リレーショナルデータベース管理システム"RDB 1"

Relational Database Management System"RDB1"

リレーショナルデータベースは、その使いやすさ、柔軟性から、最近のユーザーニーズに合致したデータベースとして急速に需要が高まっている。日立製作所は今回このユーザーニーズにこたえて、リレーショナルデータベース管理システム"RDB1"を開発した。

RDB1は、リレーショナルデータベースの考えそのものがもつ長所を製品として実現し、更にエンドユーザーにも比較的簡単に使える体系化されたコマンド言語を提供している。これにより、エンドユーザーが直接データベースを容易に作成、操作できるようになり、データベース管理者の負担を軽減するとともに、多様で変化するユーザーニーズに柔軟に対応できるデータベース管理システムとなっている。

福嶋慎一* Shin'ichi Fukushima

中村史朗** Fumio Nakamura

本間敏夫*** Toshio Homma

中川恵子* Keiko Nakagawa

11 緒言

RDB1 (Relational Database Manager 1)は、1970年にE. F.Coddが論文 $^{1)}$ で発表したリレーショナルデータモデルの考えに基づくデータベースを管理する本格的なリレーショナルDBMS(データベース管理システム)である $^{2)}$ 。RDB1では、データベースは表の集まりと考えることができ、データベースに対する操作は表に対する操作として表現できる。リレーショナルデータベースのもつ理解しやすさ、使いやすさ、柔軟性から、RDB1は多様で変化するユーザーニーズに柔軟に対応できるDBMSになっている。

本論文では、まずRDB1開発の背景と製品の特長を述べたあと、製品仕様の概略について説明する。そして、エンドユーザーから見た場合の使いやすさについて例を用いて説明するとともに、データベース管理者の負担の少なさについて述べる。

2 RDB 1 開発の背景と製品の特長

今までの階層構造やネットワーク構造を扱う構造形データベースでは、ユーザーはデータの物理的な配置や関連を事前に定義し、利用する場合にもこれらを意識する必要があった。これが、エンドユーザーによるデータベースの作成やデータの自由な利用という点で大きな阻害要因となり、データベースやプログラムの作成をEDP(Electronic Data Processing)部門の人に頼らざるを得ない結果となっていた。

しかも、EDP部門では、コンピュータ適用業務の拡大に伴って開発待ちの業務が増加していた。その解決手段として生産性が高く、突発的な業務にも簡単に対処でき、しかもエンドユーザーにも直接扱えるデータベースが強く望まれるようになってきた。

データマネジメントシステムRDB1は,このようなユーザーニーズを満たすDBMS製品として開発され,VOS3(Virtual-storage Operating System 3)及びVOS1/ES(Virtual-storage Operating System 1/Extended System)システムの下で稼動する^{3),4)}。その特長を次に述べる。

(1) リレーショナルデータベースのサポート

RDB1のデータベースは、リレーショナルデータモデルに 基づくデータベースである。データ構造の基本は、テーブル と呼ばれる表形式の構造であり、データ操作は表の操作である。ユーザーはデータの物理構造を意識する必要はなく、データだけを意識して自由に操作できる。

(2) 体系化されたデータベース言語EQL

RDB1ではデータの定義から検索, 更新・制御までデータベースに関するすべての操作をエンドユーザーにも簡単に使えるコマンド言語EQL(Extended Query Language)として一つに体系化してサポートしている。EQLコマンドを端末から使用して種々のデータベース処理を簡単に行なうことができる。

(3) 容易なデータベース構築

表形式のデータ構造の採用により、データベース設計は簡単である。データベースの定義もテーブルとそのデータ項目だけでよく、物理構造の定義は不要なため、エンドユーザーにも定義できる。また、テーブルの追加、削除、拡張もデータベースを作り直さずにシステム稼動中に行なえ、段階的なデータベースの構築が可能となる。

(4) 容易なシステム運用

RDB1ではエンドユーザーがデータベースを直接作成したり、更新したりすることを許している。このため、DBMSとして必要な排他制御、障害回復機能及びユーティリティに加えてきめの細かい機密保護機能及びデータディクショナリ機能を備えて、システム運用の負担を軽減している。

図 データベースの構成

3.1 ユーザーから見たデータ構造

(1) テーブル

RDB1のデータベースは、データモデルとしてリレーショナルモデルを採用しており、ユーザーデータはすべて表形式で表現される。この表形式をテーブルと呼ぶ。テーブルは、複数の欄と行で構成される。テーブルの例を図1に示す。ユーザーはEQLコマンドを使って、このテーブルを構成する行全体に対して一括操作ができる。

(2) インデックス

テーブル操作時に特定の行を速く検索するために、インデックスを作成することができる。1個のインデックスはインデックスの付けられた欄の値をキーとするB-treeと呼ばれる

