講 座

——データベースの基礎理論（5）——

関係データベースにおける質問処理†

上 林 弥 彦††

1. まえがき

関係データベースは自由度が高いかわりに処理効率が低いという問題があり，質問処理の効率を向上させることは非常に重要である．本稿では，関係データベースの質問処理について重要と思われる話題を紹介している．

第2章では質問処理に関する基礎的事項をまとめている．**2.1** はブール質問の処理，**2.2** では関係代数質問の処理について基本的な考えを示している．これらの処理はデータベースシステムを構成する上で非常に重要なものである．**2.3** では質問処理の見通しを良くするための質問のグラフ表現の方法と重要な質問の部分クラスについてまとめている．グラフ表現として，関係代数処理の順序を表現する質問処理グラフ，制約と結合操作を表現する質問グラフおよび自然結合のためのハイパーグラフをあげている．質問の部分クラスとしては，SPJ 質問，SJ 質問，NJ 質問，木質問，巡回質問をあげている．SPJ 質問は第4章で述べるように論理積型関係論理質問とも対応している．

第3章では，質問処理のコストが最もかかる結合操作に対する処理について示している．**3.1** は INGRES で用いられた質問分解法，**3.2** はデータベースシステムで用いられる基本的方法である，入れ子走査法，併合走査法，多重走査法，単純 TID 法および一般化アクセスパスについてまとめている．**3.3** では，分散データベースの質問処理で重要な準結合について紹介する．

第4章は，Aho, Ullman らによって導入された質問表についてまとめている．**4.1** では，SPJ 質問に対応する質問表の等価性や最適化について述べている．**4.2** では，従属性を考慮した質問の等価性や，従属性の導出判定に用いられる質問表上の追跡法についてまとめている．

質問処理のコスト評価には種々の要因を考えなければならない．これらをすべて考えることは実際的でないが，システムのモデルによってコストに一番影響する要因が異なるので，それを考慮した質問処理が重要である．一般には，個々の処理に要する計算時間の総和や必要記憶容量を減少させることが重要であるが，小さなシステムでは必要記憶容量の最大値，二次記憶を考えた場合は主記憶へのデータ転送回数，分散システムではデータ通信量を減少させることが重要となる．本稿では，いくつかの要因のからんだ場合については扱っていない．

2. 質問処理の基礎

2.1 ブール質問の処理

文献検索では，キーワードの存在に対するブール式によって質問を表現することが多い．たとえば，変数 x_1, x_2, x_3 を，それぞれ3つのキーワード“関数モデル”，“分散データベース”，“ユーザインタフェース”に対応させた場合，関数モデルでかつ，分散データベースまたはユーザインタフェースをキーワードとするような論文を求めるための質問は次のように表わせる．

$$x_1(x_2+x_3)$$

ブール質問は，このように，異なる性質に対応する変数 $x_i (i=1, \cdots, n)$ のブール式で表わされる質問で，この性質を持つレコード（上の例の論文に対応）集合を求めるためのものである．以下では，ある仮定のもとで，このようなブール式の処理を最小時間で実現する問題を扱う．

t_i と p_i を次のように定める．

t_i：x_i に対応する性質を持つレコード集合を求めるのに要する時間

† Query Processing in Relational Databases by Yahiko KAMBAYASHI (Department of Information Science, Kyoto University).

†† 京都大学工学部

p_i : $x_i=1$ となるレコードの存在確率

Hanani は，各 x_i が独立であり，1つずつ処理されてゆくという仮定のもとで，最適の処理順を求める方法を求めている.

簡単のため，質問が x_1x_2 で，$t_1=20$, $t_2=30$, $p_1=0.5$, $p_2=0.1$ とする．まず x_1 を評価し次に x_2 を評価する場合を考える．x_1 を評価するのに t_1 だけの時間がかかる．$x_1=1$ となるレコードがなければ x_2 を評価しなくてもよい．したがって x_2 の評価には，p_1x_2 だけの時間がかかるとみなせる．平均処理時間は次のようになる.

$$t_1+p_1t_2=20+0.5\times30=35$$

同様にして，x_2 を先に評価した場合は次のようになる.

$$t_2+p_2t_1=30+0.1\times20=32$$

この例の場合は，処理時間はかかるが存在確率の低い x_2 を先に処理する方が全体の処理時間がへる.

一般に，$x_1x_2\cdots x_n$ という形の**論理積質問**では，x_1, x_2, … という順に評価した場合の平均処理時間は次のようになる.

$$t_1+p_1t_2+p_1p_2t_3+\cdots+p_1p_2\cdots p_{n-1}t_n \qquad (\text{i})$$

この式は次の性質が満たされる場合に最小となる.

$$t_1/q_1\leqq t_2/q_2\leqq\cdots\leqq t_n/q_n \qquad (\text{ii})$$

ここで，$q_i=1-p_i$ である.

この証明は，任意の処理順がこの方法よりも時間が短くならないという形で与えられる．x_k と x_{k+1} の処理順序を変えると

$$p_1\cdots p_{k-1}t_k+p_1\cdots p_{k-1}p_kt_{k+1} \qquad (\text{iii})$$

の部分が次の式で置き換えられる.

$$p_1\cdots p_{k-1}t_{k+1}+p_1\cdots p_{k-1}p_{k+1}t_k \qquad (\text{iv})$$

(iii)-(iv) がこの差にあたり，減少することはない.

$$p_1\cdots p_{k-1}(t_{k+1}+p_{k+1}t_k-t_k-p_kt_{k+1})$$
$$=p_1\cdots p_{k-1}q_kq_{k+1}(t_{k+1}/q_{k+1}-t_k/q_k)\geqq0$$

どの処理順序も，このように隣りどおしが(ii)の条件を満足する部分に互換を繰り返す形で得られるため，(ii)を満足する場合が最小であるといえる.

$x_1+x_2+\cdots+x_n$ という**論理和質問**では，平均処理時間は次のようになる.

$$t_1+q_1t_2+q_1q_2t_3+\cdots+q_1q_2\cdots q_{n-1}t_n \qquad (\text{v})$$

上記と同様にこの式は次の性質が満足される場合に最小となる.

$$t_1/p_1\leqq t_2/p_2\leqq\cdots\leqq t_n/p_n \qquad (\text{vi})$$

上記の論理積質問や論理和質問で答となるレコードの存在する確率は，それぞれ，(vii)(viii)で与えられる.

$$p_1p_2\cdots p_n \qquad (\text{vii})$$

$$1-q_1q_2\cdots q_n \qquad (\text{viii})$$

論理積と論理和のまざったブール質問に対しては，論理積部分や論理和部分を評価するための時間や答の存在する確率を上記の方法を用いて求め，その結果を組合せてゆけばよい.

否定の入っている場合は，ドモルガンの法則により変数のみに否定が作用する形に変形し，x_i の評価のかわりに $\bar{x}_i$ を評価するようにするとよい．この場合，時間 t_i は同じで，p_i は q_i に置き換えられると考えてよい.

変数の値として，未定義 (undefined) を入れても，ほぼ同じように扱えることが Laird によって示されている[2].

