データベース・システム 30－2
（1982. 5. 20）

柔構造関係データベース

大保信夫，益田隆司（筑波大・電子情報工学系）
木村裕，細井拓史（筑波大・理工学研究科）

Ⅰ．はじめに

データベース技術は、古くから帳簿という形式でほぼ定型化されたデータをその対象とし、ある程度確立された処理手順を持つ事務管理部門にその応用分野を限定することにより、理論の確立が図られ、実用システムとしての地位を確保してきた。しかし、近年、設計・開発・診断・意志決定等の支援システムとしてその適用範囲を拡げるにつれ、従来の枠組内では、データ表現・処理に強い制約が課せられることが明らかになってきた。特に重要な点は以下である。

1）新しい応用分野に於ける対象世界の認識の有り方が、事務管理部門のそれと大きく異なる場合が多く、事務処理を中心対象として意識したデータモデルで対象世界を記述する場合、対象世界の認識を人為的に歪めた形式で行わざるを得なくなる。その結果、ユーザから見たデータベースの意味構造が極めて理解し難いものになりがちである。

2）新しい応用分野は、非定型的・発見的仕事の遂行を目的とする場合が多く、試行錯誤をその基本的手法としている。そのため、少くとも以下に述べる二つの動的要素を支援することが可能でなければならない。第一は、データ処理過程に於ける航行が、それまでの結果に基づいて決定される動的要素であり、第二は、外部世界の論理認識の変化に伴い、スキーマ情報の一部に生ずる動的要素である。

上記二つの制約を緩和したデータベース・システムは、事務用データのみならず、新しい応用分野の世界を、その分野のユーザの直観的認識に忠実に表現可能であるだけの表現能力と処理能力を持ち、かつ上記(2)で示された二つの動的要素を支援する能力が必要である。しかしこの要求は明らかに実行効率に対する負要素として働く。新しい分野をその適用範囲として捕えるデータベース技術の研究は、少くとも近い将来に於ける実用システムを考える場合、この表現能力・処理能力と実行効率のトレードオフをどの線に引くかが、その中心課題の一つとなるであろう。

本稿では、現在筑波大学電子情報工学系で研究開発中のデータベース管理システムTIMEの基本概念について述べる。本システムは、関係データベースにその基礎を置くが、ある程度の実行効率の下で、上記(1)(2)の制限を緩和することを目的としたシステムである。

Ⅱ．データ・モデル

Ⅱ．1．現行の関係モデルの問題とその拡張方向

ここでは、関係モデルを新しい応用分野に適用する場合の問題点と、その解決の方向を小さな例で示す。例としては、簡単な二次元図形を対象としたグラフィック・データベースを取りあげる。このグラフィック・システムでは、原始エンティティ・タイプとして、三角形，四角形，円等が存在し、これらの原始図形を基にして目的図形が生成されるものとする。また、一度生成された図形は、その後、他の図形の構成要素として利用可能とする。三角形，四角形，円等のグラフ

ィック・エンティティ・タイプを各々一つのリレーションで表現するのが自然であろう(図1.a)。ここで家というタイプの図形を生成するものとする(図1.b, 図1.c)。ここで一番目の家は、三角形と四角形から成り、二番目の家は四つの四角形から成っている。ところで この構成関係は、いかにして表現したらよいであろう。問題は、家の各エンティティ毎に意味構造が異っていることである。充分考えられる工夫としては、図1.dに示されるように、対象というリレーション及び、対象中の各エンティティと、各図形タイプのエンティティの間に、いわゆるgeneraligationの関係を表わすリレーションを用意することである。

三角形

三角形番号	X1	Y1	X2	Y2	X3	Y3
1	0	5	5	10	10	5
2	0	0	3	3	4	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

四角形

四角形番号	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1	0	0	0	5	10	5	10	0
2	10	0	10	5	15	5	15	0
3	10	5	10	9	15	9	15	5
4	12	2	12	4	14	4	14	2
5	12	6	12	7	14	7	14	6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

