

## 関係データベースシステム RDB/V1 の最適化技法†

北 上 始<sup>††</sup> 牧之内 顕文<sup>†††</sup>  
手塚 正義<sup>†††</sup> 安 達 進<sup>†††</sup>

関係データベースシステムにおいて、問合せの効率的な評価を行うためには、次の二点に注意しなければならない。一つは、射影、制限、結合などの関係演算により、問合せの評価手順を効率よく定めることであり、他の一つは、性能のよい関係演算を実現することである。前者を大局的最適化と呼び、後者を局所的最適化と呼ぶ。本論文では、上記、二つの最適化について述べる。大局的最適化では、カスケード法による最適化アルゴリズムが容易に実現された。カスケード法とは問合せの条件式をカテゴリに分類し、各カテゴリに経験的な評価の優先順位を与える方式である。RDB/V1では、各カテゴリの条件式は、関係演算を変形した本システム特有の基本演算（6種類）により評価される。局所的最適化では、等価結合演算または等価相関演算に対し、推移述語の定義に基づく最大最小法を実現した。また、最大最小法の効果を明らかにするために、この方式を適用した場合と適用しない場合の測定を行った。これにより、無駄なデータの参照回避による処理スピードの高速化が確認された。

### 1. 序 論

E. F. Codd が、データベースの関係モデルを提案して以来、それに基づいたデータベースシステムの実現可能性を実証するために、多くの実験システムが開発されてきた<sup>1), 18), 21)</sup>。これらのシステムのデータ操作言語は非手続き的な高水準言語であるため、ユーザは、データ編成や、アクセス法を意識したデータへのアクセス手順をプログラム化するという複雑で煩わしい仕事から解放される。これは、関係データベースシステムがユーザに替わって、自動的に実行計画を立て、高速に処理する能力を備えなければならないことを意味する。すなわち、関係データベースシステムでは、問合せの評価の最適化が必須である<sup>8)</sup>。

一般に、問合せの評価は、大局的な評価順序を、射影 (Projection)、制限 (Restriction)、結合 (Join)、直積 (Cartesian) などの関係演算の系列として表し、各演算を順に評価することにより達成される。これにより、最適化研究のテーマとしては、関係演算の系列を最適に定める立場の大局的最適化<sup>4), 17), 20)</sup>と、各関係演算の最適な実現を図る立場の局所的最適化がある。

RDB/V1<sup>11), 13)</sup>は、富士通研究所で開発した関係データベースシステムであり、上記、2種の最適化問題に

重点をおいて実現されたシステムである。

また、本システムの問合せ言語 RDB/QL は、SEQUEL II<sup>1)</sup>、QUEL<sup>22)</sup>に似ているが、これらの言語にない機能もつ。

本論文は、RDB/V1 に関連する3編の論文<sup>13), 14)</sup>のうちの一つであり、問合せの評価の最適化技法について述べたものである。本論文では、次の3点に着目した最適化技法について述べる。

#### (1) 大局的最適化

問合せの中に現れるレコード選択条件および集合演算などの評価順序を、RDB/V1 特有の基本演算から成る最適な演算系列により定める。

#### (2) 局所的最適化

上記、基本演算の最適な実現を図る。

#### (3) アクセスパスの選択

基本演算を評価するためのアクセスパスの選択を、アクセスコストの計算比較により行う。

(1)に関しては、カスケード法による簡易な最適化アルゴリズムを実現した。(2)に関しては、関係データベースのユーザが最も頻繁に利用する結合演算についての研究が集中的に行われてきた<sup>3), 16), 23)</sup>。本システムでは、等価結合演算および等価相関演算に対する特徴的な最適化技法として、推移述語を利用した最大最小法<sup>7), 12)</sup>を実現した。(3)に関しては、ページのI/Oコストに基づくコスト計算<sup>19)</sup>により実現した。

最後に、最大最小法の効果を明らかにするために、最大最小法を適用した場合と適用しない場合について処理スピードの性能測定を行ったので、その結果について述べる。

† Optimization Method in the Relational Database System RDB/V1 by HAJIME KITAKAMI, (Institute for New Generation Computer Technology) AKIFUMI MAKINOCHI, MASA-YOSHI TEZUKA and SUSUMU ADACHI (Fujitsu Laboratories Ltd.).

†† 新世代コンピュータ技術開発機構

††† 富士通研究所

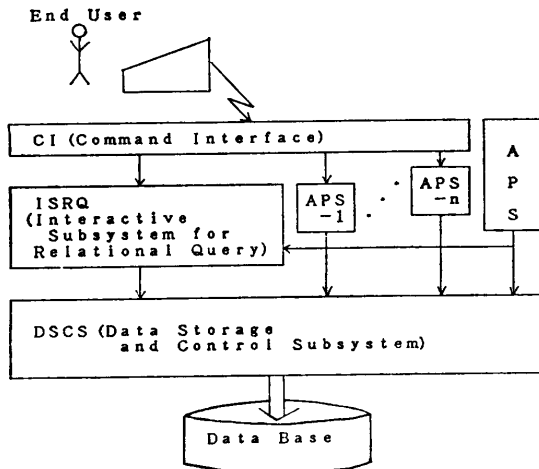


図 1 RDB/V1 のシステム構成図  
Fig. 1 Logical structure of RDB/V1.

