DNM: Dynamic Nesting Method

北川博之・国井利泰（東京大学理学部情報科学科）

＊要約＊
リレーショナル・データベース上での応用プログラムの開発をサポートするためのデータベース・エンジニアリング・ツールの基礎となる、ＤＮＭ（Dynamic Nesting Method）について述べる。ＤＮＭでは、リレーショナル・データベース中のフラットなリレーション中に、個々のデータ応用目的に適合した階層構造をユーザが手軽に導入することを可能にする。これによって、データベースの応用要求仕様の記述および応用プログラムの作成等の面での作業の軽減化が期待できる。ここでは、ＤＮＭで必要となる４つのＤＮ演算子を定義し、その基礎的性質について例を用いて説明を行なう。

１．はじめに

本講では、リレーショナル・データベース（[2],[3].[7]）上での応用プログラムの開発をサポートするためのデータベース・エンジニアリング・ツールの基礎としてのＤＮＭ（Dynamic Nesting Method）を提案する。ＤＮＭでは、リレーショナル・データベース中のフラットなリレーションに対して、ユーザが手軽に、行や列の入れ子関係を定義することを可能とする。この結果、図１に示すようなネステド・テーブル（Nested table）がユーザ・ビューとして得られる。このようなネステド・テーブルを扱うことは、次のような点で、リレーショナル・データベースのユーザ・インターフェイスの親ユーザ性を向上させる効果を持つと期待できる。

(1) 応用要求に適合したデータ構造の導入

リレーショナル・データベースにおいて取り扱われる唯一の構造体は、リレーションと呼ばれるフラット・テーブルである。フラット・テーブルのみを扱い、行や列の入れ子関係を許さないことは、データの応用に依存する要素を極力排除し、種々の応用を平等にサポートしようという哲学のもとには説得力がある。しかし、個々のデータの応用目的にとっては、それに最も適合したデータ構造が準備されていれば、応用要求仕様の記述も応用プログラムの作成も、はるかに容易になる。例えば、部品供給者と部品の組み合わせを単位として情報を取り扱いたいなら

＊部品供給表１

供給者番号	部品番号	＊供給者部品別部品供給			
		部品個数	供給日		
			年	月	日
1	1	50	80	7	25
		60	80	8	10
1	2	20	80	7	14
		30	80	8	10
2	3	100	80	7	25
		50	80	8	10
3	2	10	80	7	14
		50	80	8	10
⋮	⋮	⋮	⋮	⋮	⋮

図１　ネステド・テーブル　＊部品供給表１

ば、図1に示したネステド・テーブル「*部品供給表1」は、同一の情報をフラットな表で示したリレーション「*部品供給表2」よりも便利であると予想される。一方、日付、部品供給者、部品とその供給量という階層がデータを扱う上で都合良い場合には、図3に示すネステド・テーブル「*部品供給表3」を定義すればよい。このように、DNMではデータの各応用目的に最も適切な階層構造を、テーブルの中に導入することが可能である。

*部品供給表2					
供給者番号	部品番号	部品個数	年	月	日
1	1	50	80	7	25
1	1	60	80	8	10
1	2	20	80	7	14
1	2	30	80	8	10
2	3	100	80	7	25
2	3	50	80	8	10
3	2	10	80	7	14
3	2	50	80	8	10
⋮	⋮	⋮	⋮	⋮	⋮

図2 フラット・テーブル　*部品供給表2

(2) 通常の表形式のサポート

通常のデータの記述に用いられる表形式は、リレーショナル・モデルにおけるようなフラット・テーブルではなく、行や列の入れ子を含むネステド・テーブルであることの方が多い。これは、データベースの主たる応用分野の一つである事務処理の分野で顕著である([93])。例えば、日付というカラムは図1や図3のネステド・テーブルに示されるように、年、月、日という3つのカラムから構成されるのが普通であり、決して図2のような表にはならない。このように、事務処理等の応用を直接的にサポートする上でも、ネステド・テーブルを扱うことは重要である。

*部品供給表3					
*日付別部品供給			供給日		
供給者番号	*供給者別部品供給				
	部品番号	部品個数	年	月	日
1	1	50	80	7	25
2	3	100			
1	2	20	80	7	14
3	2	10			
1	1	60	80	8	10
	2	30			
2	3	50			
3	2	50			
⋮	⋮	⋮	⋮	⋮	⋮