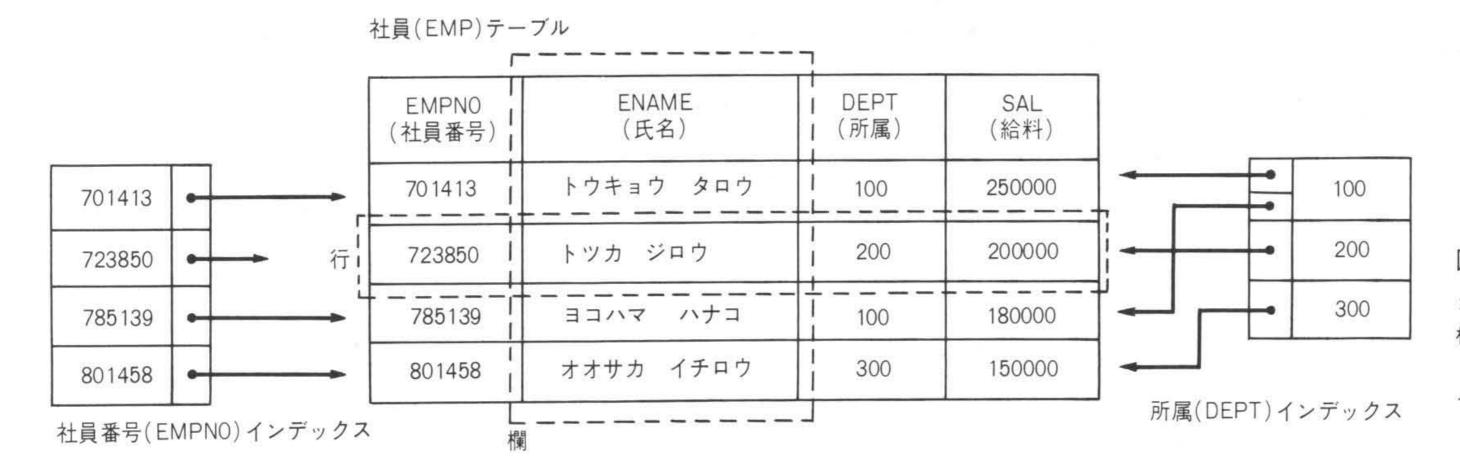


図 I テーブル, インデックス テーブルは欄と行から構成される表形式のデータ構造をもつ。テーブルの検索を速くするために, テーブルの欄にインデックス(索引)を作成できる。

木構造のデータ構造であり、任意のキー値に対して均等なアクセス速度を与える。ユーザーは必要に応じて1個のテーブルに複数個のインデックスを作成できる。インデックスはテーブル更新時にRDB1が自動的に更新し、常に最新の状態に保持される。

(3) データベース領域

RDB1データベースのテーブルは、データベース内の論理的な格納領域であるデータベース領域に格納して管理する。ユーザーのテーブルを格納するデータベース領域には用途別に次の2種がある。

(a) 公用データベース領域

複数ユーザーが共用するテーブルを作成する領域である。

(b) 私用データベース領域

そのデータベース領域の所有者だけがテーブルを作成する領域である。ユーザーは私用データベース領域にテーブルを作成することによって、そのユーザー専用のテーブルとすることができる。

(4) ディクショナリテーブル

RDB1では、ユーザーが定義したテーブル、インデックス、データベース領域、機密保護などの情報をデータベース管理情報として一括管理している。この情報はディクショナリテーブルと呼ばれるシステムのテーブルに保持している。ユーザーの定義内容に変更が生じた場合、RDB1が自動的にディクショナリテーブルの内容を更新する。ユーザーは一般のテーブルと全く同様にEQLコマンドを使ってディクショナリテーブルの情報を利用することができ、データベースの最新の状態を簡単に知ることができる。

