データベース・システム 67－3
（1988. 9. 22）

フレーム束モデル

中野 良平
ＮＴＴ情報通信処理研究所

データベースの応用分野の拡大に伴って，関係モデルから脱皮の試みが続けられている．本稿は，複合オブジェクトのためのデータモデルを論ずる．関係を２次元と見るならば，複合オブジェクトは３次元であり，より複雑になるだけに，関係モデル以上にしっかりした数学的基盤が望まれる．それをここでは束に求めた．控えめな半順序を考えることにより，性質の良い複合オブジェクトのためのデータモデルが構築できる．性質が良いとは，関係モデルのスーパーモデルとなる，検索言語が規定できる，ネスト系操作での情報損失がないことを指す．オブジェクトの間にis_a階層を張ることもできる．関係代数，関係論理をおのおの拡張して，フレーム代数，フレーム論理を構成する．また，意味的には大胆に割り切ったが，数学的には一貫した空値も導入する．

Frame Lattice Model

Ryohei Nakano
NTT Communications and Information Processing Laboratories
1-2356 Take, Yokusuka, Kanagawa, 238-03 Japan

This paper describes a data model for a complex object. While a relation is two-dimensional, a complex object is three-dimensional. Since an object to be managed in a model gets complex, sound enough mathematical foundations are expected for the model. An introduction of a partial oder makes a set of complex objects a lattice. Based on the lattice, a data model of good nature can be constructed. The model is a super model of the relational model, lossless in nest/unnest operations, and has two kinds of query languages (called frame calculus and frame algebra). Moreover, it supports an is_a hierarchy between objects and introduces rather unique null values.

1. はじめに

データベースの応用分野の拡大に伴って，関係モデルから脱皮の試みが続けられている．目標は一つではない．オフィス(OA)のためのマルチメディアＤＢ，設計(CAD等)のためのエンジニアリングＤＢ，知識処理(AI)のための複合オブジェクトＤＢの３つに大別できる．

以下，複合オブジェクトのためのデータモデルを論ずる．関係モデルを２次元と見るならば，複合オブジェクトのモデルは3次元であり，２次元動物が3次元世界に入ったときに覚えるであろうとまどいがある．しっかりとした道標がほしい．それは数学的基盤であり，それをここでは束(lattice)に求めた．控えめな半順序を考えることにより，性質の良い複合オブジェクトのためのデータモデルが構築できた．性質が良いとは，関係モデルのスーパーモデルとなる，検索言語が規定できる，ネスト系操作での情報損失がないことを指す．オブジェクトの間にis_a階層を張ることもできる．関係代数，関係論理をおのおの拡張して，フレーム代数(Frame Algebra)，フレーム論理(Frame Calculus)を構成する．それぞれ手続き的，宣言的な検索言語である．また，意味的には大胆に割り切ったが，数学的には一貫した空値も導入する．以降，関係という用語が種々出てくるので，関係モデルの関係を特に正規関係と呼んで区別する．

2. フレーム束モデルの概要

はじめに，組構成子と集合構成子を用いてオブジェクトを再帰的に定義する．ある構成条件を満たすオブジェクトをとくにフレームとして定義する．フレームおよびその元である組オブジェクトはおのおの関係モデルの正規関係，組の拡張概念である．空値も効率良く扱える．ただし，３値論理は導入しない．従来，空値の扱いに一貫性が欠けていたが，ここでは首尾一貫した空値とする．

次いで，オブジェクトに半順序を定義する．この半順序こそが平面的な関係モデルを奥行きを持つ複合オブジェクトのモデルに拡張する重要なキーである．関係モデルでは，属性値が単一値であるためこのような半順序を考える必要がない．なお，Bancilhon [1] の導入した半順序は順序づけをやりすぎるために関係モデルを包含するモデルとなりえない．

こうして規定されたオブジェクト集合は半順序集合であり，その任意の２元{x, y}が上限x∩yと下限x∪yを持つ．ゆえに，オブジェクト集合は演算∩，∪のもとで束である．フレームのみの集合を考えても同様のことが言えるので，フレーム集合も束である．これをとくにフレーム束と呼ぶ．上限と下限はおのおの積集合，和集合の拡張演算である．