## 2. システム構成

RDB/V1 のシステム構成を、図 1 に示す。本システムは、CI (Command Interface)、ISRQ (Interactive Subsystem for Relational Query) および DSCS (Data Storage and Control Subsystem) からなる。またこのシステムは APS (Application Subsystem) を容易に組み込めるオープンエンデッドなシステム構成をとっている。各サブシステムの詳細は他の論文<sup>13)</sup>に譲り本章は最適化に関連する部分を概略述べる。

問合せの最適化処理は、ISRQ 内のスケジューラおよびインタプリタで行われる。インタプリタでは、DSCS が提供するインタフェースでデータベースをアクセスする。

DSCS が提供するデータ編成には、(1)レコード到着順編成、(2)キー順編成、(3)ハッシュ編成がある。本論文では、(1)のテーブルをたんにテーブルと呼び、(1)のテーブルに対して作成された(2)のテーブルをインデックスと呼ぶ。(3)のテーブルは、ハッシュテーブルと呼ぶ。DSCS が提供するアクセス法 (DSCS のインタフェース) には、上記、(1)、(2)、(3)に対するテーブルのスキャンによるレコードの検索をはじめとして、TID\*(Tuple Identifier) によるレコードの検索/削除/更新などのオペレータがある。また、各オペレータには、乗法標準形 (Conjunctive Normal Form) で表現される簡単なレコード選択用の条件式\*\*を評価する機能がある。この条件式を、ス

\* レコードの識別子をさす。

\*\* <フィールド> <比較演算子> <定数> の形をした述語だけから構成される。ただし、比較演算子は、{= / ≠ / > / ≥ / < / ≤} である。

キャンアークメントと呼ぶ。

以下では、説明を簡単にするため、ハッシュテーブルに関連する説明を省く。

## 3. 問合せの表現形式

RDB/V1 の検索コマンドは、次のように表される。

GET 問合せブロック; (3.1)

詳細は省略するが、問合せブロックは、検索コマンド以外に、挿入 (Insert)、削除 (Delete)、更新 (Update) などのコマンドにも共通に使用される。

問合せブロックは、別の問合せブロックを、入れ子 (Nest) にした構造を許している。入れ子の構造は、たとえば、レベル *i* の問合せブロック  $Q_i$  が "フィールド  $\geq$  SUM (問合せブロック  $Q_{i+1}$ )" のような述語をもつ場合である。

また、 $Q_i$  は、 $m_i$  個の基本問合せブロック  $Q_i^j$  を、演算項とする集合演算により構成される ( $1 \leq j \leq m_i$ )。

基本問合せブロックは、検索対象とするテーブルから、レコード選択条件を満たす項目についてのデータを収集する記述 ((3.2) を参照) として定義される。

集合演算には、集合積 ( $\cap$ )、集合和 ( $\cup$ )、集合差 ( $-$ ) に加えて、単純和 ( $+$ ) が提供されている。単純和は、複数テーブルのレコードを、一つのテーブルによせ集めるだけの演算である。集合演算が存在しなければ、問合せブロックは、基本問合せブロックと一致する。

検索項目 (算術式または集合関数も可) のリスト  
FROM 検索対象とするテーブル名のリスト  
WHERE レコード選択の条件式

(3.2)

集合関数は、"SUM" (合計値)、"MIN" (最小値) などがある。レコード選択の条件式は述語要素を  $P_{ij}$  とすると、乗法標準形で次のように表される ( $1 \leq i \leq n$ ,  $1 \leq j \leq m_i$ )。ただし、論理和、論理積の演算子を、記号  $\vee$ ,  $\wedge$  で表す。

$\dots (P_{i1} \vee P_{i2} \vee \dots \vee P_{im_i}) \wedge \dots$   
 $\dots \wedge (P_{n1} \vee P_{n2} \vee \dots \vee P_{nm_n})$  (3.3)

$(P_{i1} \vee P_{i2} \vee \dots \vee P_{im_i})$  を、基本述語と呼び 7 種のカテゴリに分類している。この分類方法については、4.1 節で述べる。

\* 最上位の問合せブロックのレベルを 1 とする。

## 4. 大局的最適化

大局的最適化は、問合せの大局的な評価手順を、RDB/V1が提供する6種類の基本演算（縮小演算、等価相関演算<sup>\*</sup>、等価結合演算、直積演算、集合差演算、単純和演算）による最適な演算系列で表現することを目的としている。

問合せの評価は、次の2段階に分けて実施する。

- (1) 基本問合せブロックの評価
- (2) 問合せブロック内の集合演算の評価

(1)の評価は、基本述語を7種のカテゴリに分類し各カテゴリに対して、経験的な評価順序を与えることにより達成される。(2)の評価は、ユーザが与えた集合演算の演算順序<sup>\*\*</sup>に従って評価することで、達成されている。以上の評価順序に基づく問合せの評価方式を、カスケード法と呼び、4.2節で本方式による最適化アルゴリズムの詳細を述べる。RDB/V1特有の基本演算は、この評価順序に従って適用される。

### 4.1 基本述語のカテゴリ

基本述語は、以下のCAT1~CAT7のカテゴリに分類される。CAT2~CAT4およびCAT7の基本述語は論理和のない述語である。各カテゴリに属する基本述語の代表例を、付録1の図7に示す。

#### (CAT1) 単純制限述語

単一テーブルを制限する最も単純な形をした基本述語である。集合関数が存在するときは、問合せブロックの入れ子構造に相関がない<sup>\*\*\*</sup>という条件を満足しなければならない。

#### (CAT2) 単純制限集合述語

単一テーブルを制限するメンバ関係述語である。〈フィールド〉〈メンバ関係演算子〉SET (問合せブロック)が、このカテゴリに属する。ただし、メンバ関係演算子は、{IN/NOT IN}である。また、問合せブロックには、相関がない。

