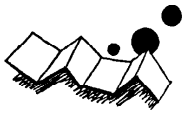


解説



関係モデルとその実現技術†

牧之内 顕文**

1. 緒言

1968年もおしつまった頃¹⁰⁾, 米国カリフォルニア州サンノゼにある IBM 研究所で「企業のデータベース中に共通して見られる情報の抽象モデルについて」の研究が開始された。この研究の目的は「データベースとデータベース管理システム (DBMS) の理論的側面を、性能面は別にして、取り扱うための理論的基礎を確立する」ことであった。この研究の成果が「関係 (リレーショナル) モデル」である。

銀行システムのような大規模な商用アプリケーションでは、ソフトウェアの開発に多数のプログラマの長期間にわたる作業が要求される。また一たん開発されたソフトウェアには不断の保守を必要とする。一方、応用プログラマに対するエンドユーザからの新規応用プログラムの開発要求は拡大の一途をたどっており、いわゆる「開発の積み残し」は業務のコンピュータ化をはばむ一つの障壁になっていた。このような状況からソフトウェアの保守を含めたソフトウェア開発の生産性向上が重要課題として意識されるようになっていた。

一方、DBMS は応用プログラマからハードウェア及びオペレーティング・システムを隠すと同時に、単純で使い易いデータビューを提供するソフトウェアとして、したがってプログラマの生産性向上の道具として、60年代後半から70年代にかけて大きな期待を担った。しかしこの期待は、Coddによれば¹¹⁾以下の理由により満足のいく成果を生まなかった。すなわち、当時の関係型でない DBMS では、

「(1) 応用プログラマは、データの検索や操作にとって本質的でない多くの概念で悩まされたばかりでなく、データの細かい構造という低次元のことを考えてプログラミングしなければならなかった。

(2) 集合処理がないため、不必要なくり返しループを考えつつプログラミングしなければならなかった。

(3) エンドユーザがデータベースに直接アクセスする必要性を正しく認識していなかった。」

これらのシステムはデータの論理的な見え方と物理的表現法とを明確に区別しなかったためにデータ独立性* を充分保証していない。特に、利用者にレコード間の構造的結合が見え、プログラマはこの結合をたどって (ナビゲートして) 目的とするレコードにたどりつかねばならない。これに対して、関係モデルの本質的部分は「これら結合を高水準演算子あるいは高水準命令 (コマンド) で置き換えたこと¹⁰⁾」である。

関係モデルの他モデルに比較した時の特長は次の3つに要約される¹¹⁾。

- (1) 高いデータ独立性。
- (2) 構造的に単純なモデル。これはエンドユーザとプログラマ間との意思の疎通を円滑にする。
- (3) 高水準のデータ操作言語。

2. データモデルとデータ操作言語**

関係モデルでは¹²⁾, データは二次元の表 (テーブル) で表現され (モデルの構造的側面), その表上の操作 (オペレーション) を通してアクセスされる。表は固定個数の列と番号づけられていない (unnumbered) 可変個数の行とからなっている。表の列の位置は意味を持たない。表上の操作を実現するデータ操作言語は集合操作を基本とした高水準言語で、関係代数と同等の能力を持つことが要請される。したがって、関係モデルの構造的側面のみを満足する DBMS は “semi-relational” と呼ばれる⁹⁾。

図-1 に表の例を示す。表 DEPT は3つの列からなり、それぞれ DNO, DNAME 及び LOC という

† Relational Data Model and its Implementation Techniques by Akifumi MAKINOCHI (Software Laboratory Fujitsu Laboratories Ltd).

** 富士通研究所

* 応用プログラムがハードウェア、オペレーティングシステム、データ記憶装置、記憶方式等の変更から保護されていること。

** data sublanguage, data manipulation language, query language 等の用語があるがこの解説では厳密に区別しては考えない。

MEMB

NAME	DNO	JOB	SAL
田中	50	10	80k
林	52	6	20k
上田	52	10	50k

DEPT

DNO	DNAME	LOC
50	医学部	千代田区
51	工学部	太田区
52	文学部	太田区

JOB

JOB	TITLE
10	教授
12	名誉教授
8	助教

図-1 大学職員データベース

名前(列名, フィールド名または属性名などと呼ばれる)で区別される。

データ操作言語としては関係論理に基づいた ALP HA⁷⁾や QUEL¹⁵⁾, 関係演算子言語⁸⁾あるいは SQL³⁾,¹⁶⁾や RDB/QL^{21), 22)}などの写像(mapping)型言語, さらにはエンドユーザインタフェースに特色のある QBE²⁷⁾(Query By Example)等がある。以下では実働化された言語の代表として QUEL RDB/QL,* QBE を選び図-1 のデータベースに対する簡単な問い合わせの例で各々の言語の特徴を示す。ALPHA, SQL 及び QBE に関しては文献 12) に詳しい。また種々の問い合わせ言語の総合的比較は文献 19) にある。