半順序のもとで，フレームの冗長性の問題を考える．その解決のためにフレーム間に同値～の概念を導入する．同値関係～はフレーム束の演算と両立する．ゆえに，フレーム束の剰余束が定義できる．とくに，代表元を定義することにより，代表元のみから成る代表束（フレーム代表束）が定義できる．

is_a階層はフレーム束の拡大に対応する．x is_a yが与えられたとき，xが下位，yが上位のフレームである．xはyから属性を継承し（属性の継承），xからyへフレームの元が遡及する（元の遡及）．元の遡及によりフレーム代表束が拡大する（is_a拡大）．

is_a拡大を終えたフレーム代表束の上に２種の検索系を考える．関係代数，関係論理を拡張したフレーム代数，フレーム論理である．フレーム代数，フレーム論理は代表フレーム束を台とする二つの代数系と考えることができる．両代数系の演算の間には全単射が存在する．その意味において，フレーム代数とフレーム論理は同類である．また，同値関係～はフレーム代数，フレーム論理の演算系と両立する．すなわち，束の演算以外に検索系を含めても代表フレーム束を対象とするのでよい．

3. フレーム束モデルの基本概念

3．1 オブジェクト

オブジェクトの概念を定義する．システム内で一意に識別できる属性名の集合を仮定する．オブジェクトを以下のように再帰的に定義する．なお，空を地と別のオブジェクトとする．この分離は重要であり，Bancilhonのモデルとの決定的な分岐点となる．

［定義３．１］オブジェクト

(1)単一値（整数，文字列，真理値など）はオブジェクトである．原子オブジェクトと呼ぶ．

(2)天(top)，地(bottom)，空(null)と呼ぶ３つの特殊オブジェクトがある．天と地はおのおの後述する半順序関係において最上位と最下位に位置する．空は不明または未定義を表わす．天，地，空はＴ，⊥，？で表わす．

(3)O1, O2, …, Onをオブジェクト，a1, a2, …, anを相異なる属性とするとき，

O = [a1: O1, a2: O2, …, an: On]

はオブジェクト（組オブジェクトと呼ぶ）である．属性値OkをO. akで表わす．

(4)O1, O2, …, Onをオブジェクトとするとき，

O = {O1, O2, …, On}

はオブジェクト（集合オブジェクトと呼ぶ）である．空集合{}も集合オブジェクトである．

(5)上記以外はオブジェクトでない．

［例３．１］オブジェクト

原子オブジェクト： 1988 ，中野 ，true

組は単純な組オブジェクト： [a:Y, c:20]　[a:S, c:?]

正規関係は集合オブジェクト： {[a:W, c:20], [a:W, c:?]}

正規関係ﾃﾞｰﾀﾍﾞｰｽも集合オブジェクト：

{[name:R1, rel: {[a:W, c:20], [a:W]}],
[name:R2, rel: {[a:Y, b:m], [a:Y, b:s]}]}

非正規関係も集合オブジェクト：

{[a:Y, b: {m, s}], [a:S, b: {m, c}]}

非正規関係ﾃﾞｰﾀﾍﾞｰｽも集合オブジェクト：

{[name:U1, rel: {[a:W, c: {20}], [a:N]}],
[name:U2, rel: {[a:Y, b: {m, j}], [a:M, b: {s}]}]}

［定義３．２］オブジェクトの等値性

(1)二つの原子オブジェクトはそれらが相等しいとき，そのときに限り等しい．なお，天，地，空は自らとのみ等しく，他のいかなるオブジェクトとも異なる．

(2)二つの組オブジェクトはすべての属性値が等しいとき，そのときに限り等しい．なお，属性値が空はその属性がないのと等価とする．

(3)二つの集合オブジェクトはすべての元が対で等しいとき，そのときに限り等しい．

［例３．２］オブジェクトの等値性

Ｔ＝Ｔ　Ｔ≠⊥　Ｔ≠？　Ｔ≠18　Ｔ≠{M}
⊥＝⊥　⊥≠Ｔ　⊥≠？　⊥≠18　⊥≠{M}
？＝？　？≠Ｔ　？≠⊥　？≠18　？≠{M}　？≠{}
[a:Y, c:20]＝[c:20, a:Y]
{音楽, 将棋, 車}＝{車, 音楽, 将棋}
[a:W]＝[a:W, b:？, c:？]　[a:？]＝[a:？]　[a:？]≠[a:W]
{[a:英樹]}≠{英樹}　{[a:英樹]}≠[a:英樹]　{英樹}≠英樹