円

円番号	中心X	中心Y	半径
1	0	0	2
⋮	⋮	⋮	⋮

⋮

図1.a グラフィック・エンティティ

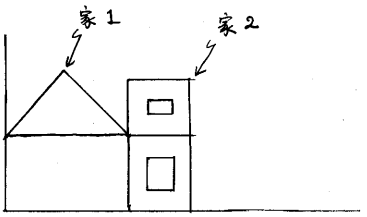


図1.b 図形 家

家

家番号	階数
1	1
2	2
⋮	⋮

図1.c 合成図形

この表現の問題点を次に述べる。

1.) エンティティを表現する各リレーションの属性として、リレーション中で一意の通し番号が用いられている。これは、関係モデルを基にしたデータベースの設計では、ユーザ制御キー属性が不可欠な要素として要求されるからである。事務用データの多くは、ユーザ制御のキー値が現実世界でそれなりの意味を有しているが、この例のような場合、単なるリンク・フィールドとしての役割しか果さず、現実世界では殆ど意味を持たない値である。データベースの対象を事務用データ以上に拡げた場合、キー属性が自然に決定されないと考えるべきである。この場合システム制御のキー(内部名としてのタプル識別子)をリンク・フィールドとして用いることを許すように

対象

対象番号	名前	番号
1	三角形	1
2	三角形	2
3	四角形	1
4	四角形	2
5	四角形	3
6	四角形	4
7	四角形	5
8	家	1
9	家	2
⋮	⋮	⋮

構成

上位対象	下位対象
8	1
8	3
9	4
9	5
9	6
9	7
⋮	⋮

図1.d generaligationの関係及び構成関係

することが自然な拡張であろう。

2) 意味構造の表現のために，リレーション名がカラム値として現われる。Codd[1]やSmith, Smith[2,3]の意味構造モデリングに於て，既にリレーション名をカラム値として許すモデルが示唆あるいは提案されている。これらの提案は，いずれも，aggregationとgeneralizationという考え方に基づくデータの意味構造を表現することを目的としている。この意味構造の表現能力も，データ・モデルの記述能力として重要であるが，前述の例に於て，リレーション名が登場せざるを得なかったのは，別の見方から見ることが出来る。外部世界でエンティティ間の関係を表現するカテゴリ名を考えると，関係モデルに於ける通常のデータベース設計理論が仮定するような，固定したエンティティ・セット間の並びを表わす場合は少なく，むしろその関係に関連するエンティティ・セットが固定されない場合が多い。例えば「AがBを構成する」，「AがBの左側にある」といった関係を考えると，AとBの属するエンティティ・セットを固定することが，かなり不自然であることが判るであろう。関係モデルに基づくデータベースでは，エンティティ間の関係は，対応する各タプルの属するリレーションのキー値を並べて表現することが自然であることは良く知られている。前述の例では，構成関係に関与するエンティティ・セットを固定する目的で，対象という多少人為的リレーションが作られている。データベース中にこのような関係が数多く存在してくると，データベースの意味構造が極めて不透明なものとなる。前述の構成関係は，図2で示されるリレーションで表現することが自然であろう。この表現では，各エンティティは，(リレーション名，通し番号)対で参照されている。前記1)の議論から，ここで通し番号は，システム制御キー値として実現さるべきである。

3) リレーション名をカラム値として許すことにより，generalizationやaggregationといった普遍的関係の記述が可能であるだけでなく，エンティティ間の関係を極めて柔軟に表現出来ることを示した。しかし関係データベースの枠内では，リレーション名に基づく航行の制御は，ユーザの完全な制御下にある。カラム値としてリレーション名を許すことは，データベースの意味構造をユーザに把え易くすることは出来ても，遂一ユーザがデータベース中のデータ値を確かめ判断しなければ航行の方向を決定出来ない。もし，カラム値がリレーション名であることがシステムに判れば，航行の方向をシステムが制御することが可能になるであろう。このことは，逆に言うと，動的に変化する航行の方向をデータベースやのデータ値として表現・処理可能であることを意味する。

Ⅱ．2 TIMEシステムに於けるデータモデルの定義。

前節に於ける関係モデルの拡張方向は，ARCと呼ぶデータモデルで実現される。本モデルは，ムロ等の，PICCOLO[4]，ELF[5]と本質的に同じである。

ARCの定義

ARC中のリレーションRは以下で定義される。

構成	合成物 リレーション名	合成物 番号	構成要素 リレーション名	構成要素 番号
	家	1	三角形	1
	家	1	四角形	1
	家	2	四角形	2
	家	2	四角形	3
	家	2	四角形	4
	家	2	四角形	5

図2 構成関係の表現例

$R \subseteq T \times T_1 \times R_1 \times T_2 \times R_2 \times \cdots \times T_k \times R_k \times D_1 \times \cdots \times D_m$