3.2 データベースの物理構造

(1) データベースの構成

データベースは物理的にはディレクトリ,ログ及びストーレジプールの3要素で構成される。図2にデータベースの構成を示す。

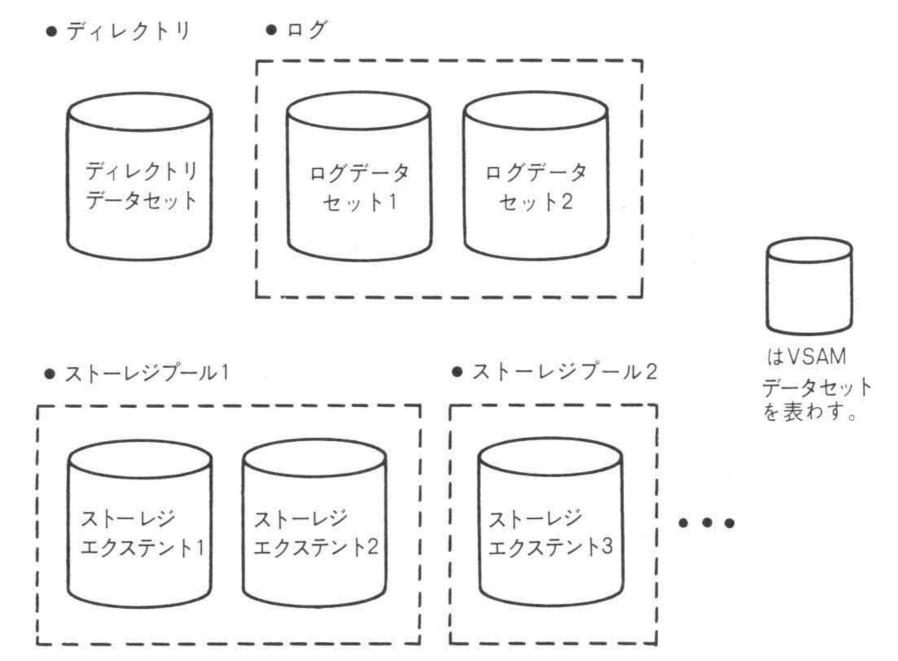
(a) ディレクトリ

ストーレジプールの領域管理,ストーレジプールの論理 構造と物理構造の管理など,ストーレジプールの管理情報 を格納している。1個のVSAM(Virtual Storage Access Method)データセットである。

(b) ログ

データベースの更新情報やユーザーのデータベース処理の開始、終了情報などを蓄積する。これらの情報は、データベースの障害回復時に使用する。ログは2個のVSAMデータセットから構成され、一方のデータセットを使い切るともう一方のデータセットを使うという形で交互に使用する。

(c) ストーレジプール



注:略語説明 VSAM(Virtual Storage Access Method)

図 2 データベースの構成 RDB I のデータベースは、ディレクトリ、ログ及びストーレジプールから構成される。

テーブル、インデックス、ディクショナリを格納する VSAMデータセット群である。1個のストーレジプールは、 1個以上のVSAMデータセットから構成される。ストーレ ジプールを構成するVSAMデータセットをストーレジエク ステントと呼ぶ。ストーレジプールは複数個定義できる。

(2) ストーレジプールの構造

ストーレジプールは、テーブル、インデックス及びディクショナリを格納する領域であり、一般ユーザー側から見た論理構造とデータベース管理者側から見た物理構造の二重構造になっている。図3にストーレジプールの構造を示す。

(a) ストーレジプールの物理構造

1個のストーレジプールは、1個以上のストーレジエクステントから構成される。ストーレジエクステントは4,096 バイトの大きさのフレームと呼ばれるブロックに分割されて管理されている。このフレームはVSAMのCI(コントロールインタバル)に対応しており、データセットに対する入出力の単位となっている。

(b) ストーレジプールの論理構造

RDB1データベースのユーザーから見ると、ストーレジプールは1個以上のデータベース領域から構成される。データベース領域は1個の連続した仮想空間(VSAMデータセットという物理的な入れ物と独立した領域)である。

1個のストーレジプールが複数個のストーレジエクステントから構成されていても、データベース領域上ではスト

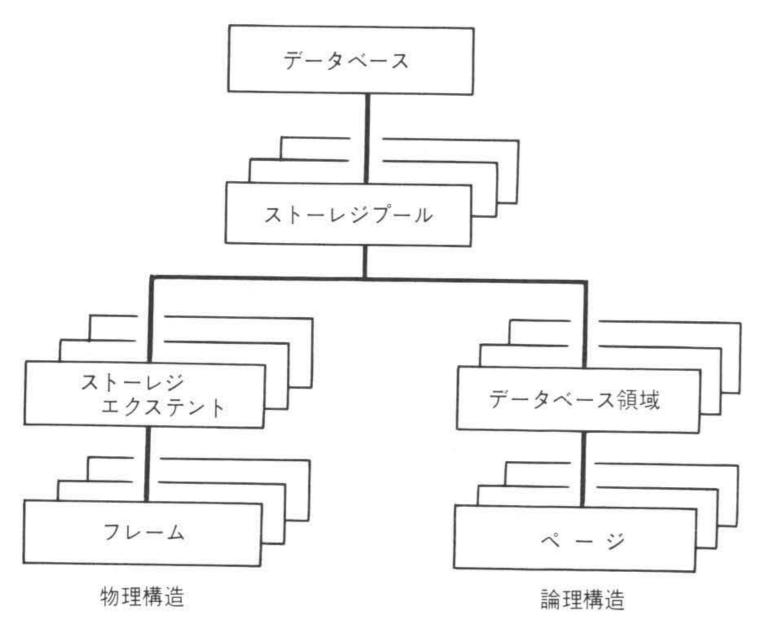


図3 ストーレジプールの構造 ストーレジプールは、一般ユーザーから見た論理構造とデータベース管理者から見た物理構造の二重構造をとっている。

ーレジエクステントの境界を意識する必要がない。

データベース領域は4,096バイトの大きさのページと呼ばれるブロックに分割されて管理されている。ページにデータが格納されるときにストーレジエクステントの一つのフレームが割り当てられる。

