

仮想化技法としてのデータベースビューについて

増永良文 (東北大学 電気通信研究所)

1. はじめに

データベースは社会の貴重な共有資源であるが、データベースを利用しようとするとき、一般にデータベースに蓄積されているデータが利用者の使用した形式で格納されているという保証はない。これはデータ操作言語についても同様である。多くの利用者がそれぞれ多様な利用形態を望むと云っても過言ではない。このようなとき、データベースシステムがビュー (view) をサポートし、データベース及びデータ操作言語を仮想化して利用者に提示し、データベースに格納されている情報を有効に利用させることができれば大変都合が良い。データベース管理システムのビューサポートサブシステムはこの機能を實現せんがためのものである。現在 CODASYL 型データベースシステムではリクエストサポート機能がこれに該当し、関係データベースシステムである SYSTEM R や INGRES, あるものは QBE でビューをサポートする機能が組み込まれているが、ほとんどがデータの検索にのみ機能し、動的なデータ更新を対象とした機能は持ち合わせておらず、これからの機能充実に待たれている。このためには解決しなければならない困難な問題がデータ意味論とのからみで指摘されているが、ポイントとしてはデータベース及びその操作言語を仮想化してユーザに提示して中二層という発想にある。

一方 ANSI/X3/SPARC は良く知られているようにデータベースシステムアーキテクチャとして内部、概念および外部のスキーマからなる層スキーマ構造を提案している。上述ビューサポートの概念はまさにデータベース管理システムがシステムの持ち合せているデータベースの論理的世界である概念スキーマ上に利用者向の外部スキーマの世界を如何にサポートするかと云うビューサポートサブシステムを構築する問題に相当している。従ってビューサポートの問題は外部スキーマと概念スキーマ間の構造的、操作的な問題と捉えることが出来るが、では一般に外部スキーマとして一体どのような様子の様なメカニズムでサポート可能であるのかについてほとんど十分な議論が行われていない。本稿では具体的にスキーム変換型ビュー、データ操作変換型ビュー、データモデル変換型ビュー、抽象化型ビュー、知識型ビューおよび時変型ビューの諸概念を提起し各々説明している。議論の数学的平明さから関係データモデルを用いて行われている所が多いが、議論の一般性は失われている。

2. ビューサポートの基本的メカニズム

一つの組織体がデータベース構築の対象とした実世界をデータベース実世界 (DBRW) とする。DBRW のデータ構造をなるべく忠実にデータベースシステム内に与り込めべく、適当なデータモデルが使用されて、データベース概念世界 (DBCW) が構築される。DBCW の構築には大別して (1) DBCW 設計者により先験的に与えられる場合と、(2) 複数の利用者ビューを統合して与えられる場合の二つに分けることが出来るがここではこれ以上言及しない。一

方、通常組織体には多数のデータベース利用者がいる。それぞれ各利用者は実世界に対してそれぞれ固有のビューを持ち、データベース外部世界 (DBEW) を形成する。1章でも述べた様にビューサポートとは DBEW ⊆ DBEW が (1) 構造的かつ (2) 操作的にサポートするに等しい。図-1 にビューサポートの概念図を示す。

さて、ビューも概念スキーマと同様にあるデータモデルに基づき (1) スキーマ構造記述 (SSD) と (2) データ操作言語 (DML) の対で表わされると考える。概念スキーマのデータモデルとしては CODASYL のネットワークモデル, IMS に代表される階層モデル, 関係モデル等が考えられるが、ビューのそれは一般に利用形態に依存し可成りの多様性が発生する。これについては次章より逐時説明してゆくことになるが、本章以下ではビューサポートの基本的定義にもう少し言及する。

R を関係名, $A_i (i=1, \dots, n)$ を属性名, $D_i (= \text{dom}(A_i)) (i=1, \dots, n)$ をドメイン名とする。

$R(A_1/D_1, A_2/D_2, \dots, A_n/D_n)$ をフリー関係スキーマとす。これを次の様に定義される。 $R \equiv \{ r \mid r \subseteq D_1 \times D_2 \times \dots \times D_n \}$ 。 C_R

を $2^{D_1 \times \dots \times D_n}$ 上の述語とし、フリー関係スキーマ R 上に課せられた保全制約条件とする。 C_R に関する R の関係スキーマ R_C (以後単に関係 R_C とする場合も) を次のように定義する。 $R_C \equiv \{ r \in R \mid r \text{ は } C_R \text{ を満たす} \}$ 。フリー関係スキーマ R の更新 (挿入, 削除, 書換) 演算 $u_R : R \rightarrow R$ とする。 u_R が C_R と両立するとは $\forall r \in R_C, u_R(r) \in R_C$ が成立するを意味する。

