数理解析研究所講究録
第695巻 1989年 135-144

複数の階層に基づくデータベースの設計

九州大学大型計算機センター 古川哲也 (Tetsuya Furukawa)
九州大学工学部 上林彌彦 (Yahiko Kambayashi)

1. まえがき

データベースの構造は、利用者の理解の容易さを考えると階層構造であることが望ましい。オブジェクト指向モデルや非正規（入れ子）関係モデルでは、階層構造を利用者ビューとしており、関係モデルに比べ属性間のつながりが分かりやすいという特徴を持っている。オブジェクト指向モデルの階層構造には、is-a, part-of, a-kind-of, member-of などがあり、属性の様々な継承が行われる。利用者は、これらの構造が混在する階層構造をデータベースのビューとする。

オブジェクト指向データベースに関する研究では、関係データベースのオブジェクト指向的な拡張[1)]や、Smalltalk などのオブジェクト指向プログラミング言語に基づいたデータベースシステムの研究[2)]がなされているが、システムの開発が中心[3)4)]で、オブジェクト指向データモデルに関する形式的な議論が行われていないのが実状である[5)]。

利用者ビューとしての階層構造が複数与えられたとき、データベースはすべて

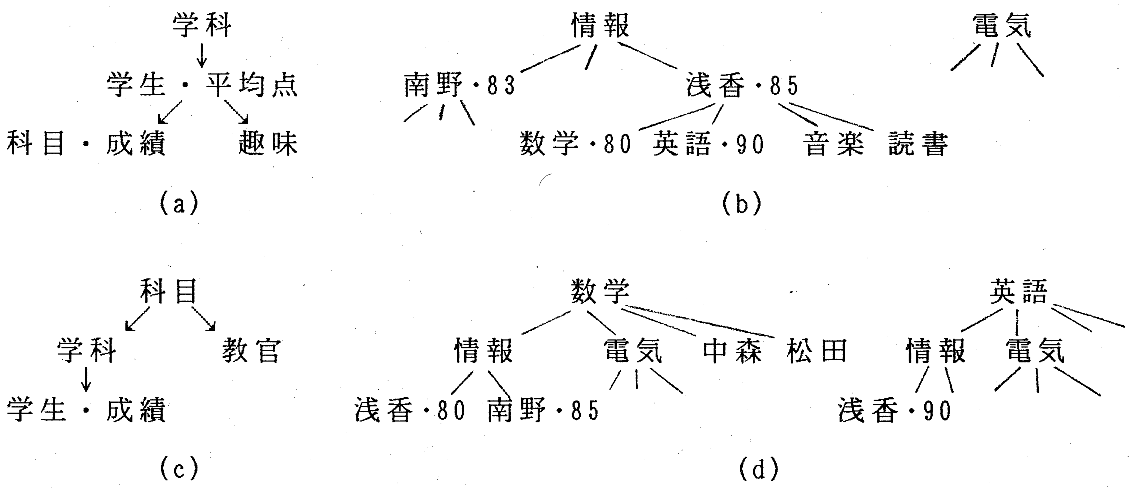

図1 階層構造による利用者ビュー

の構造を表さなければならない。図1は学生の成績に対する利用者ビューの例である。ここで有向枝は1対多の対応を表す。(a) は学科ごとに学生をまとめその成績を見るもの、(c)は科目ごとに成績を見るもので、それぞれ(b),(d)が実現値である。この様にデータベースのビューは一般に複数存在する。

ネットワーク構造を用いることにより、複数の階層を表す冗長性の少ない内部構造を実現できる。属性値の継承はオブジェクト指向モデルの特徴の1つであるが、その値はオブジェクトではなくその祖先に保持される。継承の方式にも幾通りかあり[6)]、データベース設計時の重要な問題となっている。本稿では、継承を含む階層構造の形式化を試み、そのようなネットワーク構造の設計法を示す。

2. オブジェクトによる階層

同じ属性集合からなるオブジェクトをオブジェクトの型としてまとめる。すなわち、オブジェクトはすべて型づけされているものとし、以下型を用いて議論する。メソッドもオブジェクト指向モデルの特徴の1つであるが、本稿では属性として扱う。オブジェクトによる階層構造を、有向木**H**(V_H, E_H)で表す。 ここで、V_Hは節点の集合で各節点はオブジェクト型に対応しており、E_Hは有向枝の集合で各枝は2つのオブジェクト型間のオブジェクトが枝の方向に1対多の対応している（階層がある）ことを表す。図1 (a)(c)は階層構造の例である。