検索「太田区にある学部の講師の氏名, 給与, 所属学部名及び資格を列挙せよ」

QL,** RANGE (M,D,J) IS (MEMB, DEPT, JOB)
 RETRIEVE (M. NAME, M. SAL, D. DNAME, J. JOB)
 WHERE (J. TITLE='講師') AND (D. LOC='太田区')
 AND (M. DNO=D. DNO) AND (M. JOB=J. JOB)
 RQL,** GET NAME, SAL, DNAME, JOB, JOB
 FROM MEMB, DEPT, JOB
 WHERE TITLE='講師' AND LOC='太田区'
 AND MEMB. DNO=DEPT. DNO
 AND MEMB. JOB=JOB. JOB;

QBE,**

MEMB	NAME	DNO	JOB	SAL
P. 鈴木	X	Y	P. 50	

* SQL と RDB/QL とは一部の機能を除いてほとんど同じなので SQL についてはここでは触れない。

** QL=QUEL, RQL=RDB/QL. SQL は RDB/QL に一部の機能を除いて構文意味ともほとんど同じである。QBE の例ではイタリックと太字の部分が利用者によって指定される。

DEPT	DNO	DNAME	LOC
	X	P. 医学部	太田区

JOB	JOB	TITLE
Y		講師

更新「工学部の所在地を目黒区にし, その教育すべての給与を5%上げよ」

QL, RANGE (M,D) IS (MEMB, DEPT)
 REPLACE M (SAL BY M. SAL*1.05)
 WHERE M. DNO=51
 REPLACE D (LOC BY '目黒区')
 WHERE M. DNO=51
 RQL,* UPDATE DEPT, MEMB
 FROM DEPT, MEMB.
 SAL=SAL*1.05
 LOC='目黒区'
 WHERE MEMB. DNO=DEPT. DNO
 AND DNAME='工学部';

QBE,

MEMB	NAME	DNO	JOB	SAL
		51		X
UPDATE.		51		X*1.05

DEPT	DNO	DNAME	LOC
UPDATE.	51		L
	51		目黒区

3. システム構成とストレージ構造

ここでは INGRES, SYSTEM R 及び RDB/V1 を取り上げる。これら3つのシステムはすべて研究所で**プロトタイプが開発され, その後商品化された。この3つを取り上げた理由は, これらシステムについては多数の論文が発表されており, 比較的その内容が分かっている*** からである。

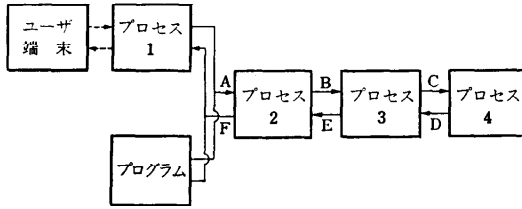
INGRES^{15), 24), 25)}

このシステムはミニコンピュータ PDP 11 上の UNIX オペレーティングシステム下で動作するように設計されたため, 番地空間として 64k バイトを持つ複数プロセスが UNIX のプロセス間通信機能を使って動くシステム構成になっている。図-2 に構成を示す。プロセス1はインタラクティブ端末モニタで命令の入力, 印刷, 編集及び実行を管理する。プロセス2は構文解析, 保全(インデグリティ)管理及び排他制

* SQL では QL や QBE と同様にこの更新は2つのコマンドによって実現される。このため SQL では QL や QBE と同様に工学部の DNO が 51 であることを利用者が指定する必要がある。

** INGRES: Electronics Research Laboratory, University of California Berkeley. SYSTEM R: IBM Research Laboratory, San Jose, California. RDB/V1: 富士通研究所。

*** 著者は RDB/V1 開発のメンバーの1人であった。



各プロセスは UNIX のパイプ (A~F) を通じてデータの授受を行う。(文献 24) より

図-2 INGRES のプロセス構成

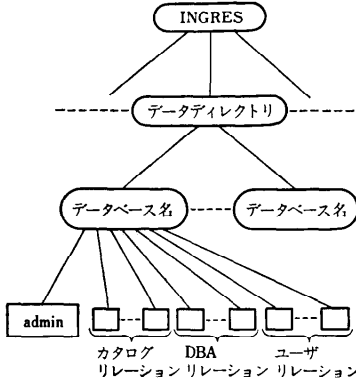


図-3 INGRES ファイル構成 (文献 24) より

御を行う。解析結果のトークン列がプロセス 3 に渡される。そこでは検索や更新命令が解釈実行される。更新された組 (タプル) の写しが特別のファイルに転送される。実更新はそのファイルを元に遅延 (deferred) 更新プロセッサが行う。プロセス 4 は表の作成・削除などのユーティリティ命令を実行する。

