の条件ブロックエリアの指定あり
CD	キー以外の条件の条件ブロックエリアの指定 1 バイトで表示されます。指定がない場合は表示されません。 Y：キー以外の条件の条件ブロックエリアの指定あり
KA	キー報告エリアの指定 1 バイトで表示されます。指定がない場合は表示されません。 Y：キー報告エリアの指定あり
FU	FOR UPDATE 1 バイトで表示されます。指定がない場合は表示されません。 Y：FOR UPDATE の指定あり
2U	二次インデクス使用フラグ 2 バイトで表示されます。二次インデクスを使用していない場合は表示されません。 01：一連番号用の二次インデクスを使用した 02：ユーザキー用の二次インデクスを使用した 二次インデクス使用フラグは，4V FMB または 4V AFM の SDB データベースの使用時に利用できる情報です。
PI	レコードポインタ情報 2 バイトで表示されます。下記に該当しない場合は表示されません。 c0：LAST ポインタおよび FIRST ポインタの示すレコードを検索または格納した 80：LAST ポインタの示すレコードを検索または格納した 40：FIRST ポインタの示すレコードを検索または格納した なお，ここでいう検索には，SD FMB の SDB データベースの子レコードのレコード型内の検索を含みません。
TM	終端検知情報 1 バイトで表示されます。下記に該当しない場合は表示されません。 Y：終端を検知した

出力内容	説明
FI	複数レコードの検索時の検索状態 2 バイトで表示されます。 01：ファミリのすべてのレコードを検索した 02：ファミリに検索していないレコードがある
NO	検索または格納したレコードの一連番号 10 バイトで表示されます。
PP	事前割り当てページ数 レコードの格納で指定した事前割り当てページ数，またはルートレコードの検索時に検索したルートレコードが属するファミリの事前割り当てページ数が 10 バイトで表示されます。
UN	USER ポインタのレコードの一連番号 USER ポインタが定義されたレコードの最初の検索時または USER ポインタの操作時に，USER ポインタのレコードの一連番号が 10 バイトで表示されます。
ROWID	検索，格納，または更新したレコードの ROWID 26 バイトで表示されます。
CONK	キーの条件の条件ブロックと値ブロックの内容（ダンプ形式）※2
COND	キー以外の条件の条件ブロックと値ブロックの内容（ダンプ形式）※2
ELM	構成要素指定エリアの内容（ダンプ形式）※2
KEY	キー報告エリア（キー部）の内容（ダンプ形式）※2
DATA	データ格納エリアの内容（ダンプ形式）※2
REQ+:STRT/FNSH※1	個別開始/終了一括要求オプション指定時に表示されます。
REQ+:STRT(0)※1	個別開始実行要求オプション指定時に表示されます。 次に示す下線部分には，SDB データベースを操作する API を実行する際に入力されたハンドル番号が表示されます。 （例） REQ+:STRT(0) REQ+:STRT(1) また，SD FMB に対する要求で，システムが個別開始を自動的に実行した場合も，ハンドル番号 0 が表示されます。
PREPROCESS FILE	プリプロセスした UAP ソースファイル名 オペレーションコードが EXSD の場合，UAP をプリプロセスしたときの HiRDB/SD のバージョンによって出力内容が異なります。 10-03-B より前の場合 アスタリスク（*） 10-03-B 以降の場合 プリプロセスファイル名

出力内容	説明
	pdsdbcbl コマンドに指定した UAP ソースファイルのパス名から，ディレクトリ名を除いたファイル名です（拡張子を含みます）。ファイル名が 30 バイトを超えるときは，ファイル名の先頭から 30 バイトが表示されます。 オペレーションコードが EXSD 以外の場合は，マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース情報の見方」を参照してください。
PREPROCESS TIME	プリプロセス時間 ポストソースファイルを特定するために，システムで使用する情報が表示されます。オペレーションコードが EXSD の場合，UAP をプリプロセスしたときの HiRDB/SD のバージョンによって出力内容が異なります。 10-03-B より前の場合 アスタリスク（*） 10-03-B 以降の場合 システムで使用する情報 オペレーションコードが EXSD 以外の場合は，マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース情報の見方」を参照してください。

注※1

個別開始/終了一括要求オプションと個別開始実行要求オプションを同時に指定した場合は，次のように並べて表示されます。

```
REQNUM:2   REQ+:STRT/FNSH
           REQ+:STRT(0)
```

注※2

ASCII コードの範囲外の文字はドット（.）で出力されます。

同じデータが 2 行以上続く場合，"--- SAME x LINES ---"が出力されます（x は行数）。

(2) パラメタトレース

SDB データベースを操作する API または DML の実行時に，パラメタトレースは出力されません。

16.5.3 クライアントエラーログ機能

HiRDB クライアントと HiRDB サーバ間の通信処理中，または X/Open で規定した XA インタフェースでエラーが発生した場合，エラー情報をクライアントエラーログとしてクライアントエラーログファイルに取得します。

SQL 文，SDB データベースを操作する API，または DML の実行時にエラーが発生した場合，クライアントエラーログ情報を参照すると，エラーの原因となる情報を特定できます。

クライアントエラーログファイルは，取得した情報で満杯になると，最も古い情報から順次新しい情報に書き替えられます。

詳細については，マニュアル「HiRDB UAP 開発ガイド」の「クライアントエラーログ機能」を参照してください。

(1)　クライアントエラーログ情報の見方

クライアントエラーログ情報の見方については，マニュアル「HiRDB UAP 開発ガイド」の「クライアントエラーログ情報の見方」を参照してください。なお，「SQL 文」は，「SDB データベースを操作する API」または「DML」に読み替えてください。「SQL 実行」は，「SDB データベースを操作する API の実行」または「DML」に読み替えてください。

(2)　クライアントエラーログへのメッセージ出力抑止

クライアントエラーログへのメッセージ出力抑止については，マニュアル「HiRDB UAP 開発ガイド」のクライアント環境定義の PDERRSKIPCODE オペランドの説明を参照してください。

16.5.4　拡張 SQL エラー情報出力機能

クライアントエラーログ機能の情報に，SQL 文，およびパラメタ情報を加えて SQL エラーレポートファイルに取得します。

SDB データベースを操作するための API または DML の実行時にエラーが発生した場合，SQL エラー情報を参照すると，エラーの原因となる情報を特定できます。

SQL エラーレポートファイルは，取得した情報で満杯になると，もう一方のファイルに出力先を切り替えます。切り替わったファイルでも，これを繰り返しながら 2 つのファイルを交互に使用します（切り替え先の古い内容は削除されます）。

詳細については，マニュアル「HiRDB UAP 開発ガイド」の「拡張 SQL エラー情報出力機能」を参照してください。

(1)　SQL エラー情報の見方

SDB データベースを操作する API または DML の実行時に出力される SQL エラー情報の出力例を図 16-5 と図 16-6 に示します。

図 16-5　SQL エラー情報の出力例（4V FMB の SDB データベースの場合）

```
>> 8784 -806521600  1140     1 2011/11/02 16:33:56 SQLCODE:-61502     3(163356096) sdifvacs.c  : 951 EXSD KFPA61502-E Set
cursor is null
> 20123 -506346240  1063     0 2011/11/08 17:13:02 KFPZ03000-I Error information, type=CONNECT STATUS, inf=CLT=09-00(Aug
25 2011):PC SVR=09-00   US:PC LIBTYPE=UNIX_XA_64K
>>20123 -506346240  1063     1 2011/11/08 17:13:02 SQLCODE:-66209     3(171302313) sdivconc.c  : 233 EXSD KFPA66209-E
Specified DBKEY is not specified for KEYDEF of SDB storage database definition
> 21233 -1949309696 11983     0 2011/11/08 17:15:27 KFPZ03000-I Error information, type=CONNECT STATUS, inf=CLT=09-00(Aug
25 2011):PC SVR=09-00   US:PC LIBTYPE=UNIX_XA_64K
>>21233 -1949309696 11983     1 2011/11/08 17:15:27 SQLCODE:-66209     3(171527937) sdivconc.c  : 233 EXSD KFPA66209-E
Specified DBKEY is not specified for KEYDEF of SDB storage database definition
UAP Unknown,PT1
SVR sdb-bs2000-1,50305,testfes1,LINX64
SQLINF 657312,1,2,2,0,0000,-0000,17:15:27.250117,17:15:27.250194,0.000077
SQL *
DMLINF
--------------------------------------------------------------------------------
*NO:4,START TIME:2014/04/18 20:41:48,END TIME:2014/04/18 20:41:49
HNDCOM 64
      DATA=2a 34 56 20 00 00 40 00 80 f2 13 02 00 00 00 00   **4V ..@.........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 2 LINES ---
4VCNTROL 128
      DATA=2a 34 56 43 80 00 00 00 01 00 00 00 00 00 00 00   **4VC............*
           10 f3 13 02 00 00 00 00 04 00 46 4d 42 31 00 00   *..........FMB1..*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 3 LINES ---
4VREQUEST 512
      DATA=2a 34 56 52 00 02 00 00 00 00 00 00 00 00 00 00   **4VR............*
           41 00 00 00 4d 4f 44 46 55 45 00 00 00 00 00 00   *A...MODFUE......*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 44   *...............D*
           00 00 00 00 59 00 00 00 00 00 00 00 00 00 00 00   *....Y...........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 6 LINES ---
           00 00 00 00 00 00 00 00 30 10 55 45 ff 7f 00 00   *........0.UE....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 4 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 70 0e 00 00   *............p...*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 12 LINES ---
KEYCONDITION 0
DATACONDITION 0
KEYVALUE 0
DATAVALUE 0
INPUTDATA 0
STORDATA 3696
      DATA=46 4d 42 31 30 30 30 5f 41 2d 46 4d 42 30 31 5f   *FMB1000_A-FMB01_*
           42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *B...............*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
ELMAREA 0
KEYREPORT 0
RESULTSAVE 0
RDAREANAME 0
4VREQUEST 512
      DATA=2a 34 56 52 00 02 00 00 00 00 00 00 00 00 00 00   **4VR............*
           41 00 00 00 4d 4f 44 46 55 45 00 00 00 00 00 00   *A...MODFUE......*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 44   *...............D*
           00 00 00 00 59 00 00 00 00 00 00 00 00 00 00 00   *....Y...........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 6 LINES ---
           00 00 00 00 00 00 00 00 30 10 55 45 ff 7f 00 00   *........0.UE....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 4 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 70 0e 00 00   *............p...*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 12 LINES ---
```

[1]

（続く）

（続き）

```
KEYCONDITION 0
DATACONDITION 0
KEYVALUE 0
DATAVALUE 0
INPUTDATA 0
STORDATA 3696
      DATA=46 4d 42 31 30 30 30 5f 41 2d 46 4d 42 30 31 5f   *FMB1000_A-FMB01_*
           42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *B..............*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
ELMAREA 0
KEYREPORT 0
RESULTSAVE 0
RDAREANAME 0
--------------------------------------------------------------------------------
*NO:3,START TIME:2014/04/18 20:41:46,END TIME:2014/04/18 20:41:47
HNDCOM 64
      DATA=2a 34 56 20 00 00 40 00 80 d2 88 00 00 00 00 00   **4V ..@.........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 2 LINES ---
4VCNTROL 128
      DATA=2a 34 56 43 80 00 00 00 01 00 00 00 00 00 00 00   **4VC...........*
           10 d3 88 00 00 00 00 00 04 00 46 4d 42 31 00 00   *..........FMB1..*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 3 LINES ---
4VREQUEST 512
      DATA=2a 34 56 52 00 02 00 00 00 00 00 00 00 00 00 00   **4VR...........*
           41 00 00 00 46 54 43 48 55 45 07 00 46 4d 42 31   *A...FTCHUE..FMB1*
           30 30 30 00 00 00 00 00 00 00 00 00 00 00 00 00   *000.............*
           00 00 00 00 00 00 00 00 00 00 52 46 00 00 4e 44   *..........RF..ND*
           00 52 00 00 59 43 59 00 59 00 00 00 00 00 00 00   *.R..YCY.Y.......*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           2a 46 44 4e 00 00 00 00 00 00 00 00 00 00 00 00   **FDN...........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 2 LINES ---
           00 00 00 00 00 00 00 00 50 d5 88 00 00 00 00 00   *........P.......*
           00 00 00 00 00 00 00 00 5c 17 45 ff 7f 00 00      *........¥.E....*
           00 00 00 00 00 00 00 e0 cf 17 45 ff 7f 00 00      *..........E....*
           00 00 00 00 00 00 00 00 90 cf 17 45 ff 7f 00 00   *..........E....*
           b0 d5 88 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 0b 00 00 00   *................*
           00 00 00 00 71 0e 00 00 00 00 00 00 1e 00 00 00   *....q...........*
           00 00 00 00 21 00 00 00 28 01 00 00 00 00 00 00   *....!...(.......*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 11 LINES ---
KEYCONDITION 0
DATACONDITION 11
      DATA=55 53 45 52 44 41 30 20 45 51 5f                  *USERDA0 EQ_     *
KEYVALUE 0
DATAVALUE 3697
      DATA=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 3 LINES ---
INPUTDATA 0
STORDATA 30
      DATA=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00         *..............  *
ELMAREA 0
KEYREPORT 33
      DATA=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
           00                                                *.              *
RESULTSAVE 296
      DATA=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 17 LINES ---
           00 00 00 00 00 00 00 00                           *........       *
RDAREANAME 0
--------------------------------------------------------------------------------
```

[1]

[2]

（続く）

（続き）

```
*NO:2,START TIME:2014/04/18 20:41:44,END TIME:2014/04/18 20:41:45
HNDCOM 64
      DATA=2a 34 56 20 00 00 40 00 80 12 06 01 00 00 00 00   **4V ..@.........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 2 LINES ---
4VCNTROL 128
      DATA=2a 34 56 43 80 00 00 00 00 00 00 00 00 00 00 00   **4VC............*
           10 13 06 01 00 00 00 00 04 00 46 4d 42 31 00 00   *..........FMB1..*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 3 LINES ---
4VREQUEST 512
      DATA=2a 34 56 52 00 02 00 00 00 00 00 00 00 00 00 00   **4VR............*
           53 00 00 00 53 54 52 54 55 45 00 00 00 00 00 00   *S...STRTUE......*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
           00 00 00 00 00 00 00 00 00 59 00 00 00 59 00 00   *.........Y...Y..*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 8 LINES ---
           00 00 00 00 00 00 00 00 40 8e 2d 94 ff 7f 00 00   *........@.-.....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 3 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 84 0e 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 11 LINES ---
KEYCONDITION 0
DATACONDITION 0
KEYVALUE 0
DATAVALUE 0
INPUTDATA 0
STORDATA 0
ELMAREA 0
KEYREPORT 0
RESULTSAVE 0
RDAREANAME 0
```

[2]

図 16-6　SQL エラー情報の出力例（SD FMB の SDB データベースの場合）

```
> 41939     0 16597     0 2018/01/19 10:08:32 KFPZ03000-I Error information, type=CONNECT STATUS, inf=CLT=09-66(Jan 17
2018):PC SVR=09-66   US:PC:10.197.12.114:35031:testfes2 LIBTYPE=UNIX_64K
>>41939     0 16597     1 2018/01/19 10:08:32 SQLCODE:-66406     4(100832165) sdismdf.c   : 396 EXSD KFPA66406-E Unable to
modify record fetched without specifying "FOR UPDATE" operand
UAP Unknown,PT1
SVR sdb-bs2500-6,21902,testfes2,LINX64
SQLINF 2756512,12385,2,2,1,0000,-0000,10:08:32.148608,10:08:32.148679,0.000071
SQL *
DMLINF
--------------------------------------------------------------------------------
*NO:4,START TIME:2018/01/19 10:08:32,END TIME:2018/01/19 10:08:32
HNDCOM 64
      DATA=2a 53 44 20 00 55 40 00 a0 09 62 00 00 00 00 00   **SD .U@...b.....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 2 LINES ---
SDCNTROL 128
      DATA=2a 53 44 43 80 00 00 00 01 00 00 00 00 00 00 00   **SDC............*
           60 0a 62 00 00 00 00 00 04 00 53 44 42 31 20 20   *`.b.......SDB1  *
           20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20   *                *
           20 20 20 20 20 20 20 20 01 00 00 00 00 00 00 00   *        ........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00   *................*
           00 00 00 00 00 00 00 00 20 08 62 00 00 00 00 00   *........ .b.....*
SDCNTROL2 128
      DATA=80 00 00 00 59 00 00 00 00 00 00 00 00 00 00 00   *....Y...........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 6 LINES ---
SDREQUEST 512
      DATA=2a 53 44 52 00 02 00 00 00 00 00 00 00 00 00 00   **SDR............*
           41 00 00 00 4d 4f 44 46 00 00 06 00 53 44 42 31   *A...MODF....SDB1*
           30 31 20 20 20 20 20 20 20 20 20 20 20 20 20 20   *01              *
           20 20 20 20 20 20 20 20 20 20 00 00 00 00 00 44   *          .....D*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 7 LINES ---
           00 00 00 00 00 00 00 00 bd fc 61 00 00 00 00 00   *..........a.....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 4 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 15 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 11 LINES ---
           00 00 00 00 00 00 00 00 a0 08 62 00 00 00 00 00   *..........b.....*
KEYCONDITION 0
DATACONDITION 0
KEYVALUE 0
DATAVALUE 0
INPUTDATA 0


STORDATA 21
      DATA=4d 4f 44 49 46 59 5f 53 44 42 31 30 31 5f 30 30   *MODIFY_SDB101_00*
           31 20 20 20 20                                    *1               *
ELMAREA 0
KEYREPORT 0
RESULTSAVE 0
RDAREANAME 0
SDREQUEST2 256
      DATA=00 01 00 00 53 00 00 00 00 00 00 00 00 00 00 00   *....S...........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 14 LINES ---
```

[1]

（続く）

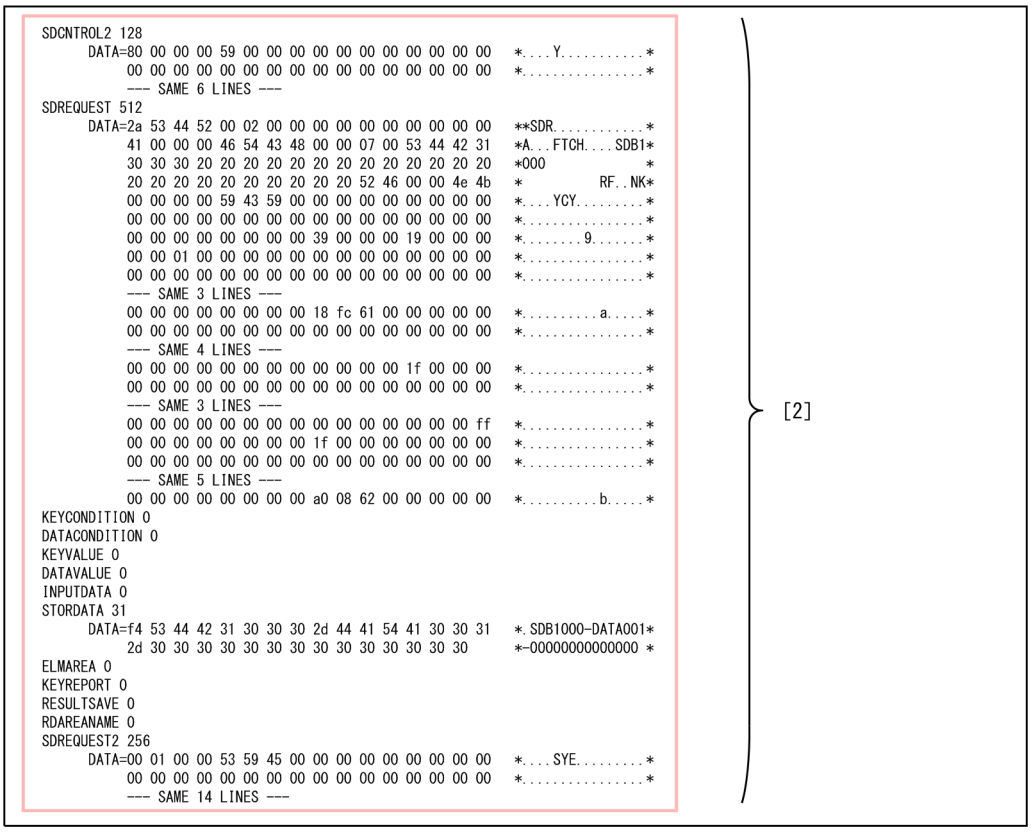
（続き）

```
--------------------------------------------------------------------------------
*NO:3,START TIME:2018/01/19 10:08:32,END TIME:2018/01/19 10:08:32
HNDCOM 64
      DATA=2a 53 44 20 00 55 40 00 a0 09 62 00 00 00 00 00   **SD .U@...b.....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 2 LINES ---
SDCNTROL 128
      DATA=2a 53 44 43 80 00 00 00 01 00 00 00 00 00 00 00   **SDC...........*
           60 0a 62 00 00 00 00 00 04 00 53 44 42 31 20 20   *`.b.......SDB1  *
           20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20   *                *
           20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00   *        ........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
           30 39 2d 36 36 20 20 20 00 00 00 04 00 00 00 00   *09-66   ........*
           00 00 00 00 00 00 00 00 20 08 62 00 00 00 00 00   *........ .b.....*
SDCNTROL2 128
      DATA=80 00 00 00 59 00 00 00 00 00 00 00 00 00 00 00   *....Y...........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 6 LINES ---
SDREQUEST 512
      DATA=2a 53 44 52 00 02 00 00 00 00 00 00 00 00 00 00   **SDR...........*
           41 00 00 00 46 54 43 48 00 00 06 00 53 44 42 31   *A...FTCH....SDB1*
           30 31 20 20 20 20 20 20 20 20 20 20 20 20 20 20   *01              *
           20 20 20 20 20 20 20 20 20 20 50 4c 00 00 4e 4b   *          PL..NK*
           00 00 00 00 59 43 59 00 00 00 00 00 00 00 00 00   *....YCY.........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 02 00 00 00 39 00 00 00 19 00 00 00   *........9.......*
           00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 3 LINES ---
           00 00 00 00 00 00 00 00 78 fc 61 00 00 00 00 00   *........x.a.....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 4 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 1a 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 3 LINES ---
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ff   *................*
           00 00 00 00 00 00 00 00 1a 00 00 00 80 00 00 00   *................*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 5 LINES ---
           00 00 00 00 00 00 00 00 a0 08 62 00 00 00 00 00   *..........b.....*
KEYCONDITION 0
DATACONDITION 0
KEYVALUE 0
DATAVALUE 0
INPUTDATA 0
STORDATA 26
      DATA=f4 02 00 00 00 53 44 42 31 30 31 2d 44 41 54 41   *.....SDB101-DATA*
           30 30 31 2d 30 30 30 30 30 32                     *001-000002      *
ELMAREA 0
KEYREPORT 0
RESULTSAVE 0
RDAREANAME 0
SDREQUEST2 256
      DATA=00 01 00 00 53 00 45 00 00 00 00 00 00 00 00 00   *....S.E.........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 14 LINES ---
--------------------------------------------------------------------------------
*NO:2,START TIME:2018/01/19 10:08:32,END TIME:2018/01/19 10:08:32
HNDCOM 64
      DATA=2a 53 44 20 00 55 40 00 a0 09 62 00 00 00 00 00   **SD .U@...b.....*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 2 LINES ---
SDCNTROL 128
      DATA=2a 53 44 43 80 00 00 00 01 00 00 00 00 00 00 00   **SDC...........*
           60 0a 62 00 00 00 00 00 04 00 53 44 42 31 20 20   *`.b.......SDB1  *
           20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20   *                *
           20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00   *        ........*
           00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   *................*
           --- SAME 1 LINES ---
           30 39 2d 36 36 20 20 20 00 00 00 04 00 00 00 00   *09-66   ........*
           00 00 00 00 00 00 00 00 20 08 62 00 00 00 00 00   *........ .b.....*
```

[2]

（続く）

（続き）



[説明]

SQL エラー情報に出力される各項目については，マニュアル「HiRDB UAP 開発ガイド」の「SQL エラー情報の見方」を参照してください。このとき，「SQL 文」は，「SDB データベースを操作する API」または「DML」に読み替えてください。

ただし，図 16-5 と図 16-6 で，枠で囲まれた部分の項目については，以下で説明します。

出力内容	説明
UAP	UAP 情報
SVR	サーバ情報 • サーバプロセスが動作しているホスト名（出力文字数は最大 30 バイト） • サーバプロセスの通信ポート番号（出力文字数は最大 5 バイト） • フロントエンドサーバのサーバ名（出力文字数は最大 8 バイト） • クライアントライブラリが対応しているプラットフォーム（出力文字数は最大 6 バイト）
SQLINF	SQL 情報 • SQL 最適化オプションの値（10 進数形式）（出力文字数は最大 10 バイト） • SQL 拡張最適化オプションの値（10 進数形式）（出力文字数は最大 10 バイト） • データ保証レベルの値（出力文字数は最大 10 バイト）
SQL	SDB データベースを操作する API または DML の実行時にはアスタリスク（*）が表示されます。
DMLINF	各エリアの情報

出力内容	説明
	各エリアのサイズが表示されます。エリアサイズが 0 以下の場合は，エリア名とエリアサイズだけが表示されます。 各エリアの内容がダンプ形式で表示されます（1 行に 16 バイト分が並びます）。同様のデータが 2 行以上続く場合は，「--- SAME n LINES ---"（n は行数）」が表示されます。ASCII 文字はアスタリスク（*）で囲まれて表示されます。
*NO	SQL カウンタ コネクト要求を 1 とし，SQL 文，SDB データベースを操作する API，または DML の実行 1 回ごとに 1 カウントアップした値が出力されます。
START TIME	DML 要求開始時刻 HiRDB クライアントからフロントエンドサーバに，SDB データベースを操作する API または DML の実行要求を送信した時刻が，YYYY/MM/DD hh:mm:ss の形式で出力されます。 YYYY：年 MM：月 DD：日 hh：時 mm：分 ss：秒
END TIME	DML 要求終了時刻 フロントエンドサーバから HiRDB クライアントに，SDB データベースを操作する API または DML の実行結果を返却した時刻が，YYYY/MM/DD hh:mm:ss の形式で出力されます。 YYYY：年 MM：月 DD：日 hh：時 mm：分 ss：秒
HNDCOM	共通エリアの内容
4VCNTROL	4V 固有エリア（管理部）の内容
SDCNTROL	SD 固有エリア（管理部）の内容
4VREQUEST※	4V 固有エリア（要求部）の内容
SDREQUEST	SD 固有エリア（要求部）の内容
KEYCONDITION※	キーの条件の条件ブロックエリアの内容
DATACONDITION※	キー以外の条件の条件ブロックエリアの内容
KEYVALUE※	キーの条件の値ブロックエリアの内容
DATAVALUE※	キー以外の条件の値ブロックエリアの内容 この項目で出力されるエリアダンプは，最大 64 バイトで打ち切られます。エリアサイズには，元々の値が出力されます。

出力内容	説明
INPUTDATA※	データ指定エリアの内容
STORDATA※	データ格納エリアの内容 この項目は，クライアント環境定義の PDUAPEXERLOGDMLDATA オペランドに YES を指定した場合に出力されます。 この項目で出力されるエリアダンプは，最大 64 バイトで打ち切られます。エリアサイズには，元々の値が出力されます。
ELMAREA※	構成要素指定エリアの内容
KEYREPORT※	キー報告エリアの内容
RESULTSAVE※	処理結果退避エリアの内容
RDAREANAME※	RD エリア名称格納エリアの内容
SDREQUEST2	SD 固有エリア（要求部 2）の内容

注※

一度の要求で複数のレコードに対してアクセスする要求の場合は，すべてのアクセス分が表示されます。複数のレコードに対してアクセスする要求については，マニュアル「TP1/Financial Service Platform プログラム作成の手引」の「SDB ハンドラ制御（ee_sdh_～）（C 言語）」の「ee_sdh_accs」を参照してください。

なお，SD FMB の SDB データベースの場合，複数のレコードに対してアクセスする要求はできません。

DMLINF 以降の出力内容については，エラー発生時の要求およびクライアント環境定義の PDUAPEXERLOGDMLSZ オペランドの指定値によって，次の情報に対応する内容が出力されます。DML 履歴用バッファについては，「(2)　DML 履歴用バッファ」を参照してください。

- SQL でのエラー発生時（PDUAPEXERLOGDMLSZ ＞ 0 の場合）

  DML 履歴用バッファに記録された SDB データベースを操作する API または DML の入力情報が出力されます。「図 16-5　SQL エラー情報の出力例（4V FMB の SDB データベースの場合）」または「図 16-6　SQL エラー情報の出力例（SD FMB の SDB データベースの場合）」の出力例の［2］が該当します。

- SQL でのエラー発生時（PDUAPEXERLOGDMLSZ=0 の場合）

  出力されません。DMLINF の文字列も出力されません。

- SDB データベースを操作する API または DML でのエラー発生時（PDUAPEXERLOGDMLSZ ＞ 0 の場合）

  エラー発生時の SDB データベースを操作する API と，DML 履歴用バッファに記録された SDB データベースを操作する API の入力情報が出力されます。

  または，エラー発生時の DML と，DML 履歴用バッファに記録された DML の入力情報が出力されます。

  「図 16-5　SQL エラー情報の出力例（4V FMB の SDB データベースの場合）」または「図 16-6　SQL エラー情報の出力例（SD FMB の SDB データベースの場合）」の出力例の［1］および［2］が該当します。

- SDB データベースを操作する API または DML でのエラー発生時（PDUAPEXERLOGDMLSZ ＝ 0 の場合）

エラー発生時の SDB データベースを操作する API または DML の入力情報が出力されます。「図 16-5　SQL エラー情報の出力例（4V FMB の SDB データベースの場合）」または「図 16-6　SQL エラー情報の出力例（SD FMB の SDB データベースの場合）」の出力例の［1］が該当します。

留意事項

クライアント環境定義の PDUAPERLOG オペランドに 0 以外の値を指定した場合，拡張 SQL エラー情報の出力が途中で中断されて「--- DATA OMITTED ---」と出力されることがあります。この場合，「(2 ＋ PDUAPEXERLOGDMLSZ オペランドの指定値）×5120」以上の値を PDUAPERLOG オペランドに指定してください。その結果，拡張 SQL エラー情報が中断されないで出力されるようになります。

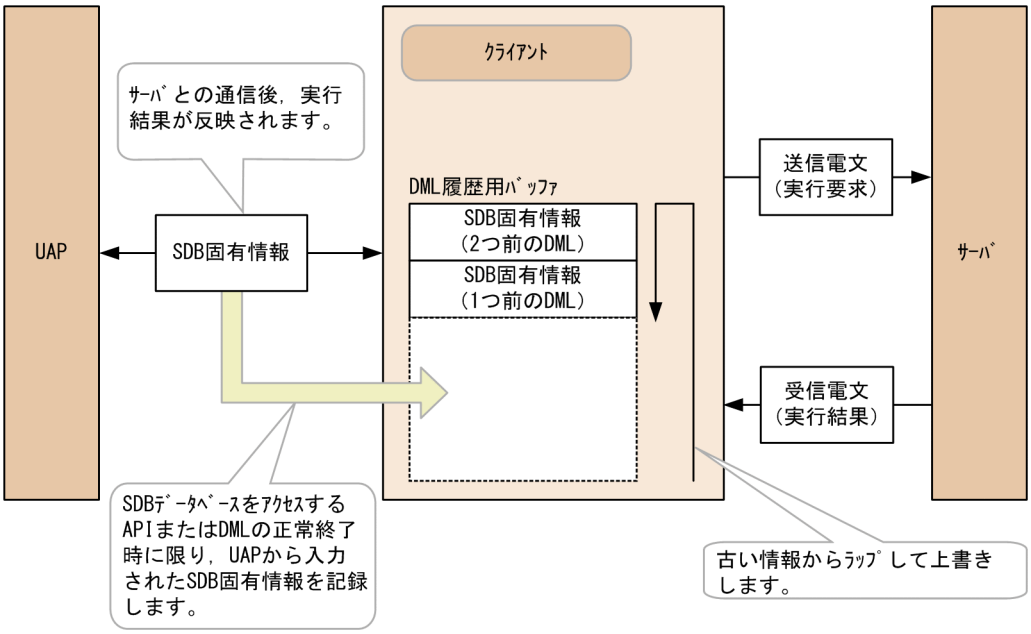
(2)　DML 履歴用バッファ

DML 履歴用バッファとは，CONNECT 文を要求してから DISCONNECT 文が要求されるまでの間に入力された情報をクライアントライブラリ内に記録するバッファのことです。

SDB データベースを操作する API または DML が正常終了してフロントエンドサーバから実行結果が返却された際に，実行結果を反映した情報をバッファ内に記録します。このとき，バッファの空きがない場合は，古い情報から上書きして記録されます。

DML 履歴用バッファの記録イメージを次の図に示します。

図 16-7　DML 履歴用バッファの記録イメージ



DML 履歴用バッファに格納された情報は，SQL 文，SDB データベースを操作する API，または DML の実行時の拡張 SQL エラー情報出力時にエリアダンプとして出力されます。「図 16-5　SQL エラー情報の出力例（4V FMB の SDB データベースの場合）」または「図 16-6　SQL エラー情報の出力例（SD FMB の SDB データベースの場合）」の出力例の［2］が該当します。

留意事項

エリアダンプの出力量が多くなるほど，拡張 SQL エラー情報の出力に掛かる時間が増加します。SQL 文，SDB データベースを操作する API，または DML でエラーが発生した際の処理時間が，要求元のタイマ監視などで問題となる場合は，次の指定値を見直してください。

- クライアント環境定義の PDUAPEXERLOGDMLSZ
  DML 履歴用バッファのサイズを小さくすると，エリアダンプの出力量が少なくなります。
- クライアント環境定義の PDUAPERLOG
  0 以外の値を指定すると，エリアダンプの出力量が指定したサイズに制限されます。

16.5.5 UAP 統計レポート機能

UAP 実行時の UAP 統計レポートを UAP 統計レポートファイルに取得します。

次に示す SDB データベースを操作するための API または DML を実行した場合，UAP 統計レポートを参照すると，アクセスパス情報および位置指示子情報を特定できます。

- レコードの検索（FETCH）
- 位置指示子の位置づけ（FIND）
- 複数レコードの検索（FETCHDB ALL）
- レコードの格納（STORE）
- レコードの更新（MODIFY）
- レコードの削除（ERASE）
- レコードの取得（GET）
- レコードの一括削除
- 構成情報取得

注意事項

次の API または DML の場合は，アクセスパス情報と位置指示子情報は出力されません。

- ユティリティが内部的に実行した API または DML
- エラーが発生した API または DML
- 終端検知（レコードの検索時に最終のレコードかどうかを判定する）のために HiRDB/SD が内部的に実行した API

UAP 統計レポートの詳細については，マニュアル「HiRDB UAP 開発ガイド」の「UAP 統計レポート機能」を参照してください。

(1) UAP 統計レポートの取得方法

UAP 統計レポートは，クライアント環境定義の PDCLTPATH，PDSQLTRACE，および PDUAPREPLVL（p または a 指定）を設定すると取得できます。

(2) UAP 統計レポートの見方

UAP 統計レポートの見方については，マニュアル「HiRDB UAP 開発ガイド」の「UAP 統計レポートの見方」を参照してください。なお，「SQL 文」は，「SDB データベースを操作する API」または「DML」に読み替えてください。

SDB データベースを操作する API または DML の実行時に出力される情報の出力形式とその説明を次に示します。

サポートしていない情報が設定されていた場合は，'(UNK)'が出力されます。

(a) アクセスパス情報

[出力形式]

```
Result of DML Optimizer :
   EXEC ID     :aa...a-bb...b-cc...c-dd...d
-------------------------------------------------------------------------------
 # DML Type    : ee...e
   Record Name : ff...f
   Set Name    : yy...y
   Bes Name    : gg...g
   RDAREA      : hh...h[iiiiiiiiii]
   Pointer Code: jj...j
   Direction   : kk...k
   Poniter Opt : BB...B
   Insert Code : ll...l
   KEYDATA     : mm...m
   Index Name  : nn...n
                     SearchCnd : oo...o [pp...p]
                     KeyCnd    : qq...q
   RowCnd      : {AA...A}
   Parameter   : (rr...r)ss...s[,(rr...r)ss...s]...
   SearchStatus: tt...t
   UnusedLen   : uu...u
   Record List : vv...v(ww...w)[,vv...v(ww...w)]...
```

SDB データベースを操作する API または DML の指定内容によっては表示されない項目があります。

[出力例]

```
Result of DML Optimizer :
   EXEC ID     : 1-1-1-1
-------------------------------------------------------------------------------
 # DML Type    : FETCH
   Record Name : FMB1000
   Bes Name    : testbesw
   RDAREA      : RDAREA74[0x00000014]
```

```
Pointer Code: F
Direction   : 0x00
KEYDATA     : 0x3030303031313131
Index Name  : FMB1IDX1
                  SearchCnd : RANGE(CS-CE) [(?(1),MIN),(MAX,MAX)]
                  KeyCnd    : FMB1000.KEYDATA2=?(2) AND FMB1000.KEYDATA1>?(1)
RowCnd      : {(FMB1000.USERDA0>?(3))}
Parameter   : (1)0x00000001, (2)0x31313131, (3)0x00000000
```

[説明]

出力内容	説明
Result of DML Optimizer	SDB データベースのアクセスパス情報が表示されます。
EXEC ID	各 ID の情報が表示されます。 それぞれ出力文字数は最大 10 バイトです。 各 ID は，トランザクション開始時に 1 から開始し，要求ごとに 1 ずつ増え，最大値 4,294,967,295 まで増えます。最大値の次の値は 0 になります。 • SDB ハンドラの要求ごとに割り振られたブロック ID 出力対象外となっている個別開始・個別終了もカウントされます。 • SDB データベース内でフロントエンドサーバからバックエンドサーバへ，SDB データベースを操作する API または DML を送信した回数を示す CALL ID • SDB ハンドラの要求の中の SDB データベースを操作する API または DML ごとに割り振られた DML ID • SDB データベース内で実行した，SDB データベースを操作する API または DML ごとに割り振られた EXEC ID
DML Type	SDB データベース内で実行したデータベース操作（出力文字数は最大 11 バイト） • FETCH：レコードの検索，または位置指示子の位置づけ※1 • STORE：レコードの格納 • ERASE：レコードの削除 • MODIFY：レコードの更新 • NULLIFY：位置指示子への空値設定 • FETCHDB ALL：複数レコードの検索 • GET：レコードの取得
Record Name※2	操作対象のレコード型名（出力文字数は最大 30 バイト） 複数レコードの検索（FETCHDB ALL）の場合はルートレコード型名が表示されます。
Set Name※2	親子集合型名（出力文字数は最大 30 バイト） DML Type が NULLIFY の場合に表示されることがあります。
Bes Name	操作対象のバックエンドサーバ名（出力文字数は最大 8 バイト）
RDAREA	次の情報が表示されます。 DML Type が NULLIFY の場合は表示されません。 • 操作対象の RD エリア名（インナレプリカ機能使用時はオリジナル RD エリア名）（出力文字数は最大 30 バイト） • 操作対象の RD エリア ID（16 進数形式）（出力文字数は最大 10 バイト）
Pointer Code	HiRDB/SD 内で実行された指示コード（出力文字数は最大 4 バイト）

出力内容	説明
	文字として表示できない場合は 16 進形式で表示されます。
Direction	HiRDB/SD 内で実行された検索コード（出力文字数は最大 4 バイト） 文字として表示できない場合は 16 進形式で表示されます。 レコードの検索時（FETCH），または位置指示子の位置づけ（FIND）のときに指示コードに U 以外が指定されている場合，0x00 が表示されます。
Pointer Opt	HiRDB/SD 内で実行したポインタオプション（出力文字数は最大 4 バイト） 文字として表示できない場合は 16 進形式で表示されます。
Insert Code	HiRDB/SD 内で実行した挿入位置検索の種別 ORDER FIRST：子レコード実現値群の先頭に格納します。 ORDER LAST：子レコード実現値群の最後に格納します。 ORDER SORTED：子レコードで指定したキー項目の値に従って，昇順または降順に格納します。
KEYDATA	KEYDATA（出力文字数は最大 512 バイト） • ルートレコードの検索（FETCH），位置指示子の位置づけ（FIND），またはレコードの取得（GET）の場合 対象となるレコードの KEYDATA（シーケンシャルインデクスキー）を 16 進形式で表示します。 • 子レコードの検索（FETCH），位置指示子の位置づけ（FIND），またはレコードの取得（GET）の場合 対象となるレコードの KEYDATA（一連番号）を 10 進形式で表示します。 • 複数レコードの検索（FETCHDB ALL）の場合 ルートレコードの KEYDATA（シーケンシャルインデクスキー）を 16 進形式で表示します。
Index Name	レコードの検索に使用したインデクス名（出力文字数は最大 30 バイト） インデクスを使用しない場合，この行は表示されません。
SearchCnd	サーチ条件の種類およびサーチ条件の絞り込み範囲（出力文字数は最大 105 バイト） インデクスを使用した検索を行わない場合，この行は表示されません。 サーチ条件については，マニュアル「HiRDB コマンドリファレンス」の「サーチ条件」を参照してください。 アクセスパス情報では，？パラメタに SDB データベース内で生成した番号が付与されます。 例：SearchCnd：AT [?(1)] 1 つのユーザの条件値を複数の条件から参照している場合，番号は等しくなります。
KeyCnd	キー条件 キー条件がない場合，この行は表示されません。 キー条件については，マニュアル「HiRDB コマンドリファレンス」の「キー条件」を参照してください。 アクセスパス情報では，？パラメタに SDB データベース内で生成した番号が付与されます。 例：MAM_CHILD1.DBNAME>?(3) 1 つのユーザの条件値を複数の条件から参照している場合，番号は等しくなります。
RowCnd	システムが使用する内部情報

出力内容	説明
Parameter	次の情報が表示されます。 サーチ条件，キー条件，ロー条件がない場合，この行は表示されません。 • ？パラメタに付与された，SDB データベース内で生成された番号（昇順）（出力文字数は最大 2 バイト） • 上記の番号に対応するサーチ条件，キー条件，ロー条件で使用する条件値（16 進数形式）（単位：バイト）
Search Status	複数レコードの検索時の検索状態 • 0x00：検索結果が NOT FOUND となった場合 • FINISH：1 つのファミリをすべて検索した場合 • HALFWAY：1 つのファミリのレコードの格納中に，データ格納エリアが満杯になった場合
Unused Len	複数レコードの検索時のデータ格納エリアの余り領域長（出力文字数は最大 5 バイト）（単位：バイト） 検索結果が NOT FOUND のときは，データ格納エリアの領域長が表示されます。
Record List	• 複数レコードの検索時に検索したレコードのレコード型名（出力文字数は最大 30 バイト） SDB ディレクトリ情報に対応するレコード型名が設定されていない場合は，*****が表示されます。 • 複数レコードの検索時に検索したレコードごとの検索レコード数（出力文字数は最大 3 バイト）