図3　ネステド・テーブル　*部品供給表3

(3) データ・セマンティック表現力の強化

ネステド・テーブルは、その中にデータの階層構造を含むことができるため、適切なネステド・テーブルを用いれば、データのセマンティクスを表現することも容易になる。例えば、リレーショナル・モデルにおいては多値依存性([5])の概念を導入しなくては扱えないデータ・セマンティクスが、ネステド・テーブルでは関数依存性([4]、[11])の概念のみで取り扱うことが可能となる。このような意味で、ネステド・テーブルはデータ・セマンティクスの表現という面でも新しいパワーを提供する。

(4) リレーショナル・モデルの包括

以上のような新しい利点をネステド・テーブルは提供するが、その一方で、リレーショナル・モデル自体をも、完全にネステド・テーブルの理論の中に包括できる点が重要である。ネステド・テーブルで行や列の入れ子を許せば、当然、それを扱う演算子の複雑性は増加する。しかし、従来のリレーショナル・モデルの簡潔さをユーザが望む場合には、それも可能なのである。この意味で、DNMは、リレーショナル・データベースが持つ特徴を保持しつつ、より多様な要求に適応できる拡張ユーザ・インターフェイスを提供することになる。

本講では、ネステド・テーブルの特徴である行や列の入れ子関係を操作するための4つの演算子の定義と、その基本的性質の解説を中心に説明を行なう。ネステド・テーブルを扱うためのその他の演算子については別の機会に発表する予定である。

2. ネステド・テーブル

ネステド・テーブルは、その行や列の入れ子関係を許すテーブルである。ネステド・テーブルの各行は、フィールドと呼ばれる。フィールドの入れ子の中で、最も外側のトップ・レベルのフィールドは、ルートと呼ばれる。ルートはネステド・テーブルのテーブル名を与える。図1に示したネステド・テーブル「*部品供給表1」では、ルートは「供給者番号」、「部品番号」、「*供給者部品別部品供給」の3つのフィールドをそのコンポーネントとして持っている。フィールド「*供給者部品別部品供給」は、「部品個数」と「供給日」をそのコンポーネントとして持っている。フィールド「供給日」はさらに「年」、「月」、「日」をコンポーネントとする。少なくとも1つ以上のコンポーネントを持つフィールドは、グループ・フィールドと呼ばれ、そうでないものはシンプル・フィールドと呼ばれる。テーブルの中味の実際のデータ値はオカレンスと呼ばれる。図1に示したネステド・テーブルでは、シンプル・フィールド「供給者番号」のオカレンスは、整数1, 1, 1, 2, …であり、「部品番号」のオカレンスは、整数1, 2, 3, 2, …である。また、グループ・フィールド「供給日」のオカレンスはタプル(80, 7, 25), (80, 8, 10), (80, 7, 14), …である。ここで、グループ・フィールド「*供給者部品別部品供給」のオカレンスは、集合{(50,(80, 7, 25)), (60,(80, 8, 10))}, {(20, (80, 7, 14)), (30, (80, 8, 10))}, …であることに注意しなくてはならない。このように、そのオカレンスが集合となるようなフィールドは、リピーティング・フィールドと呼ばれ、そうでないものは非リピーティング・フィールドと呼ばれる。この例でわかるように、リピーティング・フィールドは、その名前の先頭に*を付すことによって示される。ルートは、リピーティング・フィールドでなくてはならない。以上の通り、各フィールドは、グループかシンプルかのどちらかであり、また、リピーティングか非リピーティングかのどちらかである。ネステド・テーブル「*部品供給表1」のフィールドは、表1のように分