ここで

T, T_i はタプル識別子定義域

R_i はリレーション名の集合

D_i は単純定義域

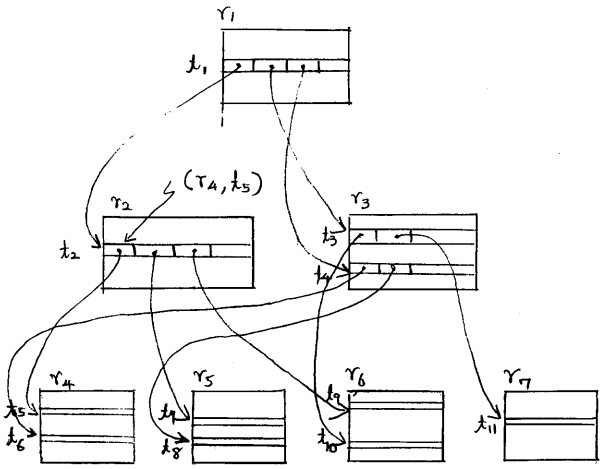
である。

リレーションの集合は、$m+1$ 個のクラス $C_0, C_1, \cdots, C_m$ に分かれる。$k=0$ のリレーションは C_0 に属する。C_i 中のリレーション R に関与するリレーション名 R_j は全て C_j $(i-1 \leq j \leq 0)$ に属さなければならない。

$(t, t_1, \gamma_1, \cdots, t_R, \gamma_R, d_1, \cdots d_m) \in R$ のとき、t はこのタプルに与えられるタプル識別子である。このタプル識別子はユーザ制御のキーとしての役割を持つ。(t_i, γ_i) は対で用いられ、他の関係 γ_i 中のタプル t_i を参照する。d_i はタプルの持つ属性値である。タプル間の k- 項関係は (t_1, γ_1), (t_2, γ_2), , (t_R, γ_R) で陽に表現可能である。しかしながら、ユーザ制御キー値による表現を防げるものではない。タプル識別子は、1つのリレーション中のみで一意であればよい。各タプル間の参照関係の例を図3に示した。このような参照関係は、設計、開発、診断、意志決定等基本的事象を含める データ間の階層構造性を素直に表現出来る。また、関係間の関係、更にその間の関係といった参照関係の表現を可能とする。

II. 3. データ操作

ARCの操作を行うリレーショナル・カリキュラスに関しては、参考文献(5)を参照されたい。そこでは、C_0, C_1 のみの場合に関する Codd のリレーショナル・カリキュラス[6]の拡張が述べられているが、C_m $(m \geq 1)$ の場合への拡張は、殆ど直接的に行えるので、ここでは操作の例を示すことにする。対象とするデータベースの例は、図1. a, 図1. c, 図2, 図4で示されたリレーションの集まりとする。図2のリレーションにはタプル識別子定義域を付加するものとする。この例では、リレーション 三角形, 四角形, 円, 家が C_0 のクラスに属し、構



C_2 クラス リレーション

C_1 クラス リレーション

C_0 クラス リレーション

図3. リレーションの参照関係

成，左がC_1のクラスに属する。操作言語はQBE[17]的言語とする。これはTIMEシステムで採用している言語である。

質問1

「座標値（0,5），（5,10），（10，5）を頂点とする三角形を構成要素とする家の階数を出力せよ」

この質問は以下の様に表現される。

三角形	タプル識別子	X1	Y1	X2	Y2	X3	Y3
	1	0	5	5	10	10	5

家	タプル識別子	階数
	2	P.

構成	合成物リレーション名	合成物番号	構成要素リレーション名	構成要素番号
	家	2	三角形	1

ここで下線で示されたのが例示要素である。カラムは，visibleカラムと，hiddenカラムに分れている。タプル識別子定義域に対応するカラムは，invisibleであり，リンク用のカラムとしてしか使用を許さないため，ここには例示要素しか書くことが許されない。その他のカラムは全てvisibleであり，原則として全ての要素を書くことが出来る。