$U_R \equiv \{ u_R : R \rightarrow R \}$ とし、 $U_R^* = \{ \Lambda \} \cup U_R \cup U_R^2 \cup \dots$, $\equiv \equiv \forall r \in R, \Lambda(r) = r$, とする。次に R_1, \dots, R_m をフリー関係スキーマとし、 $D = (R_1, R_2, \dots, R_m)$ をフリーデータベーススキーマとす。 $D \equiv \{ (r_1, \dots, r_m) \mid r_i \in R_i, i=1, \dots, m \}$ であり、 $D_C \equiv \{ (r_1, \dots, r_m) \in D \mid (r_1, \dots, r_m) \text{ は } C_D \text{ を満たす} \}$ かつ C_D に関する D のデータベーススキーマ (以後単にデータベース D_C という場合もある) とする。データベース更新演算 $u_D : D \rightarrow D$, $u_D \equiv (u_{R_1}, u_{R_2}, \dots, u_{R_m})$, $\equiv \equiv u_{R_i} : R_i \rightarrow R_i$, と定義される。つまり、 $u_D(r_1, r_2, \dots, r_m) = (u_{R_1}(r_1), u_{R_2}(r_2), \dots, u_{R_m}(r_m))$ である。 $U_D \equiv \{ u_D : D \rightarrow D \}$, $U_D^* = \{ \Lambda \} \cup U_D \cup U_D^2 \cup \dots$ とする。よってフリービュースキーマ (以後単にビューとす) V を次のように定義する。 $V \equiv (V_1,$

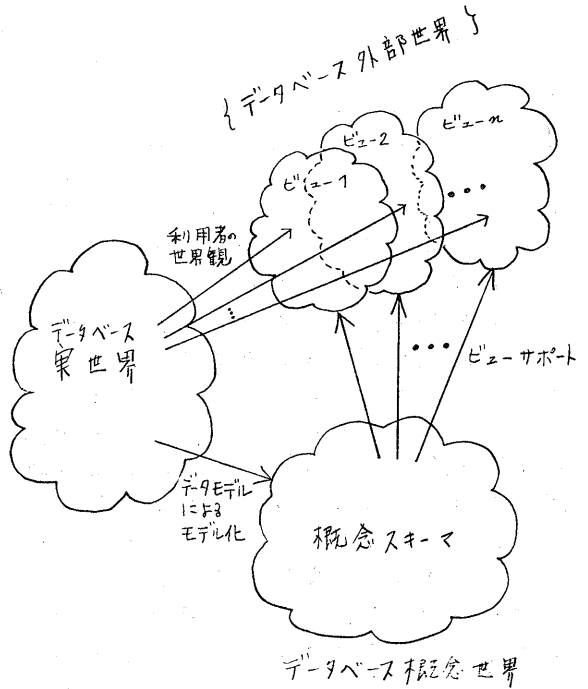


図-1 ビューサポート概念図

V_2, \dots, V_p), $i = 1, \dots, p$ の V_i ($i = 1, \dots, p$) は V との関係スキーマ。ビュ-更新演算は $u_V: V \rightarrow V$ である。 $U_V \equiv \{u_V: V \rightarrow V\}$, $U_V^* \equiv \{u_V\} \cup U_V \cup U_V^2 \cup \dots$ とする。

そこでビュ-定義写像を $f: D_C \rightarrow V$ と定義する。 f は D_C から V への構造的写像ということができる。ビュ-サポートの基本的概念を説明するには D_C の構造的写像の他に以下に導入する操作的写像が必要である。またビュ-サポートにも種々の定義概念がある。以下に要旨のみをこれらに言及する。

[定義1] $D_C' \subseteq D_C$ とする。 D_C' が V を u_V によってサポートするとは次の条件が成立すること。

$$(\exists! \theta: U_V^* \rightarrow U_D^*) (\forall d \in D_C') (u_V(f(d)) = f(\theta(u_V)(d)) \text{ かつ } (\theta(u_V)(d) \in D_C'))$$

すなわち θ を操作的写像という。一般に D_C の部分集合 D_C' の規定の仕方には大別して二通りの方法がある。一つはビュ-の更新操作 u_V に依存する場合であり、他の一つはそれに依存しなく専らビュ-の構造的写像に依存する場合である。