C プログラム中に書かれた QUEL は前処理されて INGRES の呼び出し列に変換される。C コンパイラで作られ実行モジュールはパイプ A を通じて命令をプロセス 2 に渡す。レコード処理要求のためにプロセス 3 と C プログラムとの間に特別なパイプが設定されている。

図-3 は INGRES のファイル構成である。利用者データはユーザリレーションファイルに格納される。INGRES でサポートされるファイル編成法は 5 種類あり、その内の 4 種類がキー付である。他の 1 つは到着順であり、作業用ファイルに主として使われる。キー付はハッシュ編成と ISAM 編成がある。さらに各々に圧縮オプションがある。キー値はそれを持つ組が格納されるページを識別するために利用される。

SYSTEM R^(2),5),4)

このシステムは IBM VM/370 環境下で動作するソ

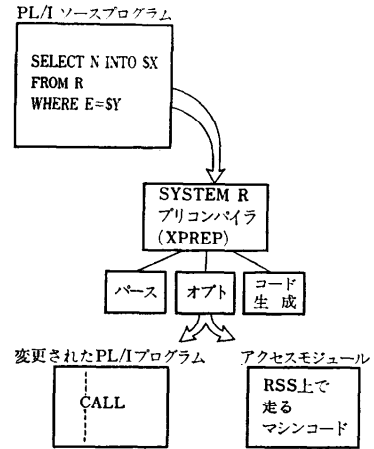


図-4 SYSTEM R のプリコンパイル段階 (文献 5) より

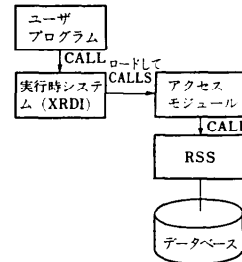
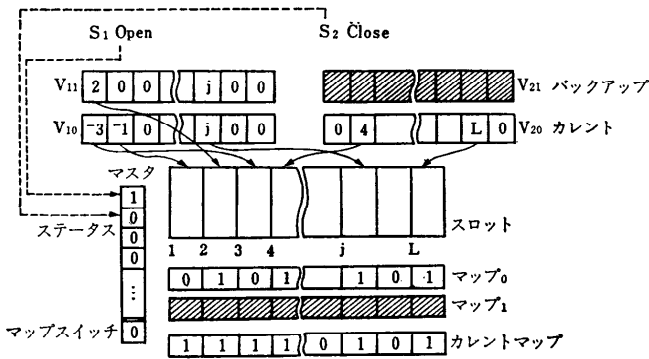


図-5 実行段階 (文献 5) より

フトウェアとして設計開発された。システムは RDS (Relational Data System) と RSS (Research Storage System) とから構成される。RDS はソースプログラム中の SQL 文を解析し、点検する。その後、評価のための最適アクセス経路を選択し、機械語のサブルーチン群 (アクセスモジュール) を作り出す (図-4)。このアクセスモジュールは RSS を通じてデータベースにアクセスする (図-5)。このコンパイル方式の採用は「SYSTEM R プロジェクトの最も成功した部分の 1 つ」⁵⁾ であるといわれている。その利点は、命令の解析、点検及び最適化がコンパイル時に 1 回実施されるだけであることとアクセスモジュールが専用・小型化され、したがって効率がよくなるからである⁵⁾。

RSS は表へのアクセス経路を提供するとともに、ロギングやロックのための機能もそなえている。アクセス経路としては (1) インデックス走査 (スキャン)、(2) セグメント走査及び (3) リンク走査が用意されている。インデックス走査では、B-木方式で



セグメント S₁ が修正された。マスタはセグメントの状態（オープンかローズか）を示す。カレントマップはスロットの使用状態を、またページテーブルのエントリにつけられた点は該当するページが現トランザクション中に変更されたことを、各々示す。

さらにマップスイッチの状態によってマップ。かマップ₁ かを選択する。（文献 20）より）。

図-6 Shadow Page 方式によるセグメントの管理

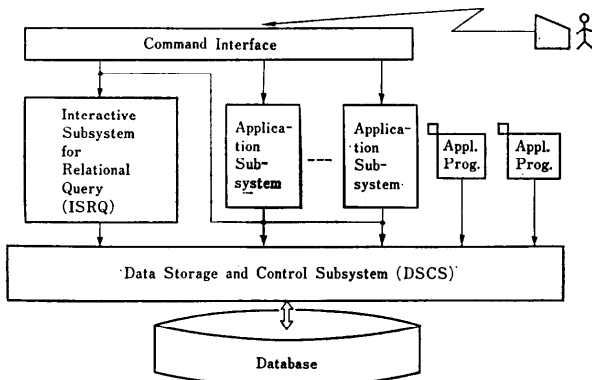


図-7 RDB/V1 システム構成

実現されているインデックスを通してキー順に表を走査する。インデックスの保守は RSS によって自動的に行われる。リンクはポインタによるレコードの結合であるが利用者データへのアクセス経路としては使われていない。利用者データ、アクセス経路構造、内部カタログ情報、RDS が使用する中間結果等のデータはすべて 4k バイト長のページから構成されるセグメントに格納される。セグメントのディスク上の物理空間へのマッピングは RSS の仕事である。ページのためのディスク空間は動的に割り当てられる。またページはディスクから仮想空間への転送単位でもある。セグメントの復旧（リカバリ）のために影ページ (shadow page) 方式が提案された (図-6)。