通常，空の導入は？＝18などを不明とする３値論理を伴うが，集合論の最も基本である所属判定（ex. 18∈{20, ?}）が不明になるため，積集合，和集合

$R \cap S = \{x \mid x \in R \wedge x \in S\}$　　$R \cup S = \{x \mid x \in R \vee x \in S\}$
などの集合演算が規定できない．すなわち，

{18, ?} ∩ {20, ?} とか {18, ?} ∪ {20, ?}

が計算できない．３値論理は通常の集合論と整合しない．

定義３．２における空の定義はユニークである．空はただ一つ存在し，他のどのオブジェクトとも異なる．？＋１も？と考えたいので？は無限大または無限小に似ている．？＝？が真とは無茶のようであるが，不明，未定義というあいまいな情報を情報空間上では一つの値に縮退して同一視して扱うと考えればよい．このとき，

{18, ?} ∩ {20, ?} = {?}　　{18, ?} ∪ {20, ?} = {18, ?, 20}

なお，オブジェクトの積集合，和集合の演算は束の上限，下限の演算として拡張される．

空は空集合とはまったく異なることに注意．

3．2 フレーム

前節のオブジェクトの定義は一般的すぎる．たとえば，次のような雑多なオブジェクト

{[a:Y], [b:{音楽, 水泳}], S, {音楽, 車}}

は正しいオブジェクトであるが，このような集まりが増えてくると，それらをうまく検索するのは容易でない．また，大量のオブジェクトをこうした雑多なオブジェクトとして管理するのは現実的でない．

オブジェクトの種類をデータベースの型の概念に基づいて制限すれば，その上に強力な検索系を構築することができる．われわれはオブジェクトに程よい制限を加えるアプローチをとる．まず，オブジェクトの型の同一性の概念を以下のように再帰的に定義する．

［定義３．３］オブジェクトの型の同一性
(1)任意の二つの原子オブジェクトは同じ型である．
(2)空は任意のオブジェクトと同じ型である．
(3)二つの組オブジェクトはすべての対応する属性値が同じ型のとき同じ型である．
(4)任意の二つの集合オブジェクトはそれらの元がすべて同じ型のとき同じ型である．
(5)上記以外のオブジェクトは同じ型でない．

［例３．３］オブジェクトの型の同一性
中野と1988　？と2000　？と[a:W]
[a:Y, c:20]と[a:?, b:{音楽, 車}]
{{音楽, 水泳}, {音楽, 車}}と{{2000}}
{[a:Y], [c:20]}と{[b:{音楽}]}
以下は同じ型でない．
Yと[a:Y]　Yと{Y}　{Y}と[a:Y]　{音楽}と{{音楽}}
[a:Y, b:音楽]と[a:S, b:{音楽}]　{[a:Y], [a:S]}と{W}

［定義３．４］フレーム
まず，定義３．１に以下の制限を加える．
(5)組オブジェクトの属性値は組オブジェクト以外とする．
(6)集合オブジェクトの元は集合オブジェクト以外とする．

このとき，すべての元が同じ型の組オブジェクトである集合オブジェクトをフレームと呼ぶ．フレームの元である組オブジェクトの属性は原子オブジェクト，フレーム以外の集合オブジェクト，フレームを値としてとる．それぞれA型，S型，F型の属性と呼ぶ（属性の型）．

［例３．４］フレーム
正規関係はフレーム　{[a:Y, c:20], [a:W, c:?]}
非正規関係もフレーム　{[a:Y, b:{音楽, 水泳}], [a:S, b:{音楽, 車}]}
以下はいずれもフレームでない．
{[a:Y, b:音楽], [a:S, b:{音楽}]}
{[a:[姓:山田, 名:太郎]]}
{[a:Y, b:{{音楽}, {水泳}}], [a:S, b:{{音楽}, {車}}]}

フレームは正規関係を複合オブジェクト用に拡張した概念である以降，特に断わらない限り，定義３．４によるオブジェクトを対象とする．

3．3 半順序とフレーム束

オブジェクト間に大小関係を定義する．オブジェクトOがO'より小さいことをO≦O'で表わす．この大小関係こそが平面的な関係モデルを奥行きを持つ複合オブジェクト用のモデルに拡張する重要なキーである．関係モデルでは属性値が単一値であるためこのような大小関係を考える必要がない．なお，大小関係の定義はBancilhon [1] と似ているが，空？を地⊥と分離し，空は他のオブジェクトと比較不能としたため実質はかなり異なり，関係モデルをサブモデルとして包含することができるようになった．