質問2

「階数2の家の左にある四形の構成要素の位置座標を全て出力せよ」

この質問は以下の様に表現される。

家	タプル識別子	階数
	1	2

左	左四形リレーション名	左四形番号	左四形リレーション名	右四形番号
	木	2	家	1

構成	合成物リレーション名	合成物番号	構成要素リレーション名	構成要素番号
	木	2	円	3

円	タプル識別子
P.	3

この例は，リレーション名を例示要素として使用するものである。TIMEの操作言語では，同じ例示要素が，テーブル名の位置及び他のリレーションのカラム値として取ることを許す。この結果上の例で言うと，構成要素の各タプル毎に，その構成要素リレーション名フィールドの値に従って決定されるリレーションを航行することになる。注意すべきことは，リレーション名定義域に対応する添字つきタプルは，リレーション名との照合しか許されないことである。この仮定の下で，Coddのリレーショナル・カリキュラスに，アルファベットとして単項の述語変数が付加することにより，カリキュラスの拡張が図られた。

左	タプル識別子	左四形リレーション名	左四形番号	右四形リレーション名	右四形番号
	1	家	1	家	2

図4. 位置関係

Ⅲ. 実行効率に関する考察

Ⅲ. 1 検索効率

通常の関係モデルに於ける問合せの大部分は、

$$\pi_{A_1 \cdots A_n}(\sigma_F(R_1 \times \cdots \times R_m))$$

の形式で表わされる。ここで $\pi_{A_1 \cdots A_n}$ は属性 $A_1, \cdots, A_n$ への射影を、σ_F は論理式 F に対する選択を、$R_1 \times \cdots \times R_m$ は、リレーション $R_1, R_2, \cdots, R_m$ の直積を表わす。この問合せを処理する場合、最適化を考慮せず、特定のアクセス・ファシリティを用いないとすると、図5に示される様なm重のネストで行われる。従って、R_i のタプルの数を r_i とすると、結局 $r_1 \times r_2 \times \cdots \times r_R$ のタプルの参照が行われることになる。

一方、ARCに於ける問合せの大部分は

$$\pi_{A_1 \cdots A_m}(\sigma_F(R_1 \times \cdots \times R_m \times L_1 \times \cdots \times L_R))$$

の形式に書ける。ここで $L_1, L_2, \cdots, L_k$ はリレーション変数（問合せ中で、リレーション名の入るべき場所に書かれた例示要素）である。これらの値は、$R_1 \times \cdots \times R_m$ の一つのインスタンス毎に決定される。その結果図6に示されるm重のネストで行われる。前記の問合せ2を例にとると、まず家×左×構成が評価され、この各タプル毎に、(構成要素リレーション名, 構成要素番号) の実現値が、次に航行すべきリレーション名とその中のタプルを一意に決定する。一般的に、この種の問合せの処理のためには、$r_1 \times \cdots \times r_m \times R$ 個のタプルの参照が必要である。

問合せ処理に於て最も計算コストの高いものは接合項の計算である。問合せ言語にリレーション変数を導入した結果は、接合項の処理には殆ど影響を与えずにすむ。いま前述のnとmが等しいと仮定すると、処理時間は定数倍にしかならない。また、最適化、及びアクセス・ファシリティに関しても、関係データベースに於ける技術をそのまま使用することが可能である。（リレーション変数、タプル識別子）は、本質的にはポインタであり、直接目的のタプルを指しているので、

```
range of t1=R1;
range of t2=R2;
    :
range of tm=Rm;

loop 1 (t1);
    loop 2 (t2);
        :
            loop n (tn);
            end loop n;
        :
    end loop2;
end loop1;
```

図5. 関係モデルに於ける問合せ処理

```
range of t1=R1;
range of t2=R2;
    :
range of tm=Rm;

loop 1 (t1);
   loop 2 (t2);
       :
          loop m (tm);
              instantiate tm+1 → L1
              instantiate tm+2 → L2
                  :
              instantiate tm+k → Lk
          end loop m;
   end loop2;
end loop 1.
```

図6 ARCに於ける問合せ処理

特別のアクセス・ファシリティを必要としないし、最適化の対象から除いて構わない。

Ⅲ.2. 一貫性保全

関係データ・モデルでは、一貫性の問題は、関数従属性、多値従属性の問題として研究が盛んである。ARCに於てもこの面は重要であるが、意味構造に素直に表現する限り、正規形（若干通常の定義とは異なる）の形で設計される。（類似の議論はCoddを参照のこと）。この問題についての議論はここでは省き、削除挿入に関する参照関係の一貫性について考える。ARCでは、二つのエンティティ間の関係は、図7のように表現することが可能である。一般に t_1 がデータベースから削除される場合、当然関係 (t_1, t_2) もデータベース中から削除されなければならない。関係データベースでは、この種の参照関係の一貫性に関しては、ユーザに全てまかされるか、非常に制限された形でしか操作を許さないのが通常である。ARCに於ては、タプル識別子が陽に現われ、かつシステム制御であるため、この種の一貫性は完全に保障されなければならない。