	リピーティング	非リピーティング
グループ	*部品供給表1 *供給者部品別部品供給	供給日
シンプル	—	供給者番号 部品番号 部品個数 年, 月, 日

表1 フィールドの分類

類できる。

各フィールドFは、そのオカレンスのドメインdom(F)を持っている。各フィールドのドメインは、次に示す条件を満たしていなくてはならない。

1) Fが非リピーティング・シンプル・フィールドの時は、dom(F)は原始的なデータ項目（例えば、整数や文字列等）の集合でなくてはならない。

2) Fがリピーティング・シンプル・フィールドの時は、dom(F)は原始的なデータ項目の集合のべき集合でなくてはならない。

3) Fが非リピーティング・グループ・フィールドで、$C_1, \cdots, C_N$ がそのコンポーネントの時は、

$$dom(F) \subseteq CP(C_1, \cdots, C_N)$$ （CPの意味については以下参照）

でなくてはならない。

4) Fがリピーティング・グループ・フィールドで、$C_1, \cdots, C_N$ がそのコンポーネントの時は、

$$dom(F) \subseteq 2^{CP(C_1, \cdots, C_N)}$$ （$CP(C_1, \cdots, C_N)$ のべき集合）

でなくてはならない。

ここで、$CP(C_1, \cdots, C_N)$ は、$dom(C_1), \cdots, dom(C_N)$ の直積を示す。直積の要素はタプルと呼ばれる。タプルtの中でのフィールド C_i のオカレンスは $t[C_i]$ で示す。また、$X \subseteq \{C_1, \cdots, C_N\}$ をフィールドの集合とする時、$t[X]$ でX中の全ての C_i に対し $U[C_i] = t[C_i]$ であるような、CP(X)のタプルUを示す。

3. DN演算子

ここで、ネステド・テーブル中の行や列の入れ子関係を定義するための4つの演算子を導入する。これらは、CE(Column Ennest)演算子、CD(Column Denest)演算子、RE(Row Ennest)演算子、RD(Row Denest)演算子である。これらは、総称してDN(Dynamic Nesting)演算子と呼ばれる。初めの2つは行の入れ子を扱う演算子であり、後の2つは列の入れ子を扱う演算子である。Ennest演算子は入れ子をより深いものにするためのものであり、Denest演算子は逆に入れ子をより浅くするためのものである。IBMのB. Housel らによって研究されている言語CONVERT([6],[13],[14])においても、階層構造をなすデータを扱うための演算子を定義しているが、ここで我々が定義する演算子の方がより基礎的なものであり、CONVERTでのそれはこれらの組み合わせとして表現することができる。

(1) CE演算子　CE[$F_1, \cdots, F_N$ INTO F]

あるグループ・フィールドGのコンポーネントの部分集合 $\{F_1, \cdots, F_N\}$ をまとめて、これらをコンポーネントとする新しいフィールドFを生成する。フィールドF自身は、Gの非リピーティングなコンポーネントとなる。

(2) CD演算子　CD[F]

あるグループ・フィールドGの非リピーティングなグループ・コンポーネントFに作用し、Fというグループ・フィールドを消去する。これはCE演算子と逆変換の関係にある。

(3) RE演算子　RE[F]

あるリピーティング・グループ・フィールドGの非リピーティングなコンポーネントFに作用し、これをリピーティング・フィールド*Fにする。Gの各オカレンスRの中では、F以外のフィールドに同じオカレンスを持つタプルが全てま