［定義３．５］オブジェクト間の大小関係
任意のオブジェクトO，O'が与えられたとき，
(1)O≦O
(2)O≦T
(3)⊥≦O
(4)O，O'が組オブジェクトで，任意の属性aに対してO.a≦O'.aのとき，O≦O'
(5)O，O'が集合オブジェクトで，任意の元e（∈O）に対してe≦e'となるe'（∈O'）があるとき，O≦O'
(6)上記以外に大小関係はない．

［例３．５］オブジェクト間の大小関係
W≦W　[a:Y, b:{0}]≦[a:Y, b:{0, 1}]　{1, ?}≦{1, 2, ?}
特に，フレーム間の大小関係：
{[a:Y]}≦{[a:Y], [a:S]}　{[b:{M}], [b:{S}]}≦{[b:{M, S}]}
{[a:2, b:{M}]}≦{[a:2, b:{M, S}]}

［定理３．１］オブジェクト間の大小関係は半順序関係である．すなわち，オブジェクト集合は半順序集合である．
［定理３．２］オブジェクト集合は束をなす．
［系１］フレーム間の大小関係は半順序関係である．
［系２］フレーム集合は束をなす．フレーム集合が作る束をフレーム束と呼ぶ．

［定義３．６］上限，下限
O_1, O_2をオブジェクト集合の元とするとき，{O_1, O_2}の上限$O_1 \cup O_2$は$O_1 \leq O$かつ$O_2 \leq O$を満たすOの中で最小のものである．また，{O_1, O_2}の下限$O_1 \cap O_2$は$O \leq O_1$かつ$O \leq O_2$を満たすOの中で最大のものである．

［例３．６］上限，下限
{[b:{M}]} ∪ {[b:{S}]} = {[b:{M, S}]}
{[b:{M}]} ∪ {[b:{M, S}]} = {[b:{M, S}]}
{[b:{M}]} ∩ {[b:{S}]} = {[b:{}]}
{[b:{M}]} ∩ {[b:{M, S}]} = {[b:{M}]}
上限，下限は関係代数の和集合，積集合の拡張である．
{[a:Y]} ∪ {[a:Y, b:m]} = {[a:Y], [a:Y, b:m]}
{[a:Y, b:m]} ∩ {[a:Y, b:s]} = {}
なお，Bancilhonの上限，下限はそうでない．

O，O_1，O_2，O_3をオブジェクト集合の任意の元とするとき，上限，下限は以下の性質を有す．
［唯一性］上限，下限はおのおの唯一つ存在する．
［巾等律］$O \cup O = O$，$O \cap O = O$
［結合律］$(O_1 \cup O_2) \cup O_3 = O_1 \cup (O_2 \cup O_3)$
$(O_1 \cap O_2) \cap O_3 = O_1 \cap (O_2 \cap O_3)$

[交換律] $O_1 \cup O_2 = O_2 \cup O_1$, $O_1 \cap O_2 = O_2 \cap O_1$
[吸収律] $(O_1 \cup O_2) \cap O_1 = O_1$, $(O1 \cap O2) \cup O1 = O1$
[分配律] $O_1 \cap (O_2 \cup O_3) = (O_1 \cap O_2) \cup (O_1 \cap O_3)$
$O_1 \cup (O_2 \cap O_3) = (O_1 \cup O_2) \cap (O_1 \cup O_3)$
[同時性] $O_1 \leqq O_2 \leftrightarrow O_1 \cup O_2 = O_2 \leftrightarrow O_1 \cap O_2 = O_1$

[定義3．7] 要素判定
オブジェクトOが集合オブジェクトSの元かどうかの要素判定
$O \in S \leftrightarrow \{O\} \leqq S$
[例3．7] 要素判定
1∈{1, 2}
[a:1]∈{[a:1], [a:2]}
[a:1, b:{10}]∈{[a:1, b:{10, 11}]}
[a:{[b:{20}, c:30]}]∈{[a:{[b:{20, 21}, c:30]}]}