同じ変数が2回以上表われる場合は評価は一度ですむので，上記の方法を変更する必要がある．さらに評価が一度で済むような変数集合を与えた場合には，上記の方法は使えなくなるので，Gudes と Hoffman はこの場合に対する発見的方法を与えている[3].

2.2 関係代数質問の処理

関係データベースシステムにおける代表的な質問処理方式は次のようなものである.

あるデータベース言語で表わされた質問

⟶関係代数による表現

⟶関係代数表現の処理

質問を関係代数による表現に変換する方法は，前回の関係完備性の項で述べた対応を用いるとよい．ここでは，Smith と Chang による関係代数表現の最適化の方法について，例を用いて説明する[4].

［例1］ 次の3つの関係を考える.

所属 (先生，学科)

講義 (先生，科目，室)

成績 (学生，学科，科目，点)

意味は自明であるので省略する．**図-1** に関係の例を示しておく．ここで次の質問を考える.

「情報工学科の先生の教えている科目で95点以上とった学生のいるような学科をすべて求めよ.」

この質問の関係代数表現を，分り易く分解して示すと次のようになる.

a. R_1=所属［先生＝先生］講義［科目＝科目］成績

$R_2=R_1$［所属.学科＝"情報工学"］［点≧95］

$R_3=R_2$［学科］

この表現では，まず3つの関係を結合して，R_1 を得

所属

先生	学科
⋮	⋮
坂井	情報工学
矢島	情報工学
萩原	情報工学
⋮	⋮

講義

先生	科目	室
坂井	パターン認識	101
坂井	音声情報処理	102
矢島	論理回路	101
矢島	符号理論	101
萩原	マイクロプログラム	102
⋮	⋮	⋮

成績

学生	学科	科目	点
山田	土木	パターン認識	97
吉田	電気	符号理論	98
金出	電子	パターン認識	100
宮田	情報	マイクロプログラム	97
⋮	⋮	⋮	⋮

図-1 例1の関係例

たあと，条件を満足する組集合を取り出して R_2 を得，さらに出力として必要な属性（学科）へ射影して解 R_3 を得ている．同じ解を得るのに次の方法も考えられる．

b．S_1＝所属［学科＝“情報工学”］
　S_2＝S_1［先生］
　S_3＝講義［先生，科目］
　S_4＝成績［点≧95］
　S_5＝S_4［学科，科目］
　S_6＝S_2［先生＝先生］S_3［科目＝科目］S_5
　S_7＝S_6［学科］

この方法では，結合する前に各関係の処理を行っているので，aの方法と比べて次の理由により処理速度を向上させることができる．

（1） 各関係を並行に処理できる．

（2） 結合によって生じる関係（aの R_1 とbの S_6）は，bの方が小さくなるため，処理に要する計算時間や記憶容量が少なくてすむ．

aの方法で表現する方が容易なので，これを処理コストの小さいbのような表現に変換する方法を考える．

質問は，基本的に次のような形式で表現できる．

［質問の関係代数表現のための基本的方法］

（1） 与えられた質問は，集合演算を内部に含まない形の部分質問に対して集合演算を適用した形で表現できる．各部分質問は（2）以下の方法で表現する．

（2） 質問に使われている関係を結合し，解が（3）（4）の操作で得られるようにする．

（3） （2）の結果の関係に制約操作を適用する．

（4） （3）の結果の関係を解に必要な属性集合へ射影する．

これに対して，処理コストの低い関係代数表現は次のようになる．

［処理コストの低い関係代数表現］

（1） 上記（1）と同じ．

（2） 質問に使われている各関係で実行できる制約の処理を行う．

（3） 各関係を射影して，結合や解のために使われない属性はすべて取り去る．

（4） 上記で得られた関係を結合する．

（5） 結合結果の射影により解を得る．

関係間にまたがる制約は，（4）で処理できる．

（2）と（3）をまとめて，各関係に対する局所処理とも呼ぶ．分散データベースにおいても，各関係は1つのサイトに記憶されていると仮定すると，（2）（3）は各サイト内の処理に対応し，（4）はサイト間のデータ伝送を要する処理に対応している．すなわち，集中型のデータベースでも分散型のデータベースでも，（4）の処理コストを低くするように局所処理を行うことが重要である．このため，次の2つの規則を使うことができる．

（1） **制約の移動**

$$(R_1[Y_1\theta Y_2]R_2)[X_1X_2\theta' C_1C_2]$$
$$=((R_1[X_1\theta_1' C_1])[Y_1\theta Y_2](R_2[X_2\theta_2' C_2]))$$

ここで，$X_1\subseteq\underline{R}_1$（$R_1$ を構成する属性集合），$X_2\subseteq\underline{R}_2$，$C_1, C_2$ は定数ベクトル，θ_1', θ_2' は θ' を分解した比較演算ベクトルで，それぞれ属性集合 X_1, X_2 に対応するものとする．

（2） **射影の移動**

$$(R_1[Y_1\theta Y_2]R_2)[X_1X_2]$$
$$=((R_1[X_1\cup Y_1])[Y_1\theta Y_2](R_2[X_2\cup Y_2]))[X_1X_2]$$

ここで，$X_1\subseteq\underline{R}_1$，$X_2\subseteq\underline{R}_2$ である．

これらの規則の適用例は次の通りである．

［例2］ 例1と同じ関係を用いる．

制約の移動

　（所属［先生＝先生］講義）［学科＝“情報工学”］
　　＝（所属［学科＝“情報工学”］）［先生＝先生］講義

結合結果に対する制約に使われている属性“学科”は，関係“所属”のみに表われているため，この制約を先に実行でき，したがって上記の変換ができる．

射影の移動

　（講義［科目＝科目］成績）［学科］

＝((講義［科目］)［科目＝科目］(成績［学科，科目］))［学科］

関係“講義”で必要な属性は結合に使われる“科目”のみであり，関係“成績”で必要な属性は“科目”と，結果に必要な“学科”であるため，先にそれぞれの関係をこれらの属性に対して射影し，小さな関係にしてから結合を行っている．

この２つの操作を適用すると，例１a の表現をbの表現に変換できる．すなわち，

（１）　R_2 を得るための制約操作を，個々の関係への制約に変換することにより，S_1 と S_4 を得る．

（２）　S_6 と S_7 の計算に用いられる属性は，先生，科目，学科である．これらの関係を得るため，各関係をこれらの属性に射影して，S_2, S_3, S_5 を得る．

（３）　結果を結合して S_6 を得て，これを学科に射影して解 S_7 を得る．

集合演算や除算に対してもコストを下げるための変換が求められている[4]．

2.3　質問のグラフ表現と部分クラス

質問をグラフで表現すると質問の構造が分り易くなるため，次のような利点がある．

（１）　質問処理方式の見通しがたてやすい．

（２）　質問を利用者に分り易い形で表示するための高水準言語としても有用である．

たとえば，自然結合質問の等価性と，その質問を表現するハイパーグラフの等価性が一致するのは，（１）の例といえる．INGRES で用いられた CUPID というグラフィックス向き言語[6]は（２）の例である．

次下では次の３種のグラフ表現について示す．

（１）　質問処理グラフ表現[4]

（２）　質問グラフ表現[5]