"SET (問合せブロック)"の評価結果は、多数のレコードから成るテーブルなのでCAT1の述語よりも弱い制限をもつ述語である。

#### (CAT3) 単純制限相関述語

単一テーブルを制限する入れ子の構造をもった述語で、述語の中に存在する問合せブロックと、現在着目

している問合せブロックの間には、相関<sup>\*</sup>がある。したがって本カテゴリは、CAT1およびCAT2に、この相関を付け加えた述語である。

#### (CAT4) 等価結合述語

本カテゴリからCAT6までは、二つ以上のテーブルに対する制限述語である。そのなかで、本カテゴリは、最も単純な形をした述語であり、二つのテーブルの等価結合を表す述語である。

#### (CAT5) 複合制限述語

二つ以上のテーブルを同時に制限する述語である。CAT4の述語よりも、複雑な形をした述語ではあるが相関をもつ問合せブロックの存在は許さない。CAT1~CAT3までの基本述語が論理和で結ばれた述語は、このカテゴリに属する。

#### (CAT6) 複合相関述語

前述までのカテゴリおよび、CAT7を除く基本述語である。この基本述語には、必ず相関をもった問合せブロックが存在する。

#### (CAT7) 等価相関述語

一段上位の問合せブロックのテーブルと等価結合する単一の述語だけが、このカテゴリに属する。詳細は、省略するが、この述語は、再帰的に評価されるという特徴をもつため、最後に評価される。それは、再帰的な評価を受けるごとに、次の述語を評価するための中間テーブルを作成しないほうがコスト安になるからである。

問合せの評価に際しては、テーブルに対し、より強い制限を与えるカテゴリの基本述語を優先的に評価したほうが処理効率がよいので、本カテゴリの分類の順序を、基本述語の評価順序としている。制限の強さは、述語の形の単純度としてとらえている。すなわち、単純な形をした述語ほど、解として得られるキー値の局在度が高いので、制限の程度が強いとしている。

### 4.2 カスケード法

本方式による最適化アルゴリズムを示す。

(1) CAT1~CAT3の基本述語を、順に縮小演算(射影および制限の機能をもつ)の適用により、評価する。

(2) CAT4の基本述語を、等価結合演算により評価する。CAT4の基本述語が複数存在するとき<sup>\*\*</sup>

\* 問合せブロック間が、お互いのフィールドで直接的に結合している場合、効率的に評価する演算。

\*\*  $(A \cap B) \cup C$  は、 $A \cap B$  を先に評価する。

\*\*\* たとえば、述語 "フィールド  $\geq$  SUM (問合せブロック)" については、述語内問合せブロックは、上位の問合せブロックのフィールドを参照しない。

\*  $a_1 \geq \text{SUM}(b_1 \text{ FROM } B \text{ WHERE } b_2 = a_2)$  がその例である。 $a_1, a_2$  は、現在着目している問合せブロックの検索テーブルを構成するフィールドであり、 $b_1, b_2$  は、述語中に存在するテーブルBのフィールドである。 $a_2$  は、( ) 内の問合せブロック内に存在する。

\*\* 複数の等価結合述語が論理積で結ばれていることを意味する。

は、同時に等価結合できるテーブル（クラスタ）をまとめ、クラスタ単位に評価する。クラスタを評価すると、中間テーブルが作成される。一般に、クラスタは、複数個あるので、等価結合結果のテーブルのレコード数が小さくなるものから順に評価される。

(3) まだ二つ以上の中間テーブルまたはベーステーブルが残っているときは、各テーブルに対し直積演算を適用する。その結果、一つの間接テーブルTが作成される。

(4) CAT5, CAT6 の基本述語を、中間テーブルTに対する縮小演算として評価する。

(5) CAT7 の基本述語を、等価相関演算で評価する。

(6) ユーザが与えた演算の順序に従い、各集合演算を、以下のように評価する。

- 集合積を、重複キーの除去と等価結合演算で評価する。ただし、等価結合キーは、演算対象テーブルの全フィールドである。

- 集合和を、単純和演算と重複キーの除去で評価する。

- 集合差を、集合差演算で評価する。

- 単純和を、単純和演算で評価する。

付録1の図8は、以上の簡易な最適化アルゴリズムにより作成された問合せの処理戦略である。

本システムでは、さらに、演算対象テーブルにインデックスが存在するときは、インデックスを使用する場合としない場合のアクセスコスト<sup>\*</sup>を比較し、基本述語の評価順序を一部変更することも行っている。

## 5. 局所的最適化

本章では、本システム特有の基本演算の実現方式とそのなかで最も特徴的な最適化技法としての最大最小法について述べる。最大最小法は、等価結合演算（または等価相関演算）に関する最適化技法であり、推移述語が利用されている（以下、等価相関演算に関する記述を省略する）。推移述語は、等価結合述語の性質から定義される述語である。そのため、推移述語は、等価結合演算の準備段階における縮小演算および、等価結合演算でテーブルを制限する条件式としても利用される。これにより、不要なレコードの早期除去を図ることができる。一方、最大最小法は、この種の最適化のみならず、スキャン範囲の局所化という効果も期待できる。

\* この算出方法は、6章を参照されたい。

## 5.1 基本演算の実現方法

6種の基本演算は、お互いに独立性が高く、いかなる問合せもこれだけで十分に評価できる。また、これらの基本演算には、すべてソート機能およびテーブルの縮小機能（射影と制限）と同時に、基本述語を評価する機能がある。さらに、等価結合演算、直積演算、集合差演算、単純和演算は、すべて多項演算の処理機能をもっているため、 $n$ 個のテーブルがある演算項になっているときは、二項演算を $n-1$ 回繰り返さずともよく、1回の演算で評価される。