オブジェクト型Cは、次の3種の要素を持つ。

(1) 固有属性：オブジェクトは、それ自身属性を持つ。 オブジェクト型Cのオブジェクトを特徴づける属性集合をCの固有属性といい、Attr(C)で表す。

(2) 保持属性：固有属性の値はオブジェクトが実際に保有せず、その親あるいは祖先から継承されることもある。オブジェクトが値を保持する属性集合を保持属性といい、Keep(C)で表す。

(3) 継承属性：オブジェクト型Cにおける固有属性と保持属性は次の意味を持つ。

Attr(C)-Keep(C)：祖先から継承されなければならない属性。

Keep(C)-Attr(C)：子孫へ継承するための属性。

従って、子孫へ継承可能な属性は、 根からCまでの経路上のオブジェクト型で保持される属性集合である。これを継承属性と呼び、Inh(C)で表す。階層構造でのオブジェクト型は、継承属性により特徴づけられる。

(4) 集約属性：子オブジェクトの属性値などから値が決定される属性で、平均，合計，子の数などがある。集約属性は、階層構造が保存されていれば計算可能なので、以下では議論しない。

(5) 識別子：オブジェクト型Cのオブジェクトを識別する属性を識別子といい、ID(C)で表す。 識別子の値は一意でない場合もある。多対多の関連を１つの階層で表すと、オブジェクトが重複される。このとき子となるオブジェクト型では、同じオブジェクトであれば同じ識別子を持つ。図１(b)の学科や学生は、科目ごとに重複して現れるが、それらは同じ識別子を持っている。データベース中でのオブジェクトの区別は固有属性の値でなされるので、$ID(C) \subseteq Attr(C)$とする。

(6) キー： オブジェクト型Cのオブジェクトを１つ特定するための極小の属性集合をCのキーと呼び、Key(C)で表す。Key(C)は根からCまでのオブジェクト型の識別子の集合の部分集合である。Key(C)＝ID(C)のとき、型Cのオブジェクトに重複はない。

3．　階層構造における従属性制約

オブジェクト階層を関係と対応させ、その性質を関係モデルにおける従属性を用いて議論する。属性集合$U=\{A_1, A_2, \cdots, A_n\}$を全属性集合とし、各属性$A_i$にはその定義域$D_i$が対応付けられているものとする。 U上の関係R(U)は、$t:\{A_1, A_2, \cdots, A_n\} \rightarrow D_1 \times D_2 \times \cdots \times D_n$で示される写像tの有限集合である。tを関係Rの組と呼び、tの属性集合Xの値をt[X]で表す。関係Rの属性集合Xへの射影を$R[X]=\{t[X] \mid t \subseteq R\}$、関係$R_1$と$R_2$の結合を$R_1(X_1) * R_2(X_2) = R(X_1 \cup X_2) = \{t \mid t[X_1] \in R_1, t[X_2] \in R_2\}$とする。関係R(U)での従属性には次のものがある。

[関数従属性] 任意の組t_1, t_2（$\in R$）に対し、$t_1[X]=t_2[X]$ならば$t_1[Y]=t_2[Y]$（$X, Y \subseteq U$）のとき、Rは関数従属性$X \rightarrow Y$を満足するという。

[多値従属性] 任意の組t_1, t_2（$\in R, t_1[X]=t_2[X]$）に対し、組t（$t[X]=t_1[X]$, $t[Y]=t_1[Y], t[U-X-Y]=t_2[U-X-Y]$）が存在するとき、Rは多値従属性$X \twoheadrightarrow Y$を満足するという。 この制約は、Xの１つの値に対応するYの値の集合がU-X-Yとは独立に定まることを意味する。

[結合従属性] $R=R[X_1]*R[X_2]*\cdots*R[X_n]$のとき、Rは結合従属性$*[X_1, X_2, \cdots, X_n]$

を満足するという。n＝2のとき、$X_1 \cap X_2 \twoheadrightarrow X_1 - X_2$で表すことができる。

階層構造**H**を関係と対応づける。**H**の各オブジェクト型のオブジェクトの集合$t = \{o_1, o_2, \cdots, o_m\}$で、$o_i$と$o_j$は異なるオブジェクト型のオブジェクト（$i \neq j$）であり、**H**の構造に従って親子関係があるとき、tを**H**におけるオブジェクトの組と呼ぶ。全属性集合上の関係Rの組が**H**におけるオブジェクトの組と1対1に対応し、各属性値が対応するオブジェクトの組の同じ属性の値と等しいとき、Rを**H**で表されるオブジェクト階層の関係と呼び、R(**H**)で表す。