（３）　自然結合質問のためのハイパーグラフ表現

質問処理グラフは，節点が関係および関係代数演算に対応し，有向枝が処理順序に対応しているような有向グラフであり，一般に木型になることが多い．分り易いように対応する演算等によって節点の形を変えており，その対応を**図-2** に示す．**図-3**（a）と（b）は，それぞれ，例１の２つの表現 a と b に対応する質問処理グラフを示している．

質問処理グラフは，質問の処理方法を表現するのに適しているが，このように同じ質問が全く異なる形で示されることが多い．

関係代数質問の中で良く用いられる重要な部分クラスを次に定義する．

（１）　**SPJ 質問**：制約，結合，射影のみよりなる質問

（２）　**SJ 質問**：制約，結合のみよりなる質問

（３）　**NJ 質問**：自然結合のみよりなる質問

これらのクラスは，特定の操作に注目して質問処理の方式を検討する場合にも重要である．また，上記のクラスで，制約 $X\theta C$ を $X=C$ のみに制限したり，

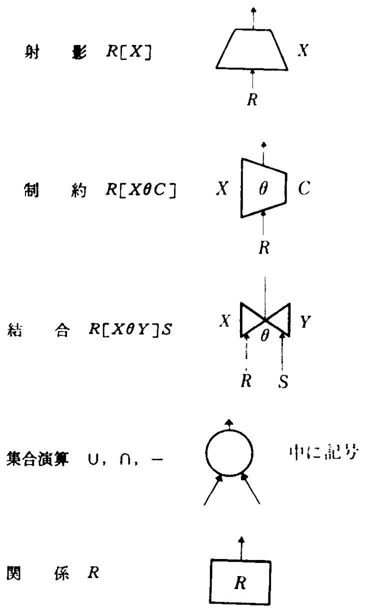

図-2　質問処理グラフの節点

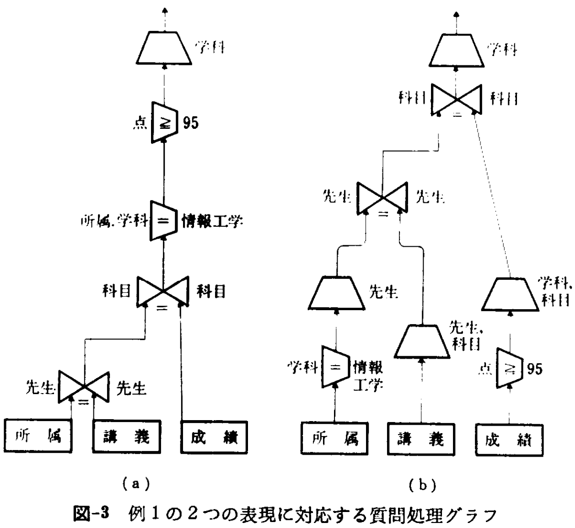

図-3　例１の２つの表現に対応する質問処理グラフ

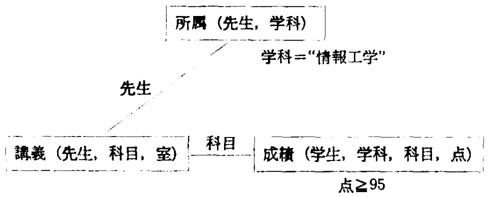

図-4 例1の質問に対する質問グラフ

結合を自然結合のみに制限することも多い．

質問グラフは SJ 質問の表現に適している．各節点は関係に対応しており，制約条件がラベルとして付けられている．結合は枝で示されており，結合の種類がラベルで示される．質問グラフに結果を求める射影条件を付けると SJP 質問の表現ともなる．**図-4** に例1の質問を表現する質問グラフを示す．ここで，目標属性：学科という表現が射影に対応している．

このように，質問グラフによる表現は質問処理グラフによる表現より分かり易いのがふつうである．

NJ 質問は，質問グラフか**ハイパーグラフ**で表現できる．ハイパーグラフの場合，同じ属性名の部分を結合するという条件を付ける必要がある．例1の場合，関係"所属"と"成績"に共通に含まれる"学科"どおしは結合されないので，成績に含まれる方を"学生の学科"というふうに変更する必要がある．制約や射影を無視した例1の質問に対応するハイパーグラフを**図-5**に示す．ハイパーグラフ表現では，属性が節点に対応し，枝が関係に対応している（ハイパーグラフの定義は，本講座（3）を参照のこと）．

質問を構成する結合がすべて自然結合で，その自然結合部分のみに注目してできるハイパーグラフが**非巡回**（定義と判定法は本講座（3）を参照）であるような質問を**木質問**と呼ぶ．これに対して，ハイパーグラフが巡回であるような質問を**巡回質問**と呼ぶ．

例1の関係に対し，次のような質問は巡回質問である．

「先生が教えている科目を受講している学生という

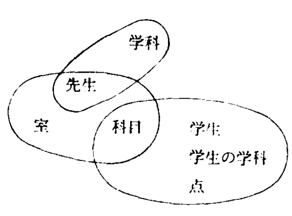

図-5 ハイパーグラフ表現

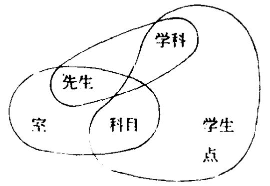

図-6 巡回ハイパーグラフ

条件のもとでの（先生，科目，学生）の組合せに，さらに先生の学科と学生の学科が同じであるという条件を加えた場合に，可能なすべての属性値の組合せを求めよ．」

対応するハイパーグラフは**図-6** に示されている．

この例の場合は質問グラフで表現しても閉路を持つ．しかし，質問グラフが非巡回であるような質問は木質問であるが，質問グラフが閉路を持っても巡回質問になるとは限らない．

図-7 は，3つの属性 A を共有する関係を自然結合する場合の質問グラフを示したもので3つとも等価である．この結果を使うと次のように自明でない等価性がある．

図-8(a)の質問グラフは，R_1 と R_2 を A で自然結合し，R_2 と R_3 を B で自然結合し，R_3 と R_1 を A で自然結合するような質問を表現している．この質問グラフには閉路が存在する．しかし，3つの関係はす

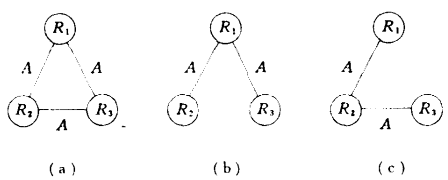

図-7 等価な質問グラフ

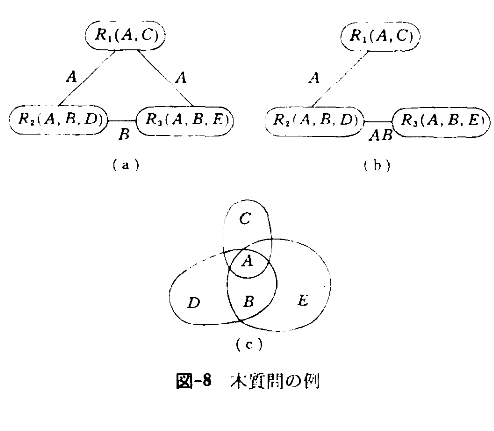

図-8 木質問の例

べて属性Aを持っているため，**図-7**(b)と(c)の等価性を使うと，**図-8**(b)に示す質問グラフが得られ，これは非巡回である．**図-8**(c)には対応するハイパーグラフが示されており，これが非巡回であるのは自明である．