注※1

FIND 文（位置指示子の位置づけ）の SDB データベースの操作種別は'FETCH'と表示されます。

注※2

アクセスパス情報には，Record Name または Set Name のどちらかが表示されます。

(b) 位置指示子情報

位置指示子情報には，SDB データベースを操作する API または DML の実行時に使用した位置指示子について，API または DML の実行後の状態が出力されます。

SDB データベースを操作する API または DML と，出力される位置指示子情報の関係を次の表に示します。

SDB データベースを操作する API または DML		レコード位置指示子	親子集合位置指示子
FETCH	ルートレコード	○	×
	子レコード	×	○
FIND		○	×
STORE	ルートレコード	○	×
	子レコード	×	○
MODIFY		○	×
ERASE		○	×

SDB データベースを操作する API または DML	レコード位置指示子	親子集合位置指示子
NULLIFY	○※	○※
FETCHDB ALL	×	×
GET	○	×

（凡例）

○：出力されます。

×：出力されません。

注※

アクセスパス情報に Record Name が表示されている場合は，レコード位置指示子が出力されます。アクセスパス情報に Set Name が表示されている場合は，親子集合位置指示子が出力されます。

位置指示子の種別（レコード位置指示子，親子集合位置指示子）によって，出力する位置指示子情報は異なります。出力する位置指示子情報がない場合はヘッダも出力しません。

位置指示子の種別ごとの出力情報を次に示します。

- レコード位置指示子の出力情報
  - 位置指示子種別
  - レコード位置指示子に対応するレコードビュー名
  - レコード位置指示子が指す SDBKEY
  - レコード位置指示子の ERASE RETAIN 状態
  - インデクス情報

    対象のインデクスについて次の情報を出力します。順序はインデクス型番号順となります。

    ・インデクス名

    ・インデクスキー長

    ・インデクスキー値

    出力対象のインデクスを次の表に示します。

対象レコード	インデクス情報	
	4V AFM または 4V FMB の SDB データベースの場合	SD FMB の SDB データベースの場合
ルートレコード	シーケンシャルインデクス	レコードに定義されたすべてのインデクス（シーケンシャルインデクスおよび二次インデクス）
子レコード	表示なし	

- 親子集合位置指示子の出力情報
  - 位置指示子種別
  - 親子集合位置指示子の親に対応するレコードビュー名
  - 親子集合位置指示子の子に対応するレコードビュー名

- 親子集合位置指示子の親が指す SDBKEY
- 親子集合位置指示子の子が指す SDBKEY（親子集合位置指示子が複合状態となった場合は，2 個出力します）
- 親子集合位置指示子の複合状態
- インデクス情報

  対象のインデクスについて次の情報を出力します。順序はインデクス型番号順となります。

  ・インデクス名

  ・インデクスキー長

  ・インデクスキー値

  出力対象のインデクスを次の表に示します。

対象レコード	インデクス情報	
	4V AFM または 4V FMB の SDB データベースの場合	SD FMB の SDB データベースの場合
ルートレコード	－	－
子レコード	二次インデクス	二次インデクス

（凡例）

－：該当しません。

[出力形式]

```
Result of SDB Cursor  :
 # Cursor Type  : aa...a
   Record Name  : bb...b
   P Rec Name   : cc...c
   C Rec Name   : dd...d
   SDBKEY       : ee...e
   SDBKEY(P)    : ff...f
   SDBKEY(C1)   : gg...g
   SDBKEY(C2)   : hh...h
   Complex      : ii...i
   Idx Name1    : kk...k
   Idx Key Len1 : ll...l
   Idx Key1     : mm...m
   Idx Name2    : kk...k
   Idx Key Len2 : ll...l
   Idx Key2     : mm...m
       :            :
   Idx Name10   : kk...k
   Idx Key Len10: ll...l
   Idx Key10    : mm...m
```

SDB データベースを操作する API または DML の指定内容によっては表示されない項目があります。

[出力例]

＜レコード位置指示子の出力例＞

```
Result of SDB Cursor :
 # Cursor Type : RECORD CURSOR
   Record Name : FMB1000
   SDBKEY      : 0x0300010014000000150000000000000100
   EraseRetain : {NO}
   Idx Name1   : FMB1IDX1
   Idx Key Len1: 8
   Idx Key1    : 0x3030303031313131
```

＜親子集合位置指示子の出力例＞

```
Result of SDB Cursor :
 # Cursor Type : SET CURSOR
   P Rec Name  : FMB1000
   C Rec Name  : FMB101
   SDBKEY(P)   : 0x0300010014000000150000000000000100
   SDBKEY(C1)  : 0x0400010014000000150000000000000200
   Complex     : NO
   Idx Name1   : FMB1IDX3
   Idx Key Len1: 8
   Idx Key1    : 0x3333333330303030
```

[説明]

出力内容	説明
Result of SDB Cursor	位置指示子情報が表示されます。
Cursor Type	位置指示子種別（出力文字数は最大 13 バイト） • RECORD CURSOR：レコード位置指示子 • SET CURSOR：親子集合位置指示子
Record Name	レコード位置指示子に対応するレコードビュー名（出力文字数は最大 30 バイト） 位置指示子種別が SET CURSOR（親子集合位置指示子）の場合，この行は表示されません。 レコードビュー名が設定されていない場合，“*****”が表示されます。
P Rec Name	親子集合位置指示子の親に対応するレコードビュー名（出力文字数は最大 30 バイト） 位置指示子種別が RECORD CURSOR（レコード位置指示子）の場合，この行は表示されません。 レコードビュー名が設定されていない場合，“*****”が表示されます。
C Rec Name	親子集合位置指示子の子に対応するレコードビュー名（出力文字数は最大 30 バイト） 位置指示子種別が RECORD CURSOR（レコード位置指示子）の場合，この行は表示されません。 レコードビュー名が設定されていない場合，“*****”が表示されます。
SDBKEY※	レコード位置指示子が指す SDBKEY（16 進数形式）（出力文字数は最大 34 バイト） 位置指示子種別が SET CURSOR（親子集合位置指示子）の場合，この行は表示されません。
SDBKEY(P)※	親子集合位置指示子が指す親の SDBKEY（16 進数形式）（出力文字数は最大 34 バイト） 位置指示子種別が RECORD CURSOR（レコード位置指示子）の場合，この行は表示されません。

出力内容	説明
SDBKEY(C1)※	親子集合位置指示子が指す子の SDBKEY（16 進数形式）（出力文字数は最大 34 バイト） 位置指示子種別が RECORD CURSOR（レコード位置指示子）の場合，この行は表示されません。
SDBKEY(C2)※	親子集合位置指示子が指す子の SDBKEY（16 進数形式）（出力文字数は最大 34 バイト） 位置指示子種別が RECORD CURSOR（レコード位置指示子）の場合，この行は表示されません。 また，位置指示子種別が SET CURSOR（親子集合位置指示子）であっても，親子集合位置指示子が複合状態でない場合，この行は表示されません。
Complex	親子集合位置指示子が複合状態であるかどうか（出力文字数は最大 3 バイト） YES：複合状態である NO：複合状態でない 位置指示子種別が RECORD CURSOR（レコード位置指示子）の場合，この行は表示されません。
EraseRetain	システムが使用する内部情報
Idx Name1 Idx Name2 ：	インデクス名（出力文字数は最大 30 バイト） 位置指示子中にインデクスの情報を保持していない場合，この行は表示されません（領域だけ存在している場合，“*****”が表示されます）。 該当するレコードのインデクスの個数分表示されます。表示順序はインデクス型番号の順番となります。
Idx Key Len1 Idx Key Len2 ：	インデクスキー長（出力文字数は最大 3 バイト） 位置指示子中にインデクスの情報を保持していない場合，この行は表示されません。 該当するレコードのインデクスの個数分表示されます。表示順序はインデクス型番号の順番となります。
Idx Key1 Idx Key2 ：	インデクスキー値（16 進数形式）（出力文字数は最大 1,016 バイト） 位置指示子中にインデクスの情報を保持していない場合，またはインデクスキー長が 0 の場合，この行は表示されません。 該当するレコードのインデクスの個数分表示されます。表示順序はインデクス型番号の順番となります。

注※

SDBKEY は空値であっても，0x00000000000000000000000000000000 のように値が出力されます。

(c) 増加するアクセスパス情報および位置指示子情報の容量

アクセスパス情報および位置指示子情報を表示すると，SQL トレースファイルの容量が次の式で示す容量分増えます。SQL トレースファイルの見積もり時には注意してください。ただし，SDB データベースの定義，および SDB データベースを操作する API または DML によって，容量は大きく変わります。次の式で見積もった値は，ある程度の目安にしてください。

```
増加量（単位：キロバイト）＝75×SDBデータベースを操作するAPIまたはDMLの実行回数
```

(d)　アクセスパス情報および位置指示子情報が省略された場合

通信量の制限によってアクセスパス情報および位置指示子情報が省略された場合，その個数が次のように出力されます。

（例）

```
The number of omitted blocks of DML Optimizer and SDB Cursor is 10.
```

(e)　UAP 単位の情報および TRANSACTION 単位の情報

[出力例]

```
UAP INFORMATION :
[1]UAPNAME(Unknown)
[2]SVHOST(sdb-bs2500-6) [3]SVPORT(37026) [4]SVNAME(testfes2) [5]CNCTNO(2)
[6]SVPID(10649) [7]CLPID(47876) [8]CLTTID(0)
[9]WAITT(0) [10]CTIME(10)
[11]ROREQ(0) [12]ROHITS(0)
[13]SOREQ(0) [14]SOHITS(0) [15]SOCRT(0) [16]SOMAX(0)
[17]COMT(0) [18]ROLB(1) [19]FROW(0) [20]DROW(0) [21]IROW(0)
[22]UROW(0) [23]SET(0) [24]OPEN(0) [25]FETC(0) [26]CLOS(0)
[27]DESC(0) [28]SEL(0) [29]INS(0) [30]UPD(0) [31]DEL(0)
[32]LOCK(0) [33]CRTT(0) [34]DRPT(0) [35]ALTT(0) [36]CRTI(0)
[37]DRPI(0) [38]CMTT(0) [39]CMTC(0) [40]CRTS(0) [41]DRPS(0)
[42]GRTR(0) [43]GRTS(0) [44]GRTA(0) [45]GRTC(0) [46]GRTD(0)
[47]RVKR(0) [48]RVKS(0) [49]RVKA(0) [50]RVKC(0) [51]RVKD(0)
[52]CRTV(0) [53]DRPV(0) [54]PRGT(0) [55]CRTP(0) [56]DRPP(0)
[57]ALTP(0) [58]CALL(0) [59]DESI(0) [60]MISC(0)
[61]MAXIO(0) [62]MAXIOM(0) [63]MINIO(0) [64]MINIOM(0)
[65]IOTIM(0) [66]IOTIMM(0)
[67]DIDRC(11) [68]DIDUC(0) [69]DIDHC(11) [70]DIDRD(0) [71]DIDWT(0)
[72]LBRFC(0) [73]LBUPC(0) [74]LBRHC(0) [75]LBUHC(0) [76]LBRDC(0)
[77]LBWTC(0) [78]BFSHC(0) [79]BRDWC(0) [80]BWTWC(0)
[81]BLKWC(0) [82]MWFN(0) [83]MWFEC(0) [84]MWFVL(0)
[85]WFRDC(0) [86]WFWTC(0) [87]WBFOC(0)
[88]MWHTS(0) [89]MBSL1(0) [90]MBSL2(0) [91]MBSL3(0)
[92]SCHSKD(0) [93]SCHCHG(0)
[94]CINSM(0) [95]CAFLS(0) [96]CAFWR(0) [97]CFMAX(0) [98]CFAVG(0)
[99]LDIRC(0) [100]LDIUC(0) [101]LDIHC(0) [102]LDIRD(0)
[103]LDIWT(0) [104]LBFSHC(0)
[105]ARREQ(0) [106]ARWC(0) [107]ARWT(0) [108]ARWTM(0)
[109]ARWTA(0) [110]ARWTMA(0) [111]ARSTA(0) [112]ARSTMA(0)
[113]HJMAX(0) [114]HJCMC(0) [115]HJHTC(0)
[116]DMLC(8) [117]DMLS(1) [118]DMLE(0) [119]DMLFT(4)
[120]DMLER(0) [121]DMLST(0) [122]DMLMO(0) [123]DMLDE(0)
[124]DMLIF(0) [125]VRTFT(0) [126]VRTST(0) [127]VRTMO(0)
[128]ONYS(1) [129]ONYE(0) [130]SDBFT(4) [131]SDBER(0)
[132]SDBST(0) [133]SDBMO(0) [134]SDFTR(4) [135]SDERR(0)
[136]SDSTR(0) [137]SDMOR(0)
[138]LCKCTC(11) [139]LCKCWC(0) [140]LCKCWT(0) [141]LCKCTM(0)
[142]LCKCWQ(0) [143]LCKCQM(0) [144]JNLFTC(1) [145]JNLFWC(0)
[146]JNLFWT(0) [147]JNLFTM(0) [148]JNLFWQ(0) [149]JNLFQM(0)
[150]BUFHTC(22) [151]BUFHWC(0) [152]BUFHWT(0) [153]BUFHTM(0)
[154]BUFHWQ(0) [155]BUFHQM(0)
[156]TUSEG(0)
[157]COMPC(0) [158]COMPT(0) [159]COMPTM(0)
[160]CMPMT(0) [161]CMPMTM(0)
[162]EXTEC(0) [163]EXTET(0) [164]EXTETM(0)
[165]EXTMT(0) [166]EXTMTM(0)
[167]SDTFC(0) [168]DMLGT(4) [169]SDBGT(4) [170]SDGTR(4)
```

UAP 単位の情報および TRANSACTION 単位の情報に出力される各項目については，マニュアル「HiRDB UAP 開発ガイド」の「UAP 統計レポートの見方」を参照してください。

ただし，上記の出力例で，枠で囲まれた部分の項目については，以下で説明します。

[説明]

番号	ヘッダ名	説明	
116	DMLC	API または DML 要求回数	SDB データベースを操作する API または DML の実行を要求した回数が出力されます。
117	DMLS	個別開始要求回数	API（個別開始）の実行を要求した回数，または DML の実行要求に対してシステムが個別開始の実行を要求した回数が出力されます。
118	DMLE	個別終了要求回数	API（個別終了）の実行を要求した回数が出力されます。
119	DMLFT	FETCH，FIND，または FETCHDB ALL 要求回数	API または DML（FETCH，FIND，または FETCHDB ALL）の実行を要求した回数が出力されます。
120	DMLER	ERASE 要求回数	API または DML（ERASE）の実行を要求した回数が出力されます。
121	DMLST	STORE 要求回数	API または DML（STORE）の実行を要求した回数が出力されます。
122	DMLMO	MODIFY 要求回数	API または DML（MODIFY）の実行を要求した回数が出力されます。
123	DMLDE	一括削除要求回数	API（一括削除）の実行を要求した回数が出力されます。
124	DMLIF	容量情報取得要求回数	API（容量情報取得）の実行を要求した回数が出力されます。
125	VRTFT	仮想ルートレコード FETCH 実行回数※	API（FETCH，STORE，一括削除，または容量情報取得）の実行要求に対して，仮想ルートレコードに FETCH を実行した回数が出力されます。
126	VRTST	仮想ルートレコード STORE 実行回数※	API（STORE）の実行要求に対して，仮想ルートレコードに STORE を実行した回数が出力されます。
127	VRTMO	仮想ルートレコード MODIFY 実行回数※	API（STORE または一括削除）の実行要求に対して，仮想ルートレコードに MODIFY を実行した回数が出力されます。
128	ONYS	個別開始実行回数	API（個別開始）の実行要求に対して，実際に個別開始を実行した回数が出力されます。または DML の実行要求に対するシステムによる個別開始の実行要求に対して，実際に個別開始を実行した回数が出力されます。

番号	ヘッダ名	説明	
129	ONYE	個別終了実行回数	API（個別終了）の実行要求に対して，実際に個別終了を実行した回数が出力されます。
130	SDBFT	FETCH，または FETCHDB ALL 実行回数※	API または DML（FETCH，FIND，または FETCHDB ALL）の実行要求に対して，実際に FETCH，または FETCHDB ALL を実行した回数が出力されます。
131	SDBER	ERASE 実行回数※	API または DML（ERASE）の実行要求に対して，実際に ERASE を実行した回数が出力されます。
132	SDBST	STORE 実行回数※	API または DML（STORE）の実行要求に対して，実際に STORE を実行した回数が出力されます。
133	SDBMO	MODIFY 実行回数※	API または DML（MODIFY）の実行要求に対して，実際に MODIFY を実行した回数が出力されます。
134	SDFTR	FETCH，または FETCHDB ALL 取得レコード件数※	API または DML（FETCH，FIND，または FETCHDB ALL）の実行要求に対して，実際に FETCH，または FETCHDB ALL を実行して取得したレコード件数が出力されます。
135	SDERR	ERASE 削除レコード件数※	API または DML（ERASE）の実行要求に対して，実際に ERASE を実行して削除したレコード件数が出力されます。 削除対象のレコードの子レコードも削除された場合，そのレコードも数に含まれます。
136	SDSTR	STORE 追加レコード件数※	API または DML（STORE）の実行要求に対して，実際に STORE を実行して追加したレコード件数が出力されます。
137	SDMOR	MODIFY 更新レコード件数※	API または DML（MODIFY）の実行要求に対して，実際に MODIFY を実行して更新したレコード件数が出力されます。
167	SDTFC	終端検知用の検索実行回数	UAP からの SDB データベースを操作する API の実行要求時，HiRDB/SD が内部的に実行する終端検知用の（レコードの検索時に最終のレコードかどうかを判定するための），FETCH または FETCHDB ALL を実行した回数が出力されます。
168	DMLGT	GET 要求回数	DML（GET）の実行を要求した回数が出力されます。
169	SDBGT	GET 実行回数	DML（GET）の実行要求に対して，実際に GET を実行した回数が出力されます。
170	SDGTR	GET 取得レコード件数	DML（GET）の実行要求に対して，実際に GET を実行して取得したレコード件数が出力されます。

注※

API または DML の実行要求に対して，FETCH，FETCHDB ALL，STORE，MODIFY，または ERASE が複数回実行された場合，実行された回数分，実行回数およびレコード件数がカウントされます。複数回実行する途中でエラーが発生した場合は，エラーが発生するまでの実行回数，レコード件数となります。

16.5.6　トラブルシュート機能に関する注意事項

次のトラブルシュート機能については，SDB データベースに関する情報は出力されません。

- SQL トレース動的取得機能
- HiRDB SQL Tuning Advisor

17 DML リファレンス【SD FMB】

この章では，SD FMB の SDB データベースを操作する DML の機能と文法について説明します。

17.1　DML の一覧

SD FMB の SDB データベースを操作する DML の一覧を次の表に示します。

表 17-1　SD FMB の SDB データベースを操作する DML の一覧

項番	分類	DML	機能
1	操作系 DML	ERASE	レコード実現値を削除します。
2		FETCH	レコード実現値を検索して，レコード実現値を取得します。また，検索したレコードに対して位置づけを行います。
3		FIND	レコード実現値に位置指示子を位置づけます。
4		GET	レコード位置指示子で位置づけられているレコード実現値を取得します。
5		MODIFY	レコード実現値を更新します。
6		STORE	レコード実現値を格納します。

注意事項

4V FMB または 4V AFM の SDB データベースを DML で操作することはできません。

17.2　基本項目

DML を使用する際の基本項目について説明します。

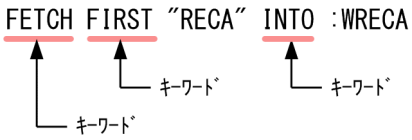
17.2.1　DML の記述形式

(1)　オペランドの指定順序

オペランドは，各 DML の形式で記述している順序に従って指定してください。

(2)　キーワードの指定

DML の名称（FETCH，MODIFY など）のように，機能を使用するために指定する語をキーワードといいます。キーワードは，システムの予約語として登録されているため，決められた位置以外には記述できません。キーワードの例を次に示します。



予約語については，「付録 O.1　DML の予約語」を参照してください。

(3)　区切り文字の挿入

区切り文字には，次の文字を指定できます。

- 空白（X'20'）
- タブ（X'09'）

区切り文字を挿入する位置については，マニュアル「HiRDB SQL リファレンス」の「区切り文字の挿入」を参照してください。

(4)　注釈（コメント）

DML 中に囲み注釈「/*　　*/」や，単純注釈「--」などを記述できません。

(5)　DML で使用できる文字

DML で使用できる文字については，マニュアル「HiRDB SQL リファレンス」の「SQL で使用できる文字」を参照してください。ただし，pdsetup コマンドで指定した文字コードが sjis の場合，全角文字に外字を含みません。

(6) DMLの最大長

UAPに記述する場合，1つのDML文は，区切り文字も含めて32,000バイトまで記述できます。

HiRDB/SDデータベースアクセスユティリティ（pdsdbexe）で実行する場合，1つのDMLコマンドは，区切り文字も含めて2,000,000バイトまで記述できます。

(7) 名前の指定

名前の指定方法には，引用符（"）で囲んで指定する方法と，引用符で囲まないで指定する方法があります。引用符で囲んで指定する方法を推奨します。

推奨する理由

名前には，予約語と同じ名前を指定できませんが，引用符で囲んだ場合は，予約語と同じ名前を指定できます。

DMLの機能拡張に伴い，将来予約語が増える可能性があります。名前をあらかじめ引用符で囲んでおくと，将来指定した名前が予約語と重複しても，DMLを書き換える必要が生じません。

注意事項

- 名前を引用符（"）で囲んだ場合，半角英小文字と半角英大文字が区別されます。
- 次に示す名前は，引用符（"）で囲まないでください。
  ・埋込み変数名
- OCCURS句を指定して生成された構成要素名を使用する場合は引用符（"）で囲んでください。OCCURS句については，「11.7.1(4)(g) OCCURS」を照してください。

名前については，「11.4.1(1) 名前の規則」を参照してください。

埋込み変数名については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAPの作成」の「埋込み変数の宣言」を参照してください。

(8) 留意事項

DMLコマンドの記述規則については，「14. HiRDB/SD データベースアクセスユティリティ(pdsdbexe)」の「14.9 規則および留意事項」も参照してください。

17.2.2 DMLのデータ型

(1) DMLのデータ型

DMLで操作するレコードのデータ型については，「2.3.4(1) データ項目」を参照してください。

(2) 変換（代入，比較）できるデータ型

変換（代入，比較）できる定数の指定形式を次の表に示します。埋込み変数を使用した変換（代入，比較）については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の作成」の「埋込み変数の宣言」を参照してください。

表 17-2 変換（代入，比較）できるデータ型

項番	DML のデータ型	指定値の形式※	指定例	備考
1	CHARACTER	文字列定数	ELM01＝'AAABBB'	• レコードの構成要素が基本項目の場合，データ型が CHARACTER であれば，DML のデータ型は CHARACTER になります。 • レコードの構成要素が集団項目の場合，すべての構成要素のデータ型が CHARACTER であれば，DML のデータ型は CHARACTER になります。
2		16 進文字列定数	ELM01＝X'414141424242'	
3	INTEGER	整数定数	ELM01＝100	• レコードの構成要素が基本項目の場合，データ型が INTEGER であれば，DML のデータ型は INTEGER になります。 • レコードの構成要素が集団項目の場合，構成要素数が 1 つでデータ型が INTEGER であれば，DML のデータ型は INTEGER になります。
4		16 進文字列定数	ELM01＝X'64000000'	
5	SMALLINT	整数定数	ELM01＝100	• レコードの構成要素が基本項目の場合，データ型が SMALLINT であれば，DML のデータ型は SMALLINT になります。 • レコードの構成要素が集団項目の場合，構成要素数が 1 つでデータ型が SMALLINT であれば，DML のデータ型は SMALLINT になります。
6		16 進文字列定数	ELM01＝X'6400'	
7	PACKED DECIMAL FIXED	10 進数定数	ELM01＝123.456	• レコードの構成要素が基本項目の場合，データ型が PACKED DECIMAL FIXED であれば，DML のデータ型は PACKED DECIMAL FIXED になります。 • レコードの構成要素が集団項目の場合，構成要素数が 1 つでデータ型が PACKED DECIMAL FIXED であれば，
8		16 進文字列定数	ELM01＝X'0123456C'	

項番	DML のデータ型	指定値の形式※	指定例	備考
				DML のデータ型は PACKED DECIMAL FIXED になります。
9	上記以外	16 進文字列 定数	ELM01＝ X'010241414164000000'	なし

注※

各定数の表記方法については，マニュアル「HiRDB SQL リファレンス」の「定数」を参照してください。

指定した定数に対して，HiRDB/SD はパディング（データを追加して長さを合わせる処理）や，切り捨てなどのデータの加工は行いません。

また，DML のデータ型によっては，次の条件を満たす必要があります。

- DML のデータ型が CHARACTER または XCHARACTER の場合

  データ長が一致している必要があります。

- DML のデータ型が PACKED DECIMAL FIXED，INTEGER，または SMALLINT の場合

  指定値は DML のデータ型で表現できる範囲の値でなければなりません。また，指定値を 16 進文字列定数で指定した場合，指定値の長さは，構成要素の長さと等しくする必要があります。

  なお，DML のデータ型が PACKED DECIMAL FIXED の場合，指定値に指定した文字数で整数部桁数および小数部桁数が決まります。

  （例）

  ```
  01234.000 → 整数部桁数5，小数部桁数3
  .12345 → 整数部桁数0，小数部桁数5
  ```

  上記の例のようにゼロサプレスはされません。指定値の整数部桁数および小数部桁数が，構成要素の整数部桁数および小数部桁数以下になるようにしてください。

(3)　PACKED DECIMAL FIXED 使用上の注意事項

DML のデータ型に PACKED DECIMAL FIXED を使用する場合の注意事項については，「2.3.4(1)　データ項目」を参照してください。

17.2.3　埋込み変数

DML と UAP との間で値の受け渡しをする際に，埋込み変数を使用します。埋込み変数の指定形式と機能について説明します。

DML に埋込み変数を指定する場合の形式を以下に示します。

形式

```
:埋込み変数
```

注意事項

コロン（:）と埋込み変数の間に区切り文字を記述しないでください。

機能および用途

埋込み変数の機能および用途を次の表に示します。

表 17-3　埋込み変数の機能および用途

機能	用途	埋込み変数を記述する DML
検索結果のレコードの値の受け取り	レコードの値を受け取ります。	FETCH 文
条件値の指定	検索条件に指定した構成要素と比較する値を指定します。	FETCH 文，FIND 文
更新値の指定	更新後のレコードの値を指定します。	MODIFY 文
格納値の指定	格納するレコードの値を指定します。	STORE 文

埋込み変数のデータ型と COBOL 言語のデータ記述の関係

埋込み変数を UAP 中に記述するときのデータ型とデータ記述の関係については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の作成」の「埋込み変数の宣言」を参照してください。

埋込み変数の修飾指定

埋込み変数を集団項目で修飾できません。

17.3　構成要素の詳細

DML 中に記述する構成要素について説明します。

17.3.1　探索条件の指定

(1)　機能

DML 中に記述した探索条件に従って論理演算が実行され，その結果が真のものだけが検索対象になります。探索条件は次の個所に指定できます。

- FETCH 文の WHERE 句
- FIND 文の WHERE 句

(2)　形式

```
探索条件：：＝比較条件 〔AND 比較条件 〕…
    比較条件：：＝構成要素名 {＝|＞|＜|＞＝|＜＝} 条件値
    条件値：：＝ {:埋込み変数
                |構成要素に対応する値
                |（構成要素に対応する値〔,構成要素に対応する値〕…）}
```

(3)　説明

●:埋込み変数

構成要素と比較する値を保持する埋込み変数の名称を指定します。

●構成要素に対応する値

構成要素と比較する値を指定します。1 つの定数を指定します。

●（構成要素に対応する値〔,構成要素に対応する値〕…）

集団項目内の基本項目と比較する値を指定します。集団項目内の各基本項目に対する値を定義順，かつ過不足なく指定する必要があります。

(4)　共通規則

- 探索条件に指定できる比較条件の個数の最大は 64 です。
- 埋込み変数を指定する場合，比較する構成要素と構成が一致していて，かつ比較できるデータ型である必要があります。埋込み変数については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「埋込み変数の宣言」を参照してください。
- 構成要素に対応する値の指定形式については，「17.2.2(2)　変換（代入，比較）できるデータ型」を参照してください。

- HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の DML コマンドに埋込み変数を指定できません。
- OCCURS 句を指定して生成された構成要素名を使用する場合は，引用符（"）で囲んでください。OCCURS 句については，「11.7.1(4)(g)　OCCURS」を参照してください。
- 比較条件のうち，比較対象の構成要素が次のどれかを満たす条件を，インデクスキーの条件と呼びます。
  - 構成要素名にインデクスの先頭の構成要素を指定した条件
  - 構成要素名に集団項目を指定した場合は次のすべてを満たす条件

    ・集団項目の先頭からインデクス定義に指定した構成要素が定義順にもれなく連続していること

    ・インデクス構成要素以外を含まないこと

    ・先頭の構成要素と ASCENDING または DESCENDING の指定が異なる構成要素を含まないこと

    なお，上記を満たしている場合，インデクス定義の構成要素をすべて含まなくてもインデクスキーの条件として扱います（前方からの部分一致でもインデクスキーの条件と見なします）。

（例）

＜スキーマ定義＞

```
RECORD REC1
        2 K1            CHARACTER  1 TYPE K,L
        2 DBKEY         INTEGER       TYPE K,N
        2 D1                  … (U1,U2,U3,U4を含む)
          3 D2                … (U1,U2,U3を含む)
            4 D3              … (U1,U2を含む)
              5 U1      CHARACTER  1 TYPE U,D
              5 U2      CHARACTER  1 TYPE U,D
            4 U3        CHARACTER  1 TYPE U,D
          3 U4          CHARACTER  1 TYPE U,D
        2 U5            CHARACTER  1 TYPE U,D
```

＜インデクス定義＞

```
SECONDARY INDEX I1 FOR RECORD ORDER KEY U1,U2
SECONDARY INDEX I2 FOR RECORD ORDER KEY U1,U2,U3
SECONDARY INDEX I3 FOR RECORD ORDER KEY U1,U2,U3,U4
SECONDARY INDEX I4 FOR RECORD ORDER KEY U1,U2,U4
SECONDARY INDEX I5 FOR RECORD ORDER KEY U1,U3
SECONDARY INDEX I6 FOR RECORD ORDER KEY U2,U1
SECONDARY INDEX I7 FOR RECORD ORDER KEY ASCENDING  U1,U2
                                        DESCENDING U3,U4
```

＜比較条件＞

インデクス	インデクスキーの条件となる比較条件
I1	U1>'A' D3>'AB'
I2	U1>'A' D3>'AB' D2>'ABC'
I3	U1>'A' D3>'AB' D2>'ABC' D1>'ABCD'
I4	U1>'A' D3>'AB'
I5	U1>'A'
I6	U2>'A'
I7	U1>'A' D3>'AB' ＜参考＞ 次の条件は，ASCENDINGとDESCENDINGの指定が混在しているため，インデクスキーの条件ではありません。 D2>'ABC' D1>'ABCD'

- レコード型内の検索では，指定されたインデクスキーの条件に従いインデクスを選択します。インデクス選択規則については「17.3.2　インデクス選択規則」を参照してください。
- 集団項目を指定した場合の比較条件の結果を次に示します。
  - ＝を指定した場合

    集団項目下のおのおのの基本項目と，対応する値を比較し，すべて「＝」である場合に真となります。それ以外の場合は偽になります。
  - ＞を指定した場合

    集団項目下のおのおのの基本項目と，対応する値を先頭から順番に「＝」の間，比較します。最初に現れた「＝」ではない基本項目と対応する値の関係が「＞」の場合に真，「＜」の場合に偽となります。なお，すべての要素が「＝」の場合は偽となります。
  - ＜を指定した場合

集団項目下のおのおのの基本項目と，対応する値を先頭から順番に「=」の間，比較します。最初に現れた「=」ではない基本項目と対応する値の関係が「<」の場合に真，「>」の場合に偽となります。なお，すべての要素が「=」の場合は偽となります。

- >=を指定した場合

  集団項目下のおのおのの基本項目と，対応する値を先頭から順番に「=」の間，比較します。最初に現れた「=」ではない基本項目と対応する値の関係が「>」の場合に真，「<」の場合に偽となります。なお，すべての要素が「=」の場合は真となります。

- <=を指定した場合

  集団項目下のおのおのの基本項目と，対応する値を先頭から順番に「=」の間，比較します。最初に現れた「=」ではない基本項目と対応する値の関係が「<」の場合に真，「>」の場合に偽となります。なお，すべての要素が「=」の場合は真となります。

(例)

<スキーマ定義>

```
RECORD REC1
        2 K1        CHARACTER  1 TYPE K,L
        2 DBKEY     INTEGER       TYPE K,N
        2 D1        … (U1,U2,U3を含む)
          3 U1      CHARACTER  1 TYPE U,D
          3 U2      INTEGER       TYPE U,D
          3 U3      CHARACTER  1 TYPE U,D
```

<比較条件の結果>

比較条件	結果	備考
D1('B',2,'b') =('B',2,'b')	真	すべての構成要素の値が「=」のため真
D1('B',2,'b') =('A',2,'b')	偽	1番目の要素が「=」でないため偽
D1('B',2,'b') =('B',2,'c')	偽	1,2番目の要素が「=」, 3番目の要素が「=」でないため偽
D1('B',2,'b') >('B',2,'b')	偽	すべての構成要素の値が「=」のため偽
D1('B',2,'b') >('A',2,'b')	真	1番目の要素が「>」のため真
D1('B',2,'b') >('B',1,'b')	真	1番目の要素が「=」, 2番目の要素が「>」のため真
D1('B',2,'b') >('B',2,'c')	偽	1,2番目の要素は「=」, 3番目の要素が「<」のため偽
D1('B',2,'b') <=('B',2,'b')	真	すべての構成要素の値が「=」のため真
D1('B',2,'b') <=('A',2,'b')	偽	1番目の要素が「>」のため偽
D1('B',2,'b') <=('B',1,'b')	偽	1番目の要素が「=」, 2番目の要素が「>」のため偽
D1('B',2,'b') <=('B',2,'c')	真	1,2番目の要素は「=」, 3番目の要素が「<」のため真

（凡例）　D1(U1の値,U2の値,U3の値)

- インデクスのサーチ範囲について説明します。

インデクスを使用した検索の場合，インデクスのサーチ範囲は指定された条件を基に決定します。インデクスのサーチ範囲を次に示します。

- ASCENDING および DESCENDING の指定が統一されたインデクスを使用した検索の場合

  すべての構成要素の条件値を使用してサーチ範囲を決定します。

  （例）

構成要素の定義

```
   2 D1
    3 U1  CHARACTER  1 TYPE K,L
    3 U2  CHARACTER  1 TYPE K,L
    3 U3  CHARACTER  1 TYPE K,L
    3 U4  CHARACTER  1 TYPE K,L
```

インデクス定義：

```
   SEQUENTIAL IDX1 FOR RECORD
     ORDER KEY ASCENDING U1,U2,U3,U4
```

DML

```
   FETCH FIRST RECA WHERE (D1>'BYby')
```

サーチ範囲：

インデクスデータ			
U1(ASC)	U2(ASC)	U3(ASC)	U4(ASC)
B	Y	b	x
B	Y	b	y
B	Y	b	z
B	Y	c	x
B	Z	a	x
C	X	a	x

検索結果

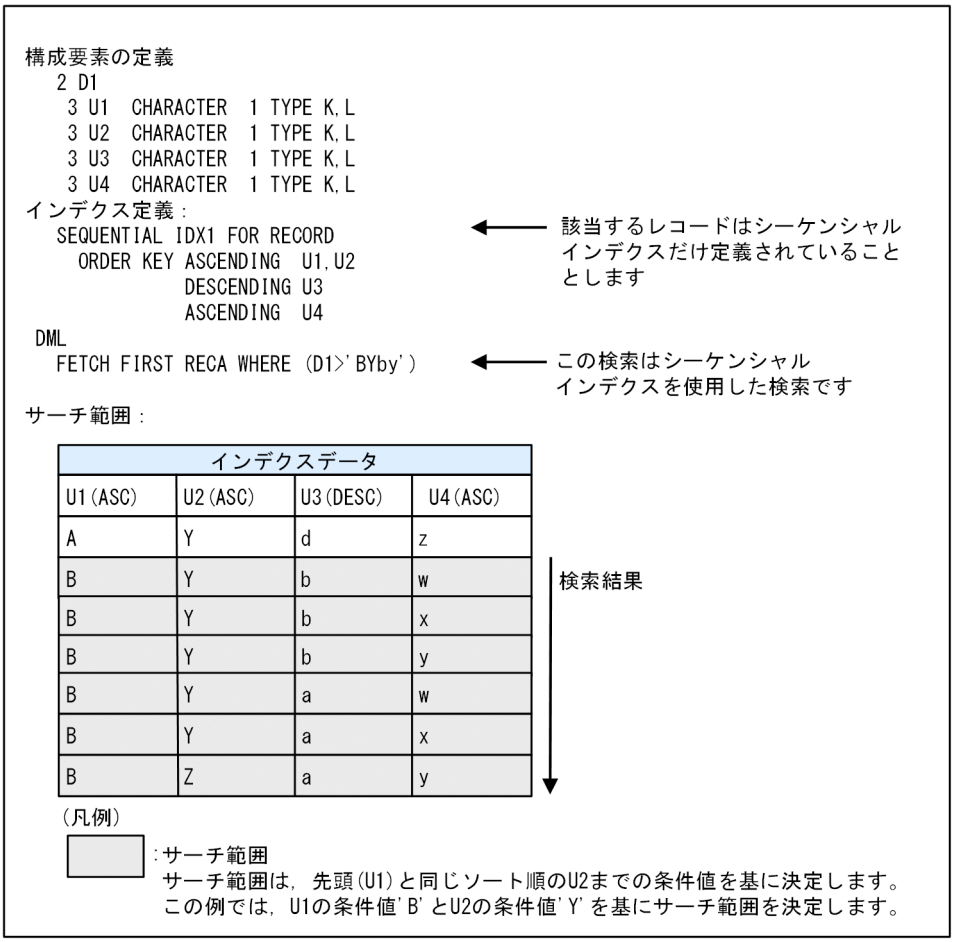
（凡例）

：サーチ範囲
サーチ範囲は，U1～U4すべての構成要素の条件値を基に決定します。

- ASCENDING および DESCENDING の指定が混在したインデクスを使用した検索の場合

  先頭の構成要素と，ASCENDING または DESCENDING が同じ指定の構成要素で先頭から連続した部分の条件値を使用してサーチ範囲を決定します。

  （例）



ASCENDING および DESCENDING の指定が混在したインデクスの検索では，指定が統一されたインデクスの検索に比べてサーチ範囲が広くなることがあります。サーチ範囲については，UAP 統計レポートの SearchCnd の内容で確認できます。

UAP 統計レポートについては，「16.5.5　UAP 統計レポート機能」を参照してください。サーチ範囲については，マニュアル「HiRDB コマンドリファレンス」の「サーチ条件」を参照してください。

17.3.2　インデクス選択規則

レコード型内の検索は，インデクスを使用した検索となります。検索時のインデクスは，DML の指定内容によってシステムが選択します。

インデクス選択規則を次に示します。

1. DML の条件に INDEXED BY を指定した場合は，指定したインデクスを使用して検索を行います。
2. DML の条件にインデクスキーの条件が指定された場合，インデクスキーの条件に対応するインデクスを使用して検索を行います。ただし，複数のインデクスが該当する場合は，最初に定義したインデクスとします。インデクスキーの条件については，「17.3.1　探索条件の指定」を参照してください。

3. DMLの条件にインデクスキーの条件が指定されていない場合，検索対象のレコードで最初に定義したインデクスを使用して検索を行います。なお，検索対象のレコードにインデクスが定義されていない場合は，エラーになります。

（例）

<スキーマ定義>
```
RECORD REC1
        2 K1          CHARACTER  1 TYPE K,L
        2 DBKEY       INTEGER      TYPE K,N
        2 D1            …（当該集団項目はU1,U2,U3を含む）
         3 D2           …（当該集団項目はU1,U2を含む）
          4 U1        CHARACTER  1 TYPE U,D
          4 U2        CHARACTER  1 TYPE U,D
         3 U3         CHARACTER  1 TYPE U,D
```

■条件例1

<インデクス定義>

次の順番で定義されているものとします。

```
SECONDARY INDEX I1 FOR RECORD ORDER KEY U1
SECONDARY INDEX I2 FOR RECORD ORDER KEY U1,U2
SECONDARY INDEX I3 FOR RECORD ORDER KEY U1,U2,U3
```

<比較条件>

比較条件	使用インデクス	備考
U1>'A'	I1	I2,I3に対してもインデクスキーの条件に該当するが，定義順でI1を選択する。
D2>'AB'	I2	I3に対してもインデクスキーの条件に該当するが，定義順でI2を選択する。
D3>'ABC'	I3	

■条件例2

<インデクス定義>

次の順番で定義されているものとします。

```
SECONDARY INDEX I1 FOR RECORD ORDER KEY U1
SECONDARY INDEX I2 FOR RECORD ORDER KEY U2
```

<比較条件>

比較条件	使用インデクス	備考
U1>'A' AND U2>'B'	I1	U1の条件はI1のインデクスキーの条件， U2の条件はI2のインデクスキーの条件に該当する。定義順でI1を選択する。

17.3.3　検索の順序，開始位置および方向

レコード型内の検索と親子集合型内の検索は，順序，開始位置および方向を決定してレコードを検索します。

順序は，レコードの並び順です。検索結果は，順序に従い検索された結果となります。開始位置は，順序で並んだレコード中の検索の開始位置です。方向は，順序で並んだレコードの検索方向（順方向，または逆順方向）です。