オブジェクト型Cにおいてオブジェクトを1つ定めると、その継承属性の値が定まる。また、重複するオブジェクトでも、識別子が等しければ固有属性の値も同じでなければならない。従って、R(**H**)は次の関数従属性を表している。

［定理1］　オブジェクト階層**H**の関係R(**H**)は、**H**の各オブジェクト型Cについて、関数従属性：$Key(C) \rightarrow Inh(C)$, $ID(C) \rightarrow Attr(C)$を満足する。□

（証明）　R(**H**)の組t_1, t_2で、$t_1[Key(C)] = t_2[Key(C)]$とする。$t_1, t_2$に対応するオブジェクトの組では、Cのキーの値が等しいことから、Cで同じオブジェクトを持つので、その祖先のオブジェクト集合も等しい。よって、$t_1[Inh(C)] = t_2[Inh(C)]$となる。$t_1[ID(C)] = t_2[ID(C)]$のとき、$t_1, t_2$に対応するオブジェクトの組のCのオブジェクトは同じか、または重複されたものである。よって、固有属性の値は等しく、$t_1[Attr(C)] = t_2[Attr(C)]$である。（証明終り）

階層構造**H**で、オブジェクト型Cの子節点のオブジェクト型を$C_1, C_2, \cdots, C_n$、C_iを根とする**H**の部分木を**H**(C_i)、**H**(C_i)のオブジェクト型で保持される属性集合をKeep(**H**(C_i))とする。Cのオブジェクトを1つ定めると、**H**(C_i)のオブジェクト集合は独立に定まるので、多値従属性を表している。

［定理2］　オブジェクト階層**H**の関係R(**H**)は、**H**の各オブジェクト型Cについて、多値従属性：$Key(C) \twoheadrightarrow Keep(\mathbf{H}(C_i))$を満足する。□

（証明）　t_1, t_2を$t_1[Key(C)] = t_2[Key(C)]$となるR(**H**)の組とする。$t_1, t_2$に対応するオブジェクトの組は、Cのキーの値が等しいことからCで同じオブジェクトoを持つ。t_1のC_iでのオブジェクトをo_iとすると、o_iはoの子となるで、t_2の組で

$\mathbf{H}(C_i)$のオブジェクトをo_iにつながるオブジェクトと置き換えたものも**H**でのオブジェクトの組である。従って、このオブジェクトの組に対応するR(**H**)の組tが存在し、$t[Key(C)]=t_1[Key(C)]$, $t[keep(\mathbf{H}(C_i))]=t_1[keep(\mathbf{H}(C_i))]$, $t[U\text{-}Key(C)\text{-}keep(\mathbf{H}(c_i))]=t_2[U\text{-}Key(C)\text{-}keep(\mathbf{H}(C_i))]$である。（証明終り）

無矛盾な多値従属性の集合は、1つの非巡回結合従属性と等価であることが知られている[7]。階層構造が表す多値従属性集合は無矛盾であり、等価な結合従属性が存在する[8]。

4．階層構造への従属性の反映

実世界のデータが満足する従属性集合をDEPとする。階層構造で実世界のデータの正しい対応を表すには、その表す従属性はDEPから導けなければならない。また、階層構造では表現されていない従属性を反映させることにより、冗長性を削減することができる。

［関数従属性の反映］　階層構造**H**で、根C_1から根以外のオブジェクト型Cまでの経路 $C_1, C_2, \cdots, C_n, C$に対し、$Key(C_k)\rightarrow X$（$\subseteq Keep(C)$）となる最小のkのC_kにXを加え、CからXを除く。□