RDB/V1^{21), 22)}

RDB/V1 は開放型 (Open-ended) システムとして

設計された。そのため対話型 RDB/V1 には応用システムをサブシステムとして容易に組み込む。したがって、特定応用分野の汎用アプリケーションシステムを RDB/V1 に組み込めば、データの検索ばかりでなく、適用分野にそったデータの処理・加工も対話的に行うことが可能になる。図-7 にシステム構成を示す。コマンドインタフェースは命令を振り分ける。ISRQ はデータ操作命令の解析、点検、スケジューリング及び解釈実行を行う。DSCS は表の編成とアクセスのための諸機能、排他制御、ログ/復旧及びバッファ管理のための機能を提供する。利用者に見える表のタイプには到着順、キー順及びハッシュの 3 種類がある。これらの区別は DSCS レベルでなされて

いる。一方、アクセス補助手段としてインデックスが利用者に提供されるが、これは ISRQ が DSCS レベルでのキー順編成の表* で実現している。

RDB/V1 の親言語インタフェースのためにプレコンパイラ**がある。これはコンパイル方式によって実現している²³⁾。

データ格納空間は 4k バイト長のページからなるセグメントである。各セグメントは自身に格納される表やインデックスに関するデータ辞書 (これはまた DSCS レベルのハッシュ編成表で実現されている) を備えているため 1 つの独立したデータベースと考えてよい。またセグメントは復旧の単位でもある。図-8 に RDB/V1 のファイル構成を示す。SYSTEM R と同様に 1 つの論理ページに 2 つの物理ページを割り当てる影ページ (shadow page) 方式が採用されているが、その機構実現に必要な管理テーブル*** (ページテーブル、スロットマップ) もまた二重化されていることが特徴である (図-9 参照)。

4. 問い合わせ評価

INGRES, SYSTEM R および RDB/V1 の 3 つのシステムで実現された問い合わせ評価法について簡単に述べる。以下の説明で使用するデータベースは 図-1

* DSCS レベルのデータは組からなる表に格納される。しかしデータへのアクセスは組 (レコード) 単位でかつ表の編成法を意識しなければならない。

** 現在 SPL/100。

*** このテーブルは関係表ではない。

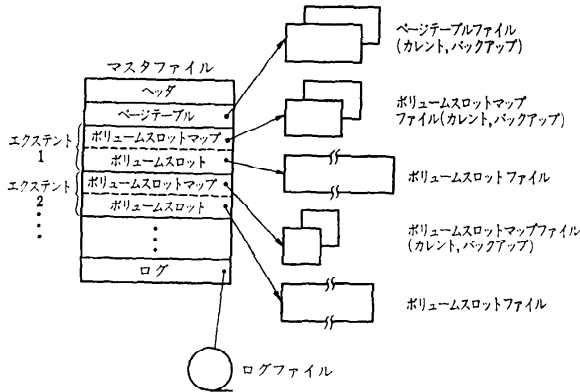


図-8 RDB/V1 セグメントのファイル構成

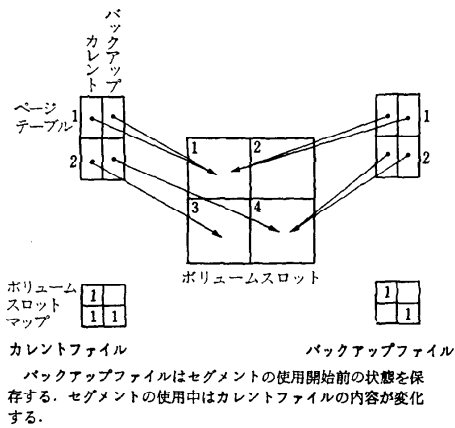
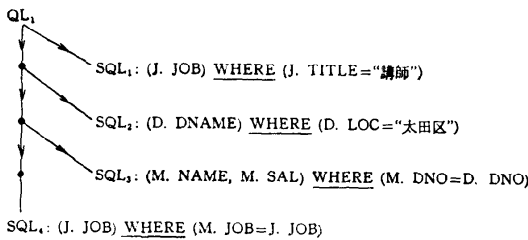