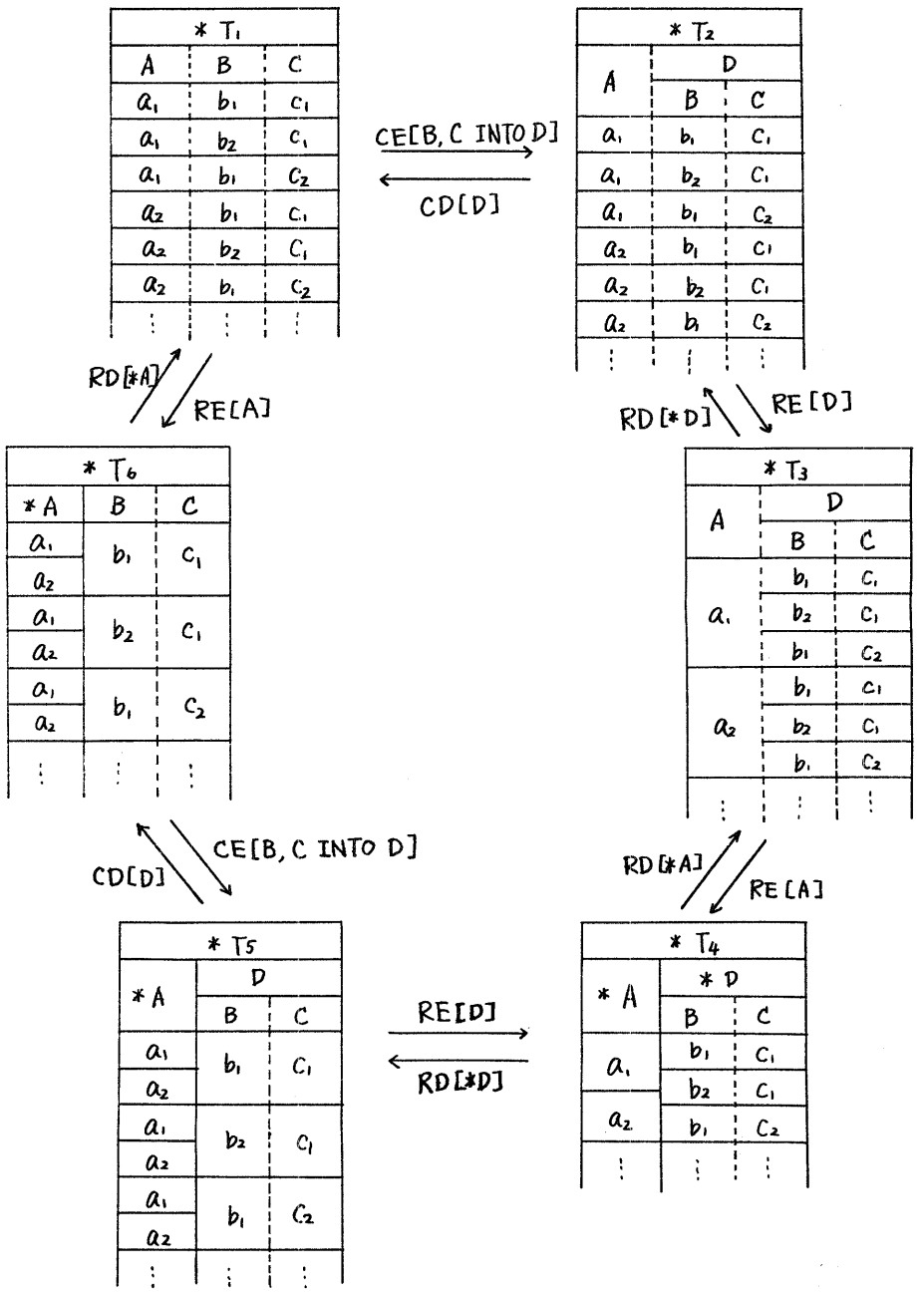

図4　DN演算子

とあられて、新しいタプルを生成する。より形式的には、次のようなR'でRは置き換えられることになる。

$$R' = \{(S(t), t[F_1], \cdots, t[F_N]) \mid t \in R\}$$

ここで、$F_1, \cdots, F_N$ はF以外のGの全てのコンポーネントであり、また

$$S(t) = \{s[F] \mid s \in R \wedge s[F_1] = t[F_1] \wedge \cdots \wedge s[F_N] = t[F_N]\}$$

である。

(4) RD演算子　RD[*F]

あるリピーティング・フィールドGのリピーティングなコンポーネント*Fに作用し、*Fを非リピーティング・フィールドFにする。オカレンスに対する影響はRE演算子の場合と逆である。より形式的には、*Fの各オカレンスRは次のようなR'で置き換えられ、さらに重複したオカレンスの除去がなされる。

$$R' = \{(s, t[F_1], \cdots, t[F_N]) \mid t \in R \wedge s \in t[*F]\}$$

ここで、$F_1, \cdots, F_N$ は *F 以外のGの全てのコンポーネントである。後に述べるように、RD演算子とRE演算子は必ずしも逆変換の関係にはならない。

各演算子の作用を図示したのが図4である。これを見れば、以上の演算子の意味は明らかであろう。

4．例題

ここで、最初に示したネステド・テーブル「*部品供給表1」と「*部品供給表3」を、フラット・テーブル「*部品供給表2」からDN演算子を用いて定義する。まず、「*部品供給表1」であるが、これは次のDN演算子の列によって定義できる。