検索ごとの順序，開始位置および方向について説明します。

(1)　レコード型内の検索

レコード型内の検索の順序，開始位置，方向は，次の３つの要素によって決定します。

- DML の指示コード
  FIRST または NEXT
- レコード位置指示子の状態
  有効，または空値
- 検索順序の状態
  有効，または無効
  前回の DML 操作で順序が決定された状態かどうかを指します。検索順序は FETCH 文，FIND 文の実行時に決定します。
  なお，レコード位置指示子が空値の場合，検索順序の状態は無効です。
  検索順序が無効になる操作を次に示します。
  - レコード型内の検索以外でレコード位置指示子を位置づけた場合（親子集合型内の検索，位置指示子指定の検索）
  - STORE 文の実行時

レコード型内の検索での順序，開始位置，方向を次の表に示します。

表 17-4　レコード型内の検索での順序，開始位置，方向

項番	指示コード	レコード位置指示子の状態	検索順序の状態	順序	開始位置	方向
1	FIRST	－	－	DML の指定内容に従い検索に使用するインデクスをシステムが選択します。順序はインデクスのキー値の順となります。※1	インデクスの先頭のキー値を持つレコード	インデクスのキー値の順方向
2	NEXT	空値	－	DML の指定内容に従い検索に使用するインデクスをシステムが選択します。順序はインデクスのキー値の順となります。※1	インデクスの先頭のキー値を持つレコード	
3		有効	無効	DML の指定内容に従い検索に使用するインデクスをシステムが選択します。順序はインデクスのキー値の順となります。※1	位置指示子が示すレコードの 1 つあとのキー値を持つレコード	
4			有効	検索に使用するインデクスのキー値の順となります。※2	位置指示子が示すレコードの 1 つあとのキー値を持つレコード	

（凡例）
－：該当しない，または状態に影響されない

注※1
インデクス選択規則については「17.3.2　インデクス選択規則」を参照してください。

注※2

INDEXED BY を指定した場合は，INDEXED BY で指定したインデクスになります。

（例）

＜レコード定義＞

```
RECORD REC1
        2 K1          CHARACTER  1 TYPE K,L
        2 DBKEY       INTEGER      TYPE K,N
        2 U1          CHARACTER  1 TYPE U,D
        2 U2          INTEGER      TYPE U,D
```

＜インデクス定義＞

```
SECONDARY INDEX I1 FOR RECORD ORDER KEY U1
SECONDARY INDEX I2 FOR RECORD ORDER KEY U2
```

＜レコードデータ（並び順I1）＞

#	K1	DBKEY	U1	U2	備考
①	'A'	1	'a'	1	
②	'B'	1	'b'	4	
③	'C'	1	'c'	5	
★	'X'	1	'd'	99	DML実行例3の追加データ
④	'D'	1	'e'	3	
⑤	'E'	1	'f'	2	

＜レコードデータ（並び順I2）＞

#	K1	DBKEY	U1	U2	備考
①	'A'	1	'a'	1	
⑤	'E'	1	'f'	2	
④	'D'	1	'e'	3	
②	'B'	1	'b'	4	
③	'C'	1	'c'	5	
★	'X'	1	'd'	99	DML実行例3の追加データ

（続く）

（続き）

```
■DML実行例1
  FETCH FIRST REC1 WHERE U1>'a'   … 検索結果：②
                                    インデクスキーの条件によってI1を使用して検索を行います。
                                    I1のキーを先頭から検索し，条件（U1>'a'）を満たす最初の
                                     レコード（②）を結果とします。
  FETCH NEXT  REC1                … 検索結果：③
                                    直前の検索で決定した順序（I1）に従い，②の次のレコード
                                     （③）を結果とします。

■DML実行例2
  FETCH FIRST REC1 WHERE U1>'a'   … 検索結果：②
                                    DML実行例1と同様
  FETCH FIRST REC1 WHERE U2>1     … 検索結果：⑤
                                    FIRSTを指定しているため，新たに検索順序を決定します。
                                    インデクスキーの条件によって，I2を使用して検索を行い
                                    ます。I2のキーを先頭から検索し，条件（U2>1）を満たす
                                    最初のレコード（⑤）を結果とします。
  FETCH NEXT  REC1 WHERE U1>'d'   … 検索結果：④
                                    直前の検索で決定した順序（I2）に従い検索を行います。
                                    I2のキー順に⑤の次のレコードから条件（U1>'d'）を満たす
                                    最初のレコード（④）を結果とします。

■DML実行例3
  STORE REC1 SET U1='d',U2=99     … 新たなレコードを追加（レコードデータの★に該当）
                                    STORE文の実行によって検索順序は無効になります。
  FETCH NEXT REC1 WHERE U1>'e'    … 検索結果：⑤
                                    検索順序が無効のため，インデクスキーの条件によって，
                                    I1を使用して検索を行います。
                                    I1のキー順に★の次のレコードから検索を行います。
                                    条件（U1>'e'）を満たす最初のレコード（⑤）を結果とします。
```

(2)　親子集合型内の検索

親子集合型内の検索の順序，開始位置，方向は次の 2 つの要素によって決定します。

- DML の指示コード
  FIRST，LAST，NEXT，または PRIOR
- 親子集合位置指示子の子レコードの状態
  - 有効，または空値
  - 複合状態，または複合状態ではない

親子集合型内の検索での順序，開始位置，方向を次の表に示します。

表 17-5　親子集合型内の検索での順序，開始位置，方向

項番	指示コード	親子集合位置指示子の子レコードの状態		順序	開始位置	方向
1	FIRST	−		親子集合型のレコード間ポインタの順序	親レコードの FIRST ポインタが指すレコード	NEXT ポインタの方向
2	LAST	−			親レコードの LAST ポインタが指すレコード	PRIOR ポインタの方向
3	NEXT	有効	複合状態ではない		位置指示子の子レコードの NEXT ポインタが指すレコード	NEXT ポインタの方向

項番	指示コード	親子集合位置指示子の子レコードの状態		順序	開始位置	方向
4			複合状態である		位置指示子の子 1 のレコードの NEXT ポインタが指すレコード	
5		空値			親レコードの FIRST ポインタが指すレコード	
6	PRIOR	有効	複合状態ではない		位置指示子の子レコードの PRIOR ポインタが指すレコード	PRIOR ポインタの方向
7			複合状態である		位置指示子の子 2 のレコードの PRIOR ポインタが指すレコード	
8		空値			親レコードの LAST ポインタが指すレコード	

（凡例）

－：状態に影響されない

注

親子集合位置指示子の複合状態については，「2.5.3　位置指示子」を参照してください。

17.4　操作系 DML

操作系 DML の機能，構文形式，および規則について説明します。

操作系 DML は，レコード型のデータを操作するとき，および位置指示子を操作するときに使用します。

17.4.1　ERASE 文

(1)　機能

レコード実現値を削除します。

(2)　形式

(a)　UAP に記述する ERASE 文

```
ERASE
    レコード名
    ALL
```

(b)　DML コマンドで記述する ERASE 文

```
ERASE
    レコード名
    ALL
```

(3)　オペランドの説明

●レコード名

～＜識別子＞((1～30 バイト))

レコード実現値を削除するレコード名を指定します。

注意事項

- レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- レコード名に空白またはハイフン（-）を使用している場合は，レコード名を引用符（"）で囲んでください。

●ALL

最下位レベルのレコードに位置づけしている場合は，位置づけしているレコードのレコード実現値だけを削除します。

最下位レベル以外のレコードに位置づけしている場合は，位置づけしているレコードのレコード実現値と，その下位レベルのレコードのレコード実現値を削除します。

(4) 共通規則

- 削除対象のレコードを，次のどれかの DML で位置づけておく必要があります。
  - FIND 文（位置決め目的に FOR UPDATE を指定する）
  - FETCH 文（位置決め目的に FOR UPDATE を指定する）
  - STORE 文

(5) 使用例

例 1

ルートレコード RECA から，位置づいているレコード実現値と子レコードを削除します。

```
ERASE "RECA" ALL
```

例 2

最下位の子レコード RECB から，位置づいているレコード実現値だけを削除します。

```
ERASE "RECB" ALL
```

例 3

DML コマンドでルートレコード RECA から，位置づいているレコード実現値と子レコードを削除します。

```
ERASE "RECA" ALL
```

17.4.2 FETCH 文

(1) 機能

レコードを検索して，レコード実現値を取得します。また，検索したレコードに対して位置づけを行います。

検索には，次の 2 つがあります。

- レコード型内の検索

  レコード位置指示子を使用した検索です。常にインデクスを使用した検索となります。
- 親子集合型内の検索

  親子集合位置指示子を使用した検索です。特定の親レコード実現値配下の子レコードを検索します。常にポインタを使用した検索となります。

検索は順序，開始位置，および検索方向を決定してレコードを検索します。詳細については，「17.3.3　検索の順序，開始位置および方向」を参照してください。

(2)　形式

(a)　UAP に記述する FETCH 文

```
FETCH
     〔FOR UPDATE〕
     {FIRST | NEXT | LAST | PRIOR}
      レコード名
     〔INDEXED BY インデクス名〕
      INTO :埋込み変数
      〔WITHIN 親子集合型名〕
      〔WHERE (探索条件)〕
```

(b)　DML コマンドで記述する FETCH 文

```
FETCH
     〔FOR UPDATE〕
     {FIRST | NEXT | LAST | PRIOR}
      レコード名
     〔INDEXED BY インデクス名〕
      〔WITHIN OWNER〕
      〔WHERE (探索条件)〕
```

(3)　オペランドの説明

●FOR UPDATE

検索したレコードを更新（ERASE，MODIFY）するかどうかを指定します。

検索したレコードを更新する場合は，FOR UPDATE を指定してください。

検索したレコードを更新しない場合は，FOR UPDATE を省略してください。

● {FIRST | NEXT | LAST | PRIOR}

検索の開始位置となるレコードと，検索の方向を示す指示コードを指定します。

FIRST：先頭から順方向に検索します。

NEXT：位置づけされているレコードから順方向に検索します。

LAST：最後から逆順方向に検索します。

PRIOR：位置づけされているレコードから逆順方向に検索します。

LAST および PRIOR は親子集合型内の検索のときだけ指定できます。

検索の開始位置，方向については，「17.3.3　検索の順序，開始位置および方向」を参照してください。

●レコード名

～<識別子>((1～30 バイト))

検索対象のレコード名を指定します。

注意事項

- レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- レコード名に空白またはハイフン（-）を使用している場合は，レコード名を引用符（"）で囲んでください。

●INDEXED BY インデクス名

～＜識別子＞((1～30 バイト))

検索で使用するインデクスを明示する場合に指定します。

レコード型内の検索で指定できます。インデクス名には，検索対象のレコードに定義されたインデクスの名称を指定してください。

注意事項

- インデクス名に英小文字を使用している場合は，インデクス名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- インデクス名に空白またはハイフン（-）を使用している場合は，インデクス名を引用符（"）で囲んでください。

●INTO :埋込み変数

検索したレコード実現値を読み込むための埋込み変数を指定します。

埋込み変数の宣言方法については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の作成」の「埋込み変数の宣言」を参照してください。

●WITHIN 親子集合型名

～＜識別子＞((1～30 バイト))

親子集合型内の検索を行う場合に指定します。

検索対象とする親子集合の親子集合型名を指定します。

注意事項

- 親子集合型名に英小文字を使用している場合は，親子集合型名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- 親子集合型名に空白またはハイフン（-）を使用している場合は，親子集合型名を引用符（"）で囲んでください。

●WITHIN OWNER

親子集合型内の検索を行う場合に指定します。

●WHERE (探索条件)

検索するレコードの条件を指定します。探索条件が真になるレコードが検索対象になります。

探索条件については，「17.3.1　探索条件の指定」を参照してください。

(4) 共通規則

- レコード型内の検索は，インデクスを使用した検索となります。インデクスが定義されていないレコードは，レコード型内の検索を実行できません。インデクス選択規則については，「17.3.2 インデクス選択規則」を参照してください。
- 親子集合型内の検索の場合，検索対象のレコードの親レコードを次のどれかの DML で位置づけておく必要があります。
  - FIND 文
  - FETCH 文
  - STORE 文

(5) 使用例

例 1

ルートレコード RECA をレコード型内の検索で検索します。

条件値に埋込み変数を指定します。

```
FETCH FOR UPDATE FIRST "RECA" INTO :WRECA
    WHERE("KEYDATA"=:WKEYDATA)
```

例 2

ルートレコード RECA をレコード型内の検索で検索します。

条件値に構成要素に対応する値を指定します。

```
FETCH FOR UPDATE FIRST "RECA" INTO :WRECA
    WHERE("KEYDATA"='0001')
```

例 3

ルートレコード RECA をレコード型内の検索で検索します。

条件値に構成要素に対応する値を複数指定します。

```
FETCH FOR UPDATE FIRST "RECA" INTO :WRECA
    WHERE("KEYDATA"=('FS','aa'))
```

例 4

子レコード CHILD_RECB を親子集合型内の検索で検索します。

```
FETCH FOR UPDATE FIRST "CHILD_RECB" INTO :WCHILD_RECB
    WITHIN "SET_AB"
```

例 5

子レコード CHILD_RECB をレコード型内の検索で検索します。

```
FETCH FOR UPDATE FIRST "CHILD_RECB" INTO :WCHILD_RECB
```

例 6

DML コマンドでルートレコード RECA をレコード型内の検索で検索します。

```
FETCH FOR UPDATE FIRST "RECA"
    WHERE("KEYDATA"='0001')
```

例 7

DML コマンドで子レコード CHILD_RECB を親子集合型内の検索で検索します。

```
FETCH FOR UPDATE FIRST "CHILD_RECB"
    WITHIN OWNER
```

例 8

DML コマンドで子レコード CHILD_RECB をレコード型内の検索で検索します。

```
FETCH FOR UPDATE FIRST "CHILD_RECB"
```

17.4.3 FIND 文

(1) 機能

レコードを検索して，レコード実現値に位置指示子を位置づけます。

位置づけには，次の 3 つの検索があります。

- レコード型内の検索

  レコード位置指示子を使用した検索です。常にインデクスを使用した検索となります。
- 親子集合型内の検索

  親子集合位置指示子を使用した検索です。特定の親レコード実現値配下の子レコードを検索します。常にポインタを使用した検索となります。
- 位置指示子指定の検索

  DML に指定した位置指示子が指すレコードを検索します。

レコード型内の検索と親子集合型内の検索は，順序，開始位置，および検索方向を決定してレコードを検索します。詳細については，「17.3.3 検索の順序，開始位置および方向」を参照してください。

(2) 形式

- レコード型内の検索，親子集合型内の検索

```
FIND
      〔FOR UPDATE〕
      {FIRST | NEXT | LAST | PRIOR}
      レコード名
       〔INDEXED BY インデクス名〕
```

```
        〔WITHIN 親子集合型名〕
       〔WHERE（探索条件）〕
```

- 位置指示子指定の検索

```
FIND
       〔FOR UPDATE〕
       CURRENT OWNER OF 親子集合型名
```

(3)　オペランドの説明

●FOR UPDATE

位置づけたレコードを更新（ERASE，MODIFY）するかどうかを指定します。

位置づけたレコードを更新する場合は，FOR UPDATE を指定してください。

位置づけたレコードを更新しない場合は，FOR UPDATE を省略してください。

● {FIRST ｜ NEXT ｜ LAST ｜ PRIOR}

レコード走査の開始位置となるレコードとレコード走査の方向を示す指示コードを指定します。

FIRST：先頭から順方向に検索します。

NEXT：位置づけされているレコードから順方向に検索します。

LAST：最後から逆順方向に検索します。

PRIOR：位置づけされているレコードから逆順方向に検索します。

LAST および PRIOR は親子集合型内の検索のときだけ指定できます。

検索の開始位置，方向については，「17.3.3　検索の順序，開始位置および方向」を参照してください。

●レコード名

～<識別子>((1～30 バイト))

位置づけるレコード名を指定します。

注意事項

- レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- レコード名に空白またはハイフン（-）を使用している場合は，レコード名を引用符（"）で囲んでください。

●INDEXED BY インデクス名

～<識別子>((1～30 バイト))

検索で使用するインデクスを明示する場合に指定します。

レコード型内の検索で指定できます。インデクス名には，検索対象のレコードに定義されたインデクスの名称を指定してください。

注意事項

- インデクス名に英小文字を使用している場合は，インデクス名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- インデクス名に空白またはハイフン（-）を使用している場合は，インデクス名を引用符（"）で囲んでください。

●WITHIN 親子集合型名

～＜識別子＞((1～30 バイト))

親子集合型内の検索を行う場合に指定します。

検索する親子集合の親子集合型名を指定します。

注意事項

- 親子集合型名に英小文字を使用している場合は，親子集合型名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- 親子集合型名に空白またはハイフン（-）を使用している場合は，親子集合型名を引用符（"）で囲んでください。

●WHERE (探索条件)

位置づけるレコードの条件を指定します。探索条件が真になるレコードが位置づけ対象になります。探索条件は，レコード型内の位置づけの際に指定できます。

探索条件については，「17.3.1　探索条件の指定」を参照してください。

●CURRENT OWNER OF 親子集合型名

位置指示子指定の検索の場合に指定します。

検索対象の親子集合型名を指定します。

指定された親子集合型の位置指示子の親レコードを検索対象とし，関連する位置指示子を更新します。

(4)　共通規則

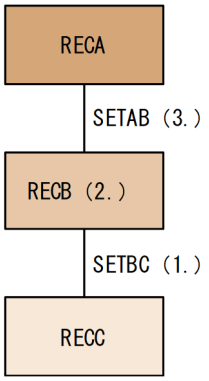
- HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の DML コマンドでは実行できません。
- レコード型内の位置づけは，インデクスを使用します。インデクスが定義されていないレコードは，レコード型内の位置づけは実行できません。インデクス選択規則については，「17.3.2　インデクス選択規則」を参照してください。
- 親子集合型内の検索の場合，検索対象のレコードの親レコードを次のどれかの DML で位置づけておく必要があります。
  - FIND 文
  - FETCH 文
  - STORE 文

- 位置指示子指定の検索で，検索対象レコードがルートレコードではない場合，SDB データベース格納定義の検索対象レコードが子レコードである親子集合の MEMBER POINTER FOR 句に OWNER を指定しておく必要があります。例を次に示します。

  DML

```
FIND CURRENT OWNER OF "SETBC"
```

  SDB データベース定義



  [説明]

  1. DML には親子集合 SETBC が指定されています。
  2. DML の検索対象レコードは RECB（SETBC の親レコード）です。
  3. RECB は親子集合 SETAB の子レコードとして定義されています。そのため，SETAB の SDB データベース格納定義の MEMBER POINTER FOR 句に OWNER を指定しておく必要があります。

- 位置指示子指定の検索の場合，指定した親子集合位置指示子の親レコードを次のどれかの DML で位置づけておく必要があります。
  - FIND 文
  - FETCH 文
  - STORE 文
- 位置指示子指定の検索を実行すると，対象レコードの検索状態は無効になります。

(5) 使用例

例 1

レコード型内の検索で，レコード実現値を更新するための位置づけを行います。

```
FIND FOR UPDATE FIRST "RECA"
    WHERE("KEYDATA"=:WKEYDATA)
```

例 2

レコード型内の検索で，レコードを検索するための位置づけを行います。

```
FIND FIRST "RECA" WHERE("KEYDATA"='0001')
```

例 3

親子集合型内の検索で，レコードを検索するための位置づけを行います。

```
FIND FIRST ”RECB” WITHIN SET1
```

例 4

位置指示子指定の検索例です。

ここでは，RECA（ルートレコード），RECB（子レコード），SETAB（RECA と RECB の親子集合型名）とします。

1. レコード型内の検索で，RECB（子レコード）の位置づけを行います。

   ```
   FIND FIRST ”RECB”
   ```

   この検索では，RECB のレコード位置指示子，SETAB の親子集合位置指示子の子および親が位置づけられます。

2. 位置指示子指定の検索で，RECA（親レコード）の位置づけを行います。

   ```
   FIND CURRENT OWNER OF ”SETAB”
   ```

3. GET 文で親レコード（RECA）のレコードデータを取得します。

   ```
   GET ”RECA” INTO :RECDATA
   ```

17.4.4 GET 文

(1) 機能

指定したレコードのレコード位置指示子が指すレコード実現値を取得します。

(2) 形式

```
GET
     レコード名
     INTO :埋込み変数
```

(3) オペランドの説明

●レコード名

～<識別子>((1～30 バイト))

レコード実現値を取得するレコード名を指定します。

注意事項

- レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。

- レコード名に空白またはハイフン（-）を使用している場合は，レコード名を引用符（"）で囲んでください。

●INTO :埋込み変数

レコード実現値を格納する埋込み変数を指定します。

埋込み変数の宣言方法については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「埋込み変数の宣言」を参照してください。

(4) 共通規則

- 検索対象のレコードを次のどれかの DML で位置づけておく必要があります。
  - FIND 文
  - FETCH 文
  - STORE 文
- HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の DML コマンドでは実行できません。

(5) 使用例

例

ルートレコード RECA に位置づいている位置指示子が指すレコード実現値を取得します。

```
GET ”RECA” INTO :WRECA
```

17.4.5 MODIFY 文

(1) 機能

レコード実現値を更新します。

(2) 形式

(a) UAP に記述する MODIFY 文

```
MODIFY
     レコード名
     FROM :埋込み変数
```

(b)　DML コマンドで記述する MODIFY 文

```
MODIFY
    レコード名
    SET 構成要素名=更新値〔,構成要素名=更新値〕…
```

(3)　オペランドの説明

●レコード名

～<識別子>((1～30 バイト))

レコード実現値を更新するレコード名を指定します。

注意事項

- レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- レコード名に空白またはハイフン（-）を使用している場合は，レコード名を引用符（"）で囲んでください。

●FROM :埋込み変数

更新対象のレコードの更新値を指定します。

埋込み変数の宣言方法については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の作成」の「埋込み変数の宣言」を参照してください。

●SET 構成要素名=更新値〔,構成要素名=更新値〕…

構成要素名：～<識別子>((1～30 バイト))

更新対象の構成要素名と更新値を指定します。

指定規則を次に示します。

- データ種別 1 が U の構成要素名を指定してください。
- 構成要素名は，更新する構成要素の個数分指定できます。
- 同じ構成要素名は指定できません。
- 構成要素名に英小文字を使用している場合は，構成要素名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- OCCURS 句を指定して生成された構成要素名を使用する場合は引用符（"）で囲んでください。OCCURS 句については，「11.7.1(4)(g)　OCCURS」を参照してください。
- 構成要素のデータ型に対応する更新値を指定してください。更新値の指定形式については，「17.2.2 DML のデータ型」を参照してください。

(4)　共通規則

- 更新対象のレコードを，次のどれかの DML で位置づけておく必要があります。

- FIND 文（位置決め目的に FOR UPDATE を指定する）
- FETCH 文（位置決め目的に FOR UPDATE を指定する）
- STORE 文

- 埋込み変数で指定したレコードデータのうち，ルートレコードのデータベースキーの構成要素，および一連番号として定義した構成要素の値は変更できません。変更していても変更後のキー値に対応するレコードではなく，位置づいているレコードが更新されます。
- 埋込み変数で指定したレコードデータのうち，ルートレコードのデータベースキーの構成要素，および一連番号の構成要素に対応するデータは無視されます。
- キー項目を更新した場合，更新した値に対応してレコードの並び順を変更します。レコード順の並びが変更された場合，親子集合位置指示子を更新します。更新内容については，「2.5.3　位置指示子」を参照してください。
- SDB データベースの横分割を指定したレコードの更新で，二次インデクスの RD エリア分割キーを更新した場合，RD エリアの移動を伴う更新はできません。

(5)　使用例

例 1

ルートレコード RECA のレコード実現値の値を更新します。

```
MODIFY ”RECA” FROM :WRECA
```

例 2

DML コマンドでルートレコード RECA のレコード実現値の値を更新します。

```
MODIFY ”RECA” SET ”USERDA0”=200
```

17.4.6　STORE 文

(1)　機能

レコード実現値を格納します。

(2)　形式

(a)　UAP に記述する STORE 文

```
STORE
    レコード名
    FROM :埋込み変数
```

(b)　DML コマンドで記述する STORE 文

```
STORE
    レコード名
    SET 構成要素名=格納値〔,構成要素名=格納値〕…
```

(3)　オペランドの説明

●レコード名

～<識別子>((1～30 バイト))

レコード実現値を格納するレコード名を指定します。

注意事項

- レコード名に英小文字を使用している場合は，レコード名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- レコード名に空白またはハイフン（-）を使用している場合は，レコード名を引用符（"）で囲んでください。

●FROM :埋込み変数

格納対象のレコードの格納値を指定します。

埋込み変数の宣言方法については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の作成」の「埋込み変数の宣言」を参照してください。

●SET 構成要素名=格納値〔,構成要素名=格納値〕…

構成要素名：～<識別子>((1～30 バイト))

格納するレコードの値を指定します。

指定規則を次に示します。

- 最下位レベルの構成要素（基本項目）に対する格納値を指定してください。
- 構成要素名は，レコードの定義で指定した順に指定してください。
- 指定が必要な構成要素をすべて指定してください。
  ・ルートレコードの場合は基本項目をすべて指定してください。
  ・子レコードの場合はデータ種別 1 が U の基本項目をすべて指定してください。
- 指定が必要な構成要素については，#USAGE コマンドの実行結果で確認できます。#USAGE コマンドの実行結果に表示されたひな形の SET オペランドに出力されている構成要素を，出力されている順にすべて指定してください。#USAGE コマンドについては，「14.8.3　#USAGE（DML コマンドのひな形の出力）」を参照してください。
- 構成要素名に英小文字を使用している場合は，構成要素名を引用符（"）で囲んでください。引用符で囲まないと，英小文字は英大文字と見なされます。
- OCCURS 句を指定して生成された構成要素名を使用する場合は引用符（"）で囲んでください。OCCURS 句については，「11.7.1(4)(g)　OCCURS」を参照してください。

- 格納値の指定形式については，「17.2.2　DML のデータ型」を参照してください。

(4)　共通規則

- 子レコードにレコードを格納する場合，次のどれかの DML で，格納レコードの親レコードまでの位置づけをしておく必要があります。
  - 格納レコードの親レコードに対する FIND 文
  - 格納レコードの親レコードに対する FETCH 文
  - 格納レコードの親レコードに対する STORE 文
- 埋込み変数を指定して，子レコードにレコードを格納する場合，埋込み変数に設定した値のうち，親レコードから引き継いだ構成要素に対応する値は無視されます。この場合，位置づいている親レコードと同じ値が格納されます。
- 埋込み変数を指定して，子レコードにレコードを格納する場合，埋込み変数に設定した値に対応する親レコードの子レコードとして格納されるのではなく，位置づいている親レコードの子レコードとして格納されます。
- 横分割しているレコードに二次インデクスが定義されている場合，レコードが格納される RD エリアと，二次インデクスが格納される RD エリアが対応しているかどうかを STORE 文の実行時にチェックします。具体的には，レコードと二次インデクスの格納先が定義上の何番目の RD エリアなのかを求めて，レコードと二次インデクスが定義上の同じ位置に格納されることをチェックします。

  なお，子レコードに定義した二次インデクスの RD エリア分割キーにキー（SDB データベース定義の TYPE 句のデータ種別 1 に K を指定した構成要素）を指定した場合，レコードが格納される RD エリアと，二次インデクスが格納される RD エリアが対応しているかどうかのチェックは，ルートレコードの格納時に行われます。

■レコードと二次インデクスが格納される RD エリアの対応をチェックする例

＜スキーマ定義＞

```
RECORD RECA                                    ...1
  2 KEY1
    3 KEYDATA1      CHARACTER  4 TYPE  K,A     ...2
    3 KEYDATA2      CHARACTER  2 TYPE  K,L
  2 UDA01           CHARACTER  2 TYPE  U,D
RECORD RECB                                    ...3
  2 KEYDATA1        CHARACTER  4 TYPE  K,A
  2 KEYDATA2        CHARACTER  2 TYPE  K,L
  2 UDB01           CHARACTER  2 TYPE  U,D
  2 UDB02           CHARACTER  2 TYPE  U,D
```

[説明]

1. ルートレコードの定義です。
2. レコード型の RD エリア分割キーの定義です。
3. 子レコードの定義です。

＜格納スキーマ定義＞

```
STORAGE RECORD RECA
  SEQUENTIAL IDX1 USED FOR RECORD                                    ...1
  DEPENDING ON KEYDATA1                                              ...2
  WITHIN ((RDAREAD01) '0100',(RDAREAD02) '0200', (RDAREAD03))
STORAGE RECORD RECB
  CLUSTERD SET_AB
  WITHIN ((RDAREAD01),(RDAREAD02),(RDAREAD03))
  SECONDARY INDEX IDX2 USED FOR RECORD
   ORDER KEY ASCENDING KEYDATA2
   DEPENDING ON KEYDATA2                                             ...3
   WITHIN((RDAREAI01) 'BB',(RDAREAI02) 'CC', (RDAREAI03))
  SECONDARY INDEX IDX3 USED FOR RECORD
   ORDER KEY ASCENDING UDB02
   DEPENDING ON UDB02                                               ....4
   WITHIN((RDAREAI01) 'XX',(RDAREAI02) 'YY', (RDAREAI03))
```

[説明]

1. シーケンシャルインデクスは省略します。
2. ルートレコードの分割を定義します。
3. 二次インデクスの分割を定義します。

   二次インデクスの RD エリア分割キーに TYPE K，L の構成要素を指定します。
4. 二次インデクスの分割を定義します。

   二次インデクスの RD エリア分割キーに TYPE U，D の構成要素を指定します。

上記の定義によって，次のように 3 つの範囲に分割されます。

```
最初の範囲： (KEYDATA1,KEYDATA2,UDB02)=(      ～'0100',      ～'BB',      ～'XX')
2番目の範囲：(KEYDATA1,KEYDATA2,UDB02)=('0101'～'0200', 'BC'～'CC', 'XY'～'YY')
最後の範囲： (KEYDATA1,KEYDATA2,UDB02)=('0201'～      , 'CD'～     , 'YZ'～    )
```

＜ルートレコードの格納時に子レコードの二次インデクスをチェックする例＞

二次インデクスの RD エリア分割キーにキーを指定している二次インデクス（IDX2）について，格納される RD エリアの対応をチェックする例です。

(例 1)

```
STORE RECA SET KEYDATA1='0090',KEYDATA2='AA',UDA01='00'
```

[説明]

レコードが格納される RD エリアは，'0090'が「最初の範囲」に該当するため，RDAREAD01（定義上の 1 番目の RD エリア）となります。二次インデクス（IDX2）が格納される RD エリアは，'AA'が「最初の範囲」に該当するため，RDAREAI01（定義上の 1 番目の RD エリア）となります。二次インデクス（IDX2）が格納される RD エリアは，RECB の KEYDATA2 に対応する RECA の KEYDATA2 の値によって取得されます。

レコードが格納される RD エリアと，二次インデクスが格納される RD エリアの両方とも定義上 1 番目の RD エリアとなり対応しているため，STORE 文は正常終了します。

(例 2)

```
STORE RECA SET KEYDATA1='0090',KEYDATA2='BC',UDA01='00'
```

[説明]

レコードが格納されるRDエリアは，'0090'が「最初の範囲」に該当するため，RDAREAD01（定義上の1番目のRDエリア）となります。二次インデクス（IDX2）が格納されるRDエリアは，'BC'が「2番目の範囲」に該当するため，RDAREAI02（定義上の2番目のRDエリア）となります。

レコードが格納されるRDエリアと，二次インデクスが格納されるRDエリアの定義上の位置が対応していないため，STORE文はエラーになります。

＜子レコードの格納時に子レコードの二次インデクスをチェックする例＞

二次インデクスのRDエリア分割キーにキーを指定していない二次インデクス（IDX3）について，格納されるRDエリアの対応をチェックする例です。この例では，ルートレコードはRDAREAD01（定義上の1番目のRDエリア）に格納されているものとします。

(例 1)

```
STORE RECB SET UDB01='10', UDB02='WW'
```

[説明]

二次インデクス（IDX3）が格納されるRDエリアは，'WW'が「最初の範囲」に該当するため，RDAREAI01（定義上の1番目のRDエリア）となります。レコードが格納されるRDエリアと，二次インデクスが格納されるRDエリアの両方とも定義上1番目のRDエリアとなり対応しているため，STORE文は正常終了します。

(例 2)

```
STORE RECB SET UDB01='10', UDB02='XY'
```

[説明]

二次インデクス（IDX3）が格納されるRDエリアは，'XY'が「2番目の範囲」に該当するため，RDAREAI02（定義上の2番目のRDエリア）となります。レコードが格納されるRDエリアと，二次インデクスが格納されるRDエリアの定義上の位置が対応していないため，STORE文はエラーになります。

(5) 使用例

例 1

ルートレコードRECAにレコードを格納します。

```
STORE "RECA" FROM :WRECA
```

例 2

DMLコマンドでルートレコードRECAにレコードを格納します。

```
STORE "RECA" SET "KEYDATA1"=X'01020304',"USERDA0"=200,"USERDB0"=100.2
```

17.5　埋込み言語文法

埋込み言語の機能，構文形式，および規則について説明します。

埋込み言語は，埋込み型の UAP を作成する際に，プログラム用 DML と一緒に使用する DML です。埋込み言語には，次の 2 つがあります。

- DML 先頭子
- DML 終了子

17.5.1　DML 先頭子

(1)　機能

DML の始まりを示します。

(2)　形式

```
EXEC DML
```

(3)　規則

- 各 DML を，DML 先頭子と DML 終了子で囲む必要があります。
- UAP 中に記述する DML については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の作成」を参照してください。

(4)　使用例

例

FETCH 文を DML 先頭子と DML 終了子で囲みます。

```
EXEC DML
    FETCH FIRST ”RECA” INTO :RECA_DATA
END-DML.
```

17.5.2　DML 終了子

(1)　機能

DML の終わりを示します。

(2) 形式

```
END-DML
```

(3) 規則

- 各 DML を，DML 先頭子と DML 終了子で囲む必要があります。
- UAP 中に記述する DML については，マニュアル「HiRDB 構造型データベース機能（UAP 開発編）」の「UAP の作成」を参照してください。

(4) 使用例

例

FETCH 文を DML 先頭子と DML 終了子で囲みます。

```
EXEC DML
    FETCH FIRST ”RECA” INTO :RECA_DATA
END-DML.
```

18 メッセージ

この章では，HiRDB が出力するメッセージについて説明します。

18.1 HiRDB が出力するメッセージ

HiRDB が出力するメッセージについては，マニュアル「HiRDB メッセージ」を参照してください。

なお，マニュアル「HiRDB メッセージ」を初めて参照する場合は，「HiRDB メッセージ」の次の個所をお読みください。

- 「メッセージの概要」の「メッセージの記述形式」
- 「メッセージの概要」の「HiRDB Structured Data Access Facility を使用している場合の留意事項」

付録

付録 A　未サポート機能の一覧

HiRDB のマニュアルに記載している機能のうち，未サポートの機能，および機能制限がある機能を次の表に示します。

表 A-1　未サポートの機能，および機能制限がある機能

項番	分類	サポートしていない，または機能制限のある機能・コマンド・インタフェース，もしくは連携できない製品	説明
1	環境設定	簡易セットアップツール	簡易セットアップツールは使用できません。
2		プログラムメンテナンス機能	単体版 HiRDB から統合版 HiRDB への入れ替え，または統合版 HiRDB から単体版 HiRDB への入れ替えには使用できません。
3	運用	インメモリデータ処理（バッチ高速化機能）	SDB データベースへのアクセスでは，バッチ高速化機能は使用できません。
4		系切り替え機能	使用できるクラスタソフトウェアは HA モニタだけです。 また，次に示す系切り替え機能は使用できません。 **系切り替えの種類** • スタンバイ型系切り替え機能 ユーザサーバホットスタンバイ • スタンバイレス型系切り替え機能 1：1 スタンバイレス型系切り替え 影響分散スタンバイレス型系切り替え **系切り替えの形態** 連動系切り替え **系切り替えの構成** • LifeKeeper と DataKeeper による複製ディスクを使用した系切り替え構成 • HA モニタと DRBD による複製ディスクを使用した系切り替え構成 ただし，Amazon EC2 環境での DRBD は使用可能 • HA モニタと複製ディスク機能を使用した系切り替え構成
5		リアルタイム SAN レプリケーション（ディザスタリカバリシステム）	次に示す処理方式が使用できます。 • 全非同期方式
6		アクセス権限	SDB データベースに対して，アクセス権限は使用できません。
7		OS ログインユーザの簡易認証機能	次に示すコマンドには，OS ログインユーザの簡易認証機能を使用できません。 • pdsdbdef • pdsdblod

項番	分類	サポートしていない，または機能制限のある機能・コマンド・インタフェース，もしくは連携できない製品	説明
			• pdsdbrog • pdsdbarc
8		HiRDB 管理者以外の OS ログインユーザからの，HiRDB/SD の運用コマンドおよびユティリティの実行	HiRDB 管理者以外の OS ログインユーザは，次に示す HiRDB/SD の運用コマンドおよびユティリティを実行できません。 • pdsdbarc • pdsdborcrt • pdsdbdef • pdsdblod • pdsdbexe • pdsdbrog
9		共用 RD エリア	SDB データベースを格納する RD エリアとして，共用 RD エリアは使用できません。
10		インデクス用グローバルバッファ	SDB データベースのインデクスに対して，インデクス用グローバルバッファは使用できません。
11		LOB 用グローバルバッファ	SDB データベースを格納する RD エリアに対して，LOB 用グローバルバッファは使用できません。
12		ローカルバッファ • データ用ローカルバッファ • インデクス用ローカルバッファ	SDB データベースを格納する RD エリア，および SDB データベースのインデクスに対して，ローカルバッファは使用できません。
13		プリフェッチ機能	SDB データベースへのアクセスに対して，プリフェッチ機能は使用できません。
14		非同期 READ 機能	SDB データベースへのアクセスに対して，非同期 READ 機能は使用できません。
15		MIB パフォーマンス情報監視機能	MIB パフォーマンス情報監視機能で構造型 DB 機能に関する情報は収集できません。
16		UAP 環境定義	SDB データベースにアクセスする場合，次の UAP 環境定義のオペランドは指定できません。 • pd_uap_wait • pdlbuffer
17		回復不要 FES	SDB データベースへのアクセスと併用できません。
18		JP1 との連携	SDB データベースは JP1 とは連携できません。
19		Cosminexus との連携	SDB データベースは Cosminexus とは連携できません。
20		データベース暗号化機能	SDB データベースに対して，暗号化機能は使用できません。

項番	分類	サポートしていない，または機能制限のある機能・コマンド・インタフェース，もしくは連携できない製品	説明
21	運用コマンド・ユティリティ	運用コマンド，ユティリティの制限については，「10.1　運用コマンドおよびユティリティの実行方法と実行可否」を参照してください。	
22	UAP の開発および実行	HiRDB OLE DB プロバイダ	HiRDB OLE DB プロバイダを使用して，SDB データベースにアクセスできません。
23		HiRDB.NET データプロバイダ（ADO.NET2.0 対応）	HiRDB.NET データプロバイダを使用して，SDB データベースにアクセスできません。
24		SQLJ	SQLJ を使用した UAP からは，SDB データベースにアクセスできません。
25		ODBC ドライバ	ODBC ドライバを使用して，SDB データベースにアクセスできません。
26		JDBC ドライバ	JDBC ドライバを使用して，SDB データベースにアクセスできません。
27		クライアント環境定義	クライアント環境定義の指定可否については，「16.3.1(3)　クライアント環境定義の一覧」を参照してください。
28		ストアドプロシジャ，ストアドファンクション	SDB データベースにアクセスする場合，ストアドプロシジャ，ストアドファンクションは使用できません。
29		Java ストアドプロシジャ，Java ストアドファンクション	SDB データベースにアクセスする場合，Java ストアドプロシジャ，Java ストアドファンクションは使用できません。
30		C ストアドプロシジャ，C ストアドファンクション	SDB データベースにアクセスする場合，C ストアドプロシジャ，C ストアドファンクションは使用できません。
31		埋込み型 UAP	4V FMB または 4V AFM の SDB データベースにアクセスする場合，埋込み型 UAP は作成できません。
32		インタフェース領域 • SQL 記述領域 • 列名記述領域 • 型名記述領域 • 文字集合名記述領域	SDB データベースにアクセスする場合，左記のインタフェース領域は使用しません。
33		空白変換機能	SDB データベースにアクセスする場合，空白変換機能は使用できません。
34		ブロック転送機能	SDB データベースにアクセスする場合，ブロック転送機能は使用できません。
35		スナップショット方式によるページアクセス	SDB データベースに対しては，スナップショット方式によるページアクセスは適用されません。

項番	分類	サポートしていない，または機能制限のある機能・コマンド・インタフェース，もしくは連携できない製品	説明
			更新可能なオンライン再編成の追い付き反映キー対応表には，スナップショット方式によるページアクセスが適用されます。
36		メモリ DB	SDB データベースにアクセスする場合，メモリ DB は使用できません。
37		XDS クライアント	XDS クライアントから SDB データベースにアクセスできません。
38		拡張 SQL エラー情報出力機能	拡張 SQL エラー出力機能を使用して，API または DML 関連情報が出力されます。 ただし，クライアント側のクライアントエラーログファイルに，API または DML の要求内容がダンプで出力されますが，サーバ側の SQL エラーレポートファイルには，API または DML の要求内容のダンプは出力されません。
39		データ保証レベル	SDB データベースにアクセスする場合，データ保証レベルの指定は無効になります。
40		UAP からのコマンド実行	次に示すコマンドは UAP から実行できません。 • pdsdbdef • pdsdblod • pdsdbrog • pdsdbexe • pdsdbarc • pdsdborcrt
41		インデクスキー値排他	インデクスキー値排他機能は使用できません。
42		残存エントリによる排他待ちの回避（行識別子の再利用抑止）	SDB データベースにアクセスする場合，残存エントリによる排他待ちの回避（行識別子の再利用抑止）機能は使用できません。
43	関連プログラムプロダクト	HiRDB Control Manager	SDB データベースを運用管理の対象にできません。
44		HiRDB RealTime Monitor	HiRDB RealTime Monitor では SDB データベースに関する情報は出力されません。
45		HiRDB SQL Executer	定義系 SQL などを実行する際に使用します。
46		HiRDB SQL Tuning Advisor	SQL Tuning Advisor では SDB データベースへのアクセスに関する情報は出力されません。
47		HiRDB SQL Tuning Advisor 用アクセスパス情報ファイル	SDB データベースへのアクセスに関する情報は出力されません。
48		HiRDB Dataextractor	SDB データベースは HiRDB Dataextractor とは連携できません。