しかしながら本稿ではこれ以上議論の詳細には立ち入らない。上記定義の最初の条件項は図-2に示す可理図が成立し得ることを主張している。

$\theta =$ 番目の条件項は $\theta(u_V)$ が制限 D_C' 上で C_D が C_D' の部分集合であることを主張している。

U_V^* を次のように分類する。
 $D_U^* \equiv \{u \in U_V^* \mid \forall v \in V, |u(v)| \leq |v|\}$,
 $I_U^* \equiv \{u \in U_V^* \mid \forall v \in V, |u(v)| \geq |v|\}$,
 $M_U^* \equiv \{u \in U_V^* \mid \forall v \in V, |u(v)| = |v|\}$.

すなわち $|u|$ は $|v|$ のタッポル数を表わす。

[定義2] D_C' が V をタッポル削除操作によってサポートするとは D_U^* の任意の元 u_V によって定義1の条件が成立すること。

タッポル挿入、書換についても同様に定義する。

[定義3] D_C' が V をサポートするとは D_C' が V をタッポル削除、挿入、書換操作によってサポートすること。

一般にビュ-サポートの問題を議論してみると、 θ が唯一に存在する場合、 θ は存在するが唯一ではない場合、 θ が存在しない場合等いろいろの場合が種々のからみで生じ、問題が極めて複雑であることがわかる。詳細な議論を展開することは本稿の目的ではないのでこれ以上行なわない。しかしながら、次章から述べるデータベースビュ-概念の多様性を整理するうえで、上述議論の工ッセスは理解する必要がある。

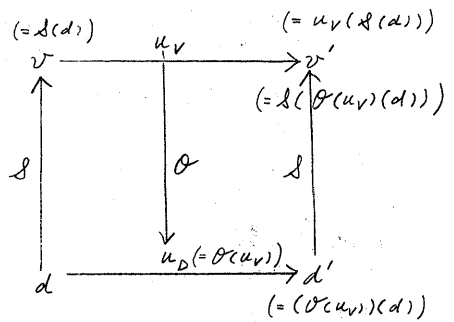


図-2 ビュ-サポートの可理図

3. 49様なデータベースビュ-

データベースシステムが各種のビュ-をサポートしようとするとは、システムが各種ユーザインタフェイスを提供しようとしていることに外ならない。関係データベースシステムを例にとれば、関係論理及び代数指向型のインタフェイス、関係論理あるいは代数指向型ではあるが、グラフィック端末での利用を指向

したインタフェース、非関係型データモデルをサポートすることを指向したインタフェース、あるいは自然言語を指向したインタフェース等各種インタフェースが開発されてきている。最近では外部インタフェースは種々の高度なアプリケーションとのからみで複雑高度化の傾向が著しい。例として歯科矯正学診断サポートシステム⁽¹⁾では頭蓋骨図形データベースに基いたタレベルの外部インタフェースを設計することを計画している。このインタフェース設計のテーマは概念の抽象化に基くものであり、スペーシャルデータベース⁽²⁾も同じ概念に基づいている。データベースの知識情報処理への適用を意識するとデータベースからの知識情報の切出しも必要となってくる。データベースのライフサイクルを考慮すると、データベースは時間的にその構造が変化すると考えるのが自然であり、利用者プログラムはどのような変化にも免疫性がほしい。例を挙げれば限りが無いが、これら各種の異なる要求を実現させる共通の道具がデータベースビュージャケットの概念である。これら数々の要求を統一的に把握するため次の主軸、副軸を手立てにする⁽³⁾。ただしこれら諸軸は数学的に直交した概念ではない場合もある。あくまで概念整理の柱のもののである。

(1) データモデル軸

(1-1) ビューは概念スキームと同一データモデルである。

(1-1-1) ビューは概念スキームとスキーム構造記述(SSD)で異なる。データ操作言語(DML)では異なる。

(1-1-2) ビューは概念スキームとDMLで異なる。SSDでは異なる。

(1-1-3) ビューは概念スキームとSSDとDMLで異なる。

(1-2) ビューは概念スキームと異なるデータモデルである。

(2) 抽象化軸

ビューは概念スキームで表現されている情報を抽象化して得られる世界、つまりデータベース外部世界の一つ、を代表するものとして定義する軸。

(3) 知識軸

ビューを定義すると使用される構造的写像が概念スキームレベルに蓄積されている情報の知識的切出しを行なっているとする軸。

(4) 時間軸

概念スキームの時間的な変化と吸収しようとするために設けられるビューの概念を説明するための軸。

これらの諸軸を用いて整理されたビューの型は次の通りである。

- (A) スキーム変換型ビュー
- (B) DML変換型ビュー
- (C) データモデル変換型ビュー
- (D) 抽象化型ビュー
- (E) 知識型ビュー
- (F) 時変型ビュー