図-9 RDB/V1 のセグメント管理



DNO	DNAME
51	工学部
52	文学部

図-10 問い合わせ QL₁ の 1 変数分離による分解と部分問い合わせ SQL₁ の評価結果 (文献 26) に準拠

に示されたものである。

INGRES^{26), 24), 15)}

複数の変数を含む問い合わせは 1 変数の問い合わせの列に分解される。1 変数の問い合わせは 1 変数問い合わせプロセッサ (One Variable Query Processor) によって評価される。分解のために使用する技法は、(1) 組代入 (tuple substitution) と (2) 1 変数分離 (one-variable detachment) である。今、第 2 章で示した検索例 QL₁ を考える。これは 3 変数問い合わせである。QL₁ は 1 変数分離によって 図-10 に示しているように分解される。SQL₁ と SQL₂ は OVQP によって評価され、各々 1 時テーブルに結果が保存される。

今 SQL₂ の評価結果が同図に示されているようになったと仮定すると、SQL₃ の変数 D への組代入は

SQL₃ (51) RETRIVE (M. NAME, M. SAL)
WHERE (M. DNO=51)

SQL₃ (52) RETRIEVE (M. NAME, M. SAL)
WHERE (M. DNO=52)

の 2 つの 1 変数問い合わせを生じさせる。これらは順次 OVQP で評価される、SQL₃ (51), SQL₃ (52) を評価する時、変数 M のレンジである表 MEMB は DNO をキーとするハッシュ表であれば評価効率はいよと考えられる。INGRES では組代入により数多くの 1 変数問い合わせが評価されるような時、その変数が動く表は、その問い合わせが評価しやすくなるように再構成 (reformat) されることがある。また 1 変数分離の結果できる表はその後の処理を考慮してハッシュ表に作られることもある。

2 個以上の変数を含む部分問い合わせでどの変数から代入を始めるのかの決定をしなければならない。この局面で最適化が有効であると考えられたが、INGRES では単純に、最小の要素数をもつ表をレンジとする変数から代入を行う。

OVQP は 1 変数問い合わせを評価する時に必要なアクセス経路の選択を行う。すなわち、単純節* (simple clause) を評価するためにハッシュドプライマリ表、ハッシュドインデックス、ISAM プライマリ表及び ISAM インデックスが使えるかどうかをこの順序で検討する。

SYSTEM R^{14), 2)}

* フィールド {=, <, >, <=, >} 定数。

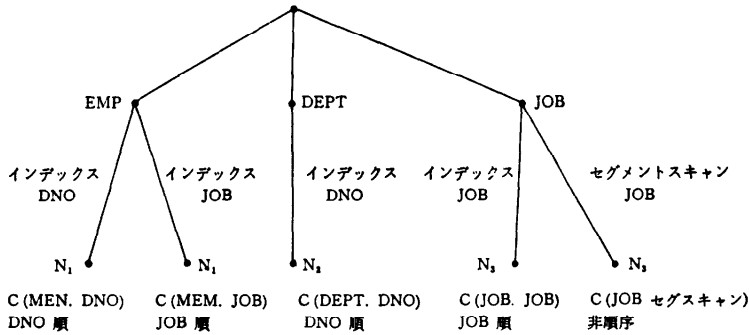


図-11 単一テーブルに関するアクセスパス選択探索木 (文献14) に準拠

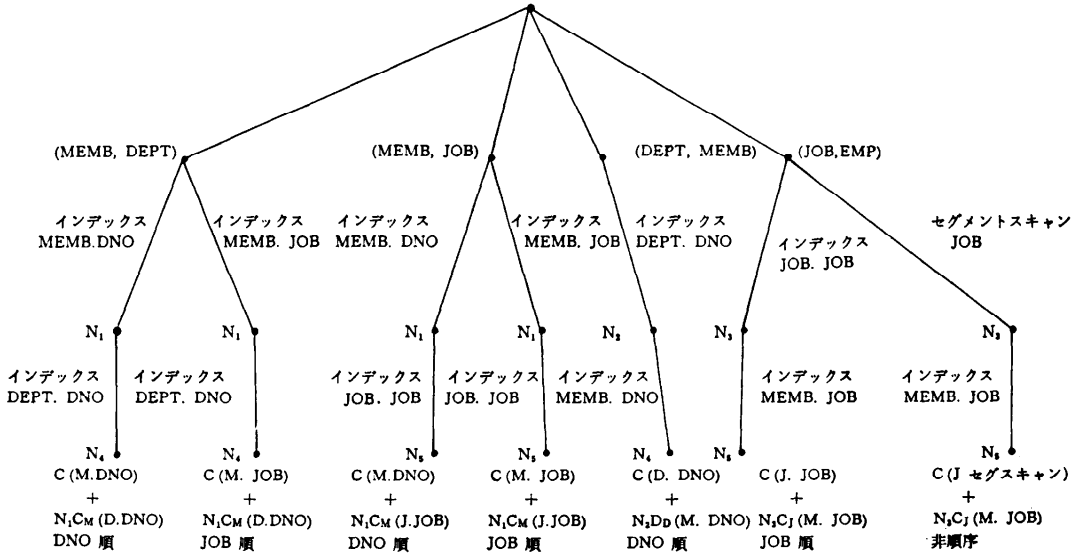


図-12 拡大探索木 (入れ子ループ法) (文献14) に準拠

使用可能なアクセス経路としてはインデックスとセグメントの走査がある。特定の表を特定のアクセス経路を使用してアクセスする時予想されるアクセスコストが下の式¹⁴⁾にしたがってシステムによって計算される。