(c) ストーレジプールの論理構造と物理構造の対応付け

ストーレジプールの論理構造と物理構造の管理及びその対応付けはディレクトリで行なう。図4にストーレジプールの論理構造、物理構造とそれを管理するディレクトリの構造例を示す。1個のストーレジプールに対応して1個のストーレジプール管理レコードが存在し、そのストーレジプールを構成するストーレジエクステント及びデータベース領域のそれぞれの管理レコードをポイントしている。データベース領域管理レコードの下には、そのデータベース領域に含まれているページとフレームの対応付けを行なうページ・フレーム変換マップがある。ストーレジプールエクステント管理レコードの下には、そのエクステントに含まれるフレームがページに割り当てられているか否かを示

す管理レコードがある。

データベース領域のあるページにアクセスする場合,ディレクトリのページ・フレーム変換マップを見て,そのページに対応するフレームのデータセット上での位置を求めて,そのフレームにアクセスする。図5にページとフレームの対応付けの様子を示す。

テーブルに行を追加してデータベース領域の新しいページが必要になった場合、ディレクトリのフレーム割当て状況管理レコードを見て、使用されていないフレームをそのページに割り当て、その対応関係をページ・フレーム交換マップに登録する。

このようなページ・フレーム管理方式を採用することによって、テーブルの行の削除などでページに割り当てられないフレームが生じても、別のページのフレームとして割り当てることができ、ストーレジエクステントの有効利用

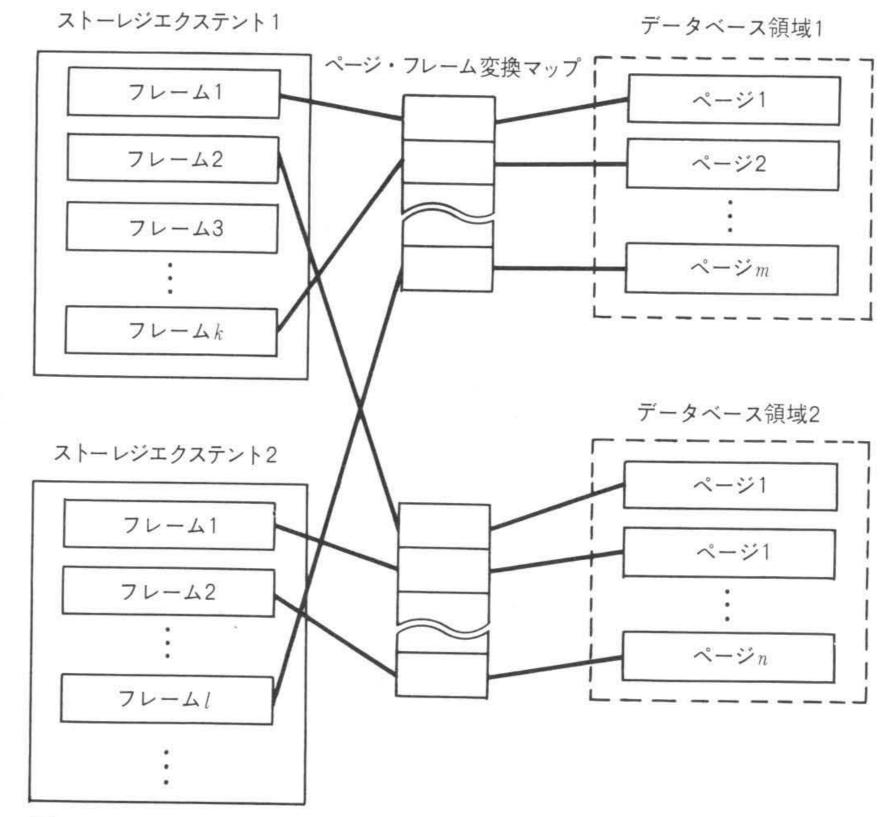
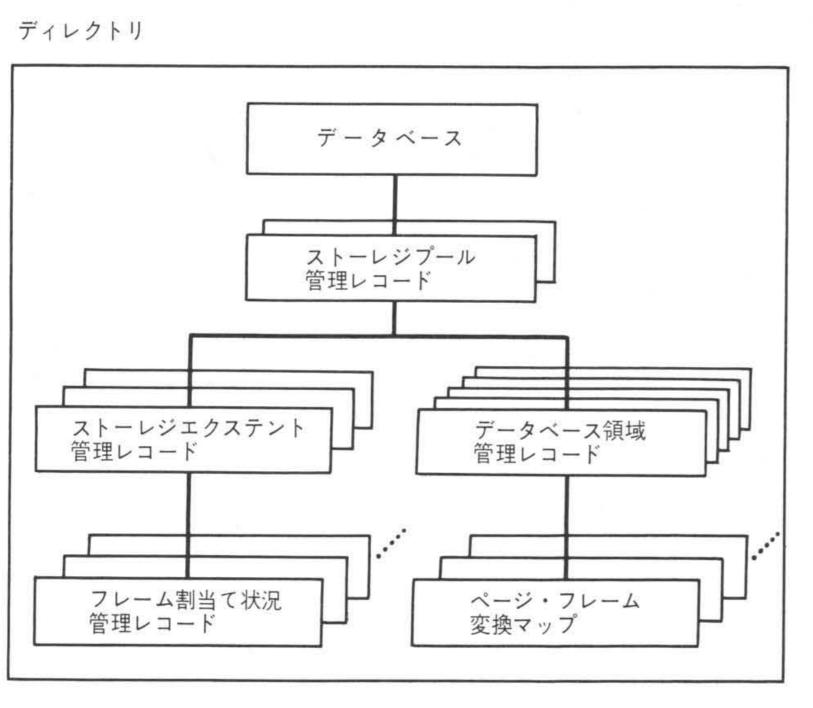