勿論これらビュー定義概念が幾つか混合して新たなビューが定義される。本稿以下では(A)~(F)についてより詳しい説明を行う。

A. スキーム変換型ビュー

このビューは概念スキーマと同一データモデルであるが、SSDにおいて異なる。INGRESやSYSTEMRがサポートしようとしたビューがこの典型例である。INGRES⁽⁴⁾はQUELの構文を使って次のようにビューを定義している。

```
RANGE OF X IS ED
RANGE OF Y IS DM
DEFINE EDM (EMP = X.EMP, DEPT = X.DEPT, MGR = Y.MGR)
WHERE X.DEPT = Y.DEPT
```

ここでED(EMP, DEPT), DM(DEPT, MGR)は基本関係又は既に定義されているビューであり、この二つのDEPT上の自然結合としてビューEDMが定義された。通常このタイプのビューはビューを定義するのに次の二つのタイプの演算が使用される。

- (1) 関係演算 (関係代数の用語では直積, 和, 差, インタセクション, 射影, 制限, (自然結合, 割り算演算等))
- (2) 非関係演算 (関係名, 属性名の変換, ドルを用いた変換する形式ドメインの変換演算, 更に複雑性によって平均, 最大, 最小等のアグリゲート関数とドメイン間の複雑な計算関数等)。

この型のビューについては関係データモデルが世にでてしばらくしてからいそぐのサポート可能性が議論されることになった⁽⁵⁾。最も単純なビュー概念と考えられるこの型のビューについては問題は極めて複雑でビューサポート機構の設計は極めて困難である。更に問題のデータ意味論とも深くかかわっていることが明らかになっている。ビューサポートとはこの様なものだということを格好の例を挙げる⁽⁶⁾。関係CP(CHILD, PARENT)を親子関係を表わしている関係とする。親ある…の子供が死亡すれば両者間に関係CPは消滅する。このとき子供と祖父母の関係を表わすビューCGを次の様に定義する。

```
RANGE OF X IS CP
RANGE OF Y IS CP
DEFINE CG (CHILD = X.CHILD, GRANDPARENT = Y.PARENT)
WHERE X.PARENT = Y.CHILD
```

いま $(a, b), (b, c) \in CP$ であったとしよう。ビューCGの定義から $(a, c) \in CG$ である。更にbの死亡による (a, b) 及び (b, c) のCPからの消滅は自動的にCGから (a, c) の消滅を誘引する。これはCGがCPとCPの自然結合の射影として定義されるビューであるからである。もしCGが本当に子供と祖父母の関係を反映しているならばaとcがその関係にあることとaの親bの生存か死亡には無関係の筈である。従ってビューCGでもって子供と祖父母の関係が維持できると言っているのはいけぬ。CGでは子供によって親が生存しているという制限付の子供と祖父母の関係が維持されているということなのである。問題がそれ程簡単であることは全く同様の形式で定義されているビューEMを考えてみるとより鮮明になる。

```
RANGE OF X IS ED
RANGE OF Y IS DM
DEFINE EM (EMP = X.EMP, MGR = Y.MGR)
WHERE X.DEPT = Y.DEPT
```

二のとはは、Cのとはと全く互反対は、(e, m)がEMの元のととは、従業員eとマネージャmとを結び付けた部署が消滅することにより(e, m)がEMから消滅すると考へるのは極めて自然である。このようにスキーム変換型ビューサポートの問題は深くデータ意味論とかがわっている。尚スキーム変換型と名付けた理由は一般にビューは概念スキーマの関係の枠組(スキーム)を変換したものとみなすことによった。CODASYLのネットワークデータモデルに準拠したデータベースシステムではサブスキーマの概念が概念スキーマとしてのネットワークスキーマからサポートされているビューの概念に相当する。ネットワークモデルの場合モデルにアクセスパスの記述を含んでいるが、これはサブスキーマサポートを複雑化する要因となっている。⁽⁷⁾