質問が木質問であるための必要十分条件は，次の(a)または(b)である．(a)対応する質問グラフと等価な質問グラフの中に非巡回のものがある．(b)対応するハイパーグラフが非巡回である．

木質問に対しては，結合処理が局所的に実現できるため，効率の良い処理が可能である．

3. 結合処理の実現

関係代数で表現された質問を処理する場合に，最もコストのかかるのは結合（または直積）の処理である．このため，結合処理の効率化はデータベース質問処理の中で重要な研究課題となっている．本節ではいくつかの代表的な方法を紹介する．

3.1 質問分解法[5]

質問分解法は，INGRES のための質問処理方式として，Wong と Youssefi によって開発されたもので，質問を中間結果の必要記憶容量の少ない部分質問に分け，それぞれを順次処理してゆくもので，記憶容量に制限のある場合に適した方法である．基本的な方法は次のとおりである．

（1） 与えられた質問を質問グラフで表現する．

（2） 質問グラフを連結成分に分け，各連結成分に対して以下の処理をする．

（3） 各関係に対する局所的な処理を適用する．

（4） 結合は，質問グラフの枝にそって実行する．この場合，特定の関係を選び，その関係の組を1つずつ取り出して処理をする．

(4)で用いる操作を**組代入**と呼ぶ．原論文では，質問を共通属性が1つしかない2つの質問に分解する操作も基本操作としているが，これは各関係の局所的処理と結合を分けて扱うことに対応している．

質問グラフを用いると3つ以上の関係の結合に対して，あきらかにコストの高い処理の順序を除外できる．図-4 で示された質問グラフでは，3つの関係の結合を行っているが，枝にそって結合すると，所属と講義を結合してから結果を成績と結合するという方法と，成績と講義を結合してから結果を所属に結合するという方法とが得られる．これらはいずれも，所属と成績を結合し（答は2つの関係の直積となる）その結果を講義と結合するよりも，中間結果が減るために優れているといえる．

図-1 と図-4 を用いて，質問分解の適用例を次に示す．

［例3］ まず各関係に制約操作を適用する．その結果，関係“所属”には，図-1 で示される3組のみが残ったとする．関係“講義”には制約操作が適用されないので組数は変化しない．関係“成績”では，点≧95 という制約で求めると20組残ったとする．結果として，関係“所属”が最も小さくなるので，この関係に対して組代入操作を適用する．

まず，（坂井，情報工学）という組を選ぶ．次に関係“講義”で先生＝“坂井”の組を検索する．この場合，（坂井，パターン認識，101）と（坂井，音声情報処理，102）という2つの組がみつかる．これらの組に対して，関係“成績”内の対応する組を求める．次に，所属の2番目の組（矢島，情報工学）に対して同じ操作を繰り返すとよい．

図-4 の質問グラフで，関係“所属”に対して組代入を行うことは，関係“講義”に先生＝“坂井”という制約（先生＝“矢島”，先生＝“萩原”）を付けた質問と等価となる．すなわち，図-4 の質問グラフは，**図-9**(a)に示す質問グラフと等価になる．このように組代入は，質問グラフで組代入の行われた関係を取り去り，その関係と結合している関係に制約条件を追加して得られる質問を，組代入を行った関係の組数だけ集めたものと等価になる．図-4 の質問グラフに対し，関係“講義”に組代入を行うと**図-9**(b)の質問グラフとなる．図-9(a)の場合は，もう一度組代入を行わなければならないが，図-9(b)の場合はもう不要である．すなわち，どの関係から順に組代入してゆくかは，質問処理全体での組代入のコストをできるだけ減

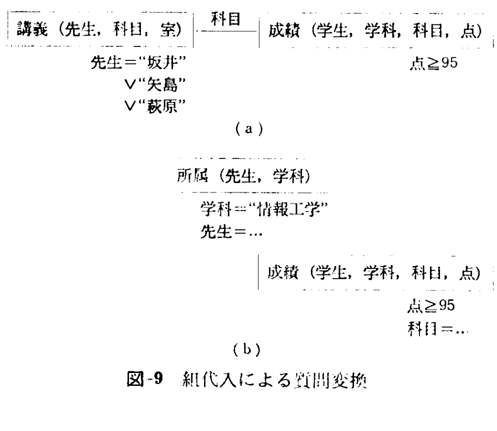

図-9 組代入による質問変換

らすように決める必要がある．上記から分かる条件は次の通りである．

（1） 組数の最も少ない関係を選ぶ（図-9(a)の場合）

（2） 組代入の結果得られる質問グラフの連結成分の数が最も多くなるように選ぶ（図-9(b)の場合）．

3.2 関係の結合の実現のための基本的方法

本節では，関係の結合を行う方法について，いくつかの代表例を示す．

関係はデータベースシステム内では，**組織別子 TID** を付けて記憶されることが多い．図-1 の関係"成績"に TID を付けたものを**図-10**(a)に示す．TID はシステムの処理のため使われるので，利用者は TID を指定して対応する組の内容を検索することはできない．

関係データベース操作の実行効率を上げるには，関係内の組を特定の属性に注目して**ソート**しておくのがよい．たとえば，図-10(a)の関係を学生名で検索することが多ければ，学生名の 50 音順にしておけば，学生名を指定した場合に対応する組集合を求めるのが容易となる．

しかし，この関係を検索するのによく用いられる属性が1つとは限らない場合には，ソートだけではあまりうまくゆかない．たとえば，学科と科目という2つの属性での検索を考えた場合，まず学科でソートして学科名の同じ部分を科目でソートした場合，同じ科目名の組が分散されてしまう．このような場合には，**索引**ないしは**逆ファイル**と呼ばれるものを構成する．**図-10**(b)は，図-10(a)で示した関係の科目に対する索引を示したものである．すなわち，同科目名に対しそれを含む組の TID をまとめて示したものである．この索引を科目でソートしておけば，元の関係をソートしたのと各様の効果が期待できる．しかし，検索結果として求まるのが TID であるため元の関係（これは TID でソートされている）の検索を行わないと最終的な結果は求まらない．

成績

TID	学生	学科	科　目	点
1	山田	土木	パターン認識	97
2	吉田	電気	符号理論	98
3	金出	電子	パターン認識	100
4	富田	情報	マイクロプログラム	97
⋮	⋮	⋮	⋮	⋮

(a)

科　目	TID 集合
パターン認識	1, 3, …
符号理論	2, …

(b)

図-10 組織別子 TID と索引

索引は，制約操作の実現に特に有効である．たとえば，「土木工学の学生でパターン認識を受講した者を求めよ」という質問は次のように処理できる．

（i） 学科に対する索引を用いて，学科名が土木であるような組に対応する TID 集合を求める．

（ii） 科目に対する索引を用いて，科目名がパターン認識であるような組に対応する TID 集合を求める．

（iii） 上記の2つの TID 集合の共通集合は求める結果に対応している．

（iv） 上記(iii)の集合に属する TID を1つずつ取り出して元の関係の対応する組を求め，この組の内容から学生名を求める．

関係の結合を行うための基本的な4つの方法について，Blasgen と Eswaran らが比較検討している[7),8)]．

（1） **入れ子走査法**：これは前節の組代入と似た方法で，2つの関係 R と S を結合する場合に，Rより1組ずつ取ってきて，結合できるSの組を探す方法である．結合属性に対する索引が用意されていることが仮定されている．