［定理3］　階層構造**H**とそれに関数従属性を反映させた階層構造**H'**で、R(**H**)＝R(**H'**)である。□

（証明）　**H**と**H'**のオブジェクトの組が1対1に対応し、対応する組での各属性の値が等しいことから明きらか。（証明終り）

関数従属性を反映させると、オブジェクト型の継承属性が変わる。$X\subseteq Keep(C)$のとき、C_i（$k\leq i\leq n$）では継承属性にXが加わるが、$Key(C_i)\rightarrow Key(C_k)$ より$Key(C_i)\rightarrow X$であり、DEPに矛盾しない。その他のオブジェクト型では、継承属性に変化はない。この変換は、オブジェクトに共通な値はその祖先で保持し、値を継承させるという階層構造の利点に合致する。$X=ID(C)$のとき、同様にDEPには矛盾しない。$ID(C)\rightarrow Attr(C)$より$Key(C_k)\rightarrow Attr(C)$なので、$Attr(C)$も$C_k$に加えることができる。この結果、オブジェクト型CのオブジェクトはC_kで保持される。

［例１］　図２(a)の階層構造で、関数従属性：C→Aが成り立つならば、この階層は図２(b) の階層構造でも表現できる。これは、Cの各オブジェクトに対してAの値は一意に定まるためである。図２(b) では、オブジェクト型ACのキー属性がCなので、C→Aを構造に反映しており、(a) に比べCの１つのオブジェクトに対するBのオブジェクトの数だけAの重複が減少している。□

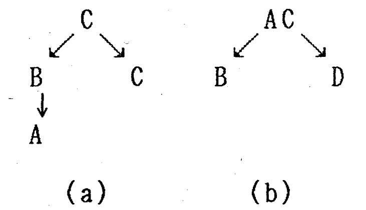

図２　関数従属性の反映

多値従属性は、データが部分的に直積に対応していることを表しており、それを反映することで冗長性を減らすことができる。

［多値従属性の反映］　階層構造**H**でオブジェクト型Cの親をC_0とする。$Key(C_0) \twoheadrightarrow X$ （$\subseteq Attr(C)$）のとき、オブジェクトCを分解し、C_1（$Attr(C_1)=X$），C_2（$Attr(C_2)=Attr(C)-X$）とする。Cの子孫はそのままC_2の子孫とするが、子孫のオブジェクト型で $Attr(C_M)\cap X\neq\emptyset$となるものは、$C_M'$（$Attr(C_M')=Attr(C_M)$）に変更する。 C, C_Mのオブジェクトは、 C_0の１つのオブジェクトに対応するC_1とC_2, C_M'のオブジェクトの集合の直積として求められる。□

［定理４］　階層構造**H**とそれに多値従属性を反映させた階層構造**H**'で、R(**H**)＝R(**H**')である。□

（証明）　Cのオブジェクトは、 C_0を介するC_1とC_2のオブジェクトの組合せに１対１に対応する。 C_MのオブジェクトもC_M'とC_1のC_0を介するオブジェクトの対応に１対１に対応する。 従って、**H**のオブジェクトの組と**H**'のオブジェクトの組は１対１に対応し、対応する組での各属性の値は等しい。（証明終り）

［例２］　図３(a) の階層構造で、多値従属性：A↠Bが常に満足されるものとする。オブジェクト型BCは分解され図２(b)とすることができる。 (a)でのAの１つのオブジェクトに対応するオブジェクト型BCのオブジェクトは、(b)でのAの１つのオブジェクトに対応するオブジェクト型B, C のオブジェクト集合の直積とし

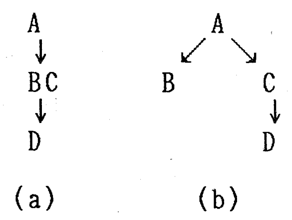

図３　多値従属性の反映

て求めることができる。□

5. 階層構造の集合を表現するネットワーク構造

ネットワーク構造は、根付き有向木でなく一般の有向グラフ**N**(V_N, E_N)で表される。V_Nは同じ属性からなる要素の集合を表す節点の集合、E_Nは2つの集合間の1対多の親子関係を表す。節点vの要素を唯一定める極小の属性集合をvのキーといい、Key(v)で表す。**N**の連結な部分グラフ**N**'で表されるネットワーク構造を**N**の部分ネットワーク構造という。**N**'の要素の対応を唯一定める極小の属性集合を**N**'のキーといいKey(**N**')で、**N**'に含まれる属性集合をAttr(**N**')で表す。