B. DML変換型ビュー

例えれば関係データベースシステムのデータ操作言語にはQUEL, SQUEL, SQL(SQL)に代表される関係論理型の言語がある。幾つかの関係代数型言語も設計されている。一般に関係論理型は非手続的であり関係代数型は手続的であり、従って前者は非専門家向き、後者は専門家向きということもできる。インタフェイス中両言語としては関係代数型を指向するかもしれない。あるいはQB Eのようにデータベースは関係モデルによって、が高度に非専門家向きのエンドユーザ言語が開発された例もある。これからのオフィスオートメーションの端末としての指向も十分意識している。これらの各々データ操作言語のサポートは一つの言語を実装し、他を仮想的にサポートするよう機能させることにより実現させる方法が考へられる。これがDML変換型ビューをサポートすることである。RENDREはVOLS⁽⁸⁾のような自然言語インタフェイスのサポートはこのタイプに入る。尚現実にはAとBの概念が混在することが多いだろう。

C. データモデル変換型ビュー

多データモデルデータベースシステムの構想⁽⁹⁾があり、種々の基礎的研究がなされてきている。元来データベースは社会の貴重な資源であり共有されるべきものである。しかしながらデータベースを記述しているデータモデルの差異によりそのデータベースへのアクセスが制限されたり不可能になってしまったのは極めて大きな社会的問題となる。このように、異なるデータモデルを念頭に作成された応用プログラムでも使用したデータベースシステムとで実行できる様システムが構成されていることが望ましい。このことを可能にするのが多データモデルデータベースシステムであり、具体的には関係データベースシステムがネットワークモデルインタフェイスや階層モデルインタフェイスを持つた、ネットワークデータベースシステムが関係モデルインタフェイスや階層モデルインタフェイスを備えることをいう。事態は更に重大で、例えれば関係データベースモデルの批判の一つにリピーティンググループに対応する概念のたりの現実世界を反映しにくくことが指摘されている。この問題の本来的データモデル論で吸収されるべき項目であるが、見方をかえるとビューとしてリピーティンググループを許すデータモデルをサポートすればよいというデータベース管理システムの構想論の問題として捉えられたことが出来る。このアプローチは可成り現実的である。APAD⁽¹⁰⁾の発想もこの範ちゅうに属する。

D. 抽象化型ビュー

データベースシステムは本来情報システムである。人はデータベースシステムを(1)必要なときに、(2)必要な形で、情報を提供してくれる(ニーズを願って)利用する。このような立場に立つと原始データとしてデータベースに格納されている情報はシステムにより必要な加工を施されて利用者に提供、利用されることとなる。例之は図形処理を行なうとして利用者に与えてビット系列としての図形データベースを直接ハンドリングすることは場合によっては大変おそろしい操作となる。それよりも図形を何らかの観測から特徴付けし、特徴量空間をシステムがサポートし、利用者何しかるべきデータ採択言語と共に有効利用することを考えるかもしれない。他の例としては管理情報システムや意思決定支援システムにおいてはオペレーショナルデータはオペレーショナル業務の遂行に必要不可欠であるが、戦略レベルあるいは戦略レベルの業務には直接必要ではなくむしろデータのマップストラクトが外部データと共に要求される。更に戦略レベルでは戦略レベルのデータのものはむしろこれらのマップストラクトで十分かもしれない。つまりデータの利用者間に存在する階層性に対応してデータベース上に階層的な抽象化型ビューサポート機構が生まれ出されるということである。図-3に歯科矯正学診断サポートシステムの開発構想で述べられている抽象化型ビューサポートの具体的な概念を示す。図-4はこのような場合に図-2に示された可換図の概念がどのように拡張されるかを示している。