### 5.1.1 縮小演算

演算対象テーブルから縮小条件<sup>\*</sup>を満足するレコードを探し、必要なフィールド値を演算結果のテーブルに納めることにより、評価が達成される。

CAT1の基本述語は、次のいずれかの評価方式で、評価される。

(1) ソートテーブルまたはインデックスに対するスキャンの開始点および終了点に変換する。

(2) スキャンアーギュメントに変換する。

(3) ISRQで、レコード選択用述語として評価する。

(1)および(2)の形で評価される基本述語は、DSCS内で評価されるので、ISRQへレコードを転送し評価するという無駄を省くことができる。

CAT2は、CAT3の形の述語に等価変換<sup>\*\*</sup>してから評価されている。CAT3は、(3)の扱いで評価される。後述するが、CAT3の述語を評価するときは、縮小演算と対象テーブルは、キーフィールドにインデックスがなければ、そのテーブルは、ソートされていなければならない。

### 5.1.2 等価相関演算

CAT7の等価相関述語 (" $b=a$ ") を評価する演算である。フィールド $b$ は、現在処理対象としている問合せブロックのフィールドであり、フィールド $a$ は、一段上位の問合せブロックのフィールドである。評価方法には、ネステッドループ法とソートマージ法<sup>(5), (9)</sup>の2種類があるが、平均的にソートマージ法が優れていると考え、この方式を採用している。演算対象テーブルは、フィールド $b$ のインデックスをもつか、フィールド $b$ でソートされている。評価は、 $b$ が $a$ に等しいレコードをテーブルから取り出すことによって、達成

\* 一般に、多くの基本述語から成る。

\*\* 基本問合せブロック内の検索テーブルをAとし、Aのフィールドを $a$ とすると、CAT3の基本述語  $a \text{ IN SET } (b \text{ FROM } B)$  は、 $\text{EMPTY} * \text{SET } (b \text{ FROM } B \text{ WHERE } b=a)$  に変換される。キーフィールドは、 $a$ である。

される。

5.1.3 等価結合演算

本演算も、ソートマージ法を採用している。本システムではこの方式をさらに強化するために、後述の最大最小法を実現した。これにより、演算対象テーブルのスキャン範囲を大幅に局所化できる。

5.1.4 直積演算

本演算対象の  $n$  個のテーブルを、 $R_1, R_2, \dots, R_n$  とする。各テーブル  $R_i$  のページ数、レコード数を、 $M_i, N_i$  とし、各テーブルを制限する基本述語の制限係数\* (Restriction Factor) を、 $f_i$  とする ( $1 \leq i \leq n$ )。直積演算の効率的な評価順序は評価関数  $\xi_i = M_i / (f_i \times M_i - 1)$  の値の降順に  $R_i$  を並べることにより得られる<sup>15)</sup>。これは制限の強い基本述語をもつテーブルから優先的に直積演算を行うことを意味する。本システムで採用した制限係数  $f_i$  の値を表2に示す。

5.1.5 集合差演算

$n$  個のテーブル  $R_1, R_2, \dots, R_n$  に対する集合差演算  $R_1 - R_2 - \dots - R_n$  の結果  $T$  は、 $R_1$  の部分集合である。 $T$  は、重複レコードの存在を許さないので事前に、 $R_1$  の重複レコードは除去される。

集合差演算でも等価結合演算と同様、演算対象テーブルは、ソートされていることが前提となる。評価の基本方針はテーブル  $R_2, \dots, R_n$  に属するどのレコードにも一致しないレコードを、 $R_1$  から取り出すアルゴリズムなので、等価結合演算とは逆のレコード選択処理を行う。

5.1.6 単純和演算

本演算対象の  $n$  個のテーブルに対し、全レコードをよせ集めることで実現している。

5.2 推移述語

推移述語は、1 クラスタ内の等価結合テーブル  $R_1, R_2, \dots, R_n$  に対して、どのテーブルを制限する述語としても推移的に利用できる述語である。この推移性は、各テーブルの等価結合フィールド  $r_1, r_2, \dots, r_n$  が、お互いに等しい ( $r_1 = r_2 = \dots = r_n$ ) という関係から導出される\*\*\*。

各テーブル  $R_i$  から推移性をもつ基本述語を選択しそれを、 $P_{i1}(r_i), P_{i2}(r_i), \dots, P_{in_i}(r_i)$  とすると、推移述

\*  $N_i, N_i'$  をおのおの制限前後のレコード数とすると、 $f_i = N_i' / N_i$  で定義されるが、実際にテーブルを制限してみないと  $N_i'$ 、すなわち  $f_i$  が定まらない。そこで、RDB/V1 では、 $f_i$  として、表2に示す経験的な値を与えている。

\*\*  $(\dots(R_1 - R_2) - \dots) - R_n$  を指す。

\*\*\* 等価結合フィールド  $r_i$  だけの関数として表される述語  $P(r_i)$  は、テーブル  $R_i$  の制限述語であると同時に、他のテーブル  $R_j$  の制限述語  $P(r_j)$  としても利用される ( $1 \leq i \leq n, 1 \leq j \leq n, i \neq j$ )。