図 5 ページとフレームの対応付け ディレクトリのページ・フレーム変換マップがページとフレームの対応付けを行なっている。



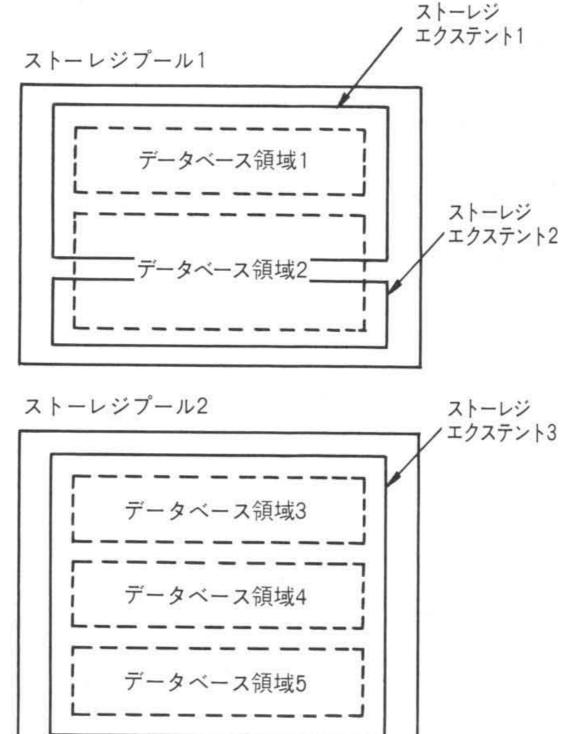


図 4 ストーレジプールの 構造とディレクトリの構造 ストーレジプールの論理構造と 物理構造は、ディレクトリ中の 管理レコードの構造で表現され ている。

を図っている。

以上述べてきたように、ストーレジプール構造は、論理構造と物理構造の二重構造を採用している。これによって、一般ユーザーがデータベースの物理的な構造を意識する必要がなくなるとともに、物理構造と独立にデータベース領域を追加できるようになっている。更に、データベース管理者から見ると、データベースの論理構造とは独立して物理領域(ストーレジエクステント)の追加ができるよになっており、データベースの領域不足に対しても簡単に対応できる構造になっている。

4 EQL言語

RDB1では、データの定義、操作、制御、権限の認可、取消しまで、データベースに関するすべての操作を、エンドユーザーにも簡単に使えるコマンド言語として一つに体系化している。このコマンド言語をEQL言語と呼ぶ。

EQL言語は次に述べるような特長をもっている。

- (1) エンドユーザーにも簡単に使えるコマンド言語である。
- (2) 高水準で非手続き形の言語である。EQLコマンドでは、 テーブル全体に対する一括操作が可能であり、ユーザーは結 果として何を行ないたいかだけを指定すればよい。
- (3) 完全なデータベース言語である。EQL言語は、データベースの処理に必要なすべての機能を包含している。
- (4) 端末から利用できるコマンド言語である。EQLコマンドは端末から対話形コマンドとして簡単に使用できるので、プログラムを作成せずに即座に必要なデータベース処理を行なえる。