$$COST = PAGE \text{ FETCHES} + W * (RSI * CALLS)$$

この評価式は I/O と CPU 使用の両方を考慮したものになっており、W は重み付け係数である。このコスト計算のために、システムは表及びアクセス経路に関する種々の統計データ (表の要素数、占有ページ数など) をカタログ中に保持する。これら統計データは 'UPDATE STATISTICS' 命令によって利用者により更新される。また論理述語に対して選択率 (selective factor) が定義される。たとえば、'column = value' に対しては、もしその column にインデック

* RSS のインタフェース。

スが張ってあればそれは 1/(インデックスの異なりキー数) とし、そうでなければ 1/10 としている。

結合演算の評価法としては入れ子ループ (nested loop) 法* とマージング・スキャン結合法とを採用している。結合述語を持つ問い合わせの評価は可能なアクセス経路とこの2つの評価法のすべての組み合わせの中からコストが最小であると予想されるものを選ぶことになる。図-11 と 図-12 にそのための探索木の一例を示す。図-11 は問い合わせ中の結合述語に着目して、その評価のために一つの表にアクセスする場合のアクセス経路、評価結果の (予想) 要素数、(予想) アクセスコスト、アクセス時の順序等を木状にまとめたものである。表 MEMB, DEPT のアクセス経路としてのセグメント走査は他のアクセス経路に比べてコストが高過ぎるとして、この時点で探索木より排除されて

* これはタプル代入法と基本的に同じである。

いる。図-12 は2つの表の入れ子ループ法による結合演算のためのアクセス経路の可能な組み合わせを示している。たとえば MEMB と DEPT の結合 ('MEMB・DNO=DEPT・DNO') は最初に MEMB の組を取り出し、次に DEPT の組を取り出すという手順とその逆が考えられる。それは、探索木上で (MEMB, DEPT) と (DEPT, MEMB) ノードに反映されている。葉ノードにはそれに到るアークに記されたアクセス経路を使用した時の結合演算コストが書かれている。このように順次拡大探索木を作ってゆき、最後にコストが最小と予想される結合手順とアクセス経路を選ぶ。

RDB/V1^(22), 18)

最適化についての詳細は本学会論文誌に発表予定¹⁸⁾なのでここでは簡単に触れるにとどめる。

アクセス経路としてはインデックス走査と表走査を備えている。前者ではインデックスの走査で得られた組ポインタの集合をソートして、同一ページは最大1回しかアクセスしないように制御している。後者はテーブルの種類¹⁹⁾に応じた走査範囲限定手段が用意されている。

最適アクセス経路の選択規準としては、取り出される論理ページ数をとっている。このコスト計算のために必要インデックスや表に関する統計データが各データ辞書中に自動的に保存される。

問い合わせの条件を構成する述語は、論理的意味と経験則にそっておおよそ5つのカテゴリに分類される。

問い合わせの評価は、その述語をカテゴリ順に評価する

PARTS	PARTNO CHAR(6)	DESCRIP CHAR(50)VAR	QOH INTEGER	QOO INTEGER	
ORDERS	ORDERNO CHAR(6)	PARTNO CHAR(6)	SUPPNO CHAR(3)	DATE CHAR(6)	QTY INTEGER
QUOTES	SUPPNO CHAR(3)	PARTNO CHAR(6)	MINQ INTEGER	MAXQ INTEGER	PRICE INTEGER

Table	No. of Records	Avg. length of record (bytes, incl. header)	Indexes Maintained
PARTS	20,000	36	PARTNO DESCRIP
ORDERS	60,000	31	ORDERNO PARTNO SUPPNO
QUOTES	180,000	27	SUPPNO PARTNO

図-13 SYSETEM R の性能評価に用いられたデータベース例 (文献4) より)

* 到着順, ハッシュ及びキー順。

ことによって行われる。各カテゴリの述語の評価の結果作られる作業用の表は次の述語の評価に必要なならソートされる。

等価結合述語と等価相関 (コリレーション) 述語の評価はマージング・スキャン結合法が使われる²⁰⁾。したがって結合される表は結合フィールドでソートされているかあるいはその上にインデックスが張られているかしなければならない。システムは結合列の最大・最小値をソート時またはインデックスより得てその値を他の表の縮小²¹⁾や走査範囲の制限に用いる。これは問い合わせの条件部に新たに '仮想述語' を付加するのと同じ効果を持つ。

表-1 SYSTEM R における問い合わせ評価例 (文献4) より)

English form:
Find the supplier number and price of all quotes for part number 010002 in quantities of 1000.