語  $TR(x)$  は、次のように定義される。

$$TR(x) = TR_1(x) \wedge TR_2(x) \wedge \dots \wedge TR_n(x)$$

$$TR_i(x) = P_{i1}(x) \wedge P_{i2}(x) \wedge \dots \wedge P_{in_i}(x)$$

ただし、 $x$  は、任意の等価結合フィールドを表す。すなわち、 $x = r_i$  とすると、 $TR(r_i)$  は  $R_i$  を制限する条件式として利用される ( $1 \leq i \leq n$ )。

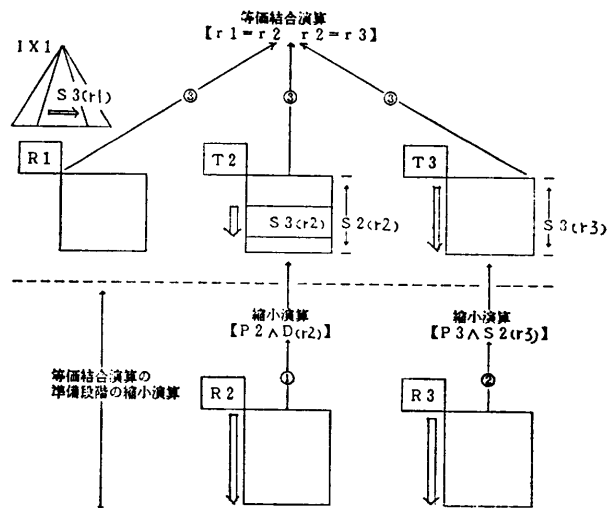
5.3 最大最小法

等価結合に関連するすべてのベーステーブルのインデックスおよびソート済みの中間テーブルから、等価結合フィールドの範囲を求め、推移述語から取り出される範囲との共通部分を計算する。このとき、共通部分の最小値および最大値の組を  $D = (C1, C2)$  とする。

最大最小法のアルゴリズムは、次のように示される。

(第一段階)

最小値および最大値の組  $D$  を検出し、それを等価結合演算における準備段階の最初の縮小演算に適用する。このとき、 $D$  は、等価結合フィールドの区間を制限する条件として利用される。2 番目以降の縮小演算に対しては、直前の縮小演算で作成されたソートテーブルから取り出される最小値および最大値を適用する。



$R_1, R_2, R_3$  は、ベーステーブルであり、 $T_2, T_3$  は、ソートされた中間テーブルである。 $R_1$  には、等価結合フィールドのインデックス  $IX1$  が存在する。 $r_1, r_2, r_3$  は、おのおの、 $R_1, R_2, R_3$  の等価結合フィールドである。演算の評価は、①→②→③の順に行われる。 $[ ]$ 内は、評価される述語である。 $S_2(r_2), S_3(r_3)$  は、おのおの、 $R_2, R_3$  の縮小演算結果から検出された最大最小値である。また、 $D(r_2)$  は、インデックス  $IX1$  および述語  $P_2, P_3$  の最大最小値により作成される述語である。

図2 等価結合演算の例

Fig. 2 An example of equi-join operation.

(第二段階)

等価結合演算の準備段階において、縮小演算が一つでもあるときは、最後の縮小演算結果から取り出される等価結合フィールドの最小値および最大値の組  $S$  を、等価結合演算対象のテーブルのスキャン範囲として利用する。このスキャン範囲は、インデックスまたはソート済みの中間テーブルに対するスキャン範囲である。

等価結合演算の準備段階としての縮小演算が一つもないときは、第一段階で求められた最小値および最大値の組  $D$  を、上記  $S$  の代わりに利用する。

以上の二段階の処理によって、一連の縮小演算および等価結合演算において、レコードの無駄な参照が回避され、処理スピードの高速化を図ることができる。

図 2 に、等価結合演算の評価例を示す。

一方、本方式は、等価相関演算にも適用される。それは、等価相関演算の基本的な評価方式が、ソートマージ法であることにほかならない。

6. アクセスパスの選択

5.1 節の基本演算の評価に際し、どのようなアクセスパスを使うのかを決定しなければならない。本章ではこの問題の基本となる縮小演算について述べる。アクセスパスの選択方式は、評価すべき条件式に対してインデックスの有無を調べるだけでなく、アクセスコストの計算に基づいている。以下の説明では、(3.3) 式の条件式を利用する。

アクセスパスは、アクセスコストが最小の基本述語  $P_i$  から選択される<sup>10)</sup>。もし、 $P_i$  の評価のためのアクセスコスト  $C_i$  が、テーブルスキャンのコスト  $M$  (テーブルの全ページ数) 以上のときは、アクセスパスとして、テーブルスキャンを選択する。そうでないとき ( $C_i < M$ ) は、各述語  $P_{ij}$  から定められた  $m_i$  個のインデックススキャンがアクセスパスになる。アクセスコストは、 $C_i = C_{i1} + C_{i2} + \dots + C_{im_i}$  である。 $C_{ij}$  は、 $P_{ij}$  を評価するときのアクセスコストである。付録 2 に、 $P_{ij}$  を評価するときのアクセスコストの計算法<sup>19)</sup>を示す。System R では、I/O コストと CPU コストをともに導入した計算を行っている<sup>2)</sup>が、RDB/V 1 では、I/O コストだけにとどめている。

7. 測定結果

本章では、等価結合演算を中心に実現された最大最小法の効果を、大型計算機 FACOM-M 200 で測定し

製品テーブル (製品番号, 品名, 単価, ...)  
 売上テーブル (製品番号, 数量, 金額, ...)  
 「単価が、20万円以上で、製品番号が49990~50010の  
 売上製品について、売上の品名と単価を、検索せよ。」

```
GET      S, 製品番号, S:品名, U, 単価
FROM    S:製品テーブル, U:売上テーブル
WHERE   S, 製品番号=U, 製品番号
AND     S, 単価 >=200000
AND     49990 <=S, 製品番号 <=50010 ;
```

U, Sは、おのおの製品テーブル, 売上テーブルの修飾子である。また、両テーブルの製品番号には、インデックスが、作成されている。

図 3 測定の間合せ例 (条件 1 の間合せ)

Fig. 3 An example of experimental query.