項番	分類	サポートしていない，または機能制限のある機能・コマンド・インタフェース，もしくは連携できない製品	説明
49		HiRDB Datareplicator	SDB データベースは HiRDB Datareplicator とは連携できません。
50		HiRDB の各種プラグイン製品	SDB データベースは HiRDB の各種プラグイン製品とは連携できません。
51		HiRDB Adapter for XML	SDB データベースは HiRDB Adapter for XML とは連携できません。
52		JP1/Performance Management - Agent Option for HiRDB	SDB データベースに関するパフォーマンスデータは収集できません。
53		DABroker	SDB データベースは DABroker とは連携できません。
54		DATAFRONT	SDB データベースは DATAFRONT とは連携できません。
55		DBPARTNER2	SDB データベースは DBPARTNER2 とは連携できません。
56		Image Database Access	SDB データベースは Image Database Access とは連携できません。
57		DocumentBroker	SDB データベースは DocumentBroker とは連携できません。
58		HiRDB Configuration Assistant	SDB データベースの環境構築はできません。

付録 B　ディクショナリ表

ディクショナリ表には，レコード型やインデクスの定義情報など，SDB データベースの定義情報が表形式で格納されています。

付録 B.1　ディクショナリ表の一覧

ディクショナリ表の一覧を次の表に示します。

表 B-1　ディクショナリ表の一覧

項番	ディクショナリ表	説明
1	SQL_SDB_DB 表	SDB データベース定義の定義情報が格納されています。
2	SQL_SDB_STORAGE_DB 表	SDB データベース格納定義の定義情報が格納されています。
3	SQL_SDB_DIR 表	SDB ディレクトリ情報が格納されています。
4	SQL_RDAREAS 表	RD エリアの定義情報が格納されています。
5	SQL_TABLES 表	スキーマ内のレコード型の情報が格納されています。
6	SQL_COLUMNS 表	レコード型の構成要素の情報が格納されています。
7	SQL_INDEXES 表	インデクスの定義情報が格納されています。
8	SQL_TABLE_PRIVILEGES 表	レコード型の所有者の情報が格納されています。
9	SQL_DIV_TABLE 表	SDB データベースの横分割の情報が格納されています。
10	SQL_INDEX_COLINF 表	インデクスを構成する構成要素名の情報が格納されています。
11	SQL_DIV_INDEX 表	インデクスの横分割の情報が格納されています。
12	SQL_AUDITS 表	監査対象の情報が格納されています。
13	SQL_DIV_TABLE_REGULARIZE 表	システムで使用する内部情報が格納されています。
14	SQL_SDB_DB_VIEW 表	システムで使用する内部情報が格納されています。
15	SQL_SDB_NAME_CODE 表	システムで使用する内部情報が格納されています。

なお，上記以外のディクショナリ表については，マニュアル「HiRDB UAP 開発ガイド」の「ディクショナリ表の詳細」を参照してください。

付録 B.2　ディクショナリ表の詳細

ディクショナリ表に格納されている情報について説明します。

なお，各ディクショナリ表に VARCHAR または MVARCHAR のデータ型の列がありますが，これはデータベース初期設定ユティリティ（pdinit），またはデータベース構成変更ユティリティ（pdmod）の dictionary datatype オペランドで，データ型をどちらにするか設定してください。

(1) SQL_SDB_DB 表

SQL_SDB_DB 表には，SDB データベース定義の定義情報が格納されています。1 行に 1SDB データベース定義の情報が格納されています。SQL_SDB_DB 表に格納されている情報を次の表に示します。

表 B-2 SQL_SDB_DB 表に格納されている情報

項番	列名	データ型	内容
1	DB_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	SDB データベースの所有者
2	DB_NAME	VARCHAR(30)，または MVARCHAR(30)	SDB データベース名
3	DB_ID	SMALLINT	SDB データベースの ID SDB データベース名ごとに HiRDB/SD が一意の管理番号を設定します。
4	CREATE_TIME	CHAR(16)	SDB データベース定義の作成日時（YYYYMMDDHHMMSSTH）
5	CHANGE_TIME	CHAR(16)	SDB データベース定義の変更日時（YYYYMMDDHHMMSSTH）
6	DEF_INF_LEN	INTEGER	SDB データベースの定義情報長
7	DEF_INF	BINARY	システムで使用する情報

(2) SQL_SDB_STORAGE_DB 表

SQL_SDB_STORAGE_DB 表には，SDB データベース格納定義の定義情報が格納されています。1 行に 1SDB データベース格納定義の情報が格納されています。SQL_SDB_STORAGE_DB 表に格納されている情報を次の表に示します。

表 B-3 SQL_SDB_STORAGE_DB 表に格納されている情報

項番	列名	データ型	内容
1	STORAGE_DB_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	SDB データベース格納定義の所有者
2	STORAGE_DB_NAME	VARCHAR(30)，または MVARCHAR(30)	SDB データベース格納名
3	DB_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	SDB データベースの所有者

項番	列名	データ型	内容
4	DB_NAME	VARCHAR(30)，または MVARCHAR(30)	SDB データベース名
5	STORAGE_DB_ID	SMALLINT	SDB データベース格納定義の ID SDB データベース格納名ごとに HiRDB/SD が一意の管理番号を設定します。
6	DB_ID	SMALLINT	SDB データベースの ID SDB データベース名ごとに HiRDB/SD が一意の管理番号を設定します。
7	CREATE_TIME	CHAR(16)	SDB データベース格納定義の作成日時（YYYYMMDDHHMMSSTH）
8	CHANGE_TIME	CHAR(16)	SDB データベース格納定義の変更日時（YYYYMMDDHHMMSSTH）
9	CHECK_TIME	CHAR(16)	SDB データベース格納定義の整合性チェック日時（YYYYMMDDHHMMSSTH） 整合性チェックをしていない場合，ナル値となります。
10	DEF_INF_LEN	INTEGER	SDB データベース格納定義の定義情報長
11	DEF_INF	BINARY	システムで使用する情報

(3) SQL_SDB_DIR 表

SQL_SDB_DIR 表には，SDB ディレクトリ情報が格納されています。1 行に 1SDB ディレクトリ情報（1SDB データベース情報または 1SDB データベースビュー情報）が格納されています。SQL_SDB_DIR 表に格納されている情報を次の表に示します。

表 B-4 SQL_SDB_DIR 表に格納されている情報

項番	列名	データ型	内容
1	DIR_INF_KIND	SMALLNT	その行に格納される SDB ディレクトリ情報種別 1：SDB データベース情報 2：SDB データベースビュー情報
2	DB_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	データベースの所有者
3	DB_NAME	VARCHAR(30)，または MVARCHAR(30)	SDB データベース名
4	DB_VIEW_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	SDB データベースビューの所有者 DIR_INF_KIND 列の値が 1（SDB データベース情報）の場合は，ナル値となります。
5	DB_VIEW_NAME	VARCHAR(30)，または MVARCHAR(30)	SDB データベースビュー名

項番	列名	データ型	内容
			DIR_INF_KIND 列の値が 1（SDB データベース情報）の場合は，ナル値となります。
6	DB_ID	SMALLNT	SDB データベースの ID SDB データベース名ごとに HiRDB/SD が一意の管理番号を設定します。
7	DB_VIEW_ID	SMALLNT	SDB データベースビュー ID SDB データベースビュー名ごとに HiRDB/SD が一意の管理番号を設定します。 DIR_INF_KIND 列の値が 1（SDB データベース情報）の場合は，ナル値となります。
8	CREATE_TIME	CHAR(16)	SDB ディレクトリ情報の作成日時（YYYYMMDDHHMMSSTH）
9	CHANGE_TIME	CHAR(16)	SDB ディレクトリ情報の更新日時（YYYYMMDDHHMMSSTH）
10	DIR_INF_LEN	INTEGER	SDB ディレクトリ情報長
11	DIR_INF	BINARY	システムで使用する情報

(4) SQL_RDAREAS 表

SQL_RDAREAS 表には，RD エリアの定義情報が格納されています。1 行に 1RD エリアの定義情報が格納されています。SQL_RDAREAS 表に格納されている情報を次の表に示します。

表 B-5 SQL_RDAREAS 表に格納されている情報

項番	列名	データ型	内容
7	N_TABLE	INTEGER	格納レコード型数（定義数） SDB データベースの定義または削除時に，格納レコード型数が更新されます。
8	N_INDEX	INTEGER	格納インデクス数（定義数） SDB データベースの定義時に，格納インデクス数が加算されます。 SDB データベースの削除時に，格納インデクス数からレコード型に定義されているインデクス数分が減算されます。
21	DATA_MODEL	CHAR(1)	データベースの形式 'S'：構造型 NULL：リレーショナル 次の場合はナル値となります。 • ユーザ用 RD エリア以外 • data model オペランド未指定時

項番	列名	データ型	内容
			• data model オペランドに relational を指定した場合

注

- 上記の表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_RDAREAS 表の内容」の表の項番と対応しています。
- 上記の表に記載されている以外の SQL_RDAREAS 表の列情報については，マニュアル「HiRDB UAP 開発ガイド」の「SQL_RDAREAS 表の内容」を参照してください。

(5) SQL_TABLES 表

SQL_TABLES 表には，スキーマ内のレコード型の情報が格納されています。1 行に 1 レコード型の定義情報が格納されています。SQL_TABLES 表に格納されている情報を次の表に示します。

表 B-6 SQL_TABLES 表に格納されている情報

項番	列名	データ型	内容
1	TABLE_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
2	TABLE_NAME	VARCHAR(30)，または MVARCHAR(30)	レコード型名 SDB データベース定義の RECORD 句に指定したレコード型名が格納されます。
3	TABLE_TYPE	CHAR(16)	レコード型のタイプ SDB RECORD TYPE：構造型 DB のレコード型
4	TABLE_ID	INTEGER	レコード型 ID SDB データベース定義の RECORDID 句の指定に関係なく，システムが設定したユニークな内部 ID が格納されます。
5	N_COLS	SMALLINT	構成要素数 SDB データベース定義の RECORD 句に指定した構成要素数が格納されます。 データ属性を持たない構成要素は，構成要素数に含まれません。 SDB データベース定義に OCCURS 句を指定した場合，繰り返し回数分生成された構成要素の数が加算されます。
6	N_INDEX	SMALLINT	インデクス定義数 • レコード型定義時 SDB データベース格納定義の SEQUENTIAL 句に指定したインデクス個数となります。 • インデクス定義時

項番	列名	データ型	内容
			インデクス定義数が更新（+1）されます。
7	DCOLUMN_NAME	VARCHAR(30)，またはMVARCHAR(30)	分割列名 SDB データベースを横分割している場合は，レコード型の RD エリア分割キーの構成要素名となります。SDB データベースを横分割していない場合は，ナル値となります。
9	FREE_AREA	SMALLINT	ページ内の未使用領域の比率（%） 0～99 となります。
10	FREE_PAGE	SMALLINT	セグメント内の空きページ（未使用ページ）の比率（%） 0～50 となります。
12	CREATE_TIME	CHAR(14)	レコード型作成時刻（YYYYMMDDHHMMSS）
13	ENQ_RESOURCE_SIZE	CHAR(1)	排他資源単位 P：ページ単位
14	DEFAULT_COLUMN	SMALLINT	既定値（DEFAULT 句または WITH DEFAULT）の指定列数 ディクショナリ表の場合はナル値となります。それ以外は 0 となります。
15	RDAREA_NAME	VARCHAR(30)，またはMVARCHAR(30)	レコードを格納している RD エリア名 SDB データベースを横分割している場合は，ナル値となります。SDB データベースを横分割していない場合は，SDB データベース格納定義の WITHIN 句に指定した RD エリア名となります。
18	N_RDAREA	INTEGER	格納先 RD エリア数 1～1,024 となります。
23	N_NOTNULL	INTEGER	構成要素数 SDB データベース定義の RECORD 句に指定した構成要素数となります。 ディクショナリ表の場合は 0 となります。 データ属性を持たない構成要素は，構成要素数に含まれません。 SDB データベース定義に OCCURS 句を指定した場合，繰り返し回数分生成された構成要素の数が加算されます。
25	DIV_TYPE	CHAR(1)	SDB データベースの横分割の種別 境界値分割の場合は'P'，それ以外はナル値となります。
29	CHANGE_TIME	CHAR(14)	レコード型の定義の変更時刻（YYYYMMDDHHMMSS）

項番	列名	データ型	内容
			インデクスを定義している場合はインデクスを定義した時間，インデクスを定義していない場合はナル値となります。

注

- 上記の表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_TABLES 表の内容」の表の項番と対応しています。
- 上記の表に記載されている以外の SQL_TABLES 表の列情報については，マニュアル「HiRDB UAP 開発ガイド」の「SQL_TABLES 表の内容」を参照してください。

(6) SQL_COLUMNS 表

SQL_COLUMNS 表には，レコード型の構成要素の情報が格納されています。1 行にレコード型の 1 構成要素の定義情報が格納されています。SQL_COLUMNS 表に格納されている情報を次の表に示します。

表 B-7 SQL_COLUMNS 表に格納されている情報

項番	列名	データ型	内容
1	TABLE_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
2	TABLE_NAME	VARCHAR(30)，または MVARCHAR(30)	構成要素を含むレコード型名 SDB データベース定義の RECORD 句に指定したレコード型名が格納されます。
3	COLUMN_NAME	VARCHAR(30)，または MVARCHAR(30)	構成要素名 SDB データベース定義の RECORD 句に指定した構成要素名が格納されます。 なお，次の名称はシステムが使用する構成要素名です。 • SYSTEM_FLAG_FIELD • SYSTEM_RECTYPE_NO また，SDB データベース定義に OCCURS 句を指定した場合，繰り返し回数分生成された構成要素名が格納されます。
4	TABLE_ID	INTEGER	レコード型 ID SDB データベース定義の RECORDID 句の指定に関係なく，システムが設定したユニークな内部 ID となります。
5	COLUMN_ID	SMALLINT	構成要素 ID 1 から始まる正の整数です。 SDB データベース定義に OCCURS 句を指定した場合，繰り返し回数分生成された構成要素に対する ID となります。

項番	列名	データ型	内容
6	DATA_TYPE	CHAR(24)	データ型 SDB データベース定義の RECORD 句に指定した各構成要素のデータ型が格納されます。 データ型によって格納する値は次のように異なります。 • CHARACTER，XCHARACTER：CHAR • PACKED DECIMAL FIXED：DECIMAL • INTEGER：INTEGER • SMALLINT：SMALLINT
7	DATA_LENGTH	CHAR(7)	列データ長 SDB データベース定義の RECORD 句に指定した文字列データの長さが格納されます。 データ型によって格納する値は次のように異なります。 PACKED DECIMAL FIXED 型の場合：(a,b)の形式で文字列にして格納します（ただし，a：整数部桁数＋小数部桁数，b：小数部桁数。上位の 0 は空白にします）。 それ以外の場合：SQL 記述領域（SQLDA）の SQLLEN と同じ値を右詰めで文字形式にして格納します（上位の 0 は空白にします）。
9	DIVIDED_KEY	CHAR(1)	レコード型の RD エリア分割キー 'Y'：レコード型の RD エリア分割キー（データ種別 1，2 の指定が K，A または K，M の構成要素） '△'：レコード型の RD エリア分割キーでない
22	DATA_TYPE_CODE	SMALLINT	データ型コード データ型によって格納する値は次のように異なります。 • CHARACTER，XCHARACTER：C4 • PACKED DECIMAL FIXED：E4 • INTEGER：F0 • SMALLINT：F4
23	DATA_LENGTH_CODE	SMALLINT	列データ長コード SDB データベース定義の RECORD 句に指定した長さを変換した値です。 SQL 記述領域（SQLDA）の SQLLEN と同じ値を格納します。データ型によって格納する値は次のように異なります。 • DECIMAL 型の場合：精度，位取りをそれぞれ 1 バイトに格納します。 • それ以外の場合：2 バイトの 2 進形式で長さを格納します。

注

- 上記の表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_COLUMNS 表の内容」の表の項番と対応しています。
- 上記の表に記載されている以外の SQL_COLUMNS 表の列情報については，マニュアル「HiRDB UAP 開発ガイド」の「SQL_COLUMNS 表の内容」を参照してください。

(7) SQL_INDEXES 表

SQL_INDEXES 表には，インデクスの定義情報が格納されています。1 行に 1 インデクスの定義情報が格納されています。SQL_INDEXES 表に格納されている情報を次の表に示します。

表 B-8 SQL_INDEXES 表に格納されている情報

項番	列名	データ型	内容
1	TABLE_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
2	TABLE_NAME	VARCHAR(30)，または MVARCHAR(30)	インデクスを含むレコード型名 SDB データベース定義の RECORD 句に指定したレコード型名
3	INDEX_NAME	VARCHAR(30)，または MVARCHAR(30)	インデクス名称 SDB データベース格納定義の SEQUENTIAL 句に指定したシーケンシャルインデクス名，または SECONDARY INDEX 句に指定した二次インデクス名
4	INDEX_ID	INTEGER	インデクス ID
5	TABLE_ID	INTEGER	レコード型 ID SDB データベース定義の RECORDID 句の指定に関係なく，システムが設定したユニークな内部 ID となります。
7	COLUMN_COUNT	SMALLINT	インデクスを構成する構成要素数
8	CREATE_TIME	CHAR(14)	インデクス作成時刻（YYYYMMDDHHMMSS）
9	RDAREA_NAME	VARCHAR(30)，または MVARCHAR(30)	レコードを格納している RD エリア名 SDB データベース格納定義の WITHIN 句に指定した RD エリア名 SDB データベースを横分割している場合はナル値となります。
11	DIV_INDEX	CHAR(1)	インデクス構成列の先頭列の種別 'Y'：レコード型の RD エリア分割キー（データ種別 1，2 の指定が K，A または K，M の構成要素） 'N'：レコード型の RD エリア分割キー以外
12	FREE_AREA	SMALLINT	ページ内の未使用領域の比率（%）

項番	列名	データ型	内容
			SDB データベース格納定義の PCTFREE 句に指定した未使用領域の比率
13	COLUMN_ID_LIST	VARCHAR(64)	インデクスを構成する構成要素 ID リスト ＋，－で昇順降順を示します。ただし，構成要素が 1 つのインデクスの降順指定時は＋となります。
26	DIV_IN_SRV	CHAR(1)	非分割キーインデクスのサーバ内横分割の有無 非分割キーインデクスをサーバ内分割している場合は'Y'となります。 上記以外の場合はナル値となります。

注

- 上記の表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_INDEXES 表の内容」の表の項番と対応しています。
- 上記の表に記載されている以外の SQL_INDEXES 表の列情報については，マニュアル「HiRDB UAP 開発ガイド」の「SQL_INDEXES 表の内容」を参照してください。

(8) SQL_TABLE_PRIVILEGES 表

SQL_TABLE_PRIVILEGES 表には，レコード型のアクセス権限の定義情報が格納されています。1 行に 1 ユーザの定義情報が格納されています。SQL_TABLE_PRIVILEGES 表に格納されている情報を次の表に示します。

表 B-9 SQL_TABLE_PRIVILEGES 表に格納されている情報

項番	列名	データ型	内容
1	GRANTOR	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
2	GRANTEE	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
3	TABLE_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
4	TABLE_NAME	VARCHAR(30)，または MVARCHAR(30)	レコード型名
5	SELECT_PRIVILEGE	CHAR(1)	SELECT 権限の有無 G（許可される（所有者の場合））となります。
6	INSERT_PRIVILEGE	CHAR(1)	INSERT 権限の有無 G（許可される（所有者の場合））となります。
7	DELETE_PRIVILEGE	CHAR(1)	DELETE 権限の有無 G（許可される（所有者の場合））となります。

項番	列名	データ型	内容
8	UPDATE_PRIVILEGE	CHAR(1)	UPDATE 権限の有無 G（許可される（所有者の場合））となります。
9	GRANT_TIME	CHAR(14)	該当する権限を受け取った時刻 （YYYYMMDDHHMMSS）

注

- 上記の表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_TABLE_PRIVILEGES 表の内容」の表の項番と対応しています。
- 上記の表に記載されている以外の SQL_TABLE_PRIVILEGES 表の列情報については，マニュアル「HiRDB UAP 開発ガイド」の「SQL_TABLE_PRIVILEGES 表の内容」を参照してください。

(9) SQL_DIV_TABLE 表

SQL_DIV_TABLE 表には，レコード型の横分割情報が格納されています。1 行または複数行に 1 レコード型の横分割情報が格納されています。SQL_DIV_TABLE 表に格納されている情報を次の表に示します。

表 B-10 SQL_DIV_TABLE 表に格納されている情報

項番	列名	データ型	内容
1	TABLE_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
2	TABLE_NAME	VARCHAR(30)，または MVARCHAR(30)	レコード型名
3	DIV_NO	INTEGER	分割条件指定順序 該当するレコード型内で 1 から始まるユニークな値で，分割条件の指定順に 1 を加えた値となります。
4	TABLE_ID	INTEGER	レコード型 ID
5	DCOND	CHAR(2)	分割条件コード 格納される値は，分割方法によって次のように異なります。 格納条件分割の場合： ＝ 境界値分割の場合： <= または 空白
6	DCVALUES	VARCHAR(256)，または MVARCHAR(256)	分割条件値 • レコード型の RD エリア分割キー値の指定なし：ナル値となります。 • 上記以外：レコード型の RD エリア分割キー値※ となります。
7	RDAREA_NAME	VARCHAR(30)，または MVARCHAR(30)	横分割した SDB データベースのレコードを格納している RD エリア名

注

- 上記の表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_DIV_TABLE 表の内容」の表の項番と対応しています。
- 上記の表に記載されている以外の SQL_DIV_TABLE 表の列情報については，マニュアル「HiRDB UAP 開発ガイド」の「SQL_DIV_TABLE 表の内容」を参照してください。

注※

レコード型の RD エリア分割キー値を X'＜16 進数字＞'形式で指定した場合，16 進数字のコードに対応する文字で設定されます。

（例）X'41414141'を指定した場合，AAAA が設定されます。

レコード型の RD エリア分割キー値に指定した 16 進数字は，次のどちらかの方法で確認してください。

- SDB 定義文ファイルを参照して確認する。
- スカラ関数 HEX を使用して，16 進文字列表現に変換して確認する。

(10) SQL_INDEX_COLINF 表

SQL_INDEX_COLINF 表には，インデクスを構成する構成要素の情報が格納されています。1 行または複数行に 1 インデクスの情報が格納されています。SQL_INDEX_COLINF 表に格納されている情報を次の表に示します。

表 B-11 SQL_INDEX_COLINF 表に格納されている情報

項番	列名	データ型	内容
1	TABLE_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
2	TABLE_NAME	VARCHAR(30)，または MVARCHAR(30)	インデクスが定義されているレコード型名 SDB データベース格納定義の STORAGE RECORD 句に指定した格納レコード名
3	INDEX_NAME	VARCHAR(30)，または MVARCHAR(30)	インデクス名称 SDB データベース格納定義の SEQUENTIAL 句に指定したシーケンシャルインデクス名，または SECONDARY INDEX 句に指定した二次インデクス名
4	INDEX_ID	INTEGER	インデクス ID
5	INDEX_ORDER	INTEGER	インデクスを構成する構成要素名順を識別する番号 SDB データベース格納定義の ORDER 句に指定した構成要素の順番
6	COLUMN_NAME	VARCHAR(30)，または MVARCHAR(30)	インデクスを構成する構成要素名 SDB データベース格納定義の ORDER 句に指定した構成要素名 なお，次の名称はシステムが使用する構成要素名です。

項番	列名	データ型	内容
			• SYSTEM_FLAG_FIELD • SYSTEM_RECTYPE_NO
7	ASC_DESC	CHAR(1)	昇順，または降順 A（昇順） D（降順） ただし，構成要素が 1 つのインデクスの降順指定時は D となりますが，インデクスキー値の並びは昇順となります。

注

上記の表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_INDEX_COLINF 表の内容」の表の項番と対応しています。

(11) SQL_DIV_INDEX 表の内容

SQL_DIV_INDEX 表には，インデクスの横分割情報が格納されています。1 行または複数行に 1 インデクスの横分割情報が格納されています。SQL_DIV_INDEX 表に格納されている情報を次の表に示します。

表 B-12 SQL_DIV_INDEX 表に格納されている情報

項番	列名	データ型	内容
1	TABLE_SCHEMA	VARCHAR(30)，または MVARCHAR(30)	レコード型の所有者
2	TABLE_NAME	VARCHAR(30)，または MVARCHAR(30)	インデクスが定義されているレコード型名 SDB データベース格納定義の STORAGE RECORD 句に指定した格納レコード名
3	INDEX_NAME	VARCHAR(30)，または MVARCHAR(30)	インデクス名称 SDB データベース格納定義の SEQUENTIAL 句に指定したシーケンシャルインデクス名，または SECONDARY INDEX 句に指定した二次インデクス名
4	DIV_NO	INTEGER	RD エリア定義順序 該当するインデクス内で 1 から始まるユニークな値で，RD エリアの定義順序に 1 を加えた値※となります。
5	INDEX_ID	INTEGER	インデクス ID
6	RDAREA_NAME	VARCHAR(30)，または MVARCHAR(30)	分割格納先 RD エリア名

注

表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_DIV_INDEX 表の内容」の表の項番と対応しています。

注※

SQL_DIV_INDEX の DIV_NO の値と，SQL_DIV_TABLE の DIV_NO の値は，関連性はありません。

(12) SQL_AUDITS 表の内容

SQL_AUDITS 表には，監査対象の情報が格納されています。1 行に，1 オブジェクトまたは 1 ユーザに対する 1 イベント分の情報が格納されています。SQL_AUDITS 表に格納されている情報を次の表に示します。

表 B-13 SQL_AUDITS 表に格納されている情報

項番	列名	データ型	内容
1	EVENT_TYPE	VARCHAR(30)	CREATE AUDIT FOR 操作種別で指定したイベントタイプの名称※1 または'ANY'
2	EVENT_SUBTYPE	VARCHAR(30)	イベントサブタイプの名称※2 または'ANY' CREATE AUDIT FOR ANY を指定した場合はナル値となります。

注

- 表の項番は，マニュアル「HiRDB UAP 開発ガイド」の「SQL_AUDITS 表の内容」の表の項番と対応しています。
- 上記の表に記載されている以外の SQL_AUDITS 表の列情報については，マニュアル「HiRDB UAP 開発ガイド」の「SQL_AUDITS 表の内容」を参照してください。

注※1

イベントタイプを次に示します。

SESSION，PRIVILEGE，DEFINITION，ACCESS，UTILITY，SDB_UTILITY，および SDB_ACCESS

注※2

イベントサブタイプを次に示します。

CONNECT，AUTHORIZATION，DISCONNECT，GRANT，REVOKE，CREATE，DROP，ALTER，SELECT，INSERT，UPDATE，DELETE，PURGE，CALL，ASSIGN LIST，LOCK，NEXT VALUE，PDLOAD，PDRORG，PDEXP，PDCONSTCK，PDSDBDEF，PDSDBLOD，PDSDBROG，FETCH，FETCH FIRST，STORE，MODIFY，ERASE，CLEAR，FETCHDB ALL，および GET

付録 C　ユティリティの最大同時実行数およびコマンドの同時接続数

付録 C.1　ユティリティの最大同時実行数

ユティリティの最大同時実行数を「表 C-1　ユティリティの最大同時実行数」に示します。この表に示す最大同時実行数は，各ユティリティを単独で同時実行したときの値です。pd_max_users オペランドの値に依存するユティリティの最大同時実行数は，ほかのユティリティ実行数と UAP の実行数に影響されます。

最大同時接続数を超えると，ユティリティは次に示すどれかの動作をします。

- 異常終了します。
- 実行中のユティリティが終了するまで無応答になります。
- プロセス割り当て時の要求電文を格納するための共用メモリ中のバッファが不足し，ユティリティが異常終了します。

表 C-1　ユティリティの最大同時実行数

ユティリティ	最大同時実行数	
	pd_utl_exec_mode=0	pd_utl_exec_mode=1
HiRDB/SD 定義ユティリティ（pdsdbdef）	1（ただし，最大同時実行数のチェックは行っていません）	
HiRDB/SD データベース作成ユティリティ（pdsdblod）	MIN（32，pd_max_users オペランドの値）	pd_max_users オペランドの値
HiRDB/SD データベース再編成ユティリティ（pdsdbrog）		
HiRDB/SD データベースアクセスユティリティ（pdsdbexe）	pd_max_users オペランドの値	

上記以外のユティリティの最大同時実行数については，マニュアル「HiRDB コマンドリファレンス」の「ユティリティの最大同時実行数」を参照してください。

■複数のユティリティの同時実行数

複数のユティリティを同時実行して，次に示す式を満たす場合，pd_utl_exec_mode オペランドに 1 を指定してください。

システム内のバックエンドサーバ数×2×すべてのユティリティの同時実行数≧824

付録 C.2　コマンドの同時接続数

コマンド（運用コマンドおよびユティリティ）の中には，HiRDB に対して内部的に接続処理をするコマンドがあります。そのため，pd_max_users オペランドの指定値を決定する場合，次の点に注意する必要があります。

- コマンドの接続時には，pd_max_users オペランドで指定した接続枠を使用するため，ユーザから接続できる数が一時的に減少します。
- コマンド実行時に，接続枠の空きよりコマンドの接続数の方が多くなる場合，接続エラーとなり，コマンドがエラー終了することがあります。

コマンドの同時接続数を次の表に示します。

表 C–2　コマンドの同時接続数

コマンド名	同時接続数
pdsdbdef	1
pdsdblod	
pdsdbrog	
pdsdbexe	
pdsdbarc	
pdsdborcrt	

上記以外のコマンドの同時接続数については，マニュアル「HiRDB コマンドリファレンス」の「コマンドの同時接続数」を参照してください。

付録 D　HiRDB/SD の最大値・最小値

HiRDB/SD の最大値・最小値を次の表に示します。そのほかの最大値・最小値については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB の最大値・最小値」を参照してください。

表 D-1　HiRDB/SD の最大値・最小値

項番	項目		最小値	最大値	単位
1	SDB データベース定義	SDB データベース名	1	30	バイト
2		レコード型名	1	30	バイト
3		レベル番号 （1 レコード型内での構成要素型の階層レベル）	2	49	－
4		構成要素名	1	30※4	バイト
5		CHARACTER （データ型）	1	30,000	バイト
6		XCHARACTER （データ型）	1	30,000	バイト
7		PACKED DECIMAL FIXED（整数部） （データ型）	0	38※1	桁
8		PACKED DECIMAL FIXED（小数部） （データ型）	0	38※1	桁
9		OCCURS 句の繰り返し回数	1	4,000	回
10		OCCURS 句の次元	1	4	次元
11		親子集合型名	1	30	バイト
12		SDB データベース定義中の親子集合型数	0	127	個
13		キー項目の構成要素数	0	4V FMB の SDB データベースの場合：1 SD FMB の SDB データベースの場合：16	個
14		キー項目の構成要素の合計長	0	254	バイト
15		1SDB データベース内のレコード型数	1	128	個
16		一連番号	1	2,147,483,647	－
17		1 レコード型が持つことができる構成要素の数	1	30,000	個

項番	項目			最小値	最大値	単位
18		1 子レコード型が持つことができる親レコード型の数		1	1	個
19		階層数		1	15	階層
20	SDB データベース格納定義	1 レコード型中の構成要素の合計長		1	30,000	バイト
21		シーケンシャルインデクスのキー長		1	255※2	バイト
22		シーケンシャルインデクスの構成要素数		1	7	個
23		格納条件分割で指定可能な RD エリア数		1	256	個
24		格納条件分割で指定可能な格納条件のレコード型の RD エリア分割キー値の数		1	4V FMB または SD FMB の SDB データベースの場合：15,000 4V AFM の SDB データベースの場合：4,000	個
25		格納条件分割で指定可能な 1RD エリア当たりの格納条件のレコード型の RD エリア分割キー値の数		1	4V FMB または SD FMB の SDB データベースの場合：15,000 4V AFM の SDB データベースの場合：4,000	個
26		境界値分割で指定可能な RD エリア数		2	3,000（1,024）※3	個
27		二次インデクスのキー長	4V FMB または 4V AFM の SDB データベースの場合	4	576	バイト
			SD FMB の SDB データベースの場合	1	255	バイト
28		二次インデクスの構成要素数	4V FMB または 4V AFM の SDB データベースの場合	2	4V FMB の SDB データベースの場合：16 4V AFM の SDB データベースの場合：8	個
			SD FMB の SDB データベースの場合	1	16	個
29		1 レコード型当たりのインデクス定義数		0	4V FMB または 4V AFM の SDB データベースの場合：2 SD FMB の SDB データベースの場合：16	個

項番	項目		最小値	最大値	単位
30		サブページ分割数	2	16	個
31	SDB データベース操作	1 ファミリ当たりの，確保可能な事前割り当てページ数	1	2,147,483,647	ページ
32		1 ファミリ当たりの，確保可能な事前割り当てサブページ数	1	2,147,483,647	サブページ

（凡例）－：該当しません。

注※1

整数部と小数部の桁数の和が 1～38 桁となります。

注※2

4V AFM の SDB データベースの場合，システム用構成要素が付加されるため，ユーザが定義した構成要素に関するキー長の最大値は 249 バイトになります。

注※3

3,000 は，境界値分割をする場合，SDB データベース格納定義の WITHIN 句に指定できる RD エリア名の延べ数の最大値です。1,024 は，WITHIN 句に RD エリア名を重複して指定する場合，RD エリア名の重複を排除して数えたときの最大値です。

ここでいう WITHIN 句は，格納レコード用 RD エリア名を指定する WITHIN 句，シーケンシャルインデクス用 RD エリア名を指定する WITHIN 句，および二次インデクス用 RD エリア名を指定する WITHIN 句を示しています。

注※4

SDB データベース定義で OCCURS 句を指定する場合，システム内で繰り返し回数分生成される構成要素の構成要素名の最大長が 30 バイトになります。

付録 E　HiRDB/SD で起動するプロセス

HiRDB/SD で起動するプロセスを次の表に示します。

なお，次の表に示すプロセス以外にも，HiRDB/SD で起動するプロセスがあります。詳細については，マニュアル「HiRDB システム導入・設計ガイド」の「HiRDB/パラレルサーバで起動するプロセス」を参照してください。

表 E-1　HiRDB/SD で起動するプロセス

サーバ種別	プロセス名		説明	プロセス数	サーバ名称
	表記	プロセス名称			
ユティリティサーバ	pdsdbdef 制御プロセス	pdsdbdef	定義ユティリティ制御	1	－
	pdsdblod 制御プロセス	pdsdblodm	データロード制御	pdsdblod コマンドの同時実行数	0msdldN※1
	pdsdbrog 制御プロセス	pdsdbrogm	データベース再編成制御	pdsdbrog コマンドの同時実行数	0msdrgN※1
	pdorend 反映プロセス	pdsdborendl	更新可能なオンライン再編成の追い付き反映処理	pdorend コマンドの-m オプションの指定値	0soreX0※2

（凡例）　－：該当しません。

注※1

末尾の N は 16 進数字になります。

注※2

サーバ名称は，pdorend コマンドの-m オプションに指定値によって決まります。例えば，-m オプションに 3 を指定した場合，サーバ名称は 0sore30 になります。

付録 F　単調増加ファイル

HiRDB/SD を使用すると単調増加するファイルを情報の種別ごとに示します。ここに説明のない単調増加ファイルについては，マニュアル「HiRDB システム導入・設計ガイド」を参照してください。

なお，*は任意の英数字です。また，ファイル名またはディレクトリ名には，標準のパス名を記載しています。システムによっては異なる場合があります。

付録 F.1　簡易ダンプ

簡易ダンプに関する単調増加ファイルを次の表に示します。

表 F-1　単調増加ファイル（簡易ダンプ）

項番	ファイル名またはディレクトリ名	説明	ファイルサイズの概算値	ファイル数	最大サイズの制限可否
1	$PDDIR/spool/pdfesdump/*	フロントエンドサーバ用の簡易ダンプファイルです。pdfes プロセスセグメンテーション障害，またはアボート時に生成されます。 自動で削除する場合： 次のどれかのオペランドを指定します。 • pd_spool_cleanup_interval • pd_spool_cleanup_interval_level • pd_spool_cleanup • pd_spool_cleanup_level 手動で削除する場合： pdcspool コマンドを実行します。 なお，$PDDIR 下が容量不足となった場合はユニットダウンします。	システム共通定義 pd_structured_shmpool_dicsize オペランド指定値 ＋ SDB 用 UAP 環境定義ワークエリアサイズ※4 ＋数 MB	ディレクトリ： $PDDIR/spool/pdfesdump，pdfesdump1，pdfesdump2 でディレクトリ 3 世代のループ ファイル： ディレクトリ下に無制限	×
2	$PDDIR/spool/pdbesdump/*	バックエンドサーバ用の簡易ダンプファイルです。pdbes プロセスセグメンテーション障害，アボート，ユーザ用 RD エリア閉塞時に生成されます。	システム共通定義 pd_structured_shmpool_dicsize オペランド指定値 ＋インデクス一括作成機能用ワークエリアサイズ※1	ディレクトリ： $PDDIR/spool/pdbesdump，pdbesdump1，	×

項番	ファイル名またはディレクトリ名	説明	ファイルサイズの概算値	ファイル数	最大サイズの制限可否
		自動で削除する場合： 次のどれかのオペランドを指定します。 • pd_spool_cleanup_interval • pd_spool_cleanup_interval_level • pd_spool_cleanup • pd_spool_cleanup_level 手動で削除する場合： pdcspool コマンドを実行します。 なお，$PDDIR 下が容量不足となった場合はユニットダウンします。	＋排他制御用ワークエリアサイズ※2 ＋ FETCHDB ALL 用ワークエリアサイズ※3 ＋数 MB	pdbesdump 2 でディレクトリ 3 世代のループ ファイル： ディレクトリ下に無制限	
3	$PDCLTPATH/pdsdbexe.abort.yyyymmddhhmmss.プロセス ID※5，※7	pdsdbexe コマンドの簡易ダンプファイルです。pdsdbexe コマンドが異常状態を検出した場合に生成されます。	概算値は，百数十 KB～数 MB になります。 **計算式（単位：バイト）** 170,000 ＋（128×SDB データベース定義数） ＋（210×個別開始した SDB データベースのレコード数） ＋（64×レコードの構成要素数の最大数×個別開始した SDB データベースのレコード数） ＋（600 ＋（操作対象のレコード長×10）） ＋（8×レコードの構成要素数の最大数）	異常状態を検出した pdsdbexe コマンドのプロセス ID ごとに 1 ファイル	×
4	$PDCLTPATH/pdsdbcbl.abort.yyyymmddhhmmss.プロセス ID※6，※7	pdsdbcbl コマンドの簡易ダンプファイルです。pdsdbcbl コマンドが異常状態を検出した場合に生成されます。	「3.6.1(6)　DML プリプロセサ（pdsdbcbl）実行時のファイルの容量【SD FMB】」を参照してください。	異常状態を検出した pdsdbcbl コマンドのプロセス ID ごとに 1 ファイル	×

（凡例）

×：最大サイズを制限できません。

注※1

「3.6.1(1)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量」にある lindxa の値です。ただし，インデクス一括作成機能を使用した pdsdblod コマンド実行中以外の場合，値は 0 です。

注※2

10×pd_max_rdarea_no の値（単位：バイト）です。

注※3

「3.4.3(2)(c)　変数 SDA の求め方」の計算式 B を参照して，算出してください。

注※4

「3.4.3(3)(a)　変数 SDA の求め方」の変数 SDA の値です。

注※5

$PDCLTPATH の指定がない場合，または$PDCLTPATH 下に書き込みできない場合は，pdsdbexe コマンドを実行した際のカレントディレクトリに生成されます。

注※6

$PDCLTPATH の指定がない場合，または$PDCLTPATH 下に書き込みできない場合は，pdsdbcbl コマンドを実行した際のカレントディレクトリに生成されます。

注※7

yyyymmddhhmmss：簡易ダンプの出力日時

プロセス ID：1～10 バイト

付録 F.2　トラブルシュート情報

トラブルシュート情報に関する単調増加ファイルを次の表に示します。

表 F-2　単調増加ファイル（トラブルシュート情報）

項番	ファイル名またはディレクトリ名	説明	ファイルサイズの概算値	ファイル数	最大サイズの制限可否
1	$PDDIR/spool/save/*コマンド名**.txt	pdsdblod コマンド，pdsdbrog コマンドのトラブルシュート情報です。コマンドのタイムアウト時に生成されます。 自動で削除する場合： 次のどれかのオペランドを指定します。 • pd_spool_cleanup_interval • pd_spool_cleanup_interval_level • pd_spool_cleanup • pd_spool_cleanup_level 手動で削除する場合： pdcspool コマンドを実行します。	数 KB～数十 KB（そのときに実行していた HiRDB/SD のプロセス数や，排他待ち資源数によって変化します）	コマンドのプロセス ID ごとに 1 ファイル	×

項番	ファイル名またはディレクトリ名	説明	ファイルサイズの概算値	ファイル数	最大サイズの制限可否
		なお，$PDDIR 下が容量不足となった場合はユニットダウンします。			

（凡例）

×：最大サイズを制限できません。

付録 F.3　ユティリティおよびコマンド実行時の出力ファイル

ユティリティおよびコマンド実行時に出力されるファイルのうち，単調増加ファイルを次の表に示します。

表 F-3　単調増加ファイル（ユティリティおよびコマンド実行時の出力ファイル）

項番	ファイル名	説明	ファイルサイズの概算値	ファイル数	最大サイズの制限可否
1	/environment 文の msglog オペランドに指定したディレクトリ※1/SDBDEF-*	pdsdbdef コマンドの実行結果を出力するファイル（実行結果ファイル）です。	SDB 定義文ファイルサイズ×1.2	毎回異なるファイル名で 1 ファイル	×
2	/load 文の storinf オペランドに指定したディレクトリ※1/SDBLOD-*	pdsdblod コマンドの実行結果を出力するファイル（実行結果ファイル）です。	「3.6.1(1)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量」を参照してください。	毎回異なるファイル名で 1 ファイル	×
3	load 文の errdata オペランドに指定した論理エラー情報ファイル	pdsdblod コマンドのエラーレコード情報（論理エラー情報ファイル）です。	「3.6.1(1)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量」を参照してください。	load 文の errdata オペランドの指定が毎回同じ場合は 1，毎回異なる場合は無制限	○
4	/ワークディレクトリ※2/PDSDBLOAD-*	インフォメーションメッセージを出力するワークファイルです。	「3.6.1(1)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量」を参照してください。	environment 文 infmsglvl=lvl2 を指定し，インデクス一括作成モードを選択する場合，毎回異なるファイル名で 1 ファイル	×
5	/unload 文の unldinf オペランドに指定した	pdsdbrog コマンドのアンロード情報を出力するファイ	「3.6.1(2)　HiRDB/SD データベース再編成ユティリティ（pdsdbrog）実行時の	毎回異なるファイル名で 1 ファイル	×