表1にEQL言語を構成するコマンドの一覧を示す。

4.1 定義系コマンド

ユーザーは、定義系コマンドを端末から使用して、データベース領域を確保して、そこにテーブルを定義することができる。図6に、DBSP1という名前の公用データベース領域を確保し、そこに図1のテーブルを定義する例を示す。INT-EGER、SMALLINT、VARCHAR、DECIMALはテーブルの欄のデータタイプを示している。表2にRDB1で扱えるデータタイプを示す。

表 I EQLコマンド データベース処理に必要な機能を包含したコマンドであり、エンドユーザーにも簡単に使用できる。

分 類	コマンド	機能
定義系	ACQUIRE DBSPACE	データベース領域の確保
	DROP DBSPACE	データベース領域の解放
	CREATE TABLE	テーブルの定義
	DROP TABLE	テーブルの削除
	ALTER TABLE	テーブルの拡張
	CREATE INDEX	インデックスの作成
	DROP INDEX	インデックスの削除
	COMMENT	注釈の付加
	SELECT	データの検索
ねんず	INSERT	データの追加
操作系	UPDATE	データの更新
	DELETE	データの削除
仕り分ので	COMMIT WORK	データ変更の書込みの指示
制御系	ROLLBACK WORK	データ変更の取消しの指示
=11 = 7	GRANT	権限の認可
認可系	REVOKE	権限の取消し

ACQUIRE PUBLIC DBSPACE
NAMED DBSP1

CREATE TABLE EMP

(EMPNO INTEGER,

ENAME VARCHAR(15),

DEPT SMALLINT,

SAL DECIMAL(8,0))

IN DBSP1

図 6 データベース領域の確保,テーブルの定義 公用データベース領域DBSP | を確保し図 | の社員(EMP)テーブルを定義する。

表 2 RDB I で扱えるデータタイプ RDB I では4種類の数値データと2種類の文字データが扱える。

形式	データタイプ	内容
数 値 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)
	DECIMAL (m, n)	固定小数点数(パック10進形式)
	FLOAT	倍精度浮動小数点数
文 字 データ	CHAR (n)	固定長文字列
	VARCHAR (n)	可変長文字列

ALTER TABLE EMP

ADD ADDRESS VARCHAR (60)

図 7 テーブルへの欄の追加 図 I の社員 (EMP) テーブルに住所 (ADDRESS)欄を追加する。

テーブルへの欄の追加も定義系コマンドを使って簡単に行 なえる。その例を**図7**に示す。

RDB1の定義系コマンドの特長は、データベースの再作成を行なわずにシステム稼動中に実行できることである。

4.2 操作系コマンド

データの検索,追加,更新,削除を行なう操作系コマンドの特長は、ユーザーがデータベースの物理的な構造やデータの取出し手順(アクセスパス)を意識する必要がないことと、テーブル全体(行の集合)に対して一括して処理が行なえることである。例えば、データ検索を行なうSELECTコマンドでは、ユーザーは得たいデータをどのテーブルからどのような条件で取り出すかだけを指定すればよい。そうすれば、RDB1自身が最適な手順を決めてデータの取り出しを行なう。

SELECTコマンドでは、複数の条件を組み合わせた検索や複数のテーブルを使った検索などを自由に行なうことができ、強力な検索機能をもっている。SELECTコマンドの例を図8に示す。

4.3 制御系コマンド

RDB1はデータベースの処理単位としてDBプロセスという概念をもっている。DBプロセスとは、ユーザーにとって一

SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL >= 200000

図8 データの検索 SELECTコマンドを使って、図 の社員(EMP)テーブルから給料が20万円以上の社員の社員番号、氏名、給料を求める。

GRANT UPDATE ON EMP TO USER1

REVOKE UPDATE ON EMP FROM USER1

図 9 アクセス権限の認可,取消し 図 | の社員(EMP)テーブルに対する更新(UPDATE)権限をユーザーUSER | に認可し、ユーザーUSER 2 から取り消す。

つの処理単位とみなす必要のある一連のEQLコマンド群であり、データベース更新の単位でもある。ユーザーはCOMMIT WORK、ROLLBACK WORKコマンドを使ってDBプロセスで行なったデータベース更新を有効にしたり無効にしたりできる。

4.4 認可系コマンド

RDB1では、ユーザー権限のクラス分けによるデータベース操作の限定とテーブルに対するきめの細かいアクセス権限の設定により、データベースの機密保護機能を実現している。ユーザー権限、アクセス権限の認可及び取消しは、認可系コマンドを使って端末から即座に行なうことができる。図9にGRANT、REVOKEコマンドの例を示す。