たので、その結果について述べる。図 3 は、測定のための間合せ例である。例で、 $\square$  を、売上げテーブルの製品番号を制限する述語に置き換えても、同じ意味をもつ。置き換える前/後を、条件 1 / 条件 2 の間合せとして区別する。製品テーブルのレコード数を 3,920 件とし、売上げテーブルのレコード数を、2,000 から 30,000 件まで変化させ、CPU 処理時間を測定した。

図 4 に、測定結果を示す。検案件数は、常に一件である。CPU 処理時間は、最大最小法の適用による測定点のなかで、売上げテーブルのレコード件数が 2,000 件の場合の CPU 処理時間で規格化してある。

最大最小法の適用により、

- (1) 高速な間合せ処理
- (2) 条件 1 と条件 2 のような間合せの形式によらない安定な処理スピード
- (3) テーブルのレコード件数に依存しない平滑な処理スピード

が達成されている。

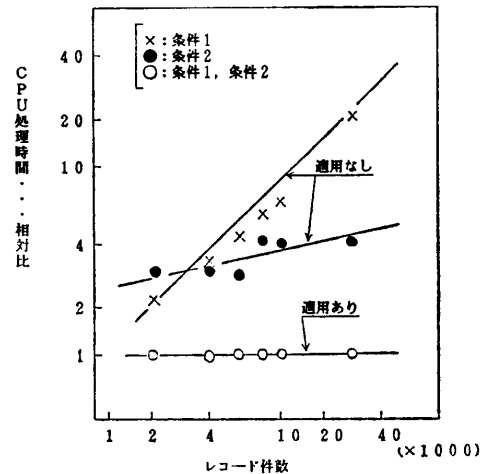


図 4 最大最小法の適用効果

Fig. 4 Effects of the Min-Max method.

## 8. 結 論

本論文では、RDB/V1の問合せ評価に関する大局的最適化と局所的最適化について述べた。とくにカスケード法による大局的な最適化アルゴリズムの簡易性および、推移述語の定義に基づく最大最小法の実現による等価結合演算（または等価相関演算）の高速化について、詳細に述べた。他のシステムでは、このような方式による最適化技法が採用されていないので、本システムの特徴といえる。

現在、以下の課題に取り組んでいる。

(1) 等価結合演算または等価相関演算に関する処理の途中で、明らかにソートマージ法よりもネステッドループ法による評価方式が有利であることがわかったときは、ネステッドループ法に切り替える。

(2) 問合せの評価のための基本演算の系列に対し基本演算間の有機的な結合を図る。たとえば、等価結合演算と等価相関演算が連続的に行われる場合は、個々に適用されている最大最小法を、一つにまとめて拡張する\*。

(3) 同じテーブルが問合せのなかに複数個存在するときは、できるだけ少ないテーブルアクセスの回数で評価を終了させる。

さて、RDB/V1は、対話型計画管理情報システム PLANNER (Planning and Management Information System based on Easy RDB)<sup>9),14)</sup>の核として商品化されている。その適用分野は、人事管理、部品管理、図書管理、地域情報管理、大学・研究所等の実験データの管理などの幅広い分野に渡っておりさまざまなユーザ（エンドユーザおよびアプリケーションプログラマ）から本システムに対する高い評価を得ている。

謝辞 RDB/V1の積極的な実用化に寄与した経営システム開発部 PLANNERグループの諸氏と種々のご指導・助言をいただいたソフトウェア研究部長林達也氏に感謝の意を表します。

## 参 考 文 献

- 1) Astrahan, M. M. et al.: System R: Relational Approach to Database Management, *ACM TODS*, Vol. 1, No. 2, pp. 487-491 (1976).
- 2) Astrahan, M. M. et al.: Performance of the System R Access Path Selection Mechanism,

Proc. of IFIP Congress 80, Tokyo, Japan, pp. 487-491 (1980).