項番	ファイル名	説明	ファイルサイズの概算値	ファイル数	最大サイズの制限可否
	ディレクトリ※1/SDBROG-*	ル（アンロードの実行結果ファイル）です。	ファイルの容量」を参照してください。		
6	/report 文の dml_skip_info オペランドに指定したディレクトリ※1/pdskipdml_*	pdorend コマンドの実行時に作成される，追い付き反映時の DML スキップ情報出力ファイルです。	700×スキップした SDB データベースを操作する API 数（単位：バイト）	pdorend コマンドの対象 RD エリアが存在する BES 数	×
7	• idxload 文の workdir オペランドの指定時 指定ディレクトリ/SDBLOD-* • idxload 文の workdir オペランドの省略時 /tmp/SDBLOD-*（pd_tmp_directory オペランド指定時は，pd_tmp_directory ディレクトリ下）	pdsdblod コマンドによるインデクス作成時のインデクス情報ファイル	idxload 文の sortdir オペランドの指定値	インデクスの作成時，毎回異なるファイル名で「インデクス数×インデクス格納用 RD エリア数」作成	×
8	• idxload 文の sortdir オペランド指定時 指定ディレクトリ/rs* • idxload 文の sortdir オペランド省略時 /tmp/sr*（pd_tmp_directory オペランド指定時は，pd_tmp_directory ディレクトリ下）	pdsdblod コマンドによるインデクス作成時のソート用ワークファイル	「3.6.1(1)　HiRDB/SD データベース作成ユティリティ（pdsdblod）実行時のファイルの容量」を参照してください。	インデクスの作成時，毎回異なるファイル名で 1 個作成	×

（凡例）

○：最大サイズを制限できます。

×：最大サイズを制限できません。

注※1

オペランドを省略している場合は，次の優先順位（1 の指定がいちばん上位）に従って出力先のディレクトリが決まります。

1. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
2. 環境変数 TMPDIR に指定したディレクトリ
3. /tmp ディレクトリ

注※2

次の優先順位（1 の指定がいちばん上位）に従って出力先のワークディレクトリが決まります。

1. システム定義の pd_tmp_directory オペランドに指定したディレクトリ
2. 環境変数 TMPDIR に指定したディレクトリ
3. /tmp ディレクトリ

付録 G　ユティリティが出力するファイル

次のユティリティが出力するファイルを次の表に示します。

- HiRDB/SD 定義ユティリティ（pdsdbdef）
- HiRDB/SD データベース作成ユティリティ（pdsdblod）
- HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

表 G-1　ユティリティが出力するファイル

項番	ユティリティ名	出力ファイルまたは出力ディレクトリ	ファイルの出力先	自動削除の対象可否
1	pdsdbdef	実行結果ファイル	environment 文の指定先	×
2		SDB ディレクトリ情報ファイル（pdsdbdir）	dirinf 文の指定先	×
3		SDB ディレクトリ情報ファイルの格納ディレクトリ		×
4		配布元一時フォルダ（SDBDEF-AS_プロセス ID_プロセス開始時刻）	ワークディレクトリ※	○
5		処理結果ファイル（SDBDEF-result ＋ホスト名+ユニット名＋生成日時）	配布元一時フォルダの直下	○
6		配布先一時フォルダ（SDBDEF-AD_ユニット名）	ワークディレクトリ※	○
7		多重起動防止ファイル（SDBDEF-run）	配布先一時フォルダの直下	○
8		処理結果ファイル（SDBDEF-result ＋ホスト名＋ユニット名＋生成日時）		○
9		リモートコピー時の一時ファイル（SDBDEF-tmp_プロセス ID_プロセスの開始時刻）		○
10		スキップファイル（SDBDEF-skip）	ユニット制御情報定義の pd_structured_directory_path オペランドに指定したディレクトリ	○
11		リモートコピー時の一時ファイル（SDBDEF-tmp_プロセス ID_プロセスの開始時刻）		○
12		pdsdbdir のバックアップ用一時ファイル（pdsdbdir_bk_生成日時）		○
13	pdsdblod	実行結果ファイル	load 文の指定先	×
14		論理エラー情報ファイル		×
15		インデクス情報ファイル	idxload 文の指定先	○
16		ソート用ワークファイル		○

項番	ユティリティ名	出力ファイルまたは出力ディレクトリ	ファイルの出力先	自動削除の対象可否
17		ワークファイル	environment 文の指定先	×
18		中間ファイル	oreload 文の指定先	○
19	pdsdbrog	アンロードの実行結果ファイル	unload 文の指定先	×
20		アンロードデータファイル		×

（凡例）

○：ユティリティの終了時，ファイルが自動的に削除されます。

×：ユティリティの終了時，ファイルが自動的に削除されることはありません。ファイルは残ります。

注 （ ）内はファイル名です。

注※ 次のディレクトリを意味しています。

- ユニット制御情報定義の pd_tmp_directory オペランドに指定したディレクトリ
- 環境変数 TMPDIR に指定したディレクトリ
- /tmp

付録 H　ユティリティの排他制御モード

次のユティリティの排他制御モードを次に示します。

- HiRDB/SD 定義ユティリティ（pdsdbdef）
- HiRDB/SD データベース作成ユティリティ（pdsdblod）
- HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

なお，ここに説明のないユティリティの排他制御モードについては，マニュアル「HiRDB コマンドリファレンス」を参照してください。

付録 H.1　HiRDB/SD 定義ユティリティ（pdsdbdef）の排他制御モード

HiRDB/SD 定義ユティリティ（pdsdbdef）の排他制御モードは，HiRDB/SD 定義ユティリティが処理の延長で発行する SQL 文によって決まります。SQL 文の排他モードについては，マニュアル「HiRDB UAP 開発ガイド」の「排他制御のモード」を参照してください。

なお，マニュアル「HiRDB UAP 開発ガイド」の「排他制御のモード」を参照する際，次の表に示す個所を参照してください。

表 H-1　HiRDB/SD 定義ユティリティ（pdsdbdef）の排他制御モード

実行する SDB 定義文		「排他制御のモード」で参照する SQL 文
*ENTRY DICTIONARY 文 *CHECK DICTIONARY 文		インデクス定義
*ENTRY DICTIONARY 文 *ENTRY DIRECTORY 文		インデクス定義
*ALTER DICTIONARY 文 *CHECK DICTIONARY 文	レコード型追加	インデクス定義※1
	格納 RD エリア変更	• 表定義変更（主キーの追加および削除は除く） • 全行削除※2
*ALTER DICTIONARY 文 *ALTER DIRECTORY 文	レコード型追加	インデクス定義※1
	格納 RD エリア変更	• 表定義変更（主キーの追加および削除は除く） • 全行削除※2
*DELETE DICTIONARY 文 *CHECK DICTIONARY 文		• 表削除 • インデクス削除
*DELETE DICTIONARY 文 *DELETE DIRECTORY 文		• 表削除 • インデクス削除

注※1

追加したレコード型にインデクスを定義する場合

注※2

「表 11-23　RD エリアの定義変更時のデータの削除有無（格納条件指定の横分割の場合）」または「表 11-24　RD エリアの定義変更時のデータの削除有無（境界値指定の横分割の場合）」の同期点を取得するケース（表の※1）が該当します。

付録 H.2　HiRDB/SD データベース作成ユティリティ（pdsdblod）の排他制御モード

HiRDB/SD データベース作成ユティリティ（pdsdblod）の排他制御モードを次の表に示します。

HiRDB/SD データベース作成ユティリティ（pdsdblod）の追い付き反映キー対応表に対する排他制御モードを「表 H-5　HiRDB/SD データベース作成ユティリティ（pdsdblod）の追い付き反映キー対応表に対する排他制御モード」に示します。

表 H-2　HiRDB/SD データベース作成ユティリティ（pdsdblod）の排他制御モード

SDB データベース種別	SDB データベースの横分割	area オペランド	資源									
			RD エリア※1				ルートレコード型	ルートレコード型 (NO WAIT 検索時)	シーケンシャルインデクス ID	二次インデクス ID	データベース複写ユティリティ※3	ページまたはサブページ
			レコード格納用	シーケンシャルインデクス格納用	二次インデクス格納用	RD エリア管理ブロック※4						
4V FMB	なし	指定なし	SU	SU	SU	SR	EX	EX※2	EX	EX	SU	−
	あり※5		EX	SU	SU	SR	SR※7	EX※2	EX※6	EX※6	SU	−
		指定あり	EX	SU	SU	SR	SR※7	−	EX※6	EX※6	SU	−
4V AFM	なし	指定なし	SU	SU	SU	SR	EX	EX※2	EX	EX	SU	−
	あり※5		EX	SU	SU	SR	−	EX※2	EX※6	EX※6	SU	−
		指定あり	EX	SU	SU	SR	−	−	EX※6	EX※6	SU	−
SD FMB	なし	指定なし	SU	SU	−	SR	EX	EX	EX	−	SU	−
	あり※5		EX	SU	−	SR	SR※7	EX※2	EX※6	−	SU	−
		指定あり	EX	SU	−	SR	SR※7	−	EX※6	−	SU	−

（凡例）

EX：排他モードの排他が掛かります。

SR：意図共用モードの排他が掛かります。

SU：意図排他モードの排他が掛かります。

−：排他は掛かりません。

注※1

インナレプリカ機能使用時は，次の RD エリアを対象とします。

pdsdblod 制御文の environment 文にある generation オペランドの指定	対象とする RD エリア
あり	指定した世代番号の RD エリア
なし	カレント RD エリア

注※2

environment 文の purge オペランドに yes を指定した場合は，データを削除するときにルートレコード型（NOWAIT 検索時）の資源に対して，EX モードで排他を掛けます。

注※3

- load 文の idxmode オペランドに create を指定した場合は，データ格納の終了処理でインデクス格納用 RD エリアのセグメントに対して，SU モードで排他を掛けます。
- environment 文の purge オペランドに yes を指定した場合は，データを削除するときに SU モードで排他を掛けます。

注※4

RD エリア中のセグメントの追加，セグメントの削除など，セグメントを操作する可能性があるプロセスが排他を掛ける領域を指します。

注※5

格納対象の RD エリアに対してだけ排他を掛けます。格納対象の RD エリアは，次の表を参照してください。

表 H-3　SDB データベースの横分割状態と area オペランドの指定の有無の組み合わせによる格納対象

SDB データベースの横分割	area オペランドの指定	格納対象
なし	なし	すべての RD エリアをデータロードの対象とします。
	あり	KFPB63301-E が出力され，エラーとなります。
あり	なし	server オペランドで指定したバックエンドサーバ上の RD エリアだけをデータロードの対象とします。
	あり	area オペランドで指定した RD エリアだけをデータロードの対象とします。

SDB データベースの横分割種別は，次のとおりです。

- 格納条件分割
- 境界値分割

注※6

SDB データベースが横分割されている場合，インデクス ID に排他を掛けるのではなく，該当するインデクス格納用 RD エリアに格納されているインデクスに対してだけ EX モードで排他が掛かるため，RD エリア単位のデータロードを同時に実行できます。

注※7

pdsdblod 制御文のオペランドの指定とルートレコード型の排他の関係を次の表に示します。

表 H-4　pdsdblod 制御文のオペランドの指定とルートレコード型の排他の関係

environment 文の purge オペランドの指定	load 文		ルートレコード型の排他
	idxmode オペランドの指定	dupkeyck オペランドの指定	
yes	create または省略	yes または省略	−
		no	−
	sync	yes または省略	−
		no	−
no または省略	create または省略	yes または省略	SR
		no	−
	sync	yes または省略	−
		no	−

（凡例）

SR：意図共用モードの排他が掛かります。

−：排他は掛かりません。

表 H-5　HiRDB/SD データベース作成ユティリティ（pdsdblod）の追い付き反映キー対応表に対する排他制御モード

排他が取得されるタイミング	資源							
	インナレプリカ※1	RD エリア			表 ID	インデクス ID	ページ	行
		表 ID	インデクス ID	RD エリア管理ブロック※2				
追い付き反映キー対応表のチェック時（KFPB63000-I 出力後）	SR	SR	SR	−	SR	−	−	PR
追い付き反映キー対応表へのデータ格納時（KFPB63042-I 出力後）	−	SU	SU	SR	SU	SU	−	EX

（凡例）

EX：排他モードの排他が掛かります。

PR：共用モードの排他が掛かります。

SR：意図共用モードの排他が掛かります。

SU：意図排他モードの排他が掛かります。

−：排他は掛かりません。

注※1

インナレプリカ構成管理情報またはレプリカグループ構成管理情報のことです。

システム共通定義の pd_inner_replica_lock_shift オペランドの指定によって，対象となる資源が次のように異なります。

- pd_inner_replica_lock_shift オペランドに Y を指定した場合は，レプリカグループ構成管理情報に排他が掛かります。
- pd_inner_replica_lock_shift オペランドに N を指定したか，または指定を省略した場合は，インナレプリカ構成管理情報に排他が掛かります。

注※2

RD エリア管理ブロックとは，RD エリア中のセグメントの追加または削除など，セグメントの操作を行う可能性があるプロセスが排他を掛ける領域のことです。

表 H-6　HiRDB/SD データベース作成ユティリティ（pdsdblod）のインデクス再作成機能使用時の排他制御モード

SDB データベース種別	SDB データベースの横分割	資源									
		RD エリア※2				ルートレコード型	ルートレコード型（NOWAIT 検索時）	シーケンシャルインデクス ID※3	二次インデクス ID※3	データベース複写ユティリティ	ページまたはサブページ
		レコード格納用	シーケンシャルインデクス格納用※3	二次インデクス格納用※3	RD エリア管理ブロック※1						
4V FMB	なし	SU	SU	SU	SR	EX	－	EX	EX	SU	－
	あり	EX	SU	SU	SR	－	－	EX	EX	SU	－
4V AFM	なし	SU	SU	SU	SR	EX	－	EX	EX	SU	－
	あり	EX	SU	SU	SR	－	－	EX	EX	SU	－
SD FMB	なし	SU	SU	－	SR	EX	EX	EX	－	SU	－
	あり	EX	SU	－	SR	－	－	EX	－	SU	－

（凡例）

EX：排他モードの排他が掛かります。

SR：意図共用モードの排他が掛かります。

SU：意図排他モードの排他が掛かります。

PR：共用モードの排他が掛かります。

－：排他は掛かりません。

注※1

RD エリア中のセグメントの追加，セグメントの削除など，セグメントを操作する可能性があるプロセスが排他を掛ける領域です。

注※2

インナレプリカ機能使用時は，次の RD エリアを対象とします。

pdsdblod 制御文の environment 文の generation オペランドの指定	対象とする RD エリア
あり	指定した世代番号の RD エリア
なし	カレント RD エリア

注※3

インデクス格納用 RD エリアおよびインデクスは，シーケンシャルインデクスまたは二次インデクスのどちらかだけに排他が掛かります。

付録 H.3　HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の排他制御モード

HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の排他制御モードを次の表に示します。

表 H-7　HiRDB/SD データベース再編成ユティリティ（pdsdbrog）の排他制御モード

SDB データベース種別	SDB データベースの横分割	資源							
		RD エリア※1				ルートレコード型	シーケンシャルインデクス ID	二次インデクス ID	ページまたはサブページ
		レコード格納用	シーケンシャルインデクス格納用	二次インデクス格納用	RD エリア管理ブロック※2				
4V FMB または SD FMB	なし	SR	SR	SR	SR	PR	－	－	－
	あり	PR	PR	PR	SR	SR	－	－	－
4V AFM	なし	SR	SR	SR	SR	PR	－	－	－
	あり	PR	PR	PR	SR	SR	－	－	－

（凡例）

PR：共用モードの排他が掛かります。

SR：意図共用モードの排他が掛かります。

－：排他は掛かりません。

注※1

インナレプリカ機能使用時は，次の RD エリアを対象とします。

pdsdblod 制御文の environment 文にある generation オペランドの指定	対象とする RD エリア
あり	指定した世代番号の RD エリア
なし	カレント RD エリア

注※2

RD エリア中のセグメントの追加，セグメントの削除など，セグメントを操作する可能性があるプロセスが排他を掛ける領域を指します。

付録 I　ユティリティ実行時の留意事項

次のユティリティ実行時の留意事項について説明します。

- HiRDB/SD 定義ユティリティ（pdsdbdef）
- HiRDB/SD データベース作成ユティリティ（pdsdblod）
- HiRDB/SD データベース再編成ユティリティ（pdsdbrog）

付録 I.1　RD エリアの状態によるユティリティの実行可否

ユティリティを実行する場合，RD エリアのオープン契機，および RD エリアの状態によって実行可否が決まります。RD エリアの状態によるユティリティの実行可否を次に示します。

インナレプリカ機能使用時の，RD エリアの状態ごとの HiRDB/SD 定義ユティリティ（pdsdbdef）の実行可否については，「付録 J.2　RD エリアの状態ごとの HiRDB/SD 定義ユティリティ（pdsdbdef）の実行可否」を参照してください。

また，HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の実行可否については，UAP の実行可否と同じになります。RD エリアの状態による UAP の実行可否については，マニュアル「HiRDB コマンドリファレンス」の「RD エリアの状態によるユティリティおよび UAP の実行可否」を参照してください。

表 I-1　RD エリアの状態によるユティリティの実行可否（オープン契機が INITIAL の場合）

RD エリアの状態		ユティリティの実行可否		
		pdsdbdef	pdsdblod	pdsdbrog
閉塞なし	オープン	○	○	○
	クローズ	×	×	×
コマンド閉塞	オープン	×	○	○
	クローズ	×	×	×
参照可能閉塞	オープン	×	○	○
	クローズ	×	×	×
参照可能バックアップ閉塞	オープン	×	×	○
	クローズ	×	×	×
更新可能バックアップ閉塞	オープン	×	×	×
障害閉塞	オープン	×	×	×
	クローズ	×	×	×

RD エリアの状態		ユティリティの実行可否		
		pdsdbdef	pdsdblod	pdsdbrog
ログレス閉塞	オープン	×	×	×
	クローズ	×	×	×
同期化閉塞	オープン	×	×	×
	クローズ	×	×	×
オンライン再編成閉塞	オープン	×	○※1，※2	○※1
	クローズ	×	×	×

（凡例）

○：実行できます。

×：実行できません。

注※1

- レプリカ RD エリアに対しては実行できません。
- 同一レプリカグループ内のオンライン再編成閉塞のオリジナル RD エリアが，カレント RD エリアの場合は実行できません。

注※2

- インデクスの再作成機能は実行できません。
- 追加データロード機能は実行できません。

表 I-2　RD エリアの状態によるユティリティの実行可否（オープン契機が DEFER または SCHEDULE の場合）

RD エリアの状態		ユティリティの実行可否		
		pdsdbdef	pdsdblod	pdsdbrog
閉塞なし	オープン	○	○	○
	クローズ	×	○	○
コマンド閉塞	オープン	×	○	○
	クローズ	×	×	×
参照可能閉塞	オープン	×	○	○
	クローズ	×	○	○
参照可能バックアップ閉塞	オープン	×	×	○
	クローズ	×	×	○
更新可能バックアップ閉塞	オープン	×	×	×

RD エリアの状態		ユティリティの実行可否		
		pdsdbdef	pdsdblod	pdsdbrog
障害閉塞	オープン	×	×	×
	クローズ	×	×	×
ログレス閉塞	オープン	×	×	×
	クローズ	×	×	×
同期化閉塞	オープン	×	×	×
	クローズ	×	×	×
オンライン再編成閉塞	オープン	×	○※1，※2	○※1
	クローズ	×	○※1，※2	○※1

（凡例）

○：実行できます。

×：実行できません。

注※1

- レプリカ RD エリアに対しては実行できません。
- 同一レプリカグループ内のオンライン再編成閉塞のオリジナル RD エリアが，カレント RD エリアの場合は実行できません。

注※2

- インデクスの再作成機能は実行できません。
- 追加データロード機能は実行できません。

付録 I.2　ユティリティ実行中の割り込みによる強制終了

ユティリティ実行中は，シグナル割り込み（例えば，CTL + C，CTL +¥，pdkill コマンドなど）を使用してユティリティを強制終了させないでください。タイミングによっては，HiRDB が終了することがあります。ユティリティは，pdcancel コマンドで終了させてください。

付録 I.3　LANG 環境変数の設定

ユティリティには，LANG 環境変数の設定が必要なものがあります。

LANG 環境変数を指定しなくても，LC_ALL または LC_*環境変数を指定していれば，こちらの設定が有効となります。また，LC_ALL または LC_*環境変数を指定していない場合は LANG 環境変数の設定が有効となります。

LANG 環境変数の設定が必要なユティリティについては，これらを考慮して実行するようにしてください。

付録 I.4　ユティリティ実行時に取得される統計情報

ユティリティ実行時に取得される統計情報は，SDB データベース定義を取得するために行うディクショナリ検索時に実行される SQL が対象となります。

ユティリティ実行時に取得される統計情報を次の表に示します。

表 I-3　ユティリティ実行時に取得される統計情報

項番	ディクショナリ検索時に取得される統計情報	ユティリティ	
		pdsdblod	pdsdbrog
1	システムの稼働に関する統計情報	○	○
2	UAP に関する統計情報	○	○
3	SQL に関する統計情報	○	○
4	グローバルバッファに関する統計情報	○	○
5	データベース操作に関する HiRDB ファイルの統計情報	○	○
6	SQL 静的最適化に関する統計情報	○	○
7	SQL 動的最適化に関する統計情報	○	○
8	SQL オブジェクト実行に関する統計情報	○	○
9	SQL 文の履歴に関する統計情報	○	○
10	CONNECT/DISCONNECT に関する統計情報	○	○
11	デファードライト処理に関する統計情報	○※	○※
12	インデクスに関する統計情報	○※	○※
13	SQL オブジェクト転送に関する統計情報	○	○

（凡例）

○：統計情報を取得します。

注※

イベントが発生しなければ値が 0 となります。

付録 J　インナレプリカ機能使用時の HiRDB/SD 定義ユティリティ（pdsdbdef）の実行【4V FMB，4V AFM】

インナレプリカ機能を使用する場合の pdsdbdef コマンドを実行するための条件，および注意事項について説明します。

付録 J.1　HiRDB/SD 定義ユティリティ（pdsdbdef）を実行するための条件

インナレプリカ機能を使用する場合，pdsdbdef コマンドはオリジナル RD エリアとレプリカ RD エリアの対象のレコード型に関連するすべての RD エリアに対して処理を行います。

pdsdbdef コマンドを実行するためには，次に示す条件を満たしている必要があります。(a)，(b) および (c) の条件を満たしていない場合，pdsdbdef コマンドはエラーになります。SDB 定義文ごとにチェック条件が異なります。チェック条件については，「表 J-3　チェックが発生する SDB 定義文の一覧とそのチェック内容」を参照してください。

(a)SDB 定義文で指定する格納 RD エリア名

指定された格納 RD エリアがオリジナル RD エリアである。

(b)対象のレコード型と関連 RD エリアのレプリカ RD エリアの複製定義

対象のレコード型と関連 RD エリアが複数になる場合，次の条件を満たすかどうか。

- 各 RD エリアのレプリカ RD エリアの定義数が同じである。
- 各 RD エリアのレプリカ RD エリアを定義した世代が同じである。

(c)RD エリアの状態

対象のレコード型と関連 RD エリアが，すべての世代（オリジナル RD エリアも含む）で HiRDB/SD 定義ユティリティを実行できる状態である。

RD エリアの状態の確認手順を次に示します。RD エリアの状態ごとの pdsdbdef コマンドの実行可否については，「付録 J.2　RD エリアの状態ごとの HiRDB/SD 定義ユティリティ（pdsdbdef）の実行可否」を参照してください。

(1)　RD エリアの状態の確認手順

*ENTRY DIRECTORY 文，*ALTER DIRECTORY 文，*CHECK DICTIONARY 文，または*DELETE DICTIONARY 文を実行する前に，次の手順に従って RD エリアの状態を確認します。

1. レコード型の関連 RD エリア（レコード型，およびインデクスを格納する RD エリア）を調べます。

<*ENTRY DIRECTORY 文，*ALTER DIRECTORY 文，または*CHECK DICTIONARY 文の場合>

SDB データベース格納定義に記述する RD エリアを確認します。

＜*DELETE DICTIONARY 文の場合＞

pdrdrefls コマンドの-t オプションに各 SDB データベースの任意のレコード型名を指定して，SDB データベースごとに実行します。

インデクスを格納する RD エリアにほかの SDB データベースのレコード型のインデクスが定義されている場合，pdrdrefls コマンドで表示される関連 RD エリアには，それらの情報も表示されます。そのため，実際に pdsdbdef コマンドが対象とする関連 RD エリアよりも範囲が広い場合があります。-a オプション指定で表示されるレコード型名またはインデクス名から対象 RD エリアを判断してください。

2. 1.で調べた RD エリアで，すべてのオリジナル RD エリアが更新可能状態であることを pddbls コマンドで確認します。

3. 1.で調べた RD エリアで，レプリカ RD エリアの状態を pddbls コマンドで確認します。pdsdbdef コマンドを実行できる条件を次に示します。

＜オリジナル RD エリアがすべてカレント RD エリアである場合＞

すべてのレプリカ RD エリアが，各世代内で次のどちらかの状態で統一されている必要があります。

・更新可能な状態

・コマンド閉塞かつクローズ状態※

＜オリジナル RD エリアがカレント RD エリアでない場合＞

(i)　pddbls コマンドに-C オプションを指定して，1.で調べたオリジナル RD エリアのレプリカ RD エリアを調べます。

(ii)　pddbls コマンドの-q オプションに(i)のレプリカ RD エリアの世代番号を指定して，すべてのレプリカ RD エリアのカレント RD エリアが更新可能状態であることを確認します。

(iii)　上記以外のレプリカ RD エリアが，各世代内で次のどちらかの状態で統一されている必要があります。

・更新可能な状態

・コマンド閉塞かつクローズ状態※

注※

世代の処理対象となるレコード型に関連するすべての RD エリアがコマンド閉塞かつクローズ状態の場合は，定義処理がスキップされます。定義処理がスキップされた場合，KFPH22032-W メッセージが出力されます。定義処理が正常に完了し，KFPH22032-W メッセージが出力されている場合は，スキップした世代の RD エリアはデータディクショナリ用 RD エリアの定義情報と不整合となります。そのため，閉塞を解除する前に RD エリアの再作成が必要になります。詳細については「付録 J.3(1)　KFPH22032-W メッセージの出力と対処」を参照してください。

付録 J.2　RD エリアの状態ごとの HiRDB/SD 定義ユティリティ（pdsdbdef）の実行可否

RD エリアの状態ごとの pdsdbdef コマンドの実行可否を，次に示します。

表 J-1　RD エリア状態ごとの HiRDB/SD 定義ユティリティ（pdsdbdef）の実行可否（オープン契機が INITIAL の場合）

RD エリアの状態		ユティリティの実行可否	
		オリジナル世代またはカレント RD エリアの世代	左記以外のレプリカ RD エリアの世代
閉塞なし	オープン	○	○
	クローズ	×	×
コマンド閉塞	オープン	×	×
	クローズ	×	○※1
参照可能閉塞	オープン	×	×
	クローズ	×	×
参照可能バックアップ閉塞	オープン	×	×
	クローズ	×	×
更新可能バックアップ閉塞	オープン	×	×
	クローズ	×	×
障害閉塞	オープン	×	×
	クローズ	×	×
ログレス閉塞	オープン	×	×
	クローズ	×	×
同期化閉塞	オープン	○※2	○※2
	クローズ	×	×
オンライン再編成閉塞	オープン	×	×
	クローズ	×	×

（凡例）

○：実行できます。

×：実行できません。

注※1

世代の処理対象となるレコード型に関連するすべての RD エリアがコマンド閉塞かつクローズ状態の場合は，定義処理がスキップされます。この場合，KFPH22032-W メッセージが出力され，閉塞を解除する前に RD エリアを再作成する必要があります。詳細は「付録 J.3(1)　KFPH22032-W メッセージの出力と対処」を参照してください。

注※2

排他待ちになります。

表 J-2　RD エリア状態ごとの HiRDB/SD 定義ユティリティ（pdsdbdef）の実行可否（オープン契機が DEFER または SCHEDULE の場合）

RD エリアの状態		ユティリティの実行可否	
		オリジナル世代またはカレント RD エリアの世代	左記以外のレプリカ RD エリアの世代
閉塞なし	オープン	○	○
	クローズ	○	○
コマンド閉塞	オープン	×	×
	クローズ	×	○※1
参照可能閉塞	オープン	×	×
	クローズ	×	×
参照可能バックアップ閉塞	オープン	×	×
	クローズ	×	×
更新可能バックアップ閉塞	オープン	×	×
	クローズ	×	×
障害閉塞	オープン	×	×
	クローズ	×	×
ログレス閉塞	オープン	×	×
	クローズ	×	×
同期化閉塞	オープン	○※2	○※2
	クローズ	○※2	○※2
オンライン再編成閉塞	オープン	×	×
	クローズ	×	×

（凡例）

○：実行できます。

×：実行できません。

注※1

世代の処理対象となるレコード型に関連するすべての RD エリアがコマンド閉塞かつクローズ状態の場合は，定義処理がスキップされます。この場合，KFPH22032-W メッセージが出力され，閉塞を解除する前に RD エリアを再作成する必要があります。詳細については「付録 J.3(1)　KFPH22032-W メッセージの出力と対処」を参照してください。

注※2

排他待ちになります。

インナレプリカ機能使用時の pdsdbdef コマンドは，処理対象となるオリジナル RD エリアとレプリカ RD エリアすべてに対して，RD エリア内の定義情報を変更します。このため，処理対象となるレコード型に関連するすべての RD エリアに対して，処理できる状態かどうかをチェックします。チェック対象となる

pdsdbdef コマンドの SDB 定義文を次の表に示します。次の表に記載のない SDB 定義文については，インナレプリカ機能の使用の有無に関係なく実行できます。

表 J-3　チェックが発生する SDB 定義文の一覧とそのチェック内容

SDB 定義文	内容	チェック内容※		
		(a)	(b)	(c)
*ENTRY DIRECTORY 文	SDB ディレクトリ情報の追加	○	○	○
*ALTER DIRECTORY 文	SDB ディレクトリ情報の変更	○	○	○
*CHECK DICTIONARY 文	データベース定義のチェック	○	○	×
*DELETE DICTIONARY 文	SDB ディクショナリ情報の削除	×	○	○

（凡例）

○：チェックします。

×：チェックしません。

注※

チェック内容については，「付録 J.1　HiRDB/SD 定義ユティリティ（pdsdbdef）を実行するための条件」を参照してください。

付録 J.3　インナレプリカ機能を使用中に HiRDB/SD 定義ユティリティ（pdsdbdef）を実行する場合の注意事項

pdsdbdef コマンドを実行するときは，次の点に注意してインナレプリカの運用方法を決定してください。

- 対象となるレコード型に関連するすべての RD エリア※1 は，複製定義数を一致させ，レプリカ RD エリアの世代番号※2 をすべてそろえて運用してください。ただし，SDB データベースのサーバ間横分割をしている場合は，複製定義数または世代番号はサーバ内でそろっていれば問題ありません。
- RD エリア内に複数のレコード型のレコードまたはインデクスがある場合は，すべてのレコード型に関連する格納 RD エリアを一組にして運用してください。
- 複数の世代に pdsdbdef コマンドを実行する場合は，世代数分の処理時間と排他資源が必要になります。
- レプリカ RD エリアの実体がオリジナル RD エリアと別になっている状態で KFPH22032-W メッセージが出力された場合，レプリカ RD エリアの再作成が必要です。詳細については「付録 J.3(1)　KFPH22032-W メッセージの出力と対処」を参照してください。

注※1

pdrdrefls コマンドで関連 RD エリアを確認できます。

注※2

レプリカ RD エリアの定義数または世代番号の確認には，pddbls コマンドを使用します。

(1) KFPH22032-W メッセージの出力と対処

(a) 出力

RD エリアがコマンド閉塞かつクローズ状態のため，定義処理がスキップされると該当する世代のサーバで，バックエンドサーバごとに KFPH22032-W メッセージが出力されます。

なお，KFPH22032-W メッセージは複数回出力されます。*ENTRY DIRECTORY 文の場合は，「(レコード型数＋インデクス数）×該当する世代数（定義処理をスキップした世代数)」の数だけ，KFPH22032-W メッセージが出力されます。例えば，レコード型数が 2 でインデクス数が 1，該当する世代が 3 の場合，KFPH22032-W メッセージは 9 回（(2 ＋ 1）×3）出力されます。

DELETE DIRECTORY 文の場合は，「レコード型数×該当する世代数」の数だけ KFPH22032-W メッセージが出力されます。

(b) 対処

コマンド閉塞かつクローズ状態のため定義処理がスキップされた RD エリアは，RD エリア内の定義情報が変更されていません。ペアを分離しているなど，オリジナル RD エリアの更新が自動でレプリカ RD エリアに反映される状態でない場合，RD エリアの内容は，データディクショナリ用 RD エリアにあるレコード型やインデクスの定義情報と不整合となります。この状態でレプリカ RD エリアの閉塞を解除して使用すると，pdsdbdef コマンドがエラーとなったり，RD エリアが障害閉塞したりします。

RD エリアの閉塞を解除する前に次のどちらかの方法で，RD エリアの内容とレコード型やインデクスの定義情報を一致させてください。

- 該当する RD エリアにオリジナル RD エリア，またはカレント RD エリアをコピーして，レプリカデータベースを再作成します。この場合，コピー元とコピー先の RD エリアの構成情報（構成ファイル数，ページ長，セグメントサイズ，およびセグメント数）が不一致のときは pdmod コマンドの RD エリアの構成情報複写で実体に合わせて RD エリアの定義情報もコピーします。
- pdmod コマンドで該当する RD エリアを再初期化します。再初期化すると RD エリアのデータはすべて削除されるため，データを再度登録します。

付録 K　検索範囲決定の契機，横分割時の検索範囲，および入力情報とキーの定義の関係

付録 K.1　検索範囲決定の契機

検索範囲（検索条件によって決定されるレコードの集合ではなく，検索対象となる RD エリアの集まりを指す）は，SDB データベース種別によって異なります。4V FMB，および 4V AFM の SDB データベースの場合は，検索要求時および格納要求時に検索範囲が決定され，SD FMB の SDB データベースの場合は，検索要求時に検索範囲が決定されます。検索範囲が決定される場合と，決定されない場合について，次に示します。

検索範囲が決定されない場合は，前回の検索/格納時に決定された範囲を検索範囲とします。検索範囲が決定された場合，「表 K-7　RD エリア指定種別，およびレコード検索時の基点条件の組み合わせによる検索範囲（4V の SDB データベースの場合）」，または「表 K-8　RD エリア指定種別，およびレコード検索時の基点条件の組み合わせによる検索範囲（SD の SDB データベースの場合）」にある RD エリアが検索範囲となります。

4V AFM の SDB データベースの場合は，「表 K-7　RD エリア指定種別，およびレコード検索時の基点条件の組み合わせによる検索範囲（4V の SDB データベースの場合）」で決定された RD エリアから，個別開始時の RD エリアの指定種別が X'00'のとき，最初に検索した仮想ルートレコードの子レコードだけに限定されます。

検索および格納の操作，および SDB データベース種別の組み合わせごとに，検索範囲の決定契機を次に示します。

表 K-1　4V FMB の SDB データベース検索時の検索範囲決定

レコードの検索時または位置指示子の位置づけ時の指定			前回の検索要求または格納要求の結果	検索範囲の決定
レコード種別	指示コード	レコード型名（指定必須）		
ルートレコード	F	指定あり	−	○
	N	指定あり	SQLCODE≧0（100 を除く）	×
			SQLCODE ＜ 0 または SQLCODE ＝ 100	○
			結果なし（前回の検索要求および格納要求が未実施）	○
子レコード	F/N/L/P/U	指定あり	−	×

（凡例）

○：検索範囲が決定されます。

×：検索範囲が決定されません。
－：該当しません。
F：FIRST ポインタから検索します。
N：NEXT ポインタから検索します。
L：LAST ポインタから検索します。
P：PRIOR ポインタから検索します。
U：USER ポインタから検索します。

表 K-2　4V AFM の SDB データベース検索時の検索範囲決定

<table>
<tr><th colspan="3">レコード検索時の指定</th><th rowspan="2">前回の検索要求または格納要求の結果</th><th colspan="4" rowspan="2">そのほかの条件</th><th rowspan="2">検索範囲の決定</th></tr>
<tr><th>レコード種別</th><th>指示コード</th><th>レコード型名</th></tr>
<tr><td rowspan="7">子レコード※</td><td rowspan="2">F/L</td><td>指定あり</td><td>－</td><td colspan="4">－</td><td>○</td></tr>
<tr><td>指定なし</td><td>－</td><td colspan="4">－</td><td>○</td></tr>
<tr><td rowspan="4">N/P</td><td rowspan="3">指定あり</td><td rowspan="2">SQLCODE≧0（100 を除く）</td><td rowspan="2">直前に一括削除を実行</td><td colspan="3">実行あり</td><td>○</td></tr>
<tr><td>実行なし</td><td>ユーザ指定のレコード型名と，前回検索および格納したレコード型名</td><td>同じ</td><td>×</td></tr>
<tr><td>SQLCODE ＜ 0 または SQLCODE ＝ 100</td><td colspan="4">－</td><td>異なる</td><td>○</td></tr>
<tr><td>結果なし（前回の検索要求または格納要求が未実施）</td><td colspan="4">－</td><td>○</td></tr>
<tr><td rowspan="3">指定なし</td><td>SQLCODE≧0（100 を除く）</td><td colspan="4">－</td><td>×</td></tr>
<tr><td>SQLCODE ＜ 0 または SQLCODE ＝ 100</td><td colspan="4">－</td><td>○</td></tr>
<tr><td>結果なし（前回の検索要求または格納要求が未実施）</td><td colspan="4">－</td><td>○</td></tr>
</table>

（凡例）

○：検索範囲が決定されます。

×：検索範囲が決定されません。

－：該当しません。

F：FIRST ポインタから検索します。

N：NEXT ポインタから検索します。

L：LAST ポインタから検索します。

P：PRIOR ポインタから検索します。

注※

ユーザは仮想ルートレコードを意識しないで，子レコードに対してアクセス要求します。

表 K-3　SD FMB の SDB データベース検索時の検索範囲決定

レコードの検索時または位置指示子の位置づけ時の指定				検索時の状態		検索範囲の決定
レコード種別	位置指示子種別	指示コード	レコード型名(指定必須)	レコード位置指示子	検索順序※1	
ルートレコード	R	F	指定あり	－	－	○
		N	指定あり	空値	－	○
				空値ではない	無効	○※2
					有効	×
子レコード	R	F	指定あり	－	－	○
		N	指定あり	空値	－	○
				空値ではない	無効	○※2
					有効	×
	P	F/N/L/P	指定あり	－	－	×

（凡例）

○：検索範囲が決定されます。

×：検索範囲が決定されません。

－：該当しません。

R：レコード位置指示子

P：親子集合位置指示子

F：FIRST ポインタから検索します。

N：NEXT ポインタから検索します。

L：LAST ポインタから検索します。

P：PRIOR ポインタから検索します。

注※1

検索順序は該当するレコードの検索範囲が決定されたときに有効になり，次の処理が実行されたときに無効になります。

・位置指示子種別に'P'を指定している子レコードの検索

・レコードの格納

注※2

レコード位置指示子が位置づけられているレコードを基点とします。

表 K-4　4V FMB の SDB データベース格納時の検索範囲決定

レコード格納時の指定		検索範囲の決定
レコード種別	レコード型名（指定必須）	
ルートレコード	指定あり	○
子レコード	指定あり	×

（凡例）

○：検索範囲が決定されます。

×：検索範囲が決定されません。

表 K-5　4V AFM の SDB データベース格納時の検索範囲決定

レコード格納時の指定		検索範囲の決定
レコード種別	レコード型名	
子レコード（ユーザは仮想ルートレコードを意識しないで，子レコードに対してアクセス要求）	指定あり，またはなし	○

（凡例）

○：検索範囲が決定されます。

表 K-6　SD FMB の SDB データベース格納時の検索範囲決定

レコード格納時の指定		検索範囲の決定
レコード種別	レコード型名（指定必須）	
ルートレコード	指定あり	×
子レコード	指定あり	

（凡例）

×：検索範囲が決定されません。

付録 K.2　横分割時の検索範囲

RD エリアの指定種別，およびレコード検索時の基点条件の組み合わせによる検索範囲について，次の表に示します。

表 K-7　RD エリア指定種別，およびレコード検索時の基点条件の組み合わせによる検索範囲（4V の SDB データベースの場合）

項番	レコード検索時※1の指定	個別開始時の指定			検索対象※3	アクセス種別	検索範囲の説明の参照先
	基点条件※2	RD エリア指定有効オプション	RD エリア指定種別※6	基点となる RD エリア			
1	なし	−	'Y'	−	指定された RD エリアを「格納値」※4の昇順に検索します。	RD エリア指定検索※5	付録 K.2(1)(a)，付録 K.2(2)(a)
2			X'00'	−	すべての RD エリアを「格納値」※4の昇順に検索します。		付録 K.2(1)(b)，付録 K.2(2)(b)
3	あり	'Y'	'Y'	含む	条件で指定された検索開始位置からその RD エリアおよび指定された RD エリアを「格納値」※4の昇順に検索します。		付録 K.2(1)(c)，付録 K.2(2)(c)
4				含まない	条件で指定された「格納値」※4を含むレコード実現値と指定された RD エリアを「格納値」※4の昇順に検索します。		付録 K.2(1)(d)，付録 K.2(2)(d)
5			X'00'	−	条件で指定された検索開始位置からすべての RD エリアを「格納値」※4の昇順に検索します。		付録 K.2(1)(e)，付録 K.2(2)(e)

項番	レコード検索時※1の指定	個別開始時の指定			検索対象※3	アクセス種別	検索範囲の説明の参照先
	基点条件※2	RD エリア指定有効オプション	RD エリア指定種別※6	基点となる RD エリア			
6		X'00'	−	−	条件で指定された検索開始位置からその RD エリアだけをインデクスの昇順に検索します。	基点検索	付録 K.2(1)(f), 付録 K.2(2)(f)