SQL form:
SELECT SUPPNO, PRICE FROM QUOTES
WHERE PARTNO = '010002'
AND MINQ <= 1000 AND MAXQ >= 1000;

Access path chosen by optimizer:
PARTNO index on QUOTES table.

Cardinality of answer set: 3

Operation	CPU time (ms on 168)	Number of I/Os
Parsing	13.3	0
Optimization	40.0	9
Code generation	10.1	0
Open cursor	3.7	5
Fetch answer set (1.5 ms per answer record)	4.6	2
Close cursor	1.2	
Total (including all of above plus formatting answer set for display)	83.4	16

(a) 問い合わせ 1

English form:
Find the order number, part number, description, date and quantity for all parts ordered from supplier number 797 during 1975.

SQL form:
SELECT ORDERNO, ORDERS.PARTNO, DESCRIP, DATE, QTY FROM ORDERS, PARTS
WHERE ORDERS.PARTNO = PARTS.PARTNO
AND DATE BETWEEN '750000' AND '751231'
AND SUPPNO = '797';

Access path chosen by optimizer:
Access ORDERS by SUPPNO index. For each qualifying ORDERS record, access corresponding PARTS record by index on PARTNO.

Cardinality of answer set: 7

Operation	CPU time (ms on 168)	Number of I/Os
Parsing	20.7	0
Optimization	73.2	9
Code generation	19.3	0
Open cursor	4.0	6
Fetch answer set (8.7 ms per answer record)	61.1	75
Close cursor	5.3	
Total (including all of above plus formatting answer set for display)	213.9	90

(b) 問い合わせ 2

* ハッシュテーブルが関与する場合、入れ子ループ法が使われる場合もある。

** restriction と projection によって表のサイズを小さくすること。

表-2 トランザクション性能測定のためのデータベース例 (文献4) より

Database	Number of PARTS records	Number of ORDERS records	Number of QUOTES records	Total size of database in megabytes (not including indexes)
1	5,000	15,000	45,000	1.86
2	20,000	60,000	180,000	7.43
3	40,000	120,000	360,000	14.87
4	70,000	210,000	630,000	26.02
5	100,000	300,000	900,000	37.17

5. 性能

性能については SYSTEM R の評価結果が種々発表されている。ここでは文献4)に沿ってそれを簡単に紹介する。

図-13 は端末からの問い合わせ評価の性能測定に用いられたデータベースである。表-1 はその測定結果例である。また PL/I プログラムをホストとするトランザクション処理効率測定の対象となったデータベースのサイズが表-2 に示してある。サイズによる CPU 時間の変化を示しているのが図-14 である。Nタイプのトランザクションは1件の新規注文を表 ORDERS に格納した後、適当な PARTS レコードの QOO フィールドを更新する。Aタイプのトランザクションでは部品の既注文が到着するとその注文中の部品番号と量を表 ORDERS 中に見つけ、それにしたがって該当する表 PARTS のレコードを更新する。その後該当 ORDERS レコードをデータベースから削除する。またタイプQのトランザクションでは部品番号が与えられると、その部品の記述、手持量及び注文量を端末に表示する。

これらの性能測定は IBM 370 Model 168 の VM/CMS 下で、マルチユーザモードの SYSTEM R を使用して実施された。

6. 結論

エンドユーザが、EDP 室を経ずに、コンピュータに格納されたデータを直接見たいという要求は非常に強い。また EDP 室の応用プログラムの生産性を高める必要性はいうまでもない。関係 DBMS はこの両者を満足させる DBMS として囑望されている。特に、既存の(非関係型)DBMS ではデータベース化がためらわれていた分野でのデータベース化が、関係データベース管理システムの出現により急速に実現されると思われる。

一方、高トランザクション処理を要求する分野への

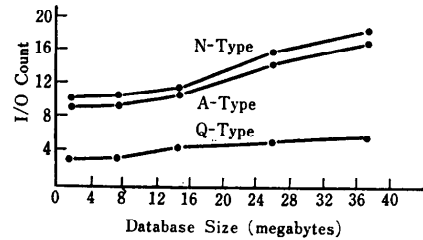
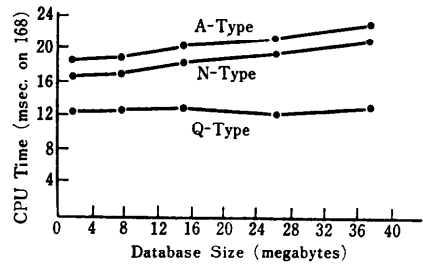