- 3) Codd, E. F.: Extending the Database Relational Model to Capture More Meaning, *ACM TODS*, Vol. 4, No. 4, pp. 397-434 (1979).
- 4) 古川: 関係データベースに対するデータアクセスの数式処理による最適化について, 情報処理学会論文誌, Vol. 22, No. 1, pp. 68-75 (1981).
- 5) Haerder, T.: A Scan-Driven Sort Facility for a Relational Database System, The 3rd International Conference on VLDB, pp. 236-243 (1977).
- 6) 神田他: RDBをベースとした計画管理情報システム PLANNER, 情報処理学会データベースシステム研究会資料, 32-3 (1982).
- 7) 北上, 手塚, 石川, 安達, 牧之内: RDB/V1に於ける問い合わせ評価の最適化について, 情報処理学会第23回全国大会, pp. 473-474 (1981).
- 8) Kim, W.: Relational Database Systems, *ACM Comput. Surv.* Vol. 11, No. 3, pp. 185-211 (1979).
- 9) King, W. F.: Relational Database Systems: Where We Stand Today, Proc. of IFIP Congress 80, Tokyo, Japan, pp. 369-381 (1980).
- 10) 牧之内, 手塚, 北上, 安達: RDB/V1のアクセスパス選択方式, 情報処理学会第20回全国大会, pp. 741-742 (1979).
- 11) 牧之内, 手塚, 北上, 佐藤, 泉田, 安達, 中田, 石川: リレーショナルデータベース管理システム RDB/V1, 情報処理学会データベース管理システム研究会資料, 19-3 (1980).
- 12) Makinouchi, A., Tezuka, M., Kitakami, H. and Adachi, S.: On the Optimization Strategy in RDB/V1, The 5th International Conference on VLDB, pp. 518-529 (1981).
- 13) 牧之内, 手塚, 北上, 安達, 佐藤, 泉田, 中田, 石川: 関係データベース管理システム RDB/V1, 情報処理学会論文誌, Vol. 24, No. 1, pp. 47-55 (1983).
- 14) 牧之内他: 関係データベースシステムを中核とした計画管理情報システム, 情報処理学会論文誌投稿中.
- 15) Pecherer, R. M.: Efficient Exploration of Product Spaces, SIGMOD Conference, pp. 44-49 (1976).
- 16) Rothnie, J. B.: Evaluating Inter-entry Retrieval Expressions in a Relational Database Management System, Proc. AFIPS, Vol. 44, Montvale, N. J., pp. 417-423 (1975).
- 17) Smith, J. M. and Chang, P. T. Y.: Optimizing the Performance of a Relational Algebra Database Interface, *CACM*, Vol. 18, No. 10, pp. 568-579 (1975).
- 18) Strand, A. J.: The Relational Approach to

\* 等価結合フィールドと等価相関フィールドが一致しているときに可能である。

Management of Databases, Proc. IFIP, pp. 901-904 (1971).

- 19) 手塚, 北上, 安達, 牧之内: RDB/V1 のアクセスコスト, 情報処理学会第20回全国大会, pp. 739-740 (1979).
- 20) Ullman, J. D.: *Principles of Database Systems*, Computer Science Press, Potomac, Md. (1979).
- 21) Whitney, V. K. M.: *Relational Data Management Implementation Techniques*, ACM SIGMOD Workshop on Data Description, Access and Control, pp. 321-348 (1974).
- 22) Wong, E. and Youssefi, K.: *Decomposition—A Strategy for Query Processing*, *ACM TODS*, Vol. 1, No. 3, pp. 223-241 (1976).
- 23) Yao, S. B.: *Optimization of Query Evaluation*

- (1) EMP (従業員テーブル)
  - ENO 従業員番号
  - YMD 従業員番号発行年月日
  - ENAME 氏名
  - SEX 性別
  - BIRTH 生年月日
- (2) DEPT (所属テーブル)
  - ENO 従業員番号
  - YMD 所属変更年月日
  - DNAME 所属変更年月日
- (3) OFAN (発令テーブル)
  - ENO 従業員番号
  - YMD 発令年月日
  - OANAME 発令名所
  - YMDEND 発令終了年月日

図5 テーブル構成

Fig. 5 Examples of logical table structure.

【例1】 「1979年10月01日までに、働いたことがある職員の氏名と所属を、検索する。」

```

GET  E·ENO, E·ENAME, D·DNAME
FROM  E:EMP, D:DEPT
WHERE D·YMD<=19791001
AND  E·ENO =D·ENO
AND  D·YMD =MAX ( YMD
                FROM  DEPT
                WHERE  YMD<=19791001
                AND   ENO =D·ENO );
    
```

【例2】 「1979年10月01日現在の休職者の従業員番号を、検索する。」

```

GET  ENO
FROM  X:OFAN
WHERE YMD<=19791001
AND  OANAME='休職'
AND  EMPTY =SET ( Y·ENO
                FROM  Y:OFAN, Z:OFAN
                WHERE  Y·YMD<=19791001
                AND   Z·YMD<=19791001
                AND   Y·OANAME='休職'
                AND   Z·OANAME='復帰復職'
                AND   Z·YMD<Y·YMDEND
                AND   Y·ENO=Z·ENO
                AND   Y·ENO=X·ENO );
    
```

図6 問合せの例

Fig. 6 An example of query.

Algorithms, *ACM TODS*, Vol. 4, No. 2, pp. 133-155 (1979).

付録1 問合せの例

ここでは、基本述語のカテゴリ分け、およびそれに基づく問合せの評価手順について、具体例を示す。

図5に、例で使用されるテーブル構成を示す。従業員の人事移動などによるテーブルのメンテナンスは、下記に基づいて実施されるとする。

(従業員テーブル)

レコードの挿入は、新しく従業員番号が発行される度に行われる。

(所属テーブル)

レコードの挿入は、入社による配属または所属変更の度に行われる。

(発令テーブル)

レコードの挿入は、休職、減給、復帰復職、退職、臨時任用がある度に行われる。

図6に、二つの問い合わせの例を示す。ただし、E·

【例1】

```

Q1 ::= E·ENO, E·ENAME, D·DNAME
      FROM  E:EMP, D:DEPT
      WHERE D·YMD<=19791001 --- (CAT1)
      AND  E·ENO =D·ENO --- (CAT4)
      AND  D·YMD =MAX (Q2) --- (CAT3)
Q2 ::= YMD
      FROM  DEPT
      WHERE YMD<=19791001 --- (CAT1)
      AND  ENO =D·ENO --- (CAT7)
    
```