CE［年、月、日 INTO 供給日］;
CE［部品個数、供給日 INTO 供給者部品別部品供給］;
RE［供給者部品別部品供給］

もちろん、この場合には次のようにしてもよい。

CE［部品個数、年、月、日 INTO 供給者部品別部品供給］;
RE［供給者部品別部品供給］;
CE［年、月、日 INTO 供給日］

この他にも「*部品供給表1」を定義するためのDN演算子の列は存在する。

次に、「*部品供給表3」であるが、これは次のDN演算子の列によって定義できる。

CE［年、月、日 INTO 供給日］;
CE［部品番号、部品個数 INTO 供給者別部品供給］;
CE［供給者番号、供給者別部品供給 INTO 日付別部品供給］;
RE［供給者別部品供給］;
RE［日付別部品供給］

この場合にも、これ以外に「*部品供給表3」を定義するためのDN演算子の列はいく通りか存在する。

5．依存性（Dependency）

DN演算子の性質を述べる上で便利なように、ここでネステド・テーブルにお

ける関数依存性（Functional Dependency）、多値依存性（Multivalued Dependency）の概念を導入する。これらは牧ノ内氏（[11]）による同様の概念をより一般的な形に拡張し定式化したものである。ここでは、後の説明のために必要な、リピーティング・グループ・フィールドのコンポーネント間の簡単な依存性の定義のみを与える。

《定義1》 Gをネステド・テーブルTのリピーティング・グループ・フィールドとし、そのコンポーネントの集合をCとする。いま、X，YをCの部分集合とする。Gのあるオカレンスの中に含まれるすべてのタプルt_1およびt_2において、$t_1[X]=t_2[X]$ならば$t_1[Y]=t_2[Y]$の関係が成立する時、関数依存性X→Yがそのオカレンス中で成立する。すべてのGのオカレンスにおいて、関数依存性X→Y*)が成立する時、ネステド・テーブル中において関数依存性X→Yが成立する。

《定義2》 Gをネステド・テーブルTのリピーティング・グループ・フィールドとし、そのコンポーネントの集合をCとする。いま、X，YをCの部分集合とし、Z＝C－X－Yとする。Gのあるオカレンスにおいて、あるタプルt中でt[X]=xかつt[Z]=zとなっているあらゆるxおよびzに対し、$R_{xz}[Y]=R_x[Y]$**)が成立する時、多値依存性X↠Yが成立する。すべてのGのオカレンスにおいて、多値依存性X↠Y*)が成立する時、ネステド・テーブル中において多値依存性X↠Yが成立する。

6．DN演算子の基本的性質

ここでDN演算子の基本的性質について述べることにする。以下の命題や系の証明は[10]に与えられている。

《命題1》 ネステド・テーブルTが与えられた時、$F_1,\cdots,F_N$をあるシンプル・グループ・フィールドGのコンポーネントとする。いま、T'=CE[$F_1,\cdots,F_N$ INTO F](T)とした時、T=CD[F](T')が成立する。

《命題2》 ネステド・テーブルTが与えられた時、Fを非リピーティング・グループ・フィールドとし、Fのコンポーネントの集合をCとする。いま、T'=CD[F](T)とした時、T=CE[C INTO F](T')が成立する。

これらの命題は、CE演算子およびCD演算子の定義より明らかである。これらの命題により、CE演算子およびCD演算子により定義されたテーブルの変換はCE演算子およびCD演算子により定義された逆変換を持つことが保証される。

---

*)各オカレンスでの依存性X⇒Yは意味的に同一のものである必要はない。単に依存性X⇒Yが成立していればよい。

**) $R_x[Y]=\{t[Y]\mid t\in R\wedge t[X]=x\}$

$R_{xz}[Y]=\{t[Y]\mid t\in R\wedge t[X]=x\wedge t[Z]=z\}$

《命題３》　ネステド・テーブルTが与えられた時、Fをあるリピーティング・グループ・フィールドの非リピーティングなコンポーネントとする。いま、T'=RE[F](T)とした時、T=RD[*F](T')が成立する。ただし、*FはT'中でFに対応するリピーティング・フィールドである。