（2） **併合走査法**：関係 R と関係 S とを共に結合属性でソートする．次にこの2つのソート系列を併合して1つのソート系列を作る操作を行うと，RとSの結合属性の等しい組を検出できる．自然結合の結果は直接求まるが，一般の θ-結合では後処理も必要である．

（3） **多重走査法**：Sが結合属性値でソートされている場合（ないしは索引のある場合），主記憶内にバッファを用意して，結合属性値の小さい順にバッファが一杯になるまで S の組を入れ，R の組を1つずつ取り出して，バッファ内のSの組と結合できるか調べる．Rの組をすべて調べ終ると，バッファに結合属性値が次に小さいSの組を入れ同じ操作を繰り返す．この操作を，Sの組がすべてバッファに入るまで繰り返すと結合が実現できる．Sが結合属性値でソートされていないで索引も存在しない場合，上記と同じようにバッファにSの組を入れるには次のようにする．Sの組を順次走査し，まずバッファを一杯にする．バッファ内の組で結合属性値の値が最も大きいものより，次に走査された組の結合属性値の方が小さければ，前者のかわりに後者を入れる．Sの走査が一回終ると，バ

ッファ内は結合属性の小さい順に選んだ組集合が入ることになる．Rとの結合可能性を調べた後，バッファ内の結合属性値の下限をすでにバッファに入った組を除外するように決めて，上記の操作を繰り返す．

（4） **単純 TID 法**：R と S の両方とも結合属性に対する索引が用意されていれば，索引を比べることにより，R と S で結合できる組の TID の対が求まる．得られた TID の対の各々に対しRとSを調べることにより結合が実現される．

Blasgen と Eswaran は，入れ子走査方式と併合走査方式が優れているという結論を得ており，これらの方式は System R をはじめとして種々のシステムで使われている．多重走査方式は，S を何度も走査する必要のある場合は効率が悪くなるが，結合前に制約操作により組数が非常に減っていたり，結合属性でソートされているか索引のある場合には有効である．

階層モデルやネットワークモデルでは，結合操作に相当する部分が**リンク**でつながれている．関係モデルでもリンクを使うことは可能であるが，結合に対する自由度が他のモデルよりはるかに高いため，リンクの保持が大変である．このため，一般に索引を用いる方がリンクより良いという結論が出されている[9]．

リンクと等価なものを効率良く実現する方法に，Header の**一般化アクセスパス**[10]がある．これは，2 つ以上の結合可能属性に対する索引をまとめたものである．たとえば，3 つの関係 R, S, T が，属性 A を持つとき，Aに対する索引の TID の部分に 3 つの関係の TID をすべて入れる形式のものである．この索引を用いると結合可能な TID の組合せが容易に求まる．関係転置構造 (RIS)[11]は，TID による方法の持つ欠点である元の関係の参照の必要性を減らし効率を上げるため，TID とハッシュ値を組合せたもので，複数の操作をまとめて実現できる等の特色がある．

データベース内の関係は一般に二次記憶に蓄えられる程大きなものであるので，結合操作の実現に対しても，二次記憶より主記憶へのデータの読み出し回数を最小にすることが重要である．複数個の関係の結合方式については Kim が論じており[12]，2 つの関係について，前処理情報をうまく利用する方法については，Merrett，上林，安浦が論じている[13]．

分散データベースでは，処理コストは通信量で決まると考えてよい．このため通信量を減らすような結合操作の実現が重要となる．このためには，次節で示す準結合が有効である．データベースマシンにおける処理でも準結合が使われている例がある．

3.3 準　結　合

準結合は，分散データベースにおける通信量の削減のため有用であり，木質問の処理に非常に適した方法でもある．

図-11 に非常に簡単な準結合の例を示す．関係RとSとは，分散データベースの異なるサイトにあるものとする．RとSとを結合するための方法としては，どちらか 1 つの関係を他の関係のあるサイトへ送ればよい．準結合を用いると，この場合の処理コストを減らすことができる．

（1） RのサイトよりSのサイトへRの結合属性Aの値集合を送る（$a_1a_2a_3a_4$ が送られてくる）．

（2） S のサイトで，S の結合属性 A の値集合と送られてきた値集合の共通集合を求める（この場合 a_2a_4）．

（3） 上記で得られた集合をRのサイトへ送る．

この結果，両方のサイトで a_2 と a_4 を含む組だけが残される．これは，$R \ast S$ を AB および AC へ射影したものとなっており，質問によってはこれで十分なこともある．$R \ast S$ が本当に必要な場合は，このように小さくした関係を片方のサイトから他方に送ればよい．このようにすると，片方の関係を前処理しないで送るよりデータ伝送量をかなり減らすことができる．この場合の（1）のような操作を準結合とよぶ．

R の S による準結合 $R \ltimes S$ は次のように定義される．

$$R \ltimes S = R \ast S\,[\underline{R} \cap \underline{S}] = (R \ast S)\,[\underline{R}]$$

$R = R_1 \ast R_2 \ast \cdots \ast R_n$ に対し，$R\,[\underline{R}_i]$ をこの結合の R_i に関する部分解という．

上記の方法は，関係が 2 つの場合に限らずもっと一般的な場合にも適用できる．図-4 の例は 3 つの関係が一本につながったものであるため，次のような処理をすればよい（制約操作は前処理として行われている

R

A	B
a_1	b_1
a_2	b_2
a_3	b_3
a_4	b_4

S

A	C
a_2	c_1
a_4	c_2
a_5	c_3
a_6	c_4

$\xrightarrow{a_1a_2a_3a_4}$ 結合属性集合

$\xleftarrow{a_2a_4}$

図-11　準結合の例

と仮定).

（1） 関係“所属”の中の先生名の集合を，関係“講義”のサイトへ送る.

（2） “講義”のサイトで，準結合 講義⋉所属を実行する.

（3） その結果の“講義”に含まれる科目名の集合を，関係“成績”のサイトへ送る.

（4） “成績”のサイトで準結合を実行すると，“成績”に対する部分解が得られる.

（5） 逆に“成績”の部分解に含まれる科目名を“講義”のサイトに送り準結合をすると，“講義”に対する部分解が求まる.

（6） “所属”の“講義”に対する部分解による準結合により，“所属”の部分解が求まる.

この方法は，質問グラフが一本につながった形の質問の処理に拡張できる．さらに，**図-12** のように質問グラフが木型になっている場合も同様である.

（1） 木の葉に相当する関係 R_1, R_2, R_4 より木の根 R_6 に近い関係 R_3, R_5 に対して準結合を適用する.

（2） 得られた R_3, R_5 により R_6 に準結合を適用する.

（3） この結果 R_6 で部分解が得られる.

（4） これを逆に葉の方向に準結合を適用してゆけば，すべての部分解が得られる.

木質問の場合，根として選ぶ節点には任意性があるので，次の条件を配慮して決める必要がある.