（凡例）

−：状態を問いません。

（補足）

レコードの格納の場合，表 K-7 の項番 6 に該当します。

注※1

レコードの検索および複数レコードの検索を指します。

注※2

基点条件は，検索範囲の開始位置を指定するキー条件のことです。キーの条件で指定する条件式の比較記号が「GT or >」，「GE or >=」，「EQ or =」の場合に基点条件が「あり」となります。

例えば，「DBKEY>B」や「DBKEY=B」は検索範囲の開始位置 B が指定されているため，基点条件が「あり」となります。「DBKEY<H」は開始位置が指定されていないので基点条件は「なし」となります。

また，格納条件値分割をしたレコードの場合，基点条件の条件値であるレコード型の RD エリア分割キー値が，SDB データベース格納定義の WITHIN 句で定義されたレコード型の RD エリア分割キー値でないときは，エラーとなります。

注※3

検索範囲の決定は，「表 K-1　4V FMB の SDB データベース検索時の検索範囲決定」，「表 K-2　4V AFM の SDB データベース検索時の検索範囲決定」に記載の契機で行われます。以降の NEXT および PRIOR 指定の検索は，そのときに決定した検索範囲を対象に行われます。

注※4

境界値分割の場合，「格納値」を「境界値に指定された値の範囲」と読み替えてください。

注※5

シーケンシャルインデクスの先頭の構成要素以外を，レコード型の RD エリア分割キーに指定し，かつ RD エリアを指定して検索する場合，次の条件を満たしていないときは，エラーとなります。

1. レコード型の RD エリア分割キーに指定した構成要素よりも前のすべての構成要素に対して KEYDEF 句を指定している。
2. 1.で指定した KEYDEF 句ごとの DATA 句で定義したキー値は 1 つだけである。

注※6

4V AFM の SDB データベースで RD エリア指定種別が X'00'の場合は，この表で決定される検索範囲から，さらに範囲が限定されます。

表 K-8　RD エリア指定種別，およびレコード検索時の基点条件の組み合わせによる検索範囲（SD の SDB データベースの場合）

項番	DML の指定	SDB 用 UAP 環境定義の指定		検索対象※2	検索範囲の説明の参照先
	基点条件※1	subschema オペランドの-r オプション			
		指定有無	基点となる RD エリア		
1	なし	あり	－	指定された RD エリアを「格納値」※3 の昇順に検索します。	付録 K.2(1)(a)，付録 K.2(2)(a)
2		なし		すべての RD エリアを「格納値」※3 の昇順に検索します。	付録 K.2(1)(b)，付録 K.2(2)(b)
3	あり	あり	含む	条件で指定された検索開始位置からその RD エリアおよび指定された RD エリアを「格納値」※3 の昇順に検索します。	付録 K.2(1)(c)，付録 K.2(2)(c)
4			含まない※4	指定された RD エリアの中で，基点条件より大きくて，最も近い値を検索開始位置として，指定された RD エリアを「格納値」※3 の昇順に検索します。	付録 K.2(1)(g)，付録 K.2(2)(g)
5		なし	－	条件で指定された検索開始位置からすべての RD エリアを「格納値」※3 の昇順に検索します。	付録 K.2(1)(e)，付録 K.2(2)(e)

（凡例）

－：状態を問いません。

（補足）

レコードの格納の場合，検索範囲を決定しません。

注※1

基点条件は，検索範囲の開始位置を指定するキー条件のことです。キーの条件で指定する条件式の構成要素がインデクスのキーの先頭構成要素，またはインデクスのキーの構成要素が先頭から連続していてインデクスのキーの構成要素以外を含まない集団項目で，比較記号が「GT or >」，「GE or >=」，「EQ or =」の条件がある場合に基点条件が「あり」となります。

インデクスのキーの先頭構成要素，またはインデクスのキーの構成要素が先頭から連続していてインデクスのキーの構成要素以外を含まない集団項目で，比較記号が「GT or >」，「GE or >=」，「EQ or =」の条件が 1 つしか指定されていない場合は，その条件が基点条件となります。ただし，条件が複数指定されている場合は，インデクスのキーの先頭構成要素に対応する値が最大の条件が基点条件となります。

インデクスのキーの先頭構成要素に対応する値が最大の条件が 1 つしか指定されていない場合は，その条件が基点条件となります。インデクスのキーの先頭構成要素に対応する値が最大の条件が複数指定されている場合の基点条件は次のとおりです。

- 比較記号が「GT or >」の条件あり

  インデクスのキーの先頭構成要素に対応する値が最大の比較記号が「GT or >」の条件が 1 つの場合は，その条件が基点条件となりますが，複数の場合は，先に指定されている条件が基点条件となります。

- 比較記号が「GT or >」の条件なし

  インデクスのキーの先頭構成要素に対応する値が最大の比較記号が「GE or >=」，「EQ or =」の条件で，先に指定されている条件が基点条件となります。

注※2

検索範囲の決定は，「表 K-3　SD FMB の SDB データベース検索時の検索範囲決定」に記載の契機で行われます。以降の NEXT および PRIOR 指定の検索は，そのときに決定した検索範囲を対象に行われます。

注※3

境界値分割の場合，「格納値」を「境界値に指定された値の範囲」と読み替えてください。

注※4

キーの条件に該当する RD エリアを指定していない場合，NOT FOUND となります。

検索対象となる範囲の例を次に示します。

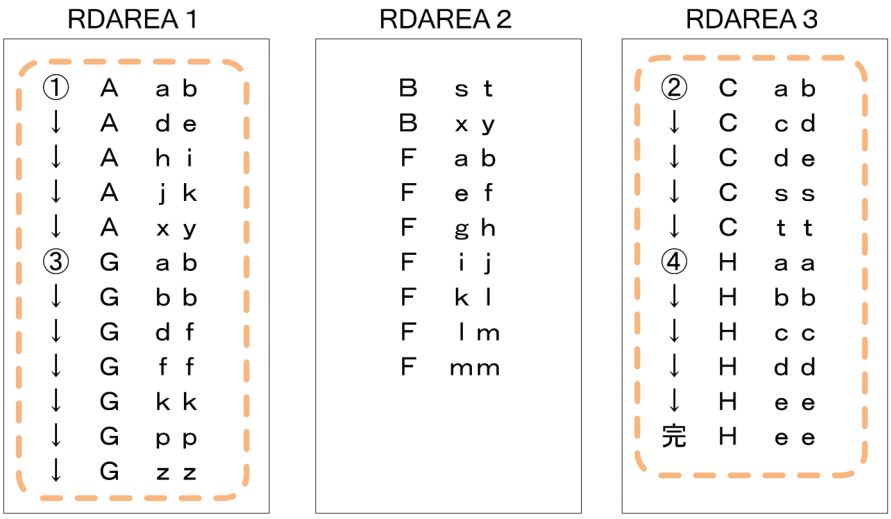
(1)　DBKEY の先頭列を指定したときの格納条件指定による横分割の定義例

DBKEY の先頭列を指定したときの格納条件指定による横分割の定義例を次に示します。

```
WITHIN ( (RDAREA1)TENBAN=(A,G)
       , (RDAREA2)TENBAN=(B,F)
       , (RDAREA3)TENBAN=(C,H))
```

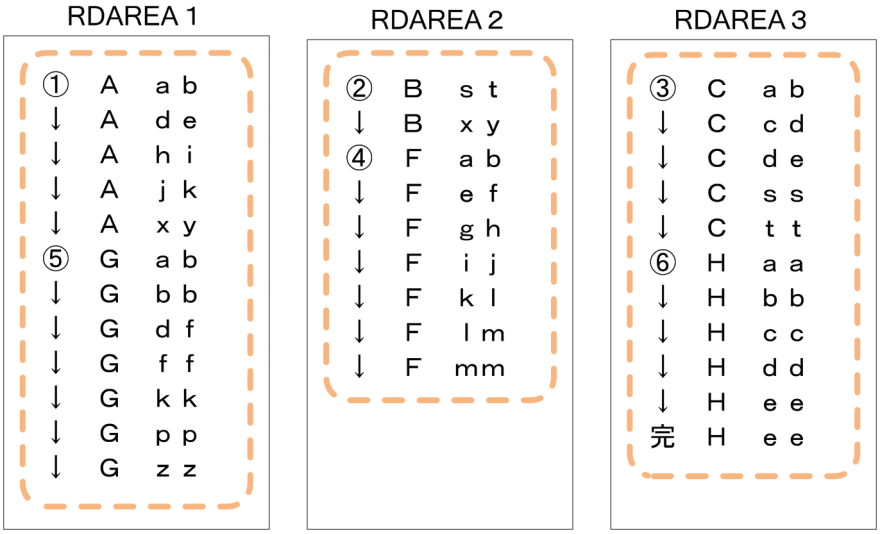
(a)　RD エリアの指定に RDAREA1 および RDAREA3 を指定した場合

RD エリアの指定に RDAREA1 および RDAREA3 を指定した場合，点線枠内が検索範囲となります。分割条件値が昇順となる A→C→G→H の順に検索します。



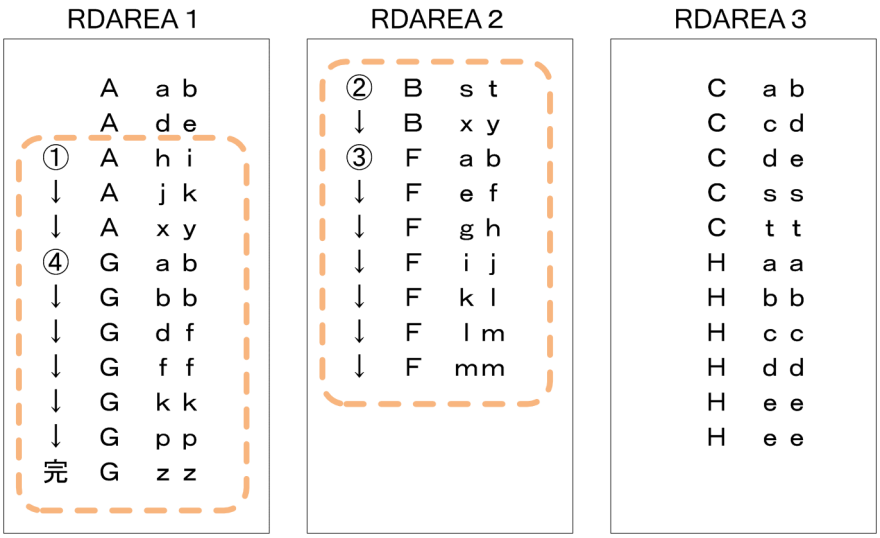
(b) すべての RD エリアを指定した場合

すべての RD エリアを指定した場合，点線枠内が検索範囲となります。分割条件値が昇順となる A→B→C→F→G→H の順に検索します。



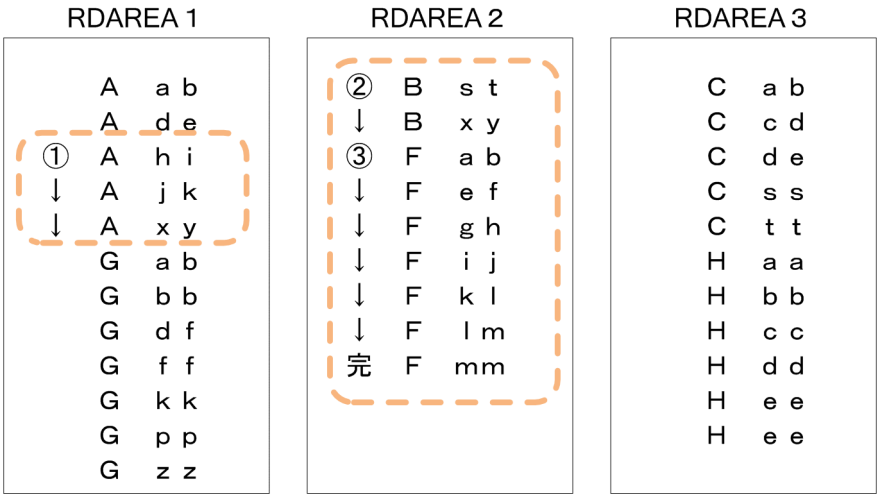
(c) 基点となる条件に「DBKEY≧Ahi」を指定し，RD エリアの指定に RDAREA1 および RDAREA2 を指定した場合

基点となる条件に「DBKEY≧Ahi」を指定した場合，RD エリアの指定に RDAREA1 および RDAREA2 を指定すると，点線枠内が検索範囲となります。分割条件値が昇順となる A→B→F→G の順に検索します。



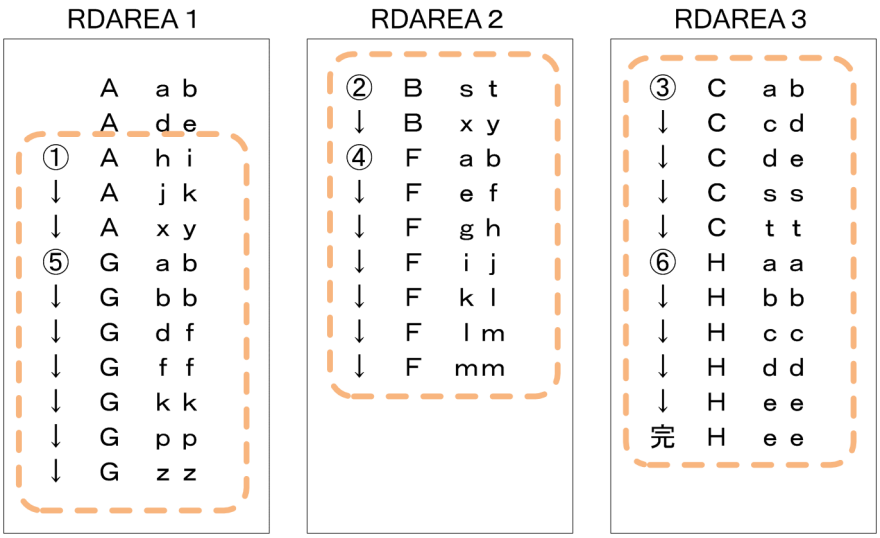
(d) 基点となる条件に「DBKEY≧Ahi」を指定し，RD エリアの指定に RDAREA2 を指定した場合

基点となる条件に「DBKEY≧Ahi」を指定した場合，RD エリアの指定に RDAREA2 を指定すると，点線枠内が検索範囲となります。分割条件値が昇順の A→B→F の順に検索します。



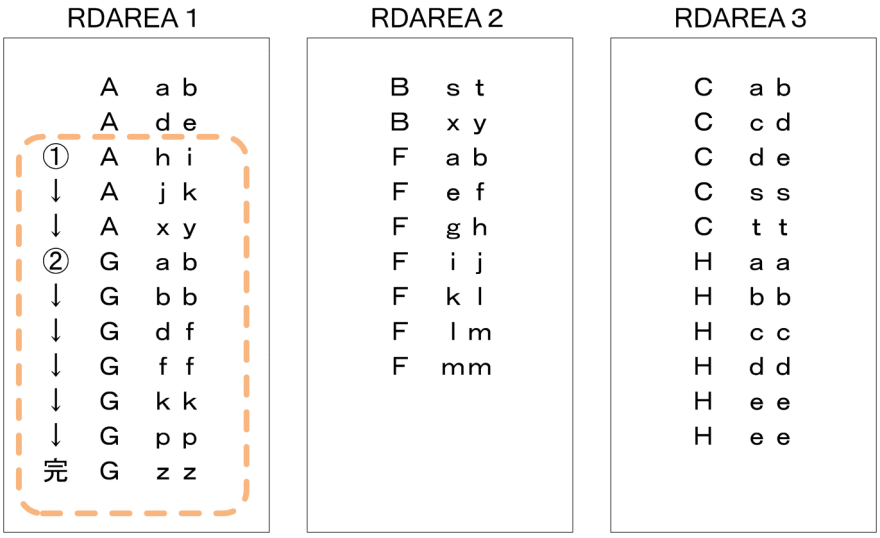
(e) 基点となる条件に「DBKEY≧Ahi」を指定し，RD エリアの指定に 0x00 を指定した場合

基点となる条件に「DBKEY≧Ahi」を指定した場合，RD エリアの指定に 0x00 を指定すると，点線枠内が検索範囲となります。分割条件値が昇順となる A→B→C→F→G→H に検索します。



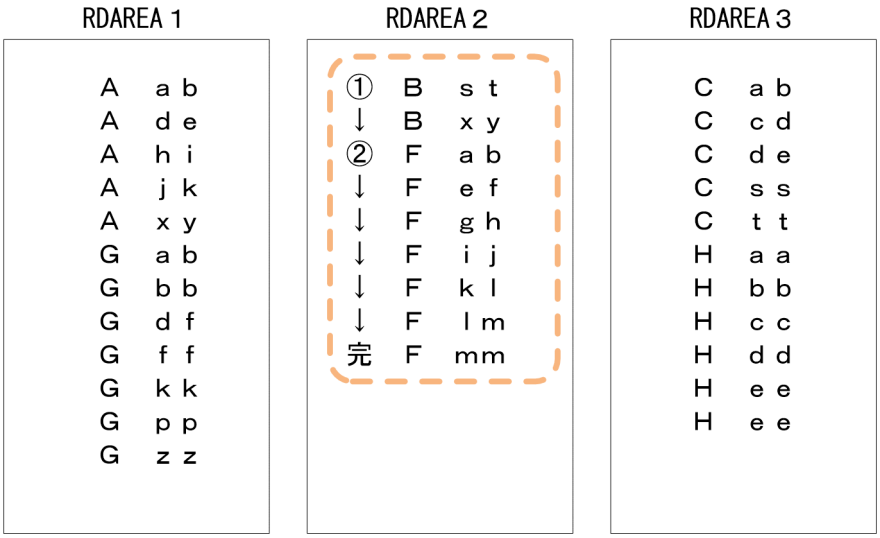
(f) 基点となる条件に「DBKEY≧Ahi」を指定した場合

基点となる条件に「DBKEY≧Ahi」を指定した場合，点線枠内が検索範囲となります。インデクスキーの昇順に検索します。



(g) 基点となる条件に「DBKEY≧Ahi」を指定し，RD エリアの指定に RDAREA2 を指定した場合

基点となる条件に「DBKEY≧Ahi」を指定した場合，RD エリアの指定に RDAREA2 を指定すると，点線枠内が検索範囲となります。分割条件値が昇順となる B→F に検索します。



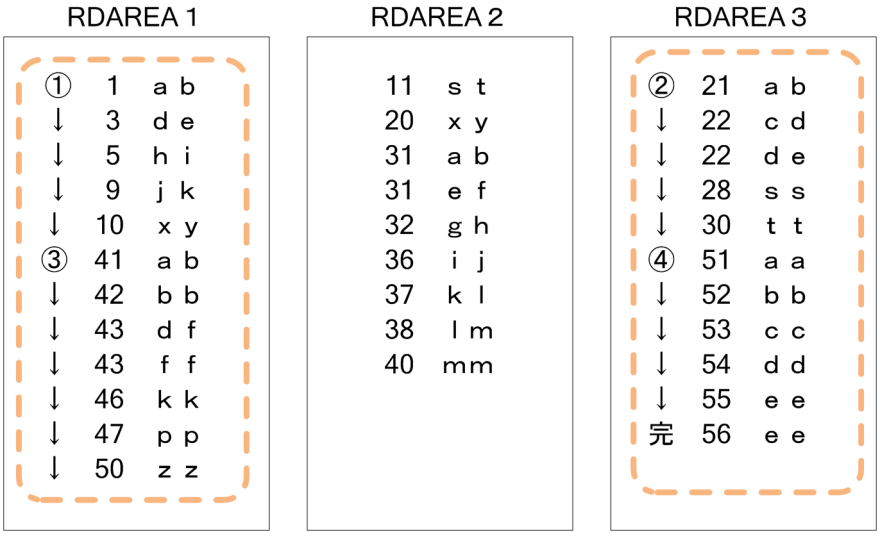
(2)　境界値指定による横分割の定義例

境界値指定による横分割の定義例を次に示します。

```
DEPENDING ON C1
 WITHIN ( (RDAREA1)10, (RDAREA2)20, (RDAREA3)30, (RDAREA2)40, (RDAREA1)50,
          (RDAREA3))
```

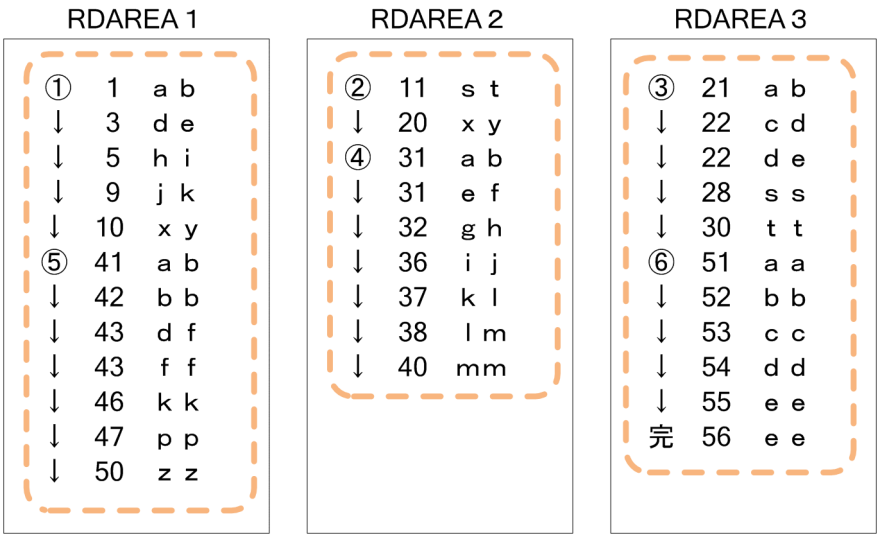
(a)　RD エリアの指定に RDAREA1 および RDAREA3 を指定した場合

RD エリアの指定に RDAREA1 および RDAREA3 を指定した場合，点線枠内が検索範囲となります。分割条件値の範囲が，1～10→21～30→41～50→51～となるように検索します。



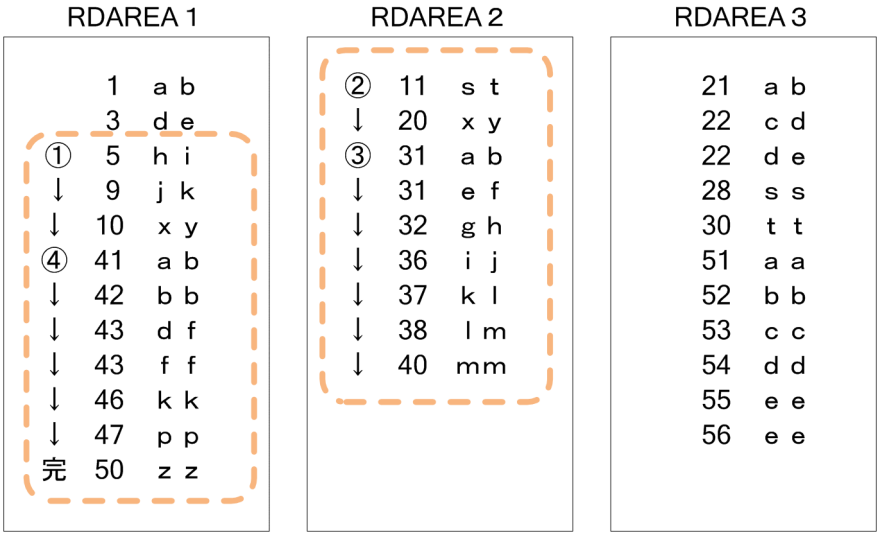
(b)　すべての RD エリアを指定した場合

すべての RD エリアを指定した場合，点線枠内が検索範囲となります。分割条件値が，1～10→11～20→21～30→31～40→41～50→51～となるように検索します。



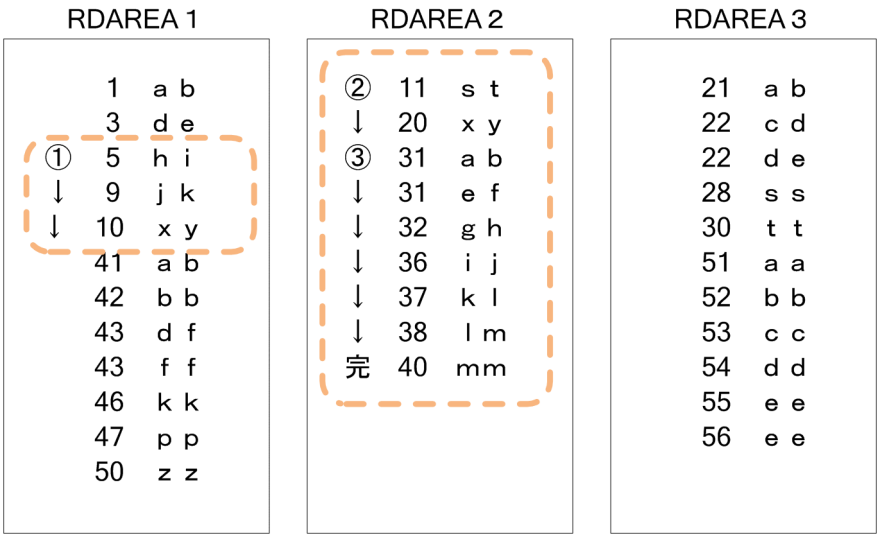
(c) 基点となる条件に「DBKEY≧5hi」を指定し，RD エリアの指定に RDAREA1 および RDAREA2 を指定した場合

基点となる条件に「DBKEY≧5hi」を指定した場合，RD エリアの指定に RDAREA1 および RDAREA2 を指定すると，点線枠内が検索範囲となります。分割条件値が，5～10→11～20→31～40→41～50 となるように検索します。



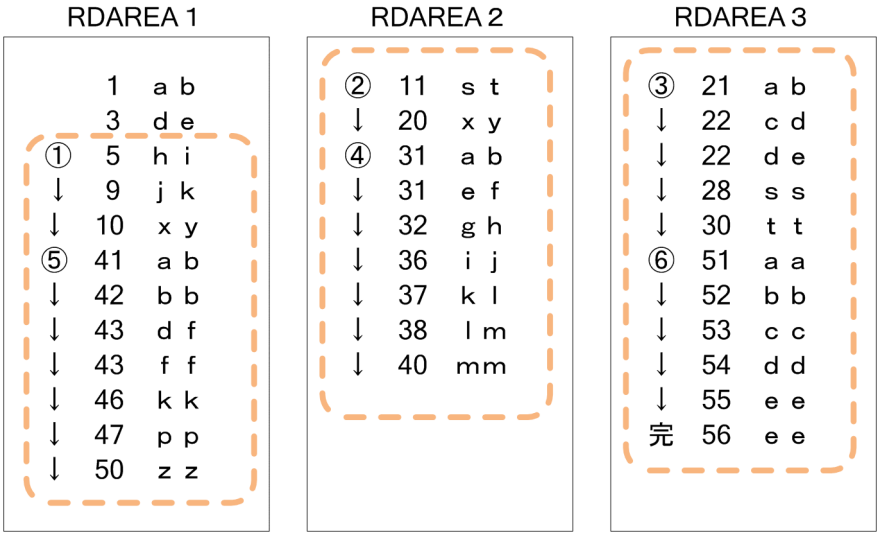
(d) 基点となる条件に「DBKEY≧5hi」を指定し，RD エリアの指定に RDAREA2 を指定した場合

基点となる条件に「DBKEY≧5hi」を指定した場合，RD エリアの指定に RDAREA2 を指定すると，点線枠内が検索範囲となります。分割条件値が，5～10→11～20→31～40 となるように検索します。



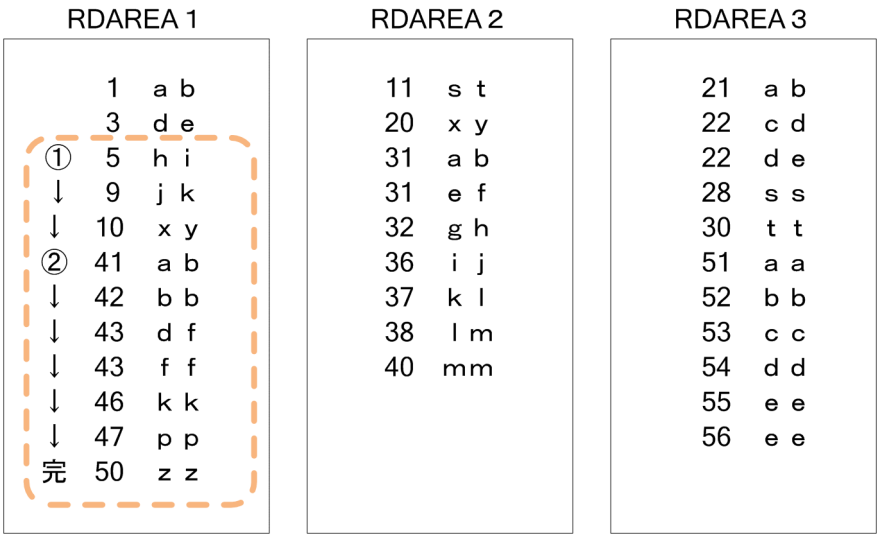
(e) 基点となる条件に「DBKEY≧5hi」を指定し，RD エリアの指定に 0x00 を指定した場合

基点となる条件に「DBKEY≧5hi」を指定した場合，RD エリアの指定に 0x00 を指定すると，点線枠内が検索範囲となります。分割条件値が，5～10→11～20→21～30→31～40→41～50→51～となるように検索します。



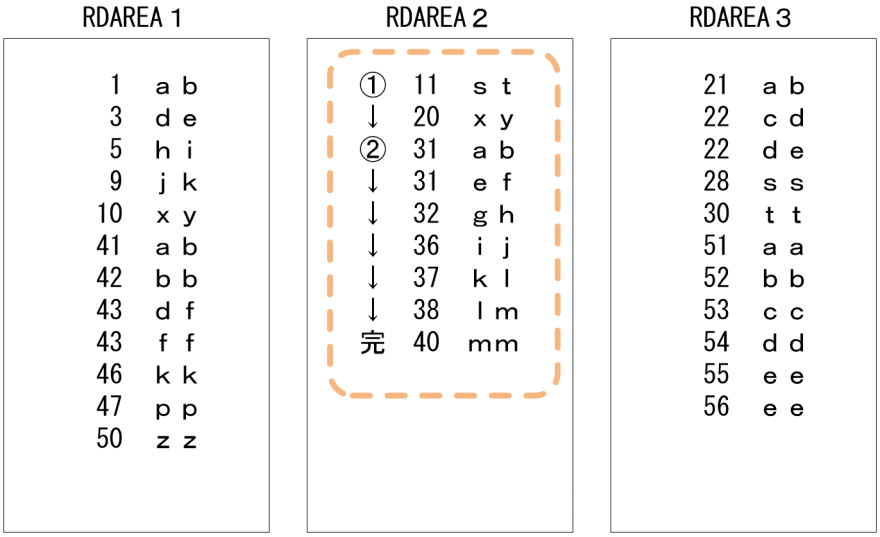
(f) 基点となる条件に「DBKEY≧5hi」を指定した場合

基点となる条件に「DBKEY≧5hi」を指定した場合，点線枠内が検索範囲となります。インデクスキーの昇順に検索します。



(g) 基点となる条件に「DBKEY≧5hi」を指定し，RD エリアの指定に RDAREA2 を指定した場合

基点となる条件に「DBKEY≧5hi」を指定した場合，RD エリアの指定に RDAREA2 を指定すると，点線枠内が検索範囲となります。分割条件値が，11～20→31～40 となるように検索します。



付録 K.3 入力情報とキーの定義の関係【4V FMB，4V AFM】

4V FMB または 4V AFM の SDB データベースの場合の，入力情報とキーの定義の関係について，次に示します。

表 K-9　入力情報とキーの定義の関係（1/2）

要求				SDB データベース種別と定義					
機能	レコード型名	DBKEY	RD エリアの指定	4V FMB	4V AFM				
					DBKEY は K，N の構成要素	DBKEY は K，L の構成要素	DBKEY は K，A の構成要素	DBKEY は K，L ＋ K，N の構成要素	DBKEY は K，A ＋ K，N の構成要素
					子レコード型数は 1				
キーの定義例					(a)	(b)	(c)	(d)	(e)
レコードの検索	指定あり	指定あり	指定あり	○	○	○	○	○	○
			指定なし	○	○	○	○	○	○
		指定なし	指定あり	○	○	○※11	○※11	○※11	○※11
			指定なし	○	○	○※1	○※1	○※1	○※1
	指定なし	指定あり	指定あり	−	○※2	○※2	○※2	○※2	○※2
			指定なし	−	○※2	○※2	○※2	○※2	○※2
		指定なし	指定あり	−	○	○※11	○※11	○※11	○※11
			指定なし	−	○※2	○※3	○※3	○※3	○※3
レコードの格納	指定あり	指定あり		○	○	○	○	○	○
		指定なし		○	○	○※4	× SQLCODE =-66214	○※4	× SQLCODE =-66214
	指定なし	指定あり		−	○※2	○※2	○※2	○※2	○※2
		指定なし		−	○※2	○※5	×SQLCOD E=-66214	○※5	×SQLCOD E=-66214
レコードの更新	指定あり	指定あり※8		−	−	−	−	−	−
		指定なし		○	○※6	○※6	○※6	○※6	○※6
	指定なし	指定あり※8		−	−	−	−	−	−
		指定なし		−	○※7	○※7	○※7	○※7	○※7
レコード	指定あり	指定あり※8		−	−	−	−	−	−

要求				SDB データベース種別と定義					
機能	レコード型名	DBKEY	RD エリアの指定	4V FMB	4V AFM				
					DBKEY は K，N の構成要素	DBKEY は K，L の構成要素	DBKEY は K，A の構成要素	DBKEY は K，L＋K，N の構成要素	DBKEY は K，A＋K，N の構成要素
					子レコード型数は 1				
キーの定義例					(a)	(b)	(c)	(d)	(e)
の削除		指定なし		○	○	○	○	○	○
レコードの一括削除	指定あり	指定あり		－	－	－	－	－	－
		指定なし		－	－	－	－	－	－
	指定なし	指定あり		－	○※2	○※2	○※2	○※2	○※2
		指定なし		－	○※2	× SQLCODE=-66213	× SQLCODE=-66213	× SQLCODE=-66213	× SQLCODE=-66213

表 K-10　入力情報とキーの定義の関係（2/2）

要求				SDB データベース種別と定義				
機能	レコード型名	DBKEY	RD エリアの指定	4V DAM または 4V MAM		4V DAM		そのほか
				DBKEY は K，R または K，M の構成要素	DBKEY は K，R＋K，N または K，M＋K，N の構成要素	DBKEY は K，R または K，M の構成要素	DBKEY は K，R＋K，N または K，M＋K，N の構成要素	
				キー値とレコードの対応は 1 対 1		キー値とレコードの対応は n（2 以上）対 1		
				子レコード型数は n				
キーの定義例				(f)	(g)	(h)	(i)	(j)
レコードの検索	指定あり	指定あり	指定あり	○	○	○	○	○
			指定なし	○	○	○	○	○
		指定なし	指定あり	○	○	○※12	○※12	○※11
			指定なし	○※9	○※9	× SQLCODE=-66213	× SQLCODE=-6 6213	× SQLCODE=-66 213

要求				SDB データベース種別と定義				
機能	レコード型名	DBKEY	RD エリアの指定	4V DAM または 4V MAM		4V DAM		そのほか
				DBKEY は K，R または K，M の構成要素	DBKEY は K，R ＋ K，N または K，M ＋ K，N の構成要素	DBKEY は K，R または K，M の構成要素	DBKEY は K，R ＋ K，N または K，M ＋ K，N の構成要素	
				キー値とレコードの対応は 1 対 1		キー値とレコードの対応は n（2 以上）対 1		
				子レコード型数は n				
キーの定義例				(f)	(g)	(h)	(i)	(j)
	指定なし	指定あり	指定あり	○※5	○※5	○※5	○※5	○※5
			指定なし	○※5	○※5	○※5	○※5	○※5
		指定なし	指定あり	○※13	○※13	○※12	○※12	○※11
			指定なし	× SQLCODE=-66217	× SQLCODE=-66217	× SQLCODE=-66217	× SQLCODE=-66217	× SQLCODE=-66217
レコードの格納	指定あり	指定あり		○	○	○	○	○
		指定なし		○※9	○※9	× SQLCODE=-66214	× SQLCODE=-66214	× SQLCODE=-66214
	指定なし	指定あり		○※10	○※10	○※10	○※10	○※10
		指定なし		× SQLCODE=-66214	× SQLCODE=-66214	× SQLCODE=-66214	× SQLCODE=-66214	× SQLCODE=-66214
レコードの更新	指定あり	指定あり		−	−	−	−	−
		指定なし		○※6	○※6	○※6	○※6	○※6
	指定なし	指定あり		−	−	−	−	−
		指定なし		○※7	○※7	○※7	○※7	○※7
レコードの	指定あり	指定あり※8		−	−	−	−	−

要求				SDB データベース種別と定義				
機能	レコード型名	DBKEY	RD エリアの指定	4V DAM または 4V MAM		4V DAM		そのほか
				DBKEY は K，R または K，M の構成要素	DBKEY は K，R ＋ K，N または K，M ＋ K，N の構成要素	DBKEY は K，R または K，M の構成要素	DBKEY は K，R ＋ K，N または K，M ＋ K，N の構成要素	
				キー値とレコードの対応は 1 対 1		キー値とレコードの対応は n（2 以上）対 1		
				子レコード型数は n				
キーの定義例				(f)	(g)	(h)	(i)	(j)
削除		指定なし		○	○	○	○	○
レコードの一括削除	指定あり	指定あり		－	－	－	－	－
		指定なし		－	－	－	－	－
	指定なし	指定あり		○※10	○※10	○※10	○※10	○※10
		指定なし		× SQLCODE=-66213	× SQLCODE=-66213	× SQLCODE=-66213	× SQLCODE=-66213	× SQLCODE=-66213

（凡例）

○：要求できます。

×：要求できません。

－：API でエラーとなります。

空白：該当しません。

(a)～(j)のキーの定義例については，「付録 K.3(1)　キーの定義例」の(a)～(j)を参照してください。

DBKEY については，4V FMB の SDB データベースの場合は「11.5.1(5)(d)　レベル番号，ITEM」を参照してください。4V AFM の SDB データベースの場合は「11.6.1(5)(d)　レベル番号，ITEM」を参照してください。

注※1

KEYDEF 句の定義からキー（K，L または K，A）を取得します。

キーが定義されていない場合は，SQLCODE=-66220 となります。キーが複数定義されている場合は，SQLCODE=-66213 となります。

注※2

SDB データベース定義からレコード型名を取得します。

注※3

SDB データベース定義からレコード型名を取得します。キー（K，L または K，A）は KEYDEF 句の定義から取得します。

キーが定義されていない場合は，SQLCODE=-66220 となります。キーが複数定義されている場合は，SQLCODE=-66213 となります。

注※4

KEYDEF 句の定義からキー（K，L）を取得します。

キーが定義されていない場合は，SQLCODE=-66220 となります。キーが複数定義されている場合は，SQLCODE=-66214 となります。

注※5

定義からレコード型名を取得します。キー（K，L）は KEYDEF 句の定義から取得します。

キーが定義されていない場合は，SQLCODE=-66220 となります。キーが複数定義されている場合は，SQLCODE=-66214 となります。

注※6

レコード型名が指定ありで，検索または格納で更新対象のレコードに位置づけしていることが前提となります。レコード型名が指定ありで，位置づけしていない場合は，SQLCODE=-66207 となります。

注※7

レコード型名が指定なしで，検索または格納で更新対象のレコードに位置づけしていることが前提となります。レコード型名が指定なしで位置づけしていない場合は，SQLCODE=-66207 となります。

注※8

条件式が指定されていても，その指定は無効となります。

注※9

レコード分割キーは，指定のレコード型名と KEYDEF 句の定義から取得します。

注※10

指定の DBKEY（レコード分割キー）からレコード型名を取得します。

注※11

4V 固有エリア（要求部）の一連番号指定，またはキー以外の条件でユーザキーを指定する場合，KEYDEF 句の定義でシーケンシャルインデクスキーのキー値が複数定義されているときは，SQLCODE=-66213 となります。

注※12

4V 固有エリア（要求部）の一連番号指定，またはキー以外の条件でユーザキーを指定する場合は，SQLCODE=-66213 となります。

注※13

4V 固有エリア（要求部）の一連番号指定，またはキー以外の条件でユーザキーを指定する場合は，SQLCODE=-66217 となります。

(1) キーの定義例

TYPE K，R または TYPE K，M を指定した場合，子レコード型は 2 つ以上指定が必要ですが，ここでは記載を省略しています。

(a) DBKEY が一連番号だけの場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_LN INTEGER       TYPE K,N
```

(b) DBKEY がそのほかのキー（TYPE K，L）だけの場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,L
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,L
  02 DBKEY_NN   INTEGER       TYPE N,N
```

(c) DBKEY がレコード型の RD エリア分割キー（TYPE K，A）だけの場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,A
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,A
  02 DBKEY_NN   INTEGER       TYPE N,N
```

(d) DBKEY がそのほかのキー（TYPE K，L）と一連番号だけの場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,L
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,L
    03 DBKEY_KN INTEGER       TYPE K,N
```

(e) DBKEY がレコード型の RD エリア分割キー（TYPE K，A）と一連番号だけの場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,A
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,A
    03 DBKEY_KN INTEGER       TYPE K,N
```

(f) DBKEY がレコード分割キー（TYPE K，R）またはレコード分割キー＋レコード型の RD エリア分割キー（TYPE K，M）だけの場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,R （またはTYPE K,M）
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
```

```
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,R （またはTYPE K,M）
  02 DBKEY_NN   INTEGER      TYPE N,N
```

(g) DBKEY がレコード分割キー（TYPE K，R）と一連番号だけの場合（またはレコード分割キー＋レコード型の RD エリア分割キー（TYPE K，M）と一連番号だけの場合）

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,R （またはTYPE K,M）
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,R （またはTYPE K,M）
    03 DBKEY_KN INTEGER      TYPE K,N
```

(h) DBKEY がレコード分割キー（TYPE K，R）またはレコード分割キー＋レコード型の RD エリア分割キー（TYPE K，M）だけの場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,R （またはTYPE K,M）
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,R （またはTYPE K,M）
  02 DBKEY_NN   INTEGER      TYPE N,N
```