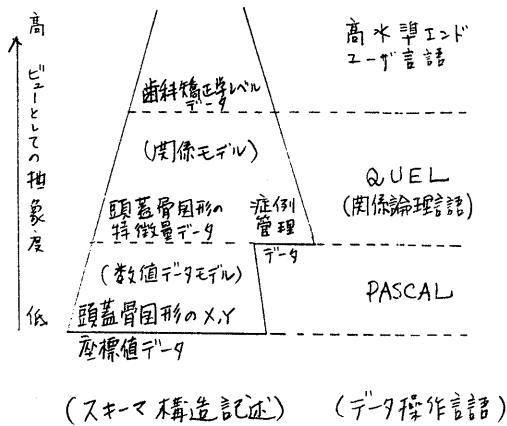


図-3 歯科矯正学診断サポートシステムにおける階層的抽象化型ビューサポート

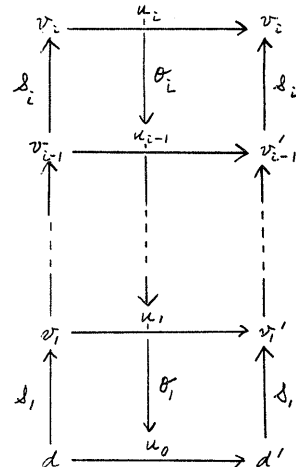


図-4 階層的ビューサポートの可換図

スเปシャル (spatial) データベース管理の概念は一層の抽象化型ビューサポートの概念に相当し、知識システムで高階論理をサポートしてゆく方向の階層的抽象化型ビューをサポートする概念になる。一般に多層の場合外部データのとり込みが各レベルで起る可能性がありビューサポートサブシステムはそのとり扱にも注意しなければならない。

E. 知識型ビュー

知識情報処理システムではデータベースシステムを中核にし、それを推論機構

がとりまき、その外側に利用者の存在する構成と考えると理解し得る。このデータベースシステムと推論機構の組み合わせをどのようにするかはシステム構成上問題となる。データベースに蓄積されている情報を出発点として知識の観点から陽に整理しておくことはシステム構成を簡素化することに貢献すると考えられる。例えば歯科矯正学診断サポートシステムにおいて、データベースから何らかの判断基準により骨格型下顎前突症の症例データのものをビューとして取り出しておくことは後の知識型情報処理の役に立つと考えられる。

F. 時変型ビュー

データベースのライフサイクルを考えると、データベースはその誕生から消滅にいたるまで時間的に構造変化してゆくと考えられるのが自然であり、遂にアプリケーションはその様な変化に対して免疫性がほしい。ライフサイクルの観点からみてアプリケーションの論理的データ独立性を達成したからである。別の側面として、関係モデルに対する批判のうちに関係データベースでは関係従属性等の制約条件の変化により例えばオズ正規形関係の集まりとして定義して下関係スキームに変化を呼び、かえってアプリケーションの論理的データ独立性を損なってしまう恐れのあることが懸念されている⁽¹²⁾。このような問題を解決するためにはデータベース管理システムは時変型ビューサポート機能を持つべきである。この機構はこの型以外のビューサポート機構が概念スキーマは不変でレポートされるビューにバリエーションがあったのに対し、逆に概念スキーマが変化することに対しビューを不変に保つようビューサポート機構が変化するという点で特徴的である。

4. おわりに

仮想化技法としてのデータベースビューの諸概念を述べた。3章A項で若干触れたがビューサポート問題はデータ意味論とかがわり完全解決にはあつかい側面もある。しかしながらデータベースシステムがこれからの情報システムの中核となることを考えると、多種多様な外部インタフェース機能を充実させることが必須で、今後のこの分野の研究開発が待たれる。

[謝辞] 本学野口正一教授、研究会等で有益な御討論を下された皆様に深謝する。尚本研究の一部文部省昭和55年度科学研究費補助金、課題番号568006の協助のもとで行われたことを付記する。

[文献] (1) 金森, 増永他, "頭蓋骨面データベースを用いた歯科...", "電通学会本邦言語研究会資料 AL80-59 (1980.12), (2) C.F. Herot, "Spatial Management of Data," ACM TODS 5, 4 (1980.12), (3) 増永他, "データベースビュー概念の一般化とその応用について," 情報処理学会第22回大会論文 2K-7 (1981.3), (4) M.R. Stonebraker 他, "The Design and Implementation of INGRES," ACM TODS 1, 3 (1976.9), (5) E.F. Codd, "Recent Investigations in Relational Data Base Systems," Information Processing 74 (1974), (6) 増永, "データベースビューについて," 京大数解研研究会 (1981.2) (7) T.W. Olte, "The CODASYL Approach to Data Base Management," (book), John Wiley & Sons, Inc. (1978), (8) E.F. Codd, "Seven Steps to Rendezvous with ...," Proc. IFZP TC-2 WC (1974.4), (9) M.F. King, invited talk in IFZP 80 Tokyo Congress (1980.10), (10) H. Kitagawa & T.L. Kunii, "APAD: Application Adaptable ...," Proc. DB Engineering, IBM Japan (1979, 11), (11) 大須賀, "知識表現のための多層論理," 情報処理学会人工知能・符号言語研究会資料 17-6 (1980.9) (12) 植村, "データベース関係モデルオズ正規形批判," 情報処理学会誌 18-1 (1977.1)