（1） 木の高さを高くする（図-12 で R_4 を根にする）と準結合が直列的に実行されるので，データ通信量の総和を減らすことができる.

（2） 木の幅を拡げる（図-12 で R_6 を根にする）と並列処理により全体の処理時間が減らせる.

（3） 質問結果として値を求めたい属性集合との関連で根を決めると，後半の処理の一部を省くことができる.

木質問に対しては必ず木型の質問グラフが存在するため，n 個の関係を対象とした木質問は $2(n-1)$ 回の準結合で（$n-1$ は木の枝数）すべての関係の部分解を求めることができる.

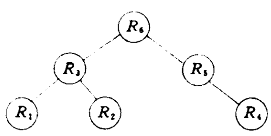

図-12 木 質 問

しかしながら，巡回質問では，準結合を n に比例した回数だけ適用しても部分解の求まらない場合のあることが知られている[14].

基本的な準結合処理に対し，次のような一般化が行われている.

（1） 自然結合だけでなく不等号結合を含む場合は，巡回質問でも，準結合のみで部分解を求めることができる[15].

（2） ローカルネットワークのような放送モードのあるネットワークでは，その性質を生かした準結合処理ができる[18),23].

（3） 準結合とデータ圧縮をうまく組合せると，さらにデータ伝送量を減らすことができ，質問の最適化が単純となる[20].

（4） 巡回質問に対しても準結合の一般化によって処理する方法を求めうる[21].

最近，著者らは，文献[21]の方法をさらに一般化し，従属性の性質も利用した巡回質問処理の方法を示している[22]．この文献で，組代入による質問分解法は文献[21]の方法の特殊な場合に相当することも示されている.

与えられた質問が木質問かどうかの判定は，本講座（3）で示した方法[24]によればよい.

Goodman と Shmueli は，巡回質問の処理はある意味で木質問への変換を含むことを示しており[26]，木質問の処理は本質的なものであると考えられる[25].

4. 質問表と追跡法

4.1 質問の等価性と質問表

例1と同じ関係集合に対し「情報工学科の先生の教えている科目で100点を取った学生のいるような学科をすべて求めよ」という質問を考えてみる．これを，定義域関係論理で表現すると次のようになる.

$$\{u_2 \mid (\exists s_1)(\exists s_2)(\exists t_1)(\exists t_2)(\exists t_3)(\exists u_1)(\exists u_3)(\exists u_4)\ \text{所属}(s_1, s_2) \wedge \text{講義}(t_1, t_2, t_3) \wedge \text{成績}(u_1, u_2, u_3, u_4) \wedge (s_1 = t_1) \wedge (t_2 = u_3) \wedge (s_2 = \text{“情報工学”}) \wedge (u_4 = 100)$$

QBE のように関係の中に条件をまとめて書くと次のようにも書ける.

$$\{a \mid (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)\ \text{所属}(b_1, \text{“情報工学”}) \wedge \text{講義}(b_1, b_2, b_3) \wedge \text{成績}(b_4, a, b_2, 100)\}$$

この質問のように，一般に次のように書けるものを**論理積型関係論理質問**と呼ぶ.

$$\{a_1 \cdots a_k \mid (\exists b_1) \cdots (\exists b_k) R_1(w_n) \wedge \cdots R_n(w_n)\}$$

ここで，w は a, b および c を要素としている．b は定義域変数，c は定数（上記の“情報工学”や 100），a はこのどちらでも良い．

Chandra と Merlin は[27]，このような論理積型関係論理質問に対して，項数が最小の表現が一意に求まることや，質問の最小性の判定問題が NP 完全であることを示した．Aho らの質問表は，この結果を一般化して従属性も扱うようにするために導入された．なお，上記の論理積型関係論理質問は，制約を $R[X=C]$ に制限し，結合を自然結合とした SPJ 質問となる．以下では，このように制限されたものを SPJ 質問と呼ぶことにする．

2つの関係代数質問が，対象とする関係集合の内容がどのように変化しても常に同じ結果となる場合に，**強等価**であるという．対象とする関係集合がすべて1つの普遍関係の射影で得られたという仮定のもとで上記の性質が成立するとき，**弱等価**であるという．

たとえば，$(R_1 * R_2)[R_1]$ は R_1 と等しくなるとは限らないが，普遍関係を仮定すると等しくなるので，この2つの表現は弱等価となる．以下では，弱等価を主に扱うため，これを等価と呼ぶことにする．この場合，属性集合を求めると関係が一意に決まるため，たとえば，属性集合 ABC 上の関係 $R(A, B, C)$ を ABC と略記するものとする．

質問表は，SPJ 質問を表の形で分かり易く示したものである．本節で用いている質問を表現する質問表を**図-13** に示す．

質問表は，上部に一行の**要約行**を持つ表である．表の要素は，**識別変数**（a_i で示される），**非識別変数**（b_i で示される）および**定数**（c_i で示される）である．識別変数は，質問結果に必要な変数を示し要約行に表われると共に同じ列のどこかに表われる．識別変数は各列にたかだか1つしか表われないので i 番目の列は a_i を使うというふうに制限できる（Chandra らの方法ではこの制約はない）．

文献[28]による質問表の構成手順を**図-14** に示す．普遍関係は属性 A, B, C よりなるとする．(a)と(b)は，AB および BC を示す．(c)は $AB*BC$ を示す．(d)は $(AB*BC)[B=1]$ を示したもので，(e)は，$(AB*BC)[B=1][AC]$ を示したものである．制約および射影に対応する質問表上の操作は自明である．自然結合の場合，もとの2つの関係の非識別変数集合どおしは互に素になるように変数名を変更する．

					a	
b_1	情報工学	b_8	b_9	b_{11}	b_{13}	b_{15}
b_1	b_6	b_2	b_3	b_{12}	b_{14}	b_{16}
b_5	b_7	b_2	b_{10}	b_4	a	100

図-13 質問表の例

A	B	C
a_1	a_2	
a_1	a_2	b_1

(a) AB

	a_2	a_3
b_1	a_2	a_3

(b) BC

a_1	a_2	a_3
a_1	a_2	b_1
b_2	a_2	a_3

(c) $AB*BC$

a_1	1	a_3
a_1	1	b_1
b_2	1	a_3

(d) $(AB*BC)[B=1]$

a_1		a_3
a_1	1	b_1
b_2	1	a_3

(e) $(AB*BC)[B=1][AC]$

図-14 質問表の逐次的構成

2つの質問表 T_1 と T_2 に対して，T_1 より T_2 への**保存写像** θ があるというのは，θ が表の要素間を次のように対応づける場合である．

(1) T_1 の任意の行に対して，各要素 i を $\theta(i)$ で置き換えると T_2 に含まれる行となる．

(2) 識別変数 a_i に対して $\theta(a_i)$ は識別変数である．

(3) 定数 c に対しては $\theta(c)=c$ となる．

(4) 非識別変数 b_i に対しては $\theta(b_i)$ については制限がない．

非識別変数は，そのようなものが存在する（$\exists b$）という条件が付いているだけであるため，識別変数になっても定数になってもよい．T_1 より T_2 への保存写像があれば，T_1 の各行の条件は T_2 の行で満足されることになる．したがって，T_1 と T_2 の表現する質問が等価であるための必要十分条件は次の(1)と(2)で与えられる．