階層構造**H**とネットワーク構造**N**で、**N**において各オブジェクトの固有属性の値とオブジェクト間の階層が求められるとき、**H**は**N**で表現されているという。オブジェクトは、多値従属性の反映と同様に、分解されている場合も表現可能であるとする。この性質は、次のように再帰的に記述できる。

［性質1］ 階層構造**H**とネットワーク構造**N**に対し、

Step 1：**H**の根オブジェクト型C_1で、Key(C_1)（＝ID(C_1)）＝Key($\mathbf{N}_1$), Attr(C_1)$\subseteq$Attr($\mathbf{N}_1$)となる**N**の部分ネットワーク構造が存在する。

Step i：**H**のオブジェクト型C_iで、Key(C_i)＝Key($\mathbf{N}_i$), Attr(C_i)$\subseteq$Attr($\mathbf{N}_i$)となり、C_iの親型C_j($j<i$)に対応する部分ネットワーク構造を含む**N**の部分ネットワーク構造が存在する。□

［定理5］ 階層構造**H**に対し、性質1を満たすネットワーク構造**N**は**H**を表現する。□

（証明） 各オブジェクト型Cに対し、Key(C)$\subseteq$Key(**N**'), Attr(C)$\subseteq$Attr(**N**')となる**N**の部分ネットワーク構造**N**'が存在することから、**N**'での要素のつながりがCのオブジェクトに1対1に対応しており、**N**'でのつながりを求めることでCのオブジェクトの固有属性の値が得られる。また、C_jがC_iの親であれば、対応する部分ネットワーク構造$\mathbf{N}_j$, $\mathbf{N}_i$で、$\mathbf{N}_j$のつながりを含む$\mathbf{N}_i$のつながりで、C_jのオブジェクトに対応するC_iのオブジェクトが得られる。（証明終り）

階層構造の集合を表現するネットワーク構造は次のようにして求めることがで

きる。

Algorithm1：オブジェクト階層の集合を表現するネットワーク構造の設計

1. 各オブジェクト階層のオブジェクト型Cの継承属性Inh(C)の集合をSとする。
2. Sを含み共通集合演算で閉じた最小の集合S^{i+}を求める
3. S^{i+}の各要素Xに対し、節点Xを作る。
4. 節点X_1, X_2（$X_1 \subseteq X_2$）で、$X_1 \subseteq X_3 \subseteq X_2$となる節点$X_3$が存在しなければ有向枝$\langle X_1, X_2 \rangle$を作る。
5. 親と共通の属性は継承するものとし、節点から除く。

［定理6］ Algorithm1の結果のネットワーク構造は、与えられたオブジェクト階層の集合をすべて表現する。□

（証明） Step1よりオブジェクト型Cに対し、Inh(C)を表す節点vが存在する。Step5で除かれた属性を持つvの祖先の節点を含むようにInh(C) に対応する部分ネットワーク構造**N'**をとることができ、**N'**の節点はすべてvの祖先なので、Key(**N'**)＝Key(C)である。また、C_jがC_iの親であれば$Inh(C_j) \subseteq Inh(C_i)$であり、対応するネットワーク構造$\mathbf{N}_j, \mathbf{N}_i$では、$Attr(\mathbf{N}_j) \subseteq Attr(\mathbf{N}_i)$となる。 定理5よりこのようなネットワーク構造は階層構造を表現する。（証明終り）

Algorithm1の結果のネットワーク構造は、オブジェクト階層を表現する。 関数従属性集合や多値従属性集合（結合従属性）を反映する方法は、次のものがある。

(1) 各オブジェクト階層に4節の方法を用いて従属性を反映し、その結果に対してAlgorithm1を適用する。

(2) 従属性もオブジェクト階層とみなしてAlgorithm1を用いる。 関数従属性：$X \rightarrow Y$に対しては、オブジェクト型をX,Y とし、Yを親、Xを子とする階層を作る。$X \twoheadrightarrow Y$に対しては、オブジェクト型をX,Y とし、Xを親、Yを子とする階層を作る。

(3) Algorithm1を修正し、節点、枝の決定時に従属性を考慮する。

(1) では、従属性制約が用いられるのは従属性がオブジェクト階層に反映できるときのみで、そうでない従属性制約はネットワーク構造に反映されない。(2)