この命題によれば、RE演算子によるテーブル変換は必ずRD演算子による逆変換を持つ。しかし、この逆は真ではない。すなわち、RD演算子によるテーブル変換は必ずしもRE演算子による逆変換を持つとは限らない。この例を示したのが図５である。ネステド・テーブル$*T_1$にRD[*C]を適用して$*T_2$を得た後、RE[C]を適用しても、$*T_1$には戻らず$*T_3$になってしまう。このような場合の逆変換の存在の条件を与えるのが次の命題である。

《命題４》ネステド・テーブルTが与えられた時、*Fをあるリピーティング・グループ・フィールドGのリピーティングなコンポーネントとし、Gのコンポーネント全体の集合をCとする。いま、T'=RD[*F](T)とした時、$C-\{*F\}\rightarrow\{*F\}$となる関数依存性がTにおいて成立することが、T=RE[F](T')となるための必要十分条件である。ただし、FはT'中で*Fに対応する非リピーティング・フィールドである。

命題４における関数依存性は、RE演算子の適用結果においては保証されていることを述べているのが、次の命題である。

《命題５》　ネステド・テーブルTが与えられた時、Fをあるリピーティング・グループ・フィールドGの非リピーティングなコンポーネントとし、Gのコンポーネント全体の集合をCとする。いま、T'=RE[F](T)とした時、$C-\{*F\}\rightarrow\{*F\}$となる関数依存性がT'において成立する。ただし、*FはT'中でFに対応するリピーティング・フィールドである。

これらの命題より次の系を得る。

《系１》　リレーショナル・データベース中のフラット・テーブルから、CE演算子，RE演算子を用いてネステド・テーブルを生成する時、このテーブル変換

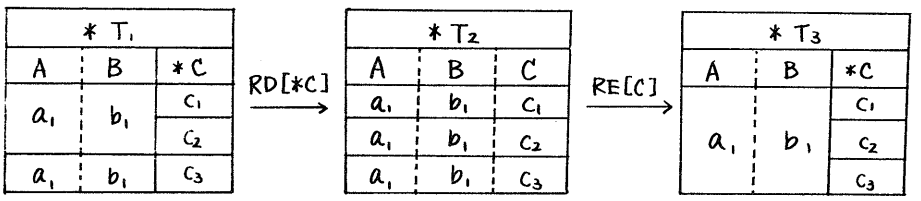

図５　可逆でないテーブル変換の例

は、ＣＤ演算子、ＲＤ演算子による逆変換を持つ。

《系２》　フラット・テーブルＵ上に、ＣＥ演算子およびＲＥ演算子を用いてネステド・テーブルＴを定義したとする。この時、すべてのリピーティング・グループ・フィールドＧの、すべてのリピーティングなコンポーネント＊Ｆについて、Ｎ→＊Ｆ（ＮはＧの非リピーティングなコンポーネントの集合）がＴで成立するような条件の下でＴを更新した時、その更新はＵに反映できる。

ＤＮ演算子同士の可換性に関しては多くの議論すべき点がある。ここでは、２つの連続したＲＥ演算子が、図６に示すように同一のテーブル形式を与えるにもかかわらず、その内部のデータが一致しないというケースのみを例として注目する。このような場合の、ＲＥ演算子の可換性を保証する十分条件は、次の命題が与えている。

《命題６》　ネステド・テーブルＴが与えられた時、Ｇをあるリピーティング・グループ・フィールドとし、Ｇのコンポーネントの集合をＣとする。ＨとＩをたいに異なるＧのリピーティングなコンポーネントとする時、もしＣ－{Ｈ，Ｉ}↠{Ｈ}がＴにおいて成立するならば、ＲＥ[Ｉ](ＲＥ[Ｈ](Ｔ))＝ＲＥ[Ｈ](ＲＥ[Ｉ](Ｔ))が成立する。

７．おわりに

本講では、データベース・エンジニアリング・ツールの基礎としてのＤＮＭを提案し、その中心となるＤＮ演算子の基本的性質について述べた。現在さらにＤＮ演算子の性質についての研究を進めている。