(1) 2つの表の要約行が同じである（識別変数の名前の付け変えは許す）．

(2) T_1 から T_2 への制約写像と，T_2 から T_1 への制約写像の2つが存在する．

図-15(a)(b)に示す2つの質問表を考える．行に図に示すような名前を付けると，次のような対応がある．

$r_1 \rightarrow s_1$, $r_2 \rightarrow s_1$, $r_3 \rightarrow s_2$

$s_1 \rightarrow r_2$, $s_2 \rightarrow r_3$

したがって，2つの質問表は等価である．

このような等価性を用いると，与えられた質問の簡

	a_1	a_2	a_3	a_4
r_1	a_1	a_2	b_1	b_2
r_2	a_1	a_2	a_3	b_3
r_3	b_4	a_2	a_3	a_4

(a)

	a_1	a_2	a_3	a_4
s_1	a_1	a_2	a_3	b_1
s_2	b_2	a_2	a_3	a_4

(b)

図-15 等価な質問表

単化ができるが，Chandra らの結果に対応して，質問表の等価性判定問題は NP-完全となる．

単純質問表は，ある列に2回以上同じ非識別変数が現われると，その例に表われる他の変数は1回しか現われないという制限のある質問表である．2回以上同じ非識別変数が現われるのは結合操作に対応しているため，上記の制限は同じ属性による関係の結合に対する制約となっている．この場合には，質問表より冗長な行を除いて最適化でき，等価性判定も多項式時間でできることが示されている[29]．

SPJ 質問に対応する質問表は容易に構成できるが，質問表の中には対応する SPJ 質問を持たないものがある．質問表が等価な SPJ 質問を持つかどうかの判定は NP 完全となることが知られている[33]．しかし，単純質問表に対しては，多項式時間で対応する SPJ 質問を求めることができる[30]．

上記の質問の等価性の考え方は次のような場合に対して一般化されている．

（1） 強等価性に対しては，タグ付き質問表を用いて扱うことができる[28]．

（2） 質問表集合を考えることにより，SPJ 質問の和集合や差集合も扱うことができる[31]．

4.2 質問表上の追跡法

もとの関係が従属性を満足している場合は，これを利用した質問の等価変換が可能である．この結果は，ある結合従属性 JD が与えられた従属性集合から導出できるかどうかという判定に使うことができる．この特別な場合として，関数従属性 FD に対する無損失結合問題が扱える．

FD に対して，次の FD 則を定義する．

FD 則：FD: $X \to Y$ に対し，質問表の X の部分の等しい2つの行があれば，これらの行の Y の部分も等しくさせる操作である．Y の値に c があれば c で，a と b があれば a に置き換える．b_i と b_j の場合は添字の小さい方で置き換える．Y の値に異なる c があるときは矛盾である．

A	B	C	D
a_1	a_2	a_3	a_4
a_1	a_2	b_1	b_2
a_1	b_5	a_3	b_3
b_4	a_2	a_3	a_4

図-16 FD 則の適用

図-16 に示す質問表で表わされる質問を考える．もし $A \to B$ という FD が存在すると，第1行と第2行の A の値が同じであるため FD 則により b_5 が a_2 に置き換えられる．すなわち，図-15(a) の質問表が得られる．これは，先に示したように同図(b)の質問表と等価である．すなわち，図-16 に対応する質問 $AB*AC*BCD$ と，図-15(b)に対応する質問 $ABC*BCD$ は，$A \to B$ という FD が存在する場合は等価である．

この考え方は，FD 集合からある JD が導けるかどうかの判定に用いることができる．

属性集合 $ABCDEFGH$ に対し，JD: $*[ABFG, BC, CDFH, AEH]$ がある FD 集合から導けるかどうかの必要十分条件は，質問 $ABFG*BC*CDFH*AEH$ と $ABCDEFGH$ がこの FD 集合のもとで等価となることである．

このように従属性の導出可能性を調べる場合は，定数は考えなくても良い，射影がないため要約行は省略できるという性質がある．

上記の問題を，次の FD が存在する場合について考察する[34]．

$B \to E$, $C \to E$, $EF \to G$, $G \to ABH$, $AH \to E$,

JD: $*[ABFG, BC, CDFH, AEH]$ に対応する質問表を**図-17**(a) に示す．簡単のため，非識別変数は必要な所以外は省いている．

$B \to E$ により $b_1=b_2$ となる．$C \to E$ により $b_2=b_3$ となる．$EF \to G$ により $b_4=a_7$ となる．次に $G \to ABH$ により，$b_5=a_1$, $b_6=a_2$, $b_7=a_8$ となる．最後に $AH \to E$ を適用すると $b_3=a_5$ となる．この結果を**図-17**(b) に示す．ここで，3行目は識別変数のみよりなっている．このため，この質問表は**図-17**(c) に示す質問表と等価である．これは，質問 $ABCDEFGH$ に対応している．

すなわち，1つの行がすべて識別変数よりなれば，与えられた JD は，FD 集合より導びかれることになる．

与えられた関係集合において，各関係が普遍関係の

A	B	C	D	E	F	G	H
a_1	a_2			b_1	a_6	a_7	b_7
	a_2	a_3		b_2			
b_5	b_6	a_3	a_4	b_3	a_6	b_4	a_8
a_1				a_5			a_8

(a)

A	B	C	D	E	F	G	H
a_1	a_2			a_5	a_6	a_7	a_8
	a_2	a_3		a_5			
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1				a_5			a_8

(b)

A	B	C	D	E	F	G	H
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8

(c)

図-17 FD による無損失結合の判定

射影によって得られている場合，これらを結合して元の普遍関係が得られるとき，この結合を無損失結合という．関係 AB と AC において FD: $A \rightarrow B$ (または $A \rightarrow C$) が成立すれば無損失結合できるが，上記の例は関係間にまたがる FD のある場合は，この無損失結合の組合せだけでは求められない無損失結合のあることを示している[34]．なお，Liu らはより効率の良い無損失性判定法を示している[35]．

より一般化して，ある JD: J_1 が与えられた FD と JD の集合より導けるかの判定は次のようになる[36]．

(1) JD: J_1 に対応する質問表を作る．

(2) これに与えられた FD と JD の集合の要素による FD 則や JD 則 (後述) を適用する．

(3) 上記の結果，識別変数のみからなる行が一つできれば，J_1 は与えられた FD と JD の集合より導けることが判る．

ここで，JD 則は次に示すものである．

JD 則：JD: $*[Y_1, \cdots, Y_m]$ に対し，質問表の行 $w_1, \cdots, w_m$ が存在し，$w[Y_j]=w_j[Y_j]$ $(j=1, \cdots, m)$ を満足する w が構成でき，そのような w が質問表にないとき，wを追加する．

このように，質問表に FD 則や JD 則を適用して変換してゆく方法を追跡法と呼んでいる．

追跡法による従属性の導出可能性検査は単純かつ強力ではあるが，普遍関係を仮定しているため，一般の場合のビューや質問結果の中で成立する従属性判定には使えない．この問題に対しては，Klug[38),39)]，Hull[40)]，著者ら[41)]，伊藤ら[42)]，Casanova[43)] 等の研究が知られている．

参 考 文 献

1) Hanani, M.Z.: An Optimal Evaluation of Boolean Expressions in an Online Query System, CACM 20, pp. 344-347 (1977).