【例2】

```

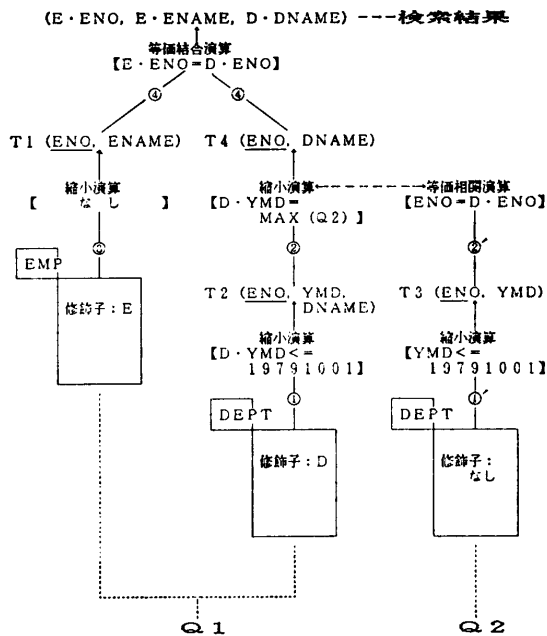
Q1 ::= ENO
      FROM  X:OFAN
      WHERE YMD<=19791001 --- (CAT1)
      AND  OANAME='休職' --- (CAT1)
      AND  EMPTY=SET·(Q2) --- (CAT3)
Q2 ::= Y·ENO
      FROM  Y:OFAN, Z:OFAN
      WHERE Y·YMD<=19791001 --- (CAT1)
      AND  Z·YMD<=19791001 --- (CAT1)
      AND  Y·OANAME='休職' --- (CAT1)
      AND  Z·OANAME='復帰復職' --- (CAT1)
      AND  Z·YMD < Y·YMDEND --- (CAT5)
      AND  Y·ENO =Z·ENO --- (CAT4)
      AND  Y·ENO =X·ENO --- (CAT7)
    
```

図7 基本述語のカテゴリ

Q1/Q2は、おのおの、レベル1/2の基本問合せブロックである。

Fig. 7 The category of primary predicate.





T1 から T4 までの中間テーブルに対して、下線部のフィールドは、ソートテーブルのキーであることをさす。【 】内は、演算で評価される基本述語である。①→②→③→④は、基本問合せブロック Q1 の評価順序であり、①'→②' は、基本問合せブロック Q2 の評価順序である。② と ②' は、同時に評価される。

図 8 問合せ (例 1) の処理戦略

Fig. 8 An evaluation strategy for query (example 1).

ENO は、テーブル EMP のフィールド ENO である。その修飾子 E は、E: EMP で定義されている。

基本述語のカテゴリ例は、図 7 に示されている。また、問合せの処理手順は、図 8 に示されている。これは、図 6 の例 1 に対する処理戦略図である。

付録 2 アクセスコストの計算法

アクセスコストを、ページアクセス回数と定義し、その計算法を示す。アクセスコストの計算に必要なコストパラメータおよび述語の制限係数を表 1 および表 2 に示す。

- (1) レコード到着順テーブル  
M
- (2) キー順テーブル  
 $H-1+f \times L$

表 1 データ編成法に基づくコストパラメータ

Table 1 Cost parameter of the data organization method.

テーブルまたはインデックス	データ編成法	コストパラメータ
テーブル	レコード到着順	<ul style="list-style-type: none"> <li>• ページ数 (M)</li> <li>• レコード数 (N)</li> </ul>
	キー順	<ul style="list-style-type: none"> <li>• ページ数 (M)</li> <li>• レコード数 (N)</li> <li>• インデックスの高さ (H)</li> <li>• キーの種類 (K)</li> <li>• キーの最大最小値 (MAX, MIN)</li> <li>• リーフページ数 (L)</li> </ul>
	ハッシュ	<ul style="list-style-type: none"> <li>• ハッシュエントリ数 (E)</li> <li>• レコード数 (N)</li> </ul>
インデックス	キー順*	<ul style="list-style-type: none"> <li>• インデックスの高さ (H)</li> <li>• キーの種類 (K)</li> <li>• キーの最大最小値 (MAX, MIN)</li> <li>• リーフページ数 (L)</li> </ul>

\*印は、レコード到着順のテーブル上で作成されていることを示す。

表 2 検索述語 (P) の制限係数 (f)

Table 2 Restriction factor (f) of the predicate.

述語 (P) のタイプ	比較演算子	制限係数 (f)
フィールド比較式	=	0.1, [1/K]
	≠	1.0
	<または<=	0.5, $\frac{定数-MIN}{MAX-MIN}$
	>または>=	0.5, $\frac{MAX-定数}{MAX-MIN}$
フィールドの区間 (定数 1 から定数 2 まで)	すべての不等号の組	0.25, $\frac{定数1-定数2}{MAX-MIN}$
メン係式	IN	0.5
	¬ IN	1.0
集合関係式	=	0.01
	≠	1.0
	CONT	0.1
	¬ CONT	1.0

[ ] は、フィールドが、インデックスキーになっているときの制限係数である。また、CONT は、CONTAINS の略記号である。その他、“フィールド 1 比較演算子 フィールド 2” の制限係数 f を、0.5 とし、等価結合述語の f は、0.1, [1/K] としている。

- (3) ハッシュテーブル
  - 1 (“フィールド=定数” の述語)
- (4) レコード到着順テーブル上のインデックス  
 $H-1+f \times (L+N)$

(昭和 57 年 6 月 15 日受付)  
(昭和 57 年 11 月 8 日採録)