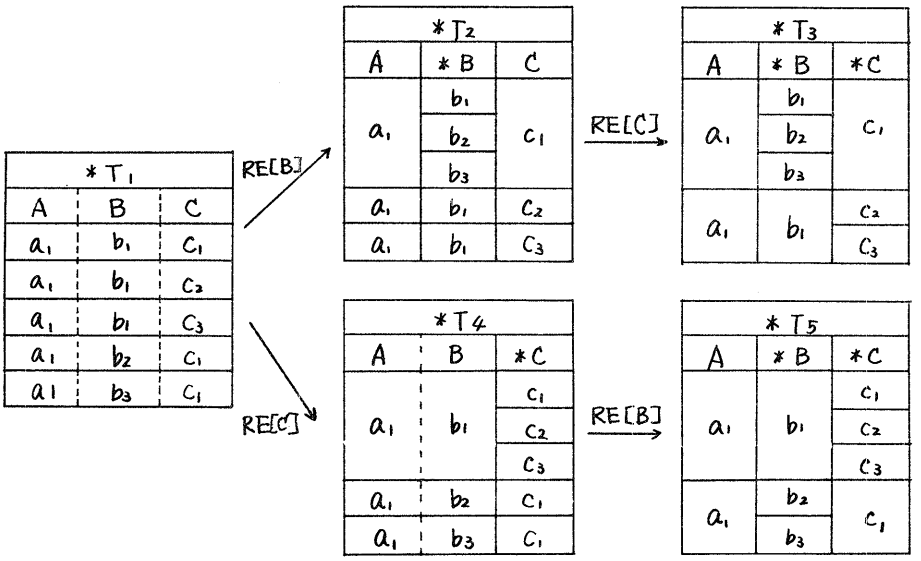

図６　可換でないＲＥ演算子の適用例

参考文献

[1] Chamberlin D. D., et al., 'Views, Authorization, and Locking in a Relational Data Base System,' Proc. AFIPS National Computer Conference, NCC '75, Anaheim, May 1975, pp. 425-430.

[2] Chamberlin D. D., et al., 'Relational Data-Base Management Systems,' ACM Computing Surveys, Vol. 8, No. 1, March 1976, pp. 43-66.

[3] Codd E. F., 'A Relational Model of Data for Large Shared Data Banks,' Comm. ACM, Vol. 13, No. 6, June 1970, pp. 377-387.

[4] Codd E. F., 'Further Normalization of the Data Base Relational Model,' in 'Data Base Systems,' Rustin R., Ed., Prentice-Hall, 1972, pp. 33-64.

[5] Fagin R., 'Multivalued Dependencies and a New Normal Form for Relational Data-Bases,' ACM Trans. on Database Systems, Vol. 2, No. 3, September 1977, pp. 262-278.

[6] Housel B. C. and Shu N. C., 'A High-Level Data Manipulation Language for Hierarchical Data Structures,' Proc. Conf. on Data Abstraction, Definition and Structure, Salt Lake City, March 1976, pp. 155-168.

[7] Kim W., 'Relational Database Systems,' ACM Computing Surveys, Vol. 11, No. 3, September 1979, pp. 185-211.

[8] Kitagawa H., 'APAD: Application-Adaptable Database System - Its Architecture and Design -,' M. Sc. Thesis, Dept. of Information Science, Univ. of Tokyo, January 1980.

[9] Kitagawa H. et al., 'A Language for Office Form Processing (OFP) - with Application to Medical Forms -,' to appear in Proc. 3rd World Conf. on Medical Informatics, Tokyo, September 1980.

[10] Kitagawa H. and Kunii T. L., 'DNM: Dynamic Nesting Method - An Aid for User Friendly Database Application Development -,' (DRAFT).

[11] Makinouchi A., 'A Consideration on Normal Form of Not-Necessarily-Normalized Relation in the Relational Data Model,' Proc. 3rd Int. Conf. on VLDB, Tokyo, October 1977, pp. 447-453.

[12] Osman I. M., 'Updating Defined Relations,' Proc. AFIPS National Computer Conference, NCC '79, New York, June 1979, pp. 733-740.

[13] Shu N. C., et al., 'CONVERT: A High Level Translation Definition Language for Data Conversion,' Comm. ACM, Vol. 18, No. 10, October 1975, pp. 557-567.

[14] Shu N. C., et al., 'EXPRESS: A Data EXtraction, Processing, and REStructuring System,' ACM Trans. on Database Systems, Vol. 2, No. 2, June 1977, pp. 134-174.