5 エンドユーザーインタフェース

RDB1では、エンドユーザーインタフェースと呼ぶ端末ユーザーのためのインタフェースを提供している。ユーザーは端末からエンドユーザーインタフェースを起動し、EQLコマンドを使って、RDB1データベースをアクセスすることができる。エンドユーザーインタフェースの起動から終了までをEQLセッションと呼ぶ。EQLセッションの開始から終了までの手順を図10に示す。

エンドユーザーインタフェースでは、端末操作を簡単にするために次の機能をもっている。

(1) 複数データの挿入

データの挿入はINSERTコマンドを使うが、TSS(Time Sharing System)端末から何件かのデータを挿入する場合は、INPUTコマンドを使用するとデータだけを繰り返し入力すればよい(図10の8~0参照)。

(2) コマンドの再実行

エンドユーザーインタフェースでは、最新のEQLコマンドを保持しているため、連続して同じコマンドを実行したい場合、STARTコマンドを使って再実行することができる(図10の①~①参照)。

(3) コマンドの実行保留

コマンドの内容を一部変更しながら使用する場合,変更部

	EQL 3	EQLセッション起動		
0	ENTER USERID: 5	ユーザー識別名応答		
4	ENTER PASSWORD:	パスワード応答		
U	AUTOCOMMIT IS ON.			
0				
8	INPUT EMP(EMPNO,ENAME,	被数データを入力するテーブル		
(9)	DEPT,SAL) (701413, トウキョウ タロウ ,100,250000)	と欄の名前を指定する。 欄に対応して,データを入力する。		
9	723850, トツカ ジロウ ,200, 2000000	はに対して、ケースを入りする。		
(10) (10)	F 63	INPUTコマンドを終了する。		
(12)				
	EMPNO ENAME SA	L		
	701413 トウキョウ タロウ 25000			
	723850 トツカ ジロウ 200000 :	0		
B	EQL :			
14	START	再度④のSELECTコマンドを		
	EMPNO ENAME SAL			
	701413 トウキョウ タロウ 250000	0		
	723850 トツカ ジロウ 200000	0		
(F)	EQL .			
16	HOLD SELECT & 01, & 02 FROM EMP	SELECTコマンドを保留する。		
0	EQL			
(18)	START (ENAME, EMPNO) ENAME FMPNO	保留したSELECTコマンドを ENAME,ENOの順に表示する。		
		ENAME, ENOUMIC 表示する。		
	トウキョウ タロウ 701413 トツカ ジロウ 723850			
	1			
(10)	STORE CATI	/P 67 + 63 0 \ 1" + 0 A T 1		
(13)	STORE DATE	保留した®のコマンドをCAT1 で登録する。		
1	EQL			
21)	START CATI (ENAME, SAL)	登録したCAT1を実行する。		
	ENAME SAL	=		
	トウキョウ タロウ 250000 トツカ ジロウ 200000			
	:			
	EQL			
23)	SELECT NAME FROM EMP	欄名NAMEは定義されていない		
24)	CHANGE/NAME/ENAME/	ためエラーとなる。 定義したENAMEに変更する。		
3.55	EQL			
(26)	START	変更したコマンドの実行		
	ENAME	V:		
	トウキョウ タロウ トツカ ジロウ			
Ø	EQL			
28	EXIT FOL SESSION NORMALLY ENDED	EQLセッションの終了		
29 EQL SESSION NORMALLY ENDED.				

図10 EQLセッションの利用手順 EQLセッションの中で、エンドユーザーインタフェースが提供するコマンドも利用できる。図の中で①はユーザーが入力する部分であり、②はエンドユーザーインタフェースが出力する部分である。

分を可変記号を使って定義し、HOLDコマンドでそのコマンドの実行を保留しておくことができる。実行する場合、STARTコマンドに可変記号の内容を指定するだけで済み、何度もコマンド全体を入力する必要はない。

(4) コマンドのカタログ機能

一つのEQLコマンドを登録するストアドEQLと、連続した 複数のEQLコマンドを一つの作業として登録するEQLプロシ ジャがある。 ストアードEQLは、STOREコマンドで登録し、その実行はSTARTコマンドで行なうことができ、登録や実行の操作が簡単になっている(図10の①~②1参照)。

EQLプロシジャは定形業務など、幾つかのコマンドで一連の作業として保存したい場合に利用すると便利である。EQLプロシジャを構成するコマンド群は、INSERTコマンド又はINPUTコマンドで登録し、RUNコマンドで実行する。