図-14 トランザクションの実行コスト (文献4) より

適用は、現在の所まだ難しいと考えられている。これもシステムの改良、メインフレームの高速化によって、そう速くない時期に実現されよう。さらに、関係データベースマシンのような専用マシン¹³⁾の登場により、実現時期は意外と早まるかもしれない。また一方データの分散によりトランザクションもまたうまく分散されるような応用分野では、分散型関係 DBMS の適用が有効となろう。

バックエンド DBMS マシンであれ分散 DBMS であれ、関係データベースの持つ高水準インタフェース(ホスト対バックエンド、ローカル DBMS 対ローカル DBMS) はそれらの実現に非常に有利に働くであろう。

以上から、関係 DBMS の重要性は、今後増すことはあれ減ることはないといえる。

参考文献

- 1) Astrahan, M. M. et al.: Performance of the System R Access Path Selection Mechanism, Proc. of IFIP 80, (1980).
- 2) Blasgen, M. W. et al.: System R: An Architectural Overview, IBM System Journal, Vol. 20, No. 1 (1981).
- 3) Chamberlin, D. D. et al.: SEQUEL 2: a unified approach to data definition, manipulation and control, IBM J. Res. Develop. (Nov. 1976).
- 4) Chamberlin, D. D. et al.: Support for Repetitive Transactions and Ad Hoc Queries in

- System R, ACM. TODS Vol. 6, No. 1 (Mar. 1981).
- 5) Chamberlin, D.D. et al.: A History and Evaluation of System R, C. ACM Vol. 24, No. 10 (Oct. 1981).
 - 6) Codd, E.F.: A Relational Model of Data for Large Shared Data Banks, C. ACM Vol. 13, No. 6 (June 1970).
 - 7) Codd, E.F.: A Data Base Sublanguage Founded on the Relational Calculus, Proc. 1971 ACM SIGFIDET Workshop on Data Description, Access and Control.
 - 8) Codd, E.F.: Relational Completeness of Data Base Sublanguages, In Data Base Systems, Courant Computer Science Symposia Series Vol. 6, Englewood Cliffs, NeJ. Prentice-Hall (1972).
 - 9) Codd, E.F.: Extending the Database Relational Model to Capture More Meaning, ACM TODS Vol. 4 No. 4 (Dec. 1979).
 - 10) Codd, E.F.: SQL/DS what it means, Computer World (Feb. 16 1981).
 - 11) Codd, E.F.: Relational Database: A Practical Foundation for Productivity, IBM Research Report RJ 3339, San Jose Calif. (Dec. 1981).
 - 12) Date, C.J.: An Introduction to Database Systems, Second Edition Addison-Wesley (1977).
 - 13) Epstein, R. and Hawthorn P.: Desing decisions for the intelligent database machine, NCC (1980).
 - 14) Griffiths Selinger, P. et al.: Access Path Selection in a Relational Database Management System, Proc. of SIGMOD Conference (1979).
 - 15) Held, G.D. et al.: INGRES—A Relational Data Base System, Proc. NCC 44 (1975).
 - 16) IBM: SQL/Data Subsystem Terminal User's Reference.
 - 17) IBM Data Processing Division: Licensed Program Announcement 5748-ZZJ (Jan. 1981).
 - 18) 北上 始他: 関係データベース管理システム RDB/V1 の最適化技法, 本学会論文誌に投稿中.
 - 19) Lacroix, M. and Pirotte, A.: Example queries in relational languages, Technical Note N107 M. B. L. E. Research Lab. Brussels, BELGIUM (Jan. 1977).
 - 20) Lorie, R. A.: Physical Integrity in a Large Segmented Database, ACM TODS Vol. 2, No. 1 (Mar. 1977).
 - 21) 牧之内顕文他: リレーショナル・データベース管理システム RDB/V1, データベース管理システム研究会 (1980年5月).
 - 22) 牧之内顕文他: 関係データベース管理システム RDB/V1, 本学会論文誌 Vol. 24, No. 1 (1983) に掲載予定.
 - 23) 佐藤秀樹他: RDB/V1 のホスト言語インターフェースの実現法, 昭和五十七年度前期情報処理学会全国大会.
 - 24) Stonebraker, M. and Wong, E.: The Design and Implementation of INGRES, Memo. ERL-M 577, Electronics Research Lab., University of California Berkely (Jan. 1976).
 - 25) Stonebraker, M.: Retrospection on a Database System, ACM TODS, Vol. 5 No. 2 (June 1980).
 - 26) Wong, E. and Youssefi, K.: Decomposition—A Strategy for Query Processing, Memo. ERL-M 574 Electronics Research Lab., University of California Berkeley (Jan. 1976).
 - 27) Zloof, M.M.: Query By Example, Proc. NCC 44 (1975).

(昭和57年6月9日受付)