ではＳの要素数が増え、Step２，３での中間結果が大きくなる。 これらの結果の節点は、ネットワーク構造の節点となるわけではなく、次の操作が必要である。

1．与えられた関数従属性を用いて各節点のキーを決定し、キーが同じ節点は併合する。
2．冗長な節点（多値従属性により分解された節点に対応する節点）を除く。

(3) は、上の２つの操作を考慮して、共通集合演算で閉じた集合を求める前に節点の併合や冗長な節点の削除をするものである（Algorithm２）。

Algorithm２：従属性制約を反映し、オブジェクト階層の集合を表現するネットワーク構造の設計

1．各オブジェクト階層のオブジェクト型Cの継承属性Inh(C)の集合をＳとする。
2．結合従属性の要素の集合をＪとする。
3．Ｓの要素で、Ｊの要素の和集合となるものを除く。この操作は、多値従属性による分解に対応する。
4．各関数従属性：$X \rightarrow Y$に対し、Y, XYを要素とする集合Ｆを求める。
5．ＳにＪとＦを加える。
6．関数従属性を用いてＳの各要素のキーを決定し、キーが同じ要素を併合する。
7．Ｓを含み共通集合演算で閉じた最小の集合S^{i+}を求める
8．S^{i+}の各要素Xに対し、節点Xを作る。
9．節点X_1, X_2（$X_1 \subseteq X_2$）で、$X_1 \subseteq X_3 \subseteq X_2$となる節点$X_3$が存在しなければ有向枝$\langle X_1, X_2 \rangle$を作る。

［定理７］ Algorithm２の結果のネットワーク構造は、与えられたオブジェクト階層の集合をすべて表現し、表現可能な従属性をすべて表現している。□

（証明） Step１，３よりオブジェクト型Cに対しInh(C)を表す節点集合が存在するので、定理６の証明と同様にすべての階層構造を表現する。また、Step４で関数従属性に対応する節点を作っており、Step４では節点は多値従属性により分解されている。これらを用いると従属性を反映した階層構造となる。（証明終り）

［例３］ 図２(a)と図３(a)を与えられた階層構造としてこれらを表現するネットワーク構造を求める。(1)や(2)の方法では、図４(a)でのABCのように、AB, AC

に分解されたオブジェクト型が中間結果にできるが、(3)の方法はそのようなものをStep3 で除いている。設計の結果は図4(b)である。ここで、下線はキー属性、括弧は継承される属性である。□

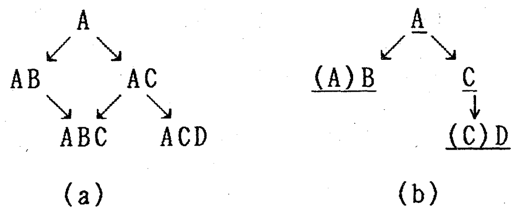

図4　階層構造の集合を表現するネットワーク構造

6. むすび

ネットワーク構造は、複数の階層利用者ビューを持つデータベースを実現できる。ネットワーク構造では、階層方向と逆の方向の親子関係も存在するため、質問処理効率を考えて継承させる属性を定めなければならない。

参考文献

1) Stonebraker, M., “Object Management in POSTGRES Using Procedures,” Proc. Int. Workshop on Object-Oriented Database Systems, pp.66-72, Sept. 1986.

2) Maier, D. and Stein, J., “Development and Implementation of Object-Oriented DBMS,” in Research Directions in Object-Oriented Programming, The MIT Press, 355-392, 1987.

3) Andrews, T. and Harris, C., “Combining Language and Database Advances in an Object-Oriented Development Environment,” Proc. OOPSLA '87, pp.430-440, 1987.

4) Lecluse, C., Richard, P., and Velez, F., “O_2, an Object-Oriented Data Model,” Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.424-433, June 1988.

5) Bancilhon, F., “Object-Oriented Database Systems,” Proc. ACM Symp. on Principles of Database Syst., pp.152-162, March 1988.

6) 佐藤英人, “統計のデータベースと知識ベース”, 情報処理学会アドバンストデータベースシステムシンポジウム, pp.71-80, 昭和62年12月.

7) Beeri, C., Faigin, R., Maier, D., Mendelzon, A., Ullman, J.D., and Yannakakis, M., “Properties of Acyclic Database Schemes,” ACM Symp. on Theorey of Computing, pp.355-362, May 1981.

8) Lien, Y.E., “Hierarchical Schemata for Relational Databases,” ACM Trans. Database Syst., Vol.6, No.1, pp.48-69, March 1981.