2) Laird, P.D.: Comment on An Optimal Evalution of Boolean Expressions in an Online Query System, CACM 22, pp. 549-550 (1979).

3) Gudes, L. and Hoffman, A.: A Note on An Optimal Evaluation of Boolean Expressions in an Online Query System, CACM 22, pp. 550-553 (1979).

4) Smith, J.M. and Chang, P. Y-T.: Optimizing the Performance of a Relational Algebra Database Interface, CACM 18, pp. 568-579 (1975).

5) Wong, E. and Youssefi: Decomposition-A Strategy for Query Processing. ACM TODS 1, pp. 223-241 (1976).

6) McDonald, N. and Stonebraber, M.: CUPID: The Friendly Query Language Electronics Research Lab Memorandum, University of Califormia, Berkeley, ERC-M 487 (1974).

7) Blasgen, M.W. and Eswaran, K.P.: Storage and Access in Relational Data Bases, IBM Syt. J 16, pp. 363-377 (1977).

8) Selinger, P.G., Astrahan, M.M., Chamberlin, D.D., Lorie, R.A. and Price, T.G.: Access Path Selection in a Relational Database Management System, ACM SIGMOD, pp. 23-34 (1979).

9) Stonebraber, M.: A Comparison of the Use of Links and Secondary Indices in a Relational Data Base System, Proc. Software Eng. Conf., pp. 527-531 (1976).

10) Header, T.: Implementing a Generalized Access Path Structure for a Relational Database System, ACM TODS 3, pp. 285-298 (1978).

11) Le Viet, C., Kambayashi, Y., Tanaka, K. and Yajima, S.: Use of Abstracted Characteristics of Data in Relational Databases, IEEE COMPSAC 3, pp. 409-414 (1979).

12) Kim, W.: A New Way to Compute the Product and Join of Relations, ACM SIGMOD, pp. 179-187 (1980).

13) Merrett, T.H., Kambayashi, Y. and Yasuura, H.: Scheduling of Page-Fetches in Join Operations, VLDB, pp. 488-498 (1981).

14) Bernstein, P.A. and Chiu, D.M.: Using Semi-Joins to Solve Relational Queries, JACM 28, pp. 25-40 (1981).

15) Bernstein, P.A. and Goodman, N.: The Power of Inequality Semijoins, Information Systems 6, pp. 255-265 (1981).

16) Bernstein, P.A. and Goodman, N.: Power of Natural Semijoins, SIAM J. Compt 10, pp

751-771 (1981).
17) Chiu, D. M. and Ho, Y. C.: A Methodology for Interpreting Tree Queries into Optimal Semi-Join Expressions, ACM SIGMOD, pp. 169-178 (1980).
18) Gouda, N. G. and Dayal, U.: Optimal Semi-join Schdules for Query Processing in Local Distributed Database Systems ACM SIGMOD, pp. 164-175 (1981).
19) Hevner, A. R. and Yao, S. B.: Query Processing in Distributed Database Systems, IEEE Trans. SE-5, pp. 177-187 (1979).
20) Kambayashi, Y.: Theory of Databases, Japan Annual Reviews in Electronics, Computer and Telecommunications, Computer Science & Technology 1982, (Ed. by Kitagawa, T.) OHM *North-Holland, pp. 206-226 (1982).
21) Kambayashi, Y., Yoshikawa, M. and Yajima, S.: Query Processing for Distributed Databases Using Generalized Semi-Joins, ACM SIGMOD, pp. 151-160 (1982).
22) Kambayashi, Y. and Yoshikawa, M.: Query Processing Utilizing Dependencies and Horizontal Decomposition, ACM SIGMOD (1983).
23) 滝沢 誠：放送通信機能を用いた分散型データベースシステム (DDBS) の通信処理実現について，情報処理学会，データベースシステム研資 31-3 (昭和57年7月).
24) Yu, C. T. and Ozsoyoglu, M. Z.: An Algorithm for Tree-Query Membership of a bistributed Query, IEEE COMPSAC 3, pp. 306-312 (1979).
25) Goodman, W. and Shmueli: Tree Queries: A Simple Class of Relational Queries, ACM TODS 7, pp. 653-677 (1982).
26) Goodman, N. and Shmueli: The Tree Property is Fundamental for Query Processing, ACM PODS, pp. 40-48 (1982).
27) Chandra, A. K. and Merlin, P. M.: Optimal Implementation of Conjunctive Queries in Relational Databases, ACMSTOC, pp. 77-90 (1976).
28) Aho, A. V., Sagiv, Y. and Ullman, J. D.: Equivalence among Relational Expression, SIAM J. comptg 8, pp. 218-246 (1979).
29) Aho, A. V., Sagiv, Y. and Ullman, J. D.: Efficient Optimization of a Class of Relational Expression, ACM TODS 4 pp. 435-454 (1979).
30) Aho, A. V., Sagiv, Y., Szymanski, G. and Ullman, J. D.: Inferring a Tree from Lowest Common Ancestors with an Application to the Optimization of Relational Expression, SIAM J. Comptg. 10, pp. 405-421 (1981).
31) Sagiv, Y. and Yannakakis: Equivalence among Relational Expression with the Union and Difference Operators, JACM 27, pp. 633-655 (1980).
32) Sagiv, Y.: Optimization of Queries in Relational batabases, UMI Research Press (1981).
33) Yannakakis, M. and Papadimitrion, C.: Algebraic Dependencies, IEEE FOCS, pp. 328-332 (1980).
34) Aho, A. V., Beeri, C. and Ullman, J. D.: The Theory of Joins in Relational Databases, ACM TODS 4, pp. 297-314 (1979).
35) Liu, L. and Demers, A.: An Algorithm for Testing Lossless Join Property in Relational Databases, Inf. Proc. Let. 11, pp. 73-76 (1980).
36) Maier, D., Mendelzon, A. O. and Sagiv, Y.: Testing Implications of Data Dependencies, ACM TODS 4, pp. 455-469 (1979).
37) Maier, D., Sagiv, Y. and Yannakakis, M.: On the Complexity of Testing Implications of Functional and Join Dependencies, JACM 28, pp. 680-695 (1981).
38) Klug, A.: Calculating Constraints on Relational Expressions, ACM TODS 5, pp. 260-290 (1980).
39) Klug, A.: On Inequality Tableaux, Univ. of Wisconsin-Madison, Computer Science Rept. No. 403 (1980).
40) Hull, R.: Implicational Dependency and Finite Specification, Univ. of Sourthern California, CS Rept (1981).
41) Tanaka, K. and Kambayashi, Y.: Logical Integration of Locally Independent Relational Databases into a Distributed Databases, VLDB, pp. 131-141 (1981).
42) 岩崎，伊藤，谷口，嵩：関係データベースにおけるビュー上の従属性，信学会，オートマトンと言語研資 AL 81-25 (昭和56年6月).
43) Casanova, M. A: A Theory of Data Dependencies over Relational Expressions, ACM PODS, pp. 189-198 (1982).

(昭和58年1月24日受付)