(i) DBKEY がレコード分割キー（TYPE K，R）と一連番号だけの場合（またはレコード分割キー＋レコード型の RD エリア分割キー（TYPE K，M）と一連番号だけの場合）

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,R （またはTYPE K,M）
RECORD A1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY
    03 DBKEY_L2 CHARACTER  4  TYPE K,R （またはTYPE K,M）
    03 DBKEY_KN INTEGER      TYPE K,N
```

(j) DBKEY がレベル 2～7 のキーと一連番号の場合

```
RECORD R1
  02 DBKEY_L1   CHARACTER  4  TYPE D,L
  02 DBKEY_L2   CHARACTER  4  TYPE K,L
  02 DBKEY_L3   CHARACTER  4  TYPE K,L
  02 DBKEY_L4   CHARACTER  4  TYPE K,L
  02 DBKEY_L5   CHARACTER  4  TYPE K,L
  02 DBKEY_L6   CHARACTER  4  TYPE K,L
  02 DBKEY_L7   CHARACTER  4  TYPE K,R
RECORD A1
```

```
    02 DBKEY_L1   CHARACTER  4  TYPE D,L
    02 DBKEY
      03 DBKEY_L2 CHARACTER  4  TYPE K,L
      03 DBKEY_L3 CHARACTER  4  TYPE K,L
      03 DBKEY_L4 CHARACTER  4  TYPE K,L
      03 DBKEY_L5 CHARACTER  4  TYPE K,L
      03 DBKEY_L6 CHARACTER  4  TYPE K,L
      03 DBKEY_L7 CHARACTER  4  TYPE K,R
      03 DBKEY_KN INTEGER       TYPE K,N
```

付録 L　SDB データベースを操作する API または DML と，データページの排他解除の有無の関係

排他自動解除機能を使用すると，SDB データベースを操作した結果，レコード位置指示子が指さなくなったレコード実現値が格納されているデータページ（ページまたはサブページ）に対する排他を，API または DML による SDB データベースの操作完了時に解除します。

SDB データベースを操作する API または DML と，データページの排他解除の有無（排他自動解除機能の対象となるデータページ）の関係を次の表に示します。

表 L-1　SDB データベースを操作する API または DML と，データページの排他解除の有無の関係

実行する SDB データベースを操作する API または DML		対象となるレコード位置指示子	対象となるレコード位置指示子に対する操作						
			生成	消滅※1		更新			
				空	X	空 → 空	空 → X	X → 空	X → X
個別開始		操作対象となる SDB データベースに対応するすべての位置指示子	△	－	－	－	－	－	－
個別終了			－	△	○	－	－	－	－
4V FMB または SD FMB	レコードの検索（FETCH）	操作対象となるレコードビュー※6 に対応する位置指示子	－	－	－	△	△	○	○
		操作対象となるレコードビューの定義上，下位に位置するレコードビューに対応する位置指示子	－	－	－	△	－	○	－
	位置指示子の位置づけ（FIND）	操作対象となるレコードビュー※6 に対応する位置指示子	－	－	－	△	△	○	○
		操作対象となるレコードビューの定義上，下位に位置するレコードビューに対応する位置指示子	－	－	－	△	－	○	－
	複数レコードの検索（FETCHDB ALL）	なし	－	－	－	－	－	－	－
	レコードの格納（STORE）	操作対象となるレコードビューに対応する位置指示子	－	－	－	－	△	－	○
		操作対象となるレコードビューの定義上，下位に位置するレコードビューに対応する位置指示子	－	－	－	△	－	○	－
	レコードの削除（ERASE）	操作対象となるレコードビューに対応する位置指示子	－	－	－	－	－	△	－

実行するSDBデータベースを操作するAPIまたはDML		対象となるレコード位置指示子		対象となるレコード位置指示子に対する操作						
				生成	消滅※1		更新			
					空	X	空 → 空	空 → X	X → 空	X → X
		操作対象となるレコードビューの定義上，下位に位置するレコードビューに対応する位置指示子		−	−	−	△	−	△	−
	レコードの更新（MODIFY）	なし		−	−	−	−	−	−	−
4V AFM	レコードの検索（FETCH）	仮想ルートレコードに対応する位置指示子	操作対象となるレコード実現値の仮想ルートレコード実現値に位置づいていない場合	−	−	−	△	△	○	○
			上記以外	−	−	−	−	−	−	−
		操作対象となるレコードビューに対応する位置指示子		−	−	−	△	△	○	○
		操作対象となるSDBデータベースに対応する，上記以外の位置指示子※3	操作対象となるレコード実現値の仮想ルートレコード実現値に位置づいていない場合	−	−	−	△	−	○	−
			上記以外	−	−	−	−	−	−	−
	レコードの格納（STORE）※2	仮想ルートレコードに対応する位置指示子	操作対象となるレコード実現値の仮想ルートレコード実現値に位置づいていない場合	−	−	−	−	△	−	○
			上記以外	−	−	−	−	−	−	−
		操作対象となるレコードビューに対応する位置指示子		−	−	−	−	△	−	○
		操作対象となるSDBデータベースに対応する，上記以	操作対象となるレコード実現値の仮想ルートレコード実現値に位	−	−	−	△	−	○	−

実行するSDBデータベースを操作するAPIまたはDML		対象となるレコード位置指示子		対象となるレコード位置指示子に対する操作						
				生成	消滅※1		更新			
					空	X	空→空	空→X	X→空	X→X
		外の位置指示子※3	置づいていない場合							
			上記以外	－	－	－	－	－	－	－
	レコードの削除（ERASE）	仮想ルートレコードに対応する位置指示子		－	－	－	－	－	－	－
		操作対象となるレコードビューに対応する位置指示子		－	－	－	－	－	△	－
		操作対象となるSDBデータベースに対応する上記以外の位置指示子※3		－	－	－	－	－	－	－
	レコードの更新（MODIFY）	なし		－	－	－	－	－	－	－
	レコードの一括削除	仮想ルートレコードに対応する位置指示子		－	－	－	△	△	○	○
		操作対象となるSDBデータベースに対応する上記以外の位置指示子※3		－	－	－	△	－	○	－
	構成情報取得	仮想ルートレコードに対応する位置指示子		－	－	－	△	△	○	○
		操作対象となるSDBデータベースに対応する上記以外の位置指示子※4		－	－	－	－	△	－	－

（凡例）

○：該当する操作があります（排他は解除されます※5）。

△：該当する操作があります（排他は解除されません）。

－：該当する操作はありません。

X：対象となるレコード位置指示子が，対応する任意のレコード実現値を指している状態

空：対象となるレコード位置指示子がどのレコード実現値も指していない状態（空値）

A→B：A（APIまたはDML実行前）→B（APIまたはDML実行後）

注※1

空値のレコード位置指示子を消滅させる場合と，レコード実現値を指している位置指示子を消滅させる場合では，排他自動解除機能の動作は変わります。

注※2

4V AFM の SDB データベースのレコードの格納（STORE）には，次の 3 種類があります。

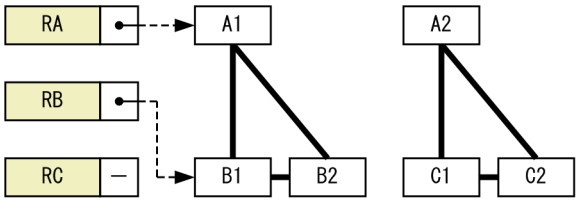
- 仮想ルートレコードのレコード実現値の追加を伴うもの
- 子レコード実現値だけを追加するもの
- 内部的にレコードの更新（MODIFY）を行うもの

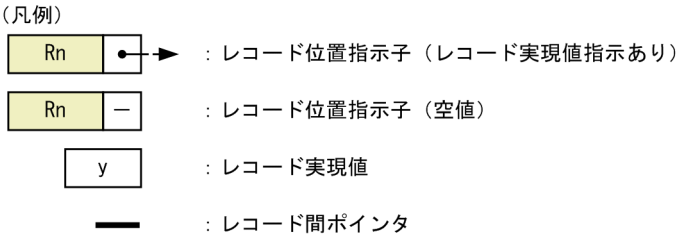
これらを実行した場合の位置指示子の動きと排他自動解除機能の動作に差はありません。

注※3

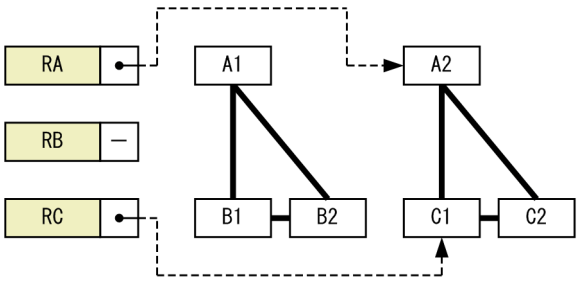
「操作対象となる SDB データベースに対応する上記以外の位置指示子」とは，次の例でのレコード位置指示子 RB のことを指します。

1. 4V AFM の SDB データベースで仮想ルートレコード型 A とその配下に子レコード型 B と C が定義された SDB データベースで，仮想ルートレコード実現値 A1，レコード実現値 B1 を検索（FETCH）します。レコード位置指示子 RA は，仮想ルートレコード実現値 A1，レコード位置指示子 RB は子レコード実現値 B1 に位置づけられます。





2. レコード実現値 C1 を検索（FETCH）します。レコード位置指示子 RA は，仮想ルートレコードのレコード実現値 A2 に位置づけられ，レコード位置指示子 RB は空値になります。



この場合，各レコード位置指示子の説明は，次のとおりです。

RA：仮想ルートレコードに対応する位置指示子

RC：操作対象となるレコードビューに対応する位置指示子

RB：（操作対象となる SDB データベースに対応する）上記以外の位置指示子

注※4

構成情報取得は SDB データベースの操作前だけ実行できるため，仮想ルートレコードに対応する位置指示子以外のレコード位置指示子が位置づいていることはありません（空値から空値への更新は発生します）。

注※5

「2.9.5(9)　排他自動解除機能の使用時に排他が解除されないケース」で説明する，解除対象外になるデータページについては，解除されません。

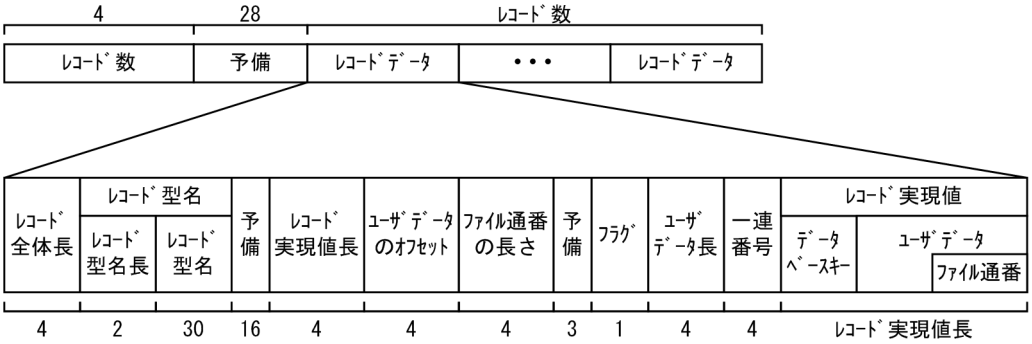
注※6

HiRDB/SD 定義ユティリティ（pdsdbdef）が SDB データベース定義の情報を基に作成した SDB データベースビュー定義で定義されたレコード型を指します。

付録 M　複数レコードの検索時の，データ格納エリアに対するレコードの格納形式【4V FMB】

複数レコードの検索時の，データ格納エリアに対するレコードの格納形式を次の図に示します。

図 M-1　複数レコードの検索時の，データ格納エリアに対するレコードの格納形式



複数レコードの検索時にデータ格納エリアに設定される値を次の表に示します。

表 M-1　複数レコードの検索時にデータ格納エリアに設定される値

項番	説明	長さ	内容
1	レコード数	4	データ格納エリアに格納したレコード数
2	予備	28	予備（0x00 が設定されます）
3	レコード全体長	4	項番 3～項番 14 の全体の長さ
4	レコード型名の長さ	2	レコード名称の長さ
5	レコード型名	30	レコード名称を左詰めで格納（余りは空白となります）
6	予備	16	予備（0x00 が設定されます）
7	レコード実現値長	4	項番 14 のレコード実現値の長さ
8	ユーザデータのオフセット	4	ユーザデータの開始位置（項番 14 のレコード実現値の先頭からのバイト数）
9	ファイル通番の長さ	4	データ種別が TYPE U，F のユーザデータがある場合に長さを設定します。TYPE U，F のユーザデータがない場合は 0 となります。
10	予備	3	予備（0x00 が設定されます）
11	フラグ	1	'U'：親レコードの USER ポインタで示されるレコードの場合に設定されます。 '△'：上記以外の場合に設定されます（ルートレコードの場合も含まれます）。
12	ユーザデータの長さ	4	項番 14 のレコード実現値のうち，ユーザデータ部分の長さ

項番	説明	長さ	内容
13	一連番号	4	一連番号の値 • ルートレコードの場合は 0 となります • データ種別が TYPE N，N の構成要素がある子レコードの場合は 1 となります。 • データ種別が TYPE K，N の構成要素がある子レコードの場合は 1 以上の値となります。
14	レコード実現値	レコード実現値の長さ	レコード実現値（データベースキー，ユーザデータ）

付録 N　障害調査のために必要な情報

問題解決支援または Q&A のサポートサービスを御利用になる場合，次の表に示す情報が必要になることがあります。性能の問題，無応答，異常終了の 3 つの場合に分けて要否を示します。そのほかに必要となる情報については，マニュアル「HiRDB システム運用ガイド」の「障害調査のために必要な情報」を参照してください。

表 N-1　障害調査のために必要な情報

区分	取得する情報	取得方法	性能	無応答	異常終了
HiRDB/SD	SDB 定義文	HiRDB/SD 定義ユティリティ（pdsdbdef）実行時に使用した SDB 定義文ファイルを取得してください。	○	－	○

（凡例）

○：必要

－：情報を取得する必要はありません。

付録 O　予約語の一覧

システムに登録されている予約語を示します。

付録 O.1　DML の予約語

DML 中に記述するキーワードは，予約語としてシステムに登録されています。システムに登録されている予約語を次の表に示します。

注意事項

次の表に示す予約語を名前として DML 中に記述する場合は，その予約語を引用符（"）で囲んでください。予約語を引用符（"）で囲むと，一般の文字列と同じように DML 中で名前として使用できます。

表 O-1　DML の予約語の一覧

頭文字	予約語
A	ACSMODE，ACSRDAREA，ADVANCING，ALL，ALLOCATE，ALSO，ALTER，AND，AREA，ASSIGN，AUTODEQ
B	BUFFER_SIZE，BY
C	CALL，CHANGE，CHILD，CLEAR，COMMENT，COMMIT，CONNECT，CONTAINS，CONTENT，CREATE，CURRENCY，CURRENT
D	DATA，DBKEY，DEALLOCATE，DEFAULT，DELETE，DIRECTION，DISCONNECT，DROP，DYNAMIC
E	EMPTY，ENQOPT，EQUAL，ERASE，EXCLUSIVE
F	FETCH，FETCHDB_ALL，FIND，FINISH，FIRST，FOR，FOUND，FROM
G	GET，GRANT，GREATER
I	INDEXED，INSERT，INTO，IS
K	KEY
L	LAST，LESS，LOCK
M	MEMBER，MODIFY
N	NEXT，NO，NONE，NOSET，NOT，NOUSE，NULL，NULLIFY
O	OCCUPY，OCCURRENCE，OF，ONLY，OR，OWNER
P	PAGENUM，PAGESWITCH，PARENT，PCTFREEVAL，PRIOR，PURGE
R	RDAREAVAL，READ，RECNAME，RECORD，RELATIVE，RETAINING，REVOKE，ROLLBACK，ROOT

頭文字	予約語
S	SDBDEFINF，SDBNAME，SECIDXSUP，SEGMENT，SELECT，SET，SHARE，SKIP，SQLCA，SQLIOA，START，STORE，SUMMARY，SYSTEM
T	T4VIF_DATA，T4VIF_KEY，T4VIF_KEYCOND，T4VIF_TABLE，TARGET，TERMOPT，THAN，TO
U	UNTIL，UPDATE，USERKEY，USERPTROPT，USING
W	WHERE，WITH，WITHIN
Y	YES
その他	#DESCRIBE，#DISPLAY，#EXIT，#HELP，#USAGE

付録 O.2　pdsdbcbl コマンドの予約語

pdsdbcbl コマンドが使用する次の予約語は，UAP ソースファイル中で名前として使用できません。使用した場合，プリプロセス，またはコンパイル時にエラーになるおそれがあります。

- COBOL2002 の予約語
- pdsdbcbl コマンド独自で使用する予約語

予約語を引用符（"）で囲むと，次の名前については，一般の文字列と同じように名前として使用できます。

- SDB データベース節で指定する SDB データベース名
- 埋込み変数名を除いた DML 中の名前

pdsdbcbl コマンド独自で使用する予約語の一覧を次の表に示します。

表 O-2　pdsdbcbl コマンド独自で使用する予約語の一覧

頭文字	予約語
D	DML
E	END-DML
S	SDB，SDB-DATABASE

付録 P　4V FMB または 4V AFM の SDB データベースと SD FMB の SDB データベースの機能差一覧

4V FMB または 4V AFM の SDB データベースと，SD FMB の SDB データベースの機能差一覧を次の表に示します。

表 P-1　4V FMB または 4V AFM の SDB データベースと，SD FMB の SDB データベースの機能差一覧

項番	分類	項目	説明	4V	SD
1	環境定義	文字コード	UTF-8 の使用	○	×
2		構造型 DB	階層構造を持たない AFM のデータベース（TMS-4V/SP がサポートしている DAM，MAM，TAM，または SAM のデータベースに相当）	○	×
3		SDB 用 UAP 環境定義	SDB 用 UAP 環境定義の適用	×	○
			SD 排他モード，アクセス目的指定	×	○
			排他自動解除指定	×	○
4		ルートレコードのデータベースキーの集団項目名	シーケンシャルインデクスの構成要素から成る集団項目の名称：任意 4V FMB または 4V AFM の場合は「DBKEY」固定です。	×	○
5	運用コマンド・ユティリティ	HiRDB/SD データベースアクセスユティリティ（pdsdbexe）	CLEAR コマンド	○	×
			FETCHDB_ALL コマンド	○	×
6		HiRDB/SD データベース作成ユティリティ（pdsdblod）	オンライン再編成中の実行	○	×
			フォーマットライト（4V DAM の機能）	○	×
			dbinf 文（4V FMB の機能）	○	×
7		HiRDB/SD データベース再編成ユティリティ（pdsdbrog）	オンライン再編成中の実行	○	×
			unload 文の unldkind=lod_4v 指定	○	×
			dbinf 文の指定	○	×
8		オンライン再編成関連コマンド	pdorbegin，pdorchg，pdorcreate，pdorgls，pdrdrefls，pdsdborcrt，pdorcheck，pdorend コマンドの実行	○	×
9		HiRDB/SD 定義ユティリティ（pdsdbdef）	DBTYPE 4V 指定	○	×
			DBTYPE SD 指定	×	○

項番	分類	項目	説明	4V	SD
			データ型の SMALLINT 指定	×	○
			TYPE U,K 指定	○	×
			TYPE N,N 指定	○	×
			OCCURS 句の指定	×	○
			FUNCTION 句の指定	○	×
			OCCURRENCE NUMBER 0 指定	○	×
			REUSE NO 指定	○	×
			SDBOPTION FUNCTION 句の指定	○	×
			POINTER AREA SIZE 句の指定	○	×
			OWNER POINTER FOR FIRST LAST USER 指定	○	×
			MEMBER POINTER FOR OWNER NEXT PRIOR 指定	×	○
			KEYDEF 句の指定	○	×
			*ALTER DICTIONARY 文の実行	○	×
10		DML プリプロセサ（pdsdbcbl）	埋込み型 UAP	×	○
11		インナレプリカ機能	pddbchg コマンドの実行	○	×
12		リアルタイム SAN レプリケーションの構成確認（pdrisechk）	全非同期方式	○	×
13	UAP の開発および実行	DML	ERASE	×	○
			FETCH [FOR UPDATE 句]	×	○
			FIND [FOR UPDATE 句]	×	○
			MODIFY	×	○
			STORE	×	○
14		SQL トレース機能	FIND	×	○
15		PRF トレース機能	FIND	×	○
			FOR UPDATE の指定の有無	×	○
16		監査証跡機能	FIND	×	○
			CLEAR	○	×
			FETCHDB ALL	○	×
17		UAP 統計レポート機能	FIND	×	○

項番	分類	項目		説明	4V	SD
18	運用	SDB データベースの操作機能		一括削除	○	×
				複数レコードの検索	○	×
				構成情報取得（容量情報）	○	×
		SDB データベースの操作機能（検索範囲）		次の条件が重なるレコード • キーの条件 • キーの条件の開始位置の RD エリア	○	×
				次の条件が重なるレコード • キーの条件 • キーの条件の開始位置の RD エリアまたはユーザが指定した RD エリア	○	×
				次の条件が重なるレコード • キーの条件 • キーの条件の開始位置のレコード実現値またはユーザが指定した RD エリア	○	×
19		個別開始	個別開始要求なしで DML の要求ができるオプション	個別開始/終了一括要求オプション	○	×
				個別開始実行要求オプション	○	×
				個別開始要求をしないオプション	×	○
			個別開始実行後の要求でのハンドル番号指定	個別開始要求をしないオプション指定ありの場合：ハンドル番号指定不要	×	○
20		UAP 実行時の排他制御	レコード格納用 RD エリア	アクセスモードの指定に従って決定されます。 • 'R'（参照モード）の場合：SR • 'U'（更新モード）の場合：SU	○	×
				SDB 用 UAP 環境定義の SD 排他モードとアクセス目的の組み合わせに従って決定されます。 • shared と retrieve の場合：SR • shared と update の場合：SU • protected と retrieve の場合：PR※1 • protected と update の場合：PU • exclusive と retrieve の場合：EX※1 • exclusive と update の場合：EX※1	×	○

項番	分類	項目		説明	4V	SD
				• nonprotected と retrieve の場合：SR※1		
			ルートレコード型	アクセスモードの指定に従って決定されます。 • 'R'（参照モード）の場合：SR • 'U'（更新モード）の場合：SU ただし，ログレスモードのときは EX（このとき，下位資源の排他は確保しない）となります。	○	×
				SDB 用 UAP 環境定義の指定に従って決定されます。 • retrieve（アクセス目的）の場合：SR※2 • update（アクセス目的）の場合：SU ログレスモードのときは EX（このとき，下位資源の排他は確保しない）となります。	×	○
			ページまたはサブページ	排他オプションの指定に従って決定されます。 排他制御モードについては，「表 2-35　排他制御のモードの組み合わせの例（4V FMB の SDB データベースの場合）」と「表 2-36　排他制御のモードの組み合わせの例（4V AFM の SDB データベースの場合）」を参照してください。	○	×
				DML と SDB 用 UAP 環境定義の指定に従って決定されます。 排他制御モードについては，「表 2-37　排他制御のモードの組み合わせの例（SD FMB の SDB データベースの場合）」を参照してください。	×	○
21		ページの割り当て		事前ページ割り当て	○	×
				ページ切り替えオプション	○	×
				PCTFREE 有効化オプション	○	×
22		インナレプリカ機能を使用したオンライン再編成			○	×
23		リアルタイム SAN レプリケーション（ディザスタリカバリシステム）		全非同期方式	○	×

（凡例）

4V：4V FMB または 4V AFM の SDB データベース

SD：SD FMB の SDB データベース

○：使用できる機能

×：使用できない機能

注※1

ページまたはサブページの排他は確保されません。

注※2

SD 排他モードが nonprotected の場合，ルートレコード型の代わりに，ルートレコード型（NOWAIT 検索時）の排他を確保します。

付録 Q　用語解説

ここに記載されている以外の用語については，マニュアル「HiRDB 解説」の「用語解説」を参照してください。

(英字)

AFM のデータベース

階層構造を持たないデータベースです。

1 つのレコード型は，1 つまたは複数の構成要素から構成されます。UAP とのデータの受け渡しは，レコード単位または構成要素単位となります。

AFM のデータベースは，メインフレームの TMS-4V/SP がサポートする DAM，MAM，TAM，および SAM のデータベースに相当します。

FMB のデータベース

階層構造を持つデータベースです。

1 つのレコード型は，複数の構成要素から構成されます。UAP とのデータの受け渡しは，レコード単位となります。

FMB のデータベースは，次のデータベースに相当します。

- メインフレームの TMS-4V/SP がサポートしている FMB のデータベース
- メインフレームの XDM/SD がサポートしている階層型モデルのデータベース

SDB 定義文

pdsdbdef コマンドで操作する SDB データベースの定義情報が指定されている制御文のことです。

SDB ディレクトリ情報ファイル

SQL_SDB_DIR 表に格納されている，SDB データベースへのアクセスに必要な情報を出力したファイルです。

SDB データベース

レコード，インデクス，親子集合など構造型 DB を構成するデータ群の総称です。

SDB データベース格納定義

構造型 DB の格納構造の定義です。インデクスの構成要素や親子関係を実現するためのポインタなどを定義します。

SDB データベース種別

SDB データベースの種類のことです。SDB データベース種別を次の表に示します。

表 Q-1　SDB データベース種別

SDB データベース種別	説明
4V FMB	階層構造を持つデータベースで，メインフレームの TMS-4V/SP がサポートしている FMB のデータベースに相当します。
4V AFM	階層構造を持たないデータベースで，メインフレームの TMS-4V/SP がサポートしている DAM，MAM，TAM，または SAM のデータベースに相当します。 4V AFM の SDB データベースは，さらに次の SDB データベース種別に分けられます（4V AFM は次の SDB データベース種別の総称です）。 • 4V DAM • 4V MAM • 4V TAM • 4V SAM
SD FMB	階層構造を持つデータベースで，メインフレームの XDM/SD がサポートしているデータベースに相当します。

SDB データベース定義

構造型 DB の論理構造の定義です。データ操作の対象となるレコードの構成やレコード間の親子関係などを定義します。

SDB データベースの横分割

SDB データベースを複数の RD エリアに分割して格納することを SDB データベースの横分割といいます。SDB データベースの横分割は，SDB データベース格納定義で定義します。

SDB データベースビュー

SDB データベース定義から，UAP が使用する部分（レコード型など）を抜き出したものです。SDB データベースビュー定義は HiRDB/SD が SDB データベース定義を基に内部的に作成します。

SDB データベースを操作するためのインタフェース

- **4V FMB または 4V AFM の SDB データベースの場合**

  4V FMB または 4V AFM の SDB データベースを操作するためのインタフェースは，TP1/FSP が提供するライブラリ関数として提供されます。また，HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の DML コマンドを使用することで SDB データベースを操作できます。

  構造型 DB を操作するためのインタフェースには，次のものがあります。

  ・個別開始

  ・個別終了

・レコードの検索（FETCH）
・レコードの格納（STORE）
・レコードの更新（MODIFY）
・レコードの削除（ERASE）
・レコードの一括削除
・複数レコードの検索（FETCHDB ALL）
・構成情報取得

詳細については，次のマニュアルを参照してください。

・「TP1/Financial Service Platform 使用の手引」
・「TP1/Financial Service Platform プログラム作成の手引」
・「TP1/Server Base Enterprise Option　プログラム作成の手引」

HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の詳細については，「14. HiRDB/SD データベースアクセスユティリティ（pdsdbexe）」を参照してください。

- SD FMB の SDB データベースの場合

SD FMB の SDB データベースを操作するためのインタフェースは，埋込み型 UAP の DML として提供されます。また，HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の DML コマンドを使用することで SDB データベースを操作できます。

構造型 DB を操作するための DML には，次のものがあります。

・レコードの検索（FETCH）
・位置指示子の位置づけ（FIND）
・レコードの格納（STORE）
・レコードの更新（MODIFY）
・レコードの削除（ERASE）

DML の詳細については，「17. DML リファレンス【SD FMB】」を参照してください。
HiRDB/SD データベースアクセスユティリティ（pdsdbexe）の詳細については，「14. HiRDB/SD データベースアクセスユティリティ（pdsdbexe）」を参照してください。

SDB 用 UAP 環境定義ファイル

SDB 用 UAP 環境定義を記述したファイルです。SDB 用 UAP 環境定義には，SD FMB の SDB データベースにアクセスする UAP の実行環境を定義します。

（ア行）

アクセスパス情報

SDB データベースを操作するための API（FETCH，STORE，MODIFY，ERASE），または DML（FETCH，FIND，STORE，MODIFY，ERASE）を実行した結果，UAP 統計レポートに出力される情報をアクセスパス情報といいます。アクセスパス情報には，SDB データベー

ス内で実行した API または DML の種類，操作対象のレコード型名やバックエンドサーバ名などが出力されます。

位置指示子

構造型 DB では，操作中のレコードや親子集合の位置を識別するための情報を保持します。この情報のことを位置指示子といいます。

一連番号

基本項目のデータベースキーのうち，HiRDB/SD がレコードの発生順に，自動的に採番して管理する番号を一連番号といいます。

インデクス

レコード実現値中の特定の構成要素の値に従ってデータを処理する場合，該当する構成要素の値と，その値を持つレコード実現値の格納位置を管理します。この管理情報をインデクスといいます。

HiRDB/SD には，次の 2 種類のインデクスがあります。

- シーケンシャルインデクス
- 二次インデクス

追い付き状態管理表

更新可能なオンライン再編成の追い付き反映処理時に，HiRDB/SD が使用する表です。

追い付き反映キー対応表

オリジナル RD エリアとレプリカ RD エリアのデータを対応づける情報が格納されている表です。更新可能なオンライン再編成のデータロード時に，追い付き反映キー対応表にデータが登録されます。登録された情報は追い付き反映処理時に使用されます。

親子集合

構造型 DB では，各レコードは互いに関連を持ちます。レコード間の論理的な関連を親子の関係で表現されます。これを親子集合といいます。

親子集合位置指示子

最後に操作した親レコード，および子レコードの位置を示す位置指示子を親子集合位置指示子といいます。親子集合位置指示子は，親子集合型ごとに生成されます。

親子集合型

親レコード型と子レコード型をモデル化したものを親子集合型といいます。親子集合型は SDB データベース定義で定義します。

親子集合型内二次インデクス

HiRDB/SD の二次インデクスの種類の 1 つです。詳細については，「二次インデクス」を参照してください。

親レコード

親子集合で，親に当たるレコードを親レコードといいます。親レコードのレコード型を親レコード型といいます。

オリジナル RD エリア

レプリカ RD エリアを作成するときに，基となる RD エリアをオリジナル RD エリアといいます。

（カ行）

回復追い付き反映処理

更新可能なオンライン再編成の追い付き反映処理時に，レプリカ RD エリア側のロールバックで回復した内容をオリジナル RD エリア側に反映する処理を回復追い付き反映処理といいます。

仮想親子集合

親子集合を構成しない（階層構造を持たない）データベースを構築する場合，HiRDB/SD では仮想のルートレコード（仮想ルートレコード）を親レコードとして設定し，仮想的な親子集合を構成します。これを仮想親子集合といいます。

仮想ルートレコード

仮想親子集合を構成する場合に，親レコードとして設定する仮想のルートレコードのことです。仮想ルートレコードのレコード型を仮想ルートレコード型といいます。

カレント RD エリア

HiRDB クライアント環境定義，UAP 環境定義，または運用コマンドで指定がないときには，決まった RD エリア（デフォルトの RD エリア）にアクセスします。この RD エリアをカレント RD エリアといいます。

キー項目

親子集合型内の子レコードの挿入順序を決定する構成要素のことです。子レコードは，親子集合の論理構造上，キー項目の順番に格納されて，検索されます。SDB データベース定義の親子集合型の KEY 句で指定します。

基準サブページ

基準レコードが格納されているサブページを基準サブページといいます。

基準ページ

基準レコードが格納されているページを基準ページといいます。

基準レコード

HiRDB/SD はレコードを格納する際に，基準とするレコードから，格納するレコードの位置を決定します。この基準とするレコードを基準レコードといいます。

空値

位置指示子がどのレコードも指していないか，または位置指示子に親子集合中のレコードを検索するために必要なデータベースキーが設定されていない状態のことを空値といいます。

更新追い付き反映処理

更新可能なオンライン再編成の追い付き反映処理時に，レプリカ RD エリア側での更新内容をオリジナル RD エリア側に反映する処理を更新追い付き反映処理といいます。

構成要素

データベースの中で取り扱える情報の最小単位であるデータ項目を，論理的な単位にまとめたものを構成要素といいます。1 つのデータ項目が 1 つの構成要素となるだけでなく，関連する 2 つ以上のデータ項目をまとめて 1 つの構成要素とする場合もあります。

子レコード

親子集合で，子に当たるレコードを子レコードといいます。子レコードのレコード型を子レコード型といいます。

（サ行）

サーバプロセス

AP の処理要求を実行するプロセスです。1 つの AP に対して，1 つのサーバプロセスが起動します。

サブページ

データページ（レコードを格納するページ）を複数の領域に分割した際の，1 つの領域をサブページといいます。1 データページは，2～16 のサブページに分割できます。

サブページ間の関連づけ

HiRDB/SD では，同一のルートレコード下のレコードを格納するサブページを関連づけて管理します。これをサブページ間の関連づけといいます。

シーケンシャルインデクス

シーケンシャルインデクスは，ルートレコード，または仮想ルートレコードの検索に使用するインデクスです。次の構成要素に定義します。

- 4V FMB または SD FMB の SDB データベースの場合
  ルートレコードの先頭の集団項目に指定した基本項目の構成要素
- 4V AFM の SDB データベースの場合
  仮想ルートレコードに指定した基本項目の構成要素

シーケンシャルインデクスは，SDB データベース格納定義の SEQUENTIAL 句で定義します。定義された構成要素の値をキーとして，キーの昇順に作成されます。

事前常駐領域

SDB データベースの定義追加または定義変更後に作成した SDB ディレクトリ情報を常駐する共用メモリ上の領域を事前常駐領域といいます。SDB ディレクトリ情報を事前常駐領域に常駐したあとに，事前常駐領域と常用常駐領域を切り替えます。HiRDB の再起動を必要としない SDB データベースの定義追加または定義変更をする際に，事前常駐領域が必要になります。

事前割り当てサブページ

ルートレコードの格納時に指定数分だけ確保する，ファミリ専用のサブページを事前割り当てサブページといいます。事前割り当てサブページをあらかじめ確保しておくことで，SDB データベースを操作する API（STORE，ERASE）を繰り返し実行した場合でも，連続したデータページ内にレコードを配置できます。

事前割り当てページ

ルートレコードの格納時に指定数分だけ確保する，ファミリ専用のデータページを事前割り当てページといいます。事前割り当てページをあらかじめ確保しておくことで，SDB データベースを操作する API（STORE，ERASE）を繰り返し実行した場合でも，連続したデータページ内にレコードを配置できます。

常用常駐領域

SDB ディレクトリ情報を常駐する共用メモリ上の領域を常用常駐領域といいます。HiRDB/SD は，常用常駐領域に常駐する SDB ディレクトリ情報を使用します。

（タ行）

データ種別

構成要素の用途を区別して，さらに詳細な種別を指定したものをデータ種別といいます。

（ナ行）

二次インデクス

HiRDB/SD には，次の 2 種類の二次インデクスがあります。

- 親子集合型内二次インデクス

  親子集合型内二次インデクスは，親子集合型内の子レコードの検索や，子レコードを格納する際のレコード挿入位置の検索に使用するインデクスです。

  子レコードの構成要素に定義するインデクスで，SDB データベース格納定義の SECONDARY INDEX 句で定義します。

  親子集合型内二次インデクスを使用すると，親子集合型内の特定の子レコードに直接アクセスできるので，NEXT ポインタまたは PRIOR ポインタに従って順アクセスするよりも，検索性能を向上させることができます。

  4V FMB または 4V AFM の SDB データベースに定義した二次インデクスを，親子集合型内二次インデクスといいます。

- レコード型内二次インデクス

  レコード型内二次インデクスは，レコード型内の検索に使用するインデクスです。次のどちらかの構成要素に定義します。

  ・ルートレコードの，シーケンシャルインデクスとは異なる構成要素

  ・子レコードの構成要素

  SDB データベース格納定義の SECONDARY INDEX 句で，USED FOR RECORD を指定します。

（ハ行）

ファミリ

ルートレコードおよびルートレコード下のレコード群を総称してファミリといいます。

ファミリ内接続サブページ

HiRDB/SD では，同一のルートレコード下のレコードを格納するサブページを関連づけて管理します。関連づけられたサブページをファミリ内接続サブページといいます。

ファミリ内接続ページ

HiRDB/SD では，同一のルートレコード下のレコードを格納するページを関連づけて管理します。関連づけられたページをファミリ内接続ページといいます。

フォーマットライト

4V DAM の SDB データベースで，pdsdbdef コマンドで定義したデータベースに初期値を格納し，インデクスを作成します。初期値は HiRDB/SD が自動的に作成します。この処理をフォーマットライトといいます。

ページ間の関連づけ

HiRDB/SD では，同一のルートレコード下のレコードを格納するページを関連づけて管理します。これをページ間の関連づけといいます。

ポインタ

親子集合で表現したレコード間の論理的な親子関係を，データベース中で関係づけることができます。これをポインタといいます。

- FIRST ポインタ

  親レコードが子レコードを持つ場合，先頭の子レコードの位置情報を親レコードが管理するために持っているポインタを FIRST ポインタといいます。

- LAST ポインタ

  親レコードが子レコードを持つ場合，最終の子レコードの位置情報を親レコードが管理するために持っているポインタを LAST ポインタといいます。

- USER ポインタ

  親レコードが子レコードを持つ場合，任意の子レコードの位置情報を子レコードごとに 1 つだけ持つことができます。これを USER ポインタといいます。

- NEXT ポインタ

  同一レコード型の子レコードが，1 つの親レコードに関連づけられたレコードの範囲内で，次のレコード（次の一連番号またはユーザキーを持つレコード）の位置情報を管理するために持っているポインタを NEXT ポインタといいます。

- PRIOR ポインタ

  同一レコード型の子レコードが，1 つの親レコードに関連づけられたレコードの範囲内で，直前のレコード（1 つ前の一連番号またはユーザキーを持つレコード）の位置情報を管理するために持っているポインタを PRIOR ポインタといいます。

- OWNER ポインタ

  子レコードに親レコードの位置情報を保持することができます。これを OWNER ポインタといいます。OWNER ポインタは子レコードごとに 1 つだけ持つことができます。

（ヤ行）

優先サブページ

SDB データベースを操作する API（STORE）によって，事前ページ割り当て機能を適用しているファミリの子レコードを格納する際，基準サブページが属するページに格納できないときに HiRDB/SD が優先的に格納を試みる事前割り当てサブページを優先サブページといいます。

優先ページ

SDB データベースを操作する API（STORE）によって，事前ページ割り当て機能を適用しているファミリの子レコードを格納する際，基準ページに格納できないときに HiRDB/SD が優先的に格納を試みる事前割り当てページを優先ページといいます。

（ラ行）

ルートレコード

階層構造の最上位のレコードをルートレコードといいます。ルートレコードのレコード型をルートレコード型といいます。

レコード

構造型 DB では，データベース中に存在する情報の基本単位のことをレコードといいます。レコードは情報の論理的な単位であると同時に，データを検索したり，更新したりするときの単位でもあります。

レコード位置指示子

最後に操作したレコードの位置を示す位置指示子をレコード位置指示子といいます。レコード位置指示子は，レコード型ごとに生成されます。

レコード型

構造型 DB では，あるレコードをモデル化したものをレコード型といいます。レコード型は SDB データベース定義で定義します。

レコード型内二次インデクス

HiRDB/SD の二次インデクスの種類の 1 つです。詳細については，「二次インデクス」を参照してください。

レコード実現値

レコード型に対応してデータベース中で，実際のデータ値を持って存在するものをレコード実現値といいます。なお，このマニュアルでは，特に誤解を与えない場合にはレコード実現値を単にレコードといいます。

レコードビュー

SDB データベースビュー定義で定義されたレコード型をレコードビューといいます。

レプリカ RD エリア

オリジナル RD エリアを基に複製した RD エリアをレプリカ RD エリアといいます。

索引

記号

*ALTER DICTIONARY 文 892
*ALTER DIRECTORY 文 896
*CHECK DICTIONARY 文 895
*DELETE DICTIONARY 文 894
*DELETE DIRECTORY 文 897
*ENTRY DICTIONARY 文 891
*ENTRY DIRECTORY 文 895
#EXIT〔pdsdbexe サブコマンド〕 1291
#HELP〔pdsdbexe サブコマンド〕 1291
#USAGE〔pdsdbexe サブコマンド〕 1294

数字

16 進データ 63

A

ACSMODE〔START コマンド〕 1257
ACSRDAREA〔START コマンド〕 1258
ADD〔SDB データベース定義〕 916, 966
afmtype〔pdsdbrog 制御文〕 1160
AFM のデータベース 60
AFM のデータベース〔用語解説〕 1491
ALLERASE〔SDB データベース定義〕 917, 967
area
  pdsdblod 制御文 1079
  pdsdbrog 制御文 1156
ASCENDING〔SDB データベース定義〕 919, 970, 1011
AUTODEQ〔START コマンド〕 1258

B

BUFFER_SIZE〔FETCHDB_ALL コマンド〕 1247

C

CHARACTER 63
CHARACTER〔SDB データベース定義〕 911, 960, 1003
checkmode 885
CLEAR
  DML コマンド 1232
CLUSTERED〔SDB データベース格納定義〕 933, 982, 1016
COMMIT〔SQL コマンド〕 1286
CONNECT〔SQL コマンド〕 1286
CONNECT 権限の付与 551
CREATE SCHEMA 61
CURRENT OWNER OF 親子集合型名 1388

D

data model
  pdinit 776
  pdmod 781
data〔pdsdblod 制御文〕 1079
DATA〔SDB データベース格納定義〕 949, 995
DBA 権限の付与 550
dbinf 文
  pdsdblod 制御文 1082
  pdsdbrog 制御文 1160
DBLODUTL〔SDB データベース定義〕 924, 975
DBSCHM
  *ALTER DIRECTORY 文 897
  *DELETE DIRECTORY 文 897
  *ENTRY DIRECTORY 文 896
DBTYPE
  SDB データベース格納定義 932, 981, 1015
  SDB データベース定義 909, 957, 1001
DECIMAL 型 415
DEPENDING ON〔SDB データベース格納定義〕 937, 1021, 1029
DESCENDING〔SDB データベース定義〕 919, 970, 1011
DIRECTION〔STORE コマンド〕 1265
dirinf 文 883
DISCONNECT〔SQL コマンド〕 1289

divermsg〔pdsdblod 制御文〕 1076
dml_error 863
dml_skip_info 865
DML コマンド 1232
  共通の規則 1270, 1285
  ひな形の出力 1294
DML 終了子 1398
DML スキップ情報出力ファイル 865
DML スキップ情報出力ファイルの出力形式 865
DML 先頭子 1398
DML で使用できる文字 1365
DML の記述形式 1365
DML の最大長 1366
DML のデータ型 1366
DML の予約語 1484
DML プリプロセサ 1306
DML リファレンス 1363
DML 履歴用バッファ 1349
DML 履歴用バッファのサイズ 1320
dupkeyck〔pdsdblod 制御文〕 1082

E

END SCHEMA〔SDB データベース定義〕 925, 977, 1013
END STORAGE SCHEMA〔SDB データベース格納定義〕 951, 998, 1032
ENQOPT〔START コマンド〕 1257
environment 文
  pdsdblod 制御文 1069
  pdsdbrog 制御文 1153
  SDB 制御文 880
ERASE
  API 94
  DML 95
  DML コマンド 1236, 1271
  SDB データベース定義 917, 967
ERASE 文 1381
errdata〔pdsdblod 制御文〕 1080
exectime 882
  pdsdblod 制御文 1072
  pdsdbrog 制御文 1153
EX モード 208

F

FETCH
  API 93
  DML 95
  DML コマンド 1239, 1273
FETCHDB_ALL
  DML コマンド 1246