(5) コマンドの修正

最新のコマンドの一部を変更したい場合,又は入力したコマンドにエラーがあった場合,必要な箇所をCHANGEコマンドで修正し,STARTコマンドで再実行することができる(図10の②3~②6参照)。

エンドユーザーインタフェースで使用できるコマンドを**表** 3に示す。

6 データベース管理者の役割

RDB1を利用したデータベースシステムでも,集中化されたデータ情報を維持,保全してゆくために,従来と同様に,

表3 エンドユーザーインタフェースのコマンド エンドユーザーインタフェースは、端末までの操作性を向上させるコマンドを提供している。

分 類	コマンド	機能
E Q L	EQL	EQLセッションの起動
セッション制御	EXIT	EQLセッションの終了
	INPUT	挿入データの入力開始
=	SAVE	入力したデータのデータベースへの格納
データ入力	BACKOUT	入力したデータの取消し
	END	INPUTコマンドの終了
	STORE	EQLコマンドの登録
	ERASE	登録EQLコマンドの消印
ストアードEQL	RENAME	登録EQLコマンドの名称変更
	RECALL	登録EQLコマンドの呼び出し
	LIST	登録EQLコマンドの表示
EQLプロシジャ	RUN	EQLプロシジャの実行
コマンド保留	HOLD	入力したEQLコマンドの実行保留
コマンドの再実行	START	最新EQLコマンド,保留したEQLコマンド,又は登録EQLコマンドの実行
コマンドの修正	CHANGE	EQLコマンドの修正

SELECT TNAME
FROM SVSTEM. SYSCATALOG

TNAME

SYSCATALOG
SYSCOLUMNS
SYSDBSPACES
. . .

図 II ディクショナリテーブルの利用 テーブルの管理情報を保持するディクショナリテーブルSYSCATALOGを検索して、テーブル名の一覧表示する。

データベース管理者は必要である。

データベース管理者の役割として,以下のことがある。

- (1) データベース領域の確保,管理
- (2) テーブルの設計, 定義, 作成及び保守
- (3) ユーザーIDの登録, 管理
- (4) 機密保護
- (5) データの保全

従来のDBMSでは、共用度の高いデータから個人的なデー タに至るすべてのデータに対して、上記の役割をデータベー ス管理者が負担していた。これに対してRDB1では、EQLコ マンドを使ってユーザー自身でテーブルを作成し、それに対 する他ユーザーのアクセス権限を設定できる機能を提供して おり、個人的なデータ(テーブル)に関する管理、運用はユー ザー自身で行なえるようになっている。したがって、RDB1 のデータベース管理者は、個人的なデータに関する管理、運 用をユーザー自身に任せることにより、データベース管理の 負担を軽減することができる。更にRDB1では、データベー スのデータに関する情報をシステムが作成した特別のテーブ ルに保持し、データベースの最新の状態を知ることができる データディクショナリ機能を組み込んでいる。このディクシ ョナリテーブルは、ユーザーのテーブルと同様にEQLコマン ドを使って検索することができる。この機能を利用してデー タベース管理者は、ユーザーが作成、管理しているテーブル を含めデータベース全体の最新の状況を簡単に端末から知る ことができ、データベース管理の負担を少なくできる。図11 に、ディクショナリテーブルを検索してデータベース中のテ ーブル名の一覧を得る例を示す。

7 結 言

以上本論文では、RDB1開発の背景と製品の特長、データベースの基本的な構成、EQL言語の特長と機能概要、エンドユーザーインタフェース及びデータベース管理者の役割について述べた。

RDB1の開発により、従来の構造形データベースにない柔軟性の高いリレーショナルデータベースシステムの構築が可能となった。RDB1は既にユーザーに出荷されており、RDB1を使ったデータベースシステムの建設が行なわれている。

今後の中・大形ホストコンピュータのデータベースシステムを考えると、集中化されたデータを多種多様な形で利用するニーズと、ユーザー個別のデータをユーザーごとに利用するというニーズが混在してくると予想される。このようなデータベースシステム環境で、リレーショナルデータベースのもつ柔軟性、即応性が更に重要になってゆくものと考えられる。RDB1は、このような今後のデータベースシステム環境に対応できる機能をもつ製品であり、今後とも拡張、改良を重ねていく考えである。

参考文献

- E.F.Codd: A Relational Model of Data for Large Shared Data Banks, Comm.ACM, 13, 6, 377~387(1970)
- 2) 福嶋, 外: リレーショナルデータベースシステムの動向, 日立評論, 64, 5, 335~338(昭57-5)
- 3) プログラムプロダクト VOS 3 データマネジメントシステム RDB 1 解説, 日立製作所(1984)
- 4) プログラムプロダクト VOS1/ES データマネジメントシステム RDB1 解説, 日立製作所(1984)