TIMEでは、挿入・削除に伴う一貫性を保全するため、相互参照関係は図8に示されるような、双方向ポインタを附随情報として持たせてある。いまレコード t_1 で表わされるエンティティが削除されると、それを関係として含むエンティティ t_2, t_2〜t_n の削除を行う。この削除に伴い、その上のクラスのリレーションで、既に削除されたエンティティを関係として含むエンティティの削除が行われる。このように削除は、階層を次々と上って実行されることになる。このようにすることにより、関係データベースで陽に支援されない、削除に伴う一貫性の保証を行う。挿入に関しても類似である。ただし、これはメモリ効率、計算コストが共に高く、実用システムとしての大きな阻害要因となろう。

Ⅲ.3. おわりに

本稿では、現在筑波大学電子情報工学系で研究開発中のデータベース管理システムTIMEの基本概念について述べた。システムの採用したデータモデルARCは、本質的に関係モデルを包含するものである。即ちCoクラスのリレーションの集まりは、関係データベースを構成す

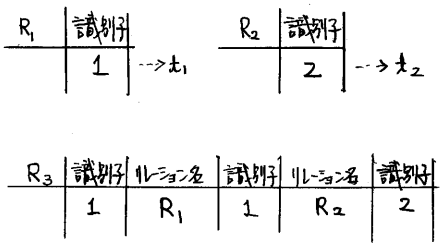


図7 ARCに於ける関係表現

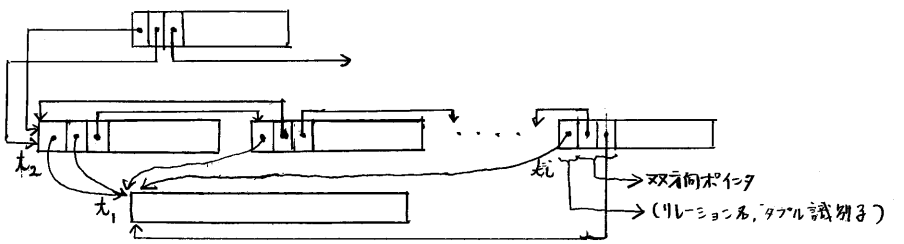


図8 TIMEに於ける参照関係表現

る．ＴＩＭＥシステム設計に於ても，Co クラスのリレーションのみでデータベースを構成するビューに対しては，通常の関係データベースとして操作可能であることが重要な要素である．Ci (i≥1) クラスのリレーションは，動的スキーマ情報，航行の動的制御，意味構造の把握情報に用いられることを想定している．試行錯誤を基本プロセスとする仕事では，動的要素が順次固定化されていくと考えられる．それに基づいて，それまで上位クラスにあったリレーションを順次下位へ落して行くことにより，実行効率が上ることが予想される．現在では，このような，リレーションの変更は，データベース管理者の手にまかされることになるが，このようなデータ構造のチューニングに対する支援システムの開発が，実用システムへの重要な道ではないかと考えられる．

参考文献

1. E. F. Codd, "Extending the Database Relational Model to Capture More Meaning," ACM TODS, Vol. 4, No. 4, 1979
2. J. M. Smith and D. C. P. Smith, "Database Abstraction: Aggregation," CACM, Vol. 20, No. 6
3. J. M. Smith and D. C. P. Smith, "Database Abstraction: Aggregation and Generalization," ACM TODS, Vol. 2, No. 2, June 1977
4. K. Yamaguchi and T. L. Kunii, "Logical Framework of a Picture Database Computer," Proc. 1981 IEEE Comp. Soc. Work. on Computer Architecture for Pattern Analysis and Image Database Management
5. K. Yamaguchi, N. Ohbo, T. L. Kunii, H. Kitagawa and M. Harada "ELF: Extended Relational Model for Large, Flexible Picture Databases," Proc. 1980 IEEE Comp. Soc. Work. on Picture Data Description and Management
6. E. F. Codd, "Relational Completeness of Data Base Sublanguages," in Data Base Systems, ed. R. Rustin, 1972, Prentice-Hall
7. M. M. Zloof, "Query by Example," Proc. AFIPS, NCC, Vol. 44, 1975