FETCHDB_ALL ONLY〔STRAT コマンド〕 1261
FETCHDB ALL 94
  排他資源数の見積もり 393
FETCH オプション
  CREATE AUDIT 664
  DROP AUDIT 670
FETCH 文 1382
FIND 95
FIND 文 1386
FINISH〔DML コマンド〕 1251, 1275
FIRST〔SDB データベース定義〕 1009
FIRST ポインタ 69
FMB のデータベース 60
FMB のデータベース〔用語解説〕 1491
FORMAT〔SDB データベース定義〕 975
FOR RECORD〔SDB データベース格納定義〕 933, 983, 1017
FOR UPDATE
  FETCH 文 1383
  FIND 文 1387
FUNCTION〔SDB データベース定義〕 916, 965

G

generate 884
generation
  pdsdblod 制御文 1076
  pdsdbrog 制御文 1155
GET 文 1390

H

HA モニタの環境設定
　高速系切り替え機能　628
HiRDB/SD からの切り離し　1289
HiRDB/SD 定義ユティリティ　868
HiRDB/SD データベースアクセスユティリティ 1202
HiRDB/SD データベース再編成ユティリティ　1146
HiRDB/SD データベース作成ユティリティ　1056
HiRDB/SD の位置づけ　49
HiRDB/SD の環境設定
　高速系切り替え機能　629
HiRDB/SD の最大値・最小値　1424
HiRDB/SD の特長　41
HiRDB/SD への接続　1286
HiRDB Advanced High Availability　48
HiRDB Server と HiRDB Structured Data Access Facility の入れ替え手順　424
HiRDB Staticizer Option　48
HiRDB Structured Data Access Facility Extension for XDM/SD type　48
HiRDB システム定義　269
HiRDB システム定義の作成
　高速系切り替え機能　629
HiRDB システム定義の変更　496
HiRDB のアンインストール　409
HiRDB のインストール　406
HiRDB の開始方法　447
　高速系切り替え機能　637
HiRDB の終了方法　447
HiRDB のバージョンアップ　410
Hitachi Unified Storage VM　723

I

idxarea　1085
idxload 文〔pdsdblod 制御文〕　1085
idxmode〔pdsdblod 制御文〕　1081
idxname　1085
idxremode　1085
INDEXED BY
　FETCH 文　1384
　FIND 文　1387
index 文〔pdsdblod 制御文〕　1084
infmsglvl〔pdsdblod 制御文〕　1073
INSERTION〔SDB データベース定義〕　918, 970, 1010
INTEGER
　SDB データベース定義　912, 960, 1003
　データ型　63
IP アドレスの構成例〔高速系切り替え機能〕　626
ITEM〔SDB データベース定義〕　910, 958, 1001

J

JP1/NETM/Audit 用監査ログ出力ファイル　838
JP1/NETM/Audit 用監査ログ出力ユティリティ 838

K

KEYDEF〔SDB データベース格納定義〕　946, 993
KEY〔SDB データベース定義〕　919, 970, 1011
KFPB31010-E メッセージ　737

L

LAST〔SDB データベース定義〕　918, 969, 1009
LAST ポインタ　69
load 文〔pdsdblod 制御文〕　1077
lockrange オペランド　742
logmode〔pdsdblod 制御文〕　1070

M

MEMBER POINTER FOR〔SDB データベース格納定義〕　946, 992, 1032
MEMBER〔SDB データベース定義〕　918, 970, 1010
midfile　1087
MODIFY
　API　94
　DML　95
　DML コマンド　1252, 1276
MODIFY 文　1391
msglog　880

N

NEXT ポインタ　69

O

OCCURRENCE NUMBER〔SDB データベース定義〕　920, 971, 1012
OCCURRENCE WARNING〔SDB データベース定義〕　924, 977, 1013
OCCURS〔SDB データベース定義〕　1007
ORDER KEY〔SDB データベース格納定義〕
　シーケンシャルインデクス　933, 983, 1017
　二次インデクス　943, 990, 1026
ORDER〔SDB データベース定義〕　918, 969, 1009
oreload 文〔pdsdblod 制御文〕　1087
OS の時刻を変更する方法　609
OWNER POINTER FOR〔SDB データベース格納定義〕　946, 992, 1032
OWNER〔SDB データベース定義〕　918, 969, 1009
OWNER ポインタ　69

P

PACKED DECIMAL FIXED
　SDB データベース定義　912, 960, 1003
　データ型　63
pagecflg
　データロード　1083
　フォーマットライト　1084
pagenum
　pdsdbrog 制御文　1161
　データロード　1084
PAGESWITCH〔STORE コマンド〕　1266
PCTFREE〔SDB データベース格納定義〕
　シーケンシャルインデクス　934, 984, 1018
　二次インデクス　944, 991, 1029
　レコード　937, 986, 1020
PCTFREEVAL〔STORE コマンド〕　1266
PCTFREE 有効化オプション
　サブページ分割をしない場合　289
　サブページ分割をする場合　307
pd_assurance_index_no　757
pd_cmd_exec_time　759
pd_cmd_process_conwaittime　760
pd_dbbuff_dev_sector_size　757
pd_dbreuse_remaining_entries　757
pd_dec_sign_normalize　754
pd_fes_lck_pool_size　757
pd_ha_agent　758
pd_indexlock_mode　755
pd_key_resource_type　755
pd_lck_pool_size　388, 756
pd_lck_until_disconnect_cnt　756
pd_lock_uncommited_delete_data　755
pd_max_access_tables　753
pd_max_commit_write_reclaim_no　754
pd_max_open_fds　759
pd_max_resident_rdarea_no　757
pd_max_users　753
pd_nowait_scan_option　755
pd_pageaccess_mode　754
pd_rise_disaster_mode　758
pd_rise_fence_level　758
pd_rise_pairvolume_combination　758
pd_rorg_predict　757
pd_rpl_hdepath　758
pd_shared_rdarea_use　757
pd_space_level　754
pd_spd_syncpoint_skip_limit　755
pd_sql_object_cache_size　757
pd_structured_advance_resident　740
pd_structured_directory_path　741
pd_structured_shmpool_dicsize　739
pd_table_def_cache_size　757
pd_tmp_directory　759
pd_uap_wait　759
pd_utl_buff_size　758
pd_utl_exec_mode　753
pd_utl_exec_time　754
pd_utl_file_buff_size　758

pd_work_buff_mode 758
pd_work_buff_size 758
pdacunlck 770
pdadmvr 765
pdaudatld 770
pdaudbegin 770
pdaudend 770
pdaudput 770, 838
pdaudrm 770
pdaudswap 770
PDAUTORECONNECT 1305
pdbkupls 773, 848
pdbuffer 328, 757
pdbufls 769
pdbufmod 769
pdcancel 769
pdcat 765
pdcbl 771
pdchgconf 765
pdchprc 769
pdclibsync 765
pdclose 768
pdclttrc 770
pdcmt 769
pdconfchk 766
pdconfchk コマンドでチェックできるオペランド 763
pdconstck 772
pdcopy 773
pdcpp 771
pdcspool 766
pddbchg 769
pddbfrz 768
PDDBLOG 266
pddbls 768, 849
pddbst 773, 796
pddef 771
pddefrev 772
pdexp 772
pdfbkup 767
pdffsck 767
pdfgt 769
pdfls 767
pdfmkfs 767
pdfrm 767
pdfrstr 767
pdfstatfs 767
pdgetcst 773
pdhold 768
pdinfoget 766
pdinit 771, 775
pditvstop 766
pditvtrc 766
pdjarsync 766
pdlbuffer 759
pdlistls 766
pdload 771
pdlodsv 766
pdlogadpf 767
pdlogatul 767
pdlogchg 767
pdlogcls 767
pdloginit 767
pdlogls 767
pdlogopen 767
pdlogrm 767
pdlogswap 768
pdlogsync 768
pdlogucat 768
pdlogunld 768
pdls 766
pdmemdb 770
pdmemsv 766
pdmod 772, 780
pdobils 771
pdobjconv 766
pdocb 771
pdocc 771
pdopen 768

pdopsetup　766
pdorbegin　769, 858
pdorcheck　769, 859
pdorchg　770
pdorcreate　770
pdorend　770, 861
pdparaload　771
pdpfresh　769
pdpgbfon　772, 788
pdplgrgst　773
pdplgset　773
pdprgcopy　769
pdprgrenew　769
pdrbal　772
pdrbk　769
pdrdconstck　769
pdrdrefls　769, 850
pdreclaim　772
pdreginit　773
pdrels　768
pdrisechk　770
pdrisedbto　770
pdriseset　770
pdrorg　772, 785
pdrpause　769
pdrplstart　769
pdrplstop　769
pdrstr　773
pdsdbarc　765, 840
pdsdbcbl　1306
pdsdbcbl コマンドの予約語　1485
pdsdbdef　771, 868
pdsdbexe　771
pdsdbexe コマンド　1202
　規則　1299
　コマンド実行前の作業　1299
　コマンドの形式　1205
　トラブルシューティング　1304
　リターンコード　1303
　留意事項　1301
pdsdbexe コマンドの終了　1291
pdsdbexe サブコマンド　1291
pdsdbexe 操作コマンド　1203
　指定規則　1299
pdsdblod　771
pdsdblod コマンド　1056
　RD エリアの容量不足　1118
　異常終了した場合　1116
　タイムアウト　1119
　無応答状態になった場合　1117
pdsdblod コマンドの異常終了〔障害対処方法〕　578
pdsdblod 制御文　1069
pdsdborcrt　770, 855
PDSDBPRMTRC　1321
pdsdbrog　772
pdsdbrog コマンド
　異常終了した場合　1178
　タイムアウト　1179
　無応答状態になった場合　1178
pdsdbrog 制御文　1153
PDSDBTRCCNDSIZE　1321
PDSDBTRCDATASIZE　1323
PDSDBTRCELMSIZE　1322
PDSDBTRCKEYSIZE　1322
PDSDBUAPDIR　1324
PDSDBUAPFILE　1323
pdsetup　767
pdsql　772
pdstart　768
pdstbegin　768
pdstedit　772, 789
pdstend　768
pdstjswap　768
pdstjsync　768
pdstop　768
pdstscls　768
pdstsinit　768
pdstsopen　768

pdstsrm 768
pdstsswap 768
pdsvhostname 767
pdtrndec 769
pdtrnqing 770
PDUAPEXERLOGDMLDATA 1321
PDUAPEXERLOGDMLSZ 1320
pdvrup 767
pdvwopt 773
pdwork 754
POINTER AREA SIZE〔SDB データベース格納定義〕 942, 989
prefix
pdsdblod 制御文 1081
pdsdbrog 制御文 1157
PRF トレース機能 614
PRF トレース情報の詳細と取得ポイント 614
PRIOR ポインタ 69
PR モード 208
purge〔pdsdblod 制御文〕 1071
PU モード 208

R

RAID Manager 723
RDAREAVAL〔START コマンド〕 1258
RD エリア指定有効オプション〔START コマンド〕 1258
RD エリア単位の状態解析
物理的解析 837
論理的解析 797
RD エリアの空き容量の確認方法 541
RD エリアの回復 555
RD エリアの再初期化 782
RD エリアの削除 542
pdmod 784
RD エリアの状態表示 849
RD エリアの設定変更 542
RD エリアの属性変更 784
RD エリアの追加 542
RD エリアの容量拡張 542
RD エリアの容量不足〔pdsdblod〕 1118
recfree〔pdsdblod 制御文〕 1075
recnomsg
pdsdblod 制御文 1073
pdsdbrog 制御文 1154
record
pdsdblod 制御文 1078
pdsdbrog 制御文 1156
RECORD 909, 958, 1001
RECORDID〔SDB データベース定義〕 917, 968, 1008
REFER〔SDB データベース定義〕 916, 966
reflection 文〔pdorend コマンド〕 863
report 文〔pdorend コマンド〕 864
Result of DML Optimizer 1352
Result of SDB Cursor 1357
RETENTION〔SDB データベース定義〕 919, 970, 1011
REUSE〔SDB データベース定義〕 922, 974
ROLLBACK〔SQL コマンド〕 1290

S

schema
pdsdblod 制御文 1070
pdsdbrog 制御文 1153
SCHEMA〔SDB データベース定義〕909, 957, 1000
SDBOPTION
SDB データベース格納定義 946, 993
SDB データベース定義 923, 974, 1012
SDB 制御文 877
SDB 定義文 888
SDB 定義文の記述規則 888
SDB 定義文〔用語解説〕 1491
SDB ディクショナリ情報 90
SDB ディクショナリ情報の削除 872
*DELETE DICTIONARY 文 894
SDB ディクショナリ情報の追加 870
*ENTRY DICTIONARY 文 891

SDB ディクショナリ情報の変更　871
*ALTER DICTIONARY 文　892
SDB ディレクトリ情報　90
SDB ディレクトリ情報が一致しているか確認する方法　490
SDB ディレクトリ情報長の計算式　333
SDB ディレクトリ情報の削除　874
*DELETE DIRECTORY 文　897
SDB ディレクトリ情報の常駐化　90
SDB ディレクトリ情報の常駐領域サイズの見直し 418
SDB ディレクトリ情報の追加　873
*ENTRY DIRECTORY 文　895
SDB ディレクトリ情報のファイル出力　874
SDB ディレクトリ情報の不一致が発生したときの対処　492
SDB ディレクトリ情報の変更　874
*ALTER DIRECTORY 文　896
SDB ディレクトリ情報の変更の流れ　452
SDB ディレクトリ情報ファイルの配布方法　486
SDB ディレクトリ情報ファイル〔用語解説〕　1491
SDB データベース　61
SDB データベース格納定義
4V AFM　977
4V FMB　925
SD FMB　1013
SDB データベース格納定義〔用語解説〕　1491
SDB データベース種別　61
SDB データベース種別〔用語解説〕　1492
SDB データベース操作イベント　650
SDB データベース操作イベント種別
CREATE AUDIT　664
DROP AUDIT　670
SDB データベース定義
4V AFM　952
4V FMB　902
SD FMB　999
SDB データベース定義〔用語解説〕　1492
SDB データベースの再編成
インナレプリカ機能を使用しない場合　497
オンライン再編成　502
更新可能なオンライン再編成　507
SDB データベースの再編成の手順　498
SDB データベースの操作　93
SDB データベースの操作と位置指示子の関係　127
位置指示子の位置づけ（位置指示子指定の検索）　130
位置指示子の位置づけ（レコード型内の検索）　130
個別開始　127
個別終了およびトランザクションの終了　135
子レコードの格納（子レコードの追加）　132
子レコードの検索　128, 129
子レコードの更新　133
子レコードの削除　134
複数レコードの検索　135
ルートレコードの格納（ルートレコードの追加）　131
ルートレコードの検索　128
ルートレコードの削除　134
SDB データベースの定義　90
SDB データベースの定義削除〔HiRDB/SD の再起動を必要とする場合〕　466
SDB データベースの定義追加
HiRDB/SD の再起動を必要としない場合　474
HiRDB/SD の再起動を必要とする場合　461
SDB データベースの定義追加，定義変更，または定義削除
HiRDB/SD の再起動を必要とする場合　452
ディザスタリカバリシステム　728
SDB データベースの定義追加または定義変更
HiRDB/SD の再起動を必要としない場合　467
ディザスタリカバリシステム　732
SDB データベースの定義変更
HiRDB/SD の再起動を必要としない場合　476
HiRDB/SD の再起動を必要とする場合　463
SDB データベースの定義例
4V DAM　1044
4V FMB　1039
SD FMB　1050
SDB データベースの横分割　70

SDB データベースの横分割数を増やす方法　544
SDB データベースの横分割〔用語解説〕　1492
SDB データベースビュー〔用語解説〕　1492
SDB データベースへのデータロードの手順　571
SDB データベース〔用語解説〕　1491
SDB データベースを操作する API　93
SDB データベースを操作するためのインタフェース〔用語解説〕　1492
SDB ハンドラ　1035
SDB ユティリティ操作イベント　650
SDB ユティリティ操作イベント種別
　CREATE AUDIT　665
　DROP AUDIT　670
SDB 用 UAP 環境定義　269, 742
　指定例　749
　文法規則　750
SDB 用 UAP 環境定義の指定による無排他検索機能　243
SDB 用 UAP 環境定義の変更方法　496
SDB 用 UAP 環境定義ファイル
　格納ディレクトリの作成　440
SDB 用 UAP 環境定義ファイルの作成　438
SDB 用 UAP 環境定義ファイル〔用語解説〕　1493
SD 排他モード　744
SECIDXSUP
　FETCH コマンド　1243
　STORE コマンド　1267
SECONDARY INDEX〔SDB データベース格納定義〕　942, 989, 1025
seqkeyck〔pdsdblod 制御文〕　1081
SEQUENTIAL〔SDB データベース格納定義〕　933, 982, 1016
server
　pdsdblod 制御文　1078
　pdsdbrog 制御文　1156
SET
　SDB データベース格納定義　946, 992, 1032
　SDB データベース定義　917, 969, 1008
SETOPTION〔SDB データベース定義〕　920, 971, 1011
SMALLINT
　SDB データベース定義　1003
　データ型　63
sortdir〔pdsdblod 制御文〕　1086
SORTED DUPLICATES FIRST〔SDB データベース定義〕　1009
SORTED DUPLICATES LAST〔SDB データベース定義〕　1009
SORTED DUPLICATES PROHIBITED〔SDB データベース定義〕　918, 969, 1009
source 文　879
SQL_AUDITS 表　1421
SQL_COLUMNS 表　1414
SQL_DIV_INDEX 表　1420
SQL_DIV_TABLE 表　1418
SQL_INDEX_COLINF 表　1419
SQL_INDEXES 表　1416
SQL_RDAREAS 表　1411
SQL_SDB_DB 表　1409
SQL_SDB_DIR 表　1410
SQL_SDB_STORAGE_DB 表　1409
SQL_TABLE_PRIVILEGES 表　1417
SQL_TABLES 表　1412
SQL エラー情報の出力例　1340
SQL トレース機能　1326
SQL トレース情報の出力例　1326
SR モード　208
START〔DML コマンド〕　1256, 1279
STORAGE RECORD〔SDB データベース格納定義〕　932, 982, 1016
STORAGE SCHEMA〔SDB データベース格納定義〕　932, 981, 1015
STORE
　API　94
　DML　95
　DML コマンド　1264, 1281
STORE 文　1393
storinf〔pdsdblod 制御文〕　1080
SUBPAGE NUMBER　936, 985, 1020
subschema オペランド　743

SU モード　208

T

TAMMODE〔SDB データベース定義〕　976
TAM のデータベースの無排他検索機能　240
TARGET〔ERASE コマンド〕　1237
TERMOPT〔FETCHDB_ALL コマンド〕　1248
TERMOPT〔FETCH コマンド〕　1243
TMS-4V/SP 情報　1151
TP1/FSP　47
TRANSACTION 単位の情報〔UAP 統計レポート〕　1359
transfer　884
TrueCopy　719
type〔pdsdblod 制御文〕　1078
TYPE〔SDB データベース定義〕　912, 960, 1003

U

UAP 単位の情報〔UAP 統計レポート〕　1359
UAP 統計レポート機能　1350
UAP の障害対策　1326
UAP の設計　1310
unit〔dirinf 文〕　886
Universal Replicator　719
unldfile〔pdsdbrog 制御文〕　1156
unldinf〔pdsdbrog 制御文〕　1157
unldkind〔pdsdbrog 制御文〕　1159
unload 文〔pdsdbrog 制御文〕　1155
UNTIL NOT FOUND〔FETCHDB_ALL コマンド〕　1248
UPDATE〔SDB データベース定義〕　916, 966
USED FOR〔SDB データベース格納定義〕　1026
userpflg
　pdsdblod 制御文　1083
　pdsdbrog 制御文　1161
USERPTROPT
　MODIFY コマンド　1253
　STORE コマンド　1266
USER ポインタ　69
USER ポインタフラグ　1101
USER ポインタフラグの開始位置　1083

W

WARNING〔SDB データベース定義〕　920, 971, 1012
WITHIN〔SDB データベース格納定義〕
　シーケンシャルインデクス　935, 984, 1018
　二次インデクス　945, 991, 1029
　レコード　938, 986, 1021
WITHIN 親子集合型名
　FETCH 文　1384
　FIND 文　1388
workdir〔pdsdblod 制御文〕　1086
write〔pdsdblod 制御文〕　1078

X

XCHARACTER
　SDB データベース定義　912, 960, 1003
　データ型　63

あ

空きセグメント　280
アクセス順序　41
アクセスパス情報〔用語解説〕　1493
アクセスモード〔START コマンド〕　1257
アンインストール　409
アンロード　1147
　SDB データベースが横分割されている場合　1148
　実行結果ファイル　1148
　実行結果ファイルの出力形式　1175
アンロードおよびデータロードの例
　4V FMB　1181
　4V MAM　1190
　SD FMB　1194
　オンライン再編成の場合　1185
アンロードデータファイル　1147
アンロードデータファイルの出力形式　1168
アンロードデータファイルを UAP の入力情報として使用する　1151

アンロードの例〔アンロードデータファイルを UAP の入力情報として使用する場合〕
4V FMB 1197
4V MAM 1199

い

位置指示子 95
位置指示子の位置づけ 114
位置指示子の種類 96
位置指示子〔用語解説〕 1494
一連番号 66
一連番号使用比率の指定
OCCURRENCE WARNING 924, 977, 1013
WARNING 920, 972, 1012
一連番号の監視 610
一連番号〔用語解説〕 1494
意図共用モード 208
意図排他モード 208
イベントサブタイプ 688
イベントタイプ 688
入れ替え手順 424
インストール
HiRDB 406
付加 PP 407
インデクス 76
インデクス一括作成モード 1060
インデクス検索時の排他制御 260
インデクス更新モード 1060
インデクス情報ファイル 1059
インデクス情報ファイルの名称 1116
インデクス選択規則 1375
インデクスの再作成 1065
インデクスの再作成手順 574
インデクスの再作成の例
4V FMB 1140
SD FMB 1143
インデクスの作成モード 1060
インデクスの残存エントリ 262
インデクスの未完状態 782
インデクスページ 56
インデクス〔用語解説〕 1494
インデクスを格納するページ数の見積もり 372
インナレプリカ機能 420
環境設定 420
更新可能なオンライン再編成 507

う

埋込み言語文法 1398
埋込み変数 95, 1368
運用項目の一覧 445

お

追い付き状態管理表〔用語解説〕 1494
追い付き状態管理表を格納する RD エリア 525
追い付き反映キー対応表 510
pdsdborcrt コマンド 855
格納 RD エリアを確認する 538
用語解説 1494
追い付き反映キー対応表のインデクス 510
追い付き反映キー対応表の操作 855
追い付き反映キー対応表を格納する RD エリアの容量見積もり 373
追い付き反映処理 510
API のエラーをスキップしたい場合 540
格納配置制御 518
状態を確認する 538
追い付き反映制御ファイル 863
オペレーティングシステムパラメタの見積もり 386
親子集合 66
親子集合位置指示子 97
親子集合位置指示子〔用語解説〕 1494
親子集合型 67
親子集合型内二次インデクス 80
親子集合型内二次インデクス〔用語解説〕 1495
親子集合型内の検索 1379
親子集合型名 1384, 1388
親子集合型〔用語解説〕 1494
親子集合〔用語解説〕 1494

親レコード　67
　ポインタの種類　69
　用語解説　1495
親レコード型　67
オリジナル RD エリア〔用語解説〕　1495
オンライン再編成の手順　503

か

カーネルパラメタの見積もり　386
開始方法　447
　高速系切り替え機能　637
階層構造　39
下位の位置指示子　98
回復追い付き反映処理〔用語解説〕　1495
拡張 SQL エラー情報出力機能　1340
格納位置を検索する方向　1265
格納条件指定による横分割　71
格納配置制御〔追い付き反映処理〕　518
仮想親子集合　68
仮想親子集合〔用語解説〕　1495
仮想ルートレコード　68
仮想ルートレコード〔用語解説〕　1495
カレント RD エリア〔用語解説〕　1495
環境設定方法〔セキュリティ監査機能〕　661
環境変数の設定　1313
監査証跡　643
監査証跡の参照方法　671
監査証跡の絞り込み　691
監査証跡の出力契機の例
　一括削除をした場合　656
　複数レコードを検索した場合　657
　レコードを格納した場合　653
　レコードを検索した場合　652
　レコードを更新した場合　654
　レコードを削除した場合　654
　レコードを取得した場合　658
監査証跡の取得契機　644
監査証跡のレコード項目
　pdsdbdef コマンドの場合　694
　pdsdblod コマンドの場合　696
　pdsdbrog コマンドの場合　698
　複数レコードの検索の場合　710
　レコードの一括削除の場合　708
　レコードの格納の場合　703
　レコードの検索の場合　700
　レコードの更新の場合　704
　レコードの削除の場合　706
　レコードの取得の場合　712
監査証跡表の自動データロード機能　650
監査証跡表の列構成　674
監査証跡表へのデータ登録　673
監査証跡表を格納する RD エリア　661
監査証跡ファイル
　出力される情報　651
監査証跡ファイルの運用　672
　高速系切り替え機能　631
監査証跡ファイルの容量見積もり　356
監査対象イベント　649
監査対象イベントの定義　661
監査対象イベントの定義の削除　668
監査対象になるイベント　649
監査人　643
関連 RD エリアの表示　850
関連するプログラムプロダクト　47

き

キー項目更新時の親子集合位置指示子の状態　101
キー項目〔用語解説〕　1495
キーワード　1300
キーワードの指定　1365
基準サブページ　173
基準サブページ〔用語解説〕　1495
基準ページ　140
基準ページ〔用語解説〕　1496
基準レコード
　サブページ分割をしない場合　140
　サブページ分割をする場合　173
　用語解説　1496

基本項目　64
機密保護機能　272
境界値指定による横分割　72
共用意図排他モード　208
共用メモリの見積もり　332
共用モード　208
近傍配置　42

く

空値　98
空値〔用語解説〕　1496
区切り文字の挿入　1365
クライアントエラーログ機能　1339
クライアント環境定義　1313
グローバルバッファ　206
グローバルバッファ常駐化ユティリティ　788
グローバルバッファの先読み入力　331
グローバルバッファの設計　328
グローバルバッファの定義
　高速系切り替え機能　631
グローバルバッファの割り当て方法　328

け

計画系切り替えの手順
　高速系切り替え機能　638
検索コード　107
検索の基点となるレコード　106
検索の順序，開始位置および方向　1376
検索の方向　107

こ

更新追い付き反映処理〔用語解説〕　1496
更新可能なオンライン再編成　420, 507
　運用手順　528
　追い付き反映　861
　追い付き反映キー対応表の操作　855
　準備作業　518
　障害対処　587
　操作を取り消す　539
　対象 RD エリアの状態を確認する　538
　注意事項　512
　データベース静止化　858
　適用条件チェック　859
更新コピー　719
更新バッファ　206
更新ログ取得方式　265
構成情報取得　94
構成要素　64
構成要素の詳細　1370
構成要素〔用語解説〕　1496
構造型 DB　39
構造型 DB 機能に関するユーザ権限の設定　549
高速系切り替え機能　271
　HA モニタの環境設定　628
　IP アドレスの構成例　626
　環境設定　624
　システム構成例　625
個別開始
　API　93
　DML　95
　START コマンド　1256, 1279
個別開始/終了一括要求オプション　110
個別開始実行要求オプション　136
個別終了
　API　93
　DML　95
　FINISH コマンド　1251, 1275
コマンドの同時接続数　1423
コミットしていない削除データの排他制御　258
コメント　1365
子レコード　67
　ポインタの種類　69
　用語解説　1496
子レコード型　67
子レコードの検索
　親子集合型内の検索　128
　レコード型内の検索　129

さ

サーバプロセス〔用語解説〕 1496
サーバモード 271
最小値 1424
最初に作成するファイル 404
最大値 1424
最大同時実行数〔ユティリティ〕 1422
再編成
  インナレプリカ機能を使用しない場合 497
  オンライン再編成 502
  更新可能なオンライン再編成 507
再編成状態〔更新可能なオンライン再編成〕 587
再編成の手順 498
作業表用ファイル 268
作業表用ファイル〔HiRDB ファイルシステム領域の設計〕 274
サブページ 56
サブページ間の関連づけ 174
サブページ間の関連づけ〔用語解説〕 1496
サブページ切り替え 182
サブページ長の設計 366
サブページ長の設計方針 287
サブページ内の未使用領域の比率 286
  適用基準 314
サブページの解放 286
サブページの確保 286
サブページの状態 285
サブページの設計 284
サブページの適用基準 284
サブページ分割 56
サブページ分割数 936, 985, 1020
サブページ分割数の設計方針 287
サブページ〔用語解説〕 1496
参照バッファ 206
残存エントリ 262

し

シーケンシャルインデクス 78
シーケンシャルインデクス〔用語解説〕 1496
時刻を変更する方法 609
指示コード
  FETCH 文 1383
  FIND 文 1387
システム共通定義
  オペランドの説明 739
システム構成を変更する方法 622
システム構築手順 403
システムの状態監視 448
システムファイル 268
システムファイルの作成 412
システムマネジャ 52
システム用 RD エリア 53
システムログ量の見積もり 346
事前常駐領域 92
事前常駐領域〔用語解説〕 1497
事前ページ割り当て機能 156
  ファミリの格納例 166
事前ページ割り当て機能〔サブページ分割している場合〕 190
  ファミリの格納例 200
事前割り当てサブページ数の算出 366
事前割り当てサブページ〔用語解説〕 1497
事前割り当てページ数の開始位置 1084
事前割り当てページ数の算出 366
事前割り当てページ〔用語解説〕 1497
実行結果ファイル
  pdsdbdef コマンド 881
  pdsdblod コマンド 1059
  pdsdbrog コマンド 1148
実行結果ファイルの出力形式
  pdsdbdef コマンド 1038
  pdsdblod コマンド 1107
  pdsdbrog コマンド 1175
自動再接続機能 1305
自動データロード機能〔セキュリティ監査機能〕 650
集団項目 64
終了方法 447
障害調査のために必要な情報 1483

障害発生時の対処方法　577
pdsdblod コマンドの異常終了時　578
更新可能なオンライン再編成　587
高速系切り替え機能　641
定義追加または定義変更前の状態に戻す場合　480
ディザスタリカバリシステム　737
障害閉塞　266
使用中空きセグメント　280
使用中サブページ　285
使用中セグメント　280
使用中ページ　282
使用中満杯サブページ　285
使用中満杯ページ　282
常用常駐領域　91
常用常駐領域〔用語解説〕　1497
初期値〔フォーマットライト〕　1063
初期データロード　1057
初期データロードの例
4V FMB　1120
4V MAM　1126
SD FMB　1129
二次インデクスを定義している場合　1123

す

スキーマ　60
スキーマ定義権限の付与　550
スキーマの削除　553
スキーマの追加　553

せ

制御文ファイル
pdsdblod コマンド　1059
pdsdbrog コマンド　1147
制限事項
更新可能なオンライン再編成　512
整数データ　63
セキュリティ　549
セキュリティ監査機能　643
運用方法　671
環境設定方法　661
セグメント　56
セグメントサイズの設計　370
セグメント内の空きページの比率　280, 314
セグメントの確保と解放　281
セグメントの状態　280
セグメントの設計　279
セマフォ所要量の見積もり　387
選択オプション　663, 670
全非同期方式　721

そ

操作系 DML　1381
操作種別　663
ソート用ワークファイル　1059
属性　63

た

タイムアウト
pdsdbdef コマンド　882
pdsdblod コマンド　1119
pdsdbrog コマンド　1179
探索条件の指定　1370
単体版 HiRDB　424
単体版 HiRDB から統合版 HiRDB への入れ替え手順　424
単調増加ファイル　1428

ち

中間ファイル　1060
容量見積もり式　375
中間ファイルの格納ディレクトリ　1087
注釈　1365

つ

追加データロード　1057
追加データロードの例
4V FMB　1131
SD FMB　1135

て

ディクショナリサーバ　52
ディクショナリ操作機能　870
ディクショナリ表　54
ディクショナリ表のインデクス　54
ディクショナリ表の参照権限の設定　551
ディザスタリカバリシステム　719
　SDB データベースの定義追加，定義変更，または定義削除　728
　SDB データベースの定義追加または定義変更　732
　運用　727
　構築　725
ディスク容量の見積もり　346
ディレクトリ操作機能　873
ディレクトリページ　56
データ格納エリア　94
データ型　63
データ型〔SDB データベース定義〕　911, 959, 1002
データ構造　41
データ項目　63
データ種別 1〔SDB データベース定義〕　912, 960, 1003
データ種別 2〔SDB データベース定義〕　912, 960, 1004
データ種別 3〔SDB データベース定義〕　960
データ種別〔用語解説〕　1497
データディクショナリ用 RD エリア　54
データディレクトリ用 RD エリア　54
データ反映方式〔ディザスタリカバリシステム〕　721
データページ　56
データベースキー　65
データベース構成変更ユティリティ　780
データベース再編成ユティリティ　785
データベース状態解析ユティリティ　796
データベース初期設定ユティリティ　775
データベース静止化　858
データベース定義のチェック　872
　*CHECK DICTIONARY 文　895
データベースの回復　555
データ用グローバルバッファ　206
データロード　1057
　SDB データベースが横分割されている場合　1060
データロードの手順　571
デッドロック　244
デッドロックプライオリティ値　255

と

統計解析情報の取得
　高速系切り替え機能　638
統計解析ユティリティ　789
統計情報の取得　449
統合版 HiRDB　424
統合版 HiRDB から単体版 HiRDB への入れ替え手順　430
動作条件　46
同時接続数〔コマンド〕　1423
同時にバックアップを取得する必要がある RD エリア　450
トラブルシューティング
　pdsdbexe コマンド　1304
　pdsdblod コマンド　1116
　pdsdbrog コマンド　1178
トランザクションの正常終了　1286
トランザクションの取り消し　1290

な

名前　1300
名前の規則　890
名前の指定　1366

に

二次インデクス
　4V FMB，4V AFM　80
　SD FMB　88
　親子集合型内二次インデクス　80, 1498
　効果が期待できるケース　82
　子レコード格納時の使用有無　86
　子レコード検索時の使用有無　82
　使用有無の確認　87

用語解説　1497
レコード型内二次インデクス　88, 1498
入力データの論理エラー　1112
入力データファイル　1059
入力データファイルの形式　1100

は

排他オプション〔START コマンド〕　1257
排他資源　207
排他資源数　388
排他資源数の見積もり　388
排他自動解除オプション〔START コマンド〕　1258
排他自動解除機能　228
排他制御　207
排他制御モード
pdsdbdef コマンド　1436
pdsdblod コマンド　1437
pdsdbrog コマンド　1441
排他の期間　228
排他モード　208
排他モード指定による無排他検索機能　241
配布方法〔SDB ディレクトリ情報ファイル〕　486
バックアップの取得　450
バックアップファイルの情報表示　848
バックエンドサーバ　52
パラメタトレース　1339

ひ

日立ディスクアレイサブシステム　719
非同期コピーの特徴　721
一筆書き順　904

ふ

ファイルの容量見積もり
pdcopy コマンド　377
pddbst コマンド　379
pdsdbcbl コマンド　379
pdsdblod コマンド　375
pdsdbrog コマンド　376
pdstedit コマンド　378
ファミリ　68
ファミリ内接続サブページ　174
ファミリ内接続サブページ〔用語解説〕　1498
ファミリ内接続ページ　141
ファミリ内接続ページ〔用語解説〕　1498
ファミリ〔用語解説〕　1498
フォーマットライト　1061
SDB データベースが横分割されている場合　1062
用語解説　1498
フォーマットライトの例　1137
付加 PP　48
付加 PP のインストール　407
複合状態　99
複数レコードの検索　120
FETCHDB_ALL コマンド　1246
排他資源数の見積もり　393
符号正規化機能　415
符号付きパック形式　415
符号付きパック形式 10 進データ　63
符号部　415
符号部の変換規則〔データ格納時〕　1097
物理的解析　837
プラットフォーム　46
プリフィクス部　1101
プリフィクス部のデータ形式　1170
プロセス固有メモリの見積もり　338
フロントエンドサーバ　52
分割格納条件の変更機能　73

へ

ページ　56
ページ間の関連づけ　141
ページ間の関連づけ〔用語解説〕　1498
ページ切り替え　149
ページ切り替えオプション
サブページ分割をしない場合　149
サブページ分割をする場合　182
ページ切り替えオプション〔STORE コマンド〕　1266

ページ切り替えフラグ　1101
ページ切り替えフラグの扱い　1177
ページ切り替えフラグの開始位置　1083
ページ長の設計　366
ページ長の設計方針　283
ページ内の未使用領域の比率　282
ページの解放　283
ページの解放〔サブページ分割している場合〕　287
ページの確保　283
ページの確保〔サブページ分割している場合〕　287
ページの状態　282
ページの設計　281
ヘルプ情報の出力〔pdsdbexe サブコマンド〕　1291

ほ

ポインタ　68
ポインタオプション〔STORE コマンド〕　1266
ポインタ管理領域　58
ポインタ管理領域の予備領域　58
ポインタ検索　81
ポインタ〔用語解説〕　1499

ま

マスタディレクトリ用 RD エリア　53
マルチ HiRDB の設計　437
満杯セグメント　280

み

未完状態〔インデクス〕　782
未使用サブページ　285
未使用セグメント　280
未使用ページ　282

む

無排他検索機能　240, 241
無排他検索機能〔SDB 用 UAP 環境定義の指定〕　243

め

メインサイト　719
メインフレーム　415
メッセージ　1401
メッセージキューの見積もり　387
メモリ所要量の見積もり
　pdcopy コマンド　384
　pddbst コマンド　384
　pdrstr コマンド　384
　pdsdbcbl コマンド　385
　pdsdbexe コマンド　383
　pdsdblod コマンド　380
　pdsdbrog コマンド　382
メモリ容量の見積もり　332

も

文字コード　46
文字コードの指定　408
文字列データ　63
モニタモード　271

ゆ

ユーザキー　66
ユーザ権限　272
ユーザデータ　66
ユーザ用 RD エリア　54
ユーザ用 RD エリアの容量見積もり　364
　追い付き反映キー対応表を格納する RD エリア　373
優先サブページ〔用語解説〕　1499
優先ページ〔用語解説〕　1499
ユティリティが出力するファイル　1434
ユティリティの最大同時実行数　1422
ユニット制御情報定義
　オペランドの説明　741
ユニットの状態を確認する
　高速系切り替え機能　638

よ

横分割　70
横分割数を増やす手順
　HiRDB/SD の再起動を必要としない場合　546

HiRDB/SD の再起動を必要とする場合　547
再定義をする場合　544
予約語　890
予約語の一覧　1484

り

リアルタイム SAN レプリケーション　719
リターンコード
pdsdbdef コマンド　1037
pdsdbexe コマンド　1303
pdsdblod コマンド　1099
pdsdbrog コマンド　1167
リモートサイト　719
リレーショナル DB を格納する RD エリア　330

る

ルートレコード　68
ルートレコード型　68
ルートレコード〔用語解説〕　1500

れ

レコード　63
レコード位置指示子　97
レコード位置指示子〔用語解説〕　1500
レコード格納時のサブページ切り替え　182
レコード格納時のページ切り替え　148
レコード型　63
レコード型内二次インデクス　88
レコード型内二次インデクス〔用語解説〕　1500
レコード型内の検索　1376
レコード型〔用語解説〕　1500
レコード識別コード〔SDB データベース定義〕　917, 968, 1008
レコード実現値　63
レコード実現値の削除範囲　1237
レコード実現値〔用語解説〕　1500
レコード操作キー情報　686
レコード長　365
レコードのアクセス順序　104
レコードのアンロード　1147
レコードの位置づけ　96, 109
レコードの一括削除　120
CLEAR コマンド　1232
レコードの格納　116
STORE コマンド　1264, 1281
レコードの格納順序
アンロードデータファイル　1168
入力データファイル　1104
レコードの検索
4V FMB，4V AFM　104
FETCH コマンド　1239, 1273
SD FMB　112
レコードの検索範囲　107
レコードの更新　119
MODIFY コマンド　1252, 1276
レコードの削除　119
ERASE コマンド　1236, 1271
レコードの集約度の見積もり方法　832
レコードのデータ形式
アンロードデータファイル　1168
入力データファイル　1100
レコードの特定方法　109
レコードの配置制御
サブページ分割をしない場合　139
サブページ分割をする場合　172
レコードビュー〔用語解説〕　1500
レコード〔用語解説〕　1500
レコードを格納するデータページ数　370
レプリカ RD エリアの作成　420
レプリカ RD エリア〔用語解説〕　1500
レベル番号〔SDB データベース定義〕　910, 958, 1001

ろ

ログ取得モード　265
logmode　1070
ログレス閉塞　266
ログレスモード　265

logmode　1070
論理エラーが発生したときの対処　1112
論理エラー情報ファイル　1059
論理エラー情報ファイルの出力形式　1112
論理エラーの詳細　1113
論理エラー番号　1113
論理的解析　797