3．4 同値関係と代表束

フレーム間に同値の概念を導入すれば，フレームの冗長性の問題がきれいに解決できる．

[定義3．8] フレームの冗長性
フレームFの2元の下限においてS型またはF型属性の属性値が空集合でないものがあれば，そのフレームは冗長である（A型属性のみから成るフレームは等しい元を持たないので冗長でありえない）．冗長なフレームを内包するフレームもまた冗長である．
[例3．8] フレームの冗長性
{[a:S, b:{1, 2}], [a:S, b:{1}]}
2元の下限は[a:S, b:{1}]となり，定義よりこのフレームは冗長である．冗長性を除いたフレームとしては以下がある．
{[a:S, b:{1, 2}]}
{[a:S, b:{1}], [a:S, b:{2}]}
また，別の例として
{[a:S, b:{1, 2, 3}], [a:S, b:{2, 3}], [a:S, b:{3}]}
この場合，冗長性を除いたフレームとして以下がある．
{[a:S, b:{1}], [a:S, b:{2}], [a:S, b:{3}]}
{[a:S, b:{1}], [a:S, b:{2, 3}]}
{[a:S, b:{2}], [a:S, b:{1, 3}]}
{[a:S, b:{3}], [a:S, b:{1, 2}]}
{[a:S, b:{1, 2, 3}]}

冗長性の除き方は一般に何種類もあるので，冗長性を除いたフレームが何通りもできる．これらは同一視すべきである．そこで，フレームの同値の概念を導入する．

[定義3．9] フレームの同値
フレームF_1, F_2が同値であることを$F_1 \sim F_2$で表わす．
(1) $F_1 = F_2 \ \rightarrow \ F_1 \sim F_2$
(2) $F_1 = \{[x:\{y||Z\}]||W\}$, $F_2 = \{[x:\{y\}], [x:Z]||W\}$
$\rightarrow \ F_1 \sim F_2$かつ$F_2 \sim F_1$
ただし，集合オブジェクト{p||Q}は元pを除いた残りがQであることを表わす．
(3) $F_1 = F_2$, $F_2 = F_3 \ \rightarrow \ F_1 \sim F_3$
[例3．9] フレームの同値
{[a:S, b:{1, 2}], [a:S, b:{1}]}
～{[a:S, b:{1, 2}]}～{[a:S, b:{1}], [a:S, b:{2}]}
{[a:S, b:{1, 2, 3}], [a:S, b:{2, 3}], [a:S, b:{3}]}
～{[a:S, b:{1}], [a:S, b:{2}], [a:S, b:{3}]}
～{[a:S, b:{1}], [a:S, b:{2, 3}]}～{[a:S, b:{1, 2, 3}]}
{[a:{1}, b:{p}], [a:{1}, b:{q}], [a:{2}, b:{p}]}
～{[a:{1}, b:{p, q}], [a:{2}, b:{p}]}
～{[a:{1, 2}, b:{p}], [a:{1}, b:{q}]}

[定理3．3] フレームの同値は同値関係である．

[定理3．4] フレーム束の同値関係～はフレーム束における合同関係である．すなわち，フレーム束の同値関係はフレーム束の上限下限の演算と両立する．すなわち，F_1, F_2などを任意のフレームとするとき，
$F_1 \sim F_2, F_3 \sim F_4 \rightarrow F_1 \cup F_3 \sim F_2 \cup F_4$
$F_1 \sim F_2, F_3 \sim F_4 \rightarrow F_1 \cap F_3 \sim F_2 \cap F_4$

定理3．4より，フレーム束Lの同値関係～に関する商集合L／～においても上限，下限の演算が定義されて，L／～もフレーム束となる．

[定義3．10] 極大フレーム
フレームFの同値類の中でそれ以上大きな同値フレームがないものをその同値類の極大フレームと呼ぶ．
[例3．10] 極大フレーム
例3．9において極大フレームは
{[a:S, b:{1, 2}]}　{[a:S, b:{1, 2, 3}]}
極大フレームは一つに定まらないこともある．
{[a:{1}, b:{p, q}], [a:{2}, b:{p}]}
{[a:{1, 2}, b:{p}], [a:{1}, b:{q}]}

[定義3．11] フレーム代表束
一つの極大フレームを同値類の代表元とするとき，その集合をフレーム束Lの同値関係～によるフレーム代表束と呼ぶ．
[例3．11] フレーム代表束
以下のフレームから成る集合はフレーム代表束である．
{[a:{}]}　{[a:{1}]}　{[a:{2}]}　{[a:{1, 2}]}

3．5 is_a拡大

フレーム間のis_a階層を定義する．is_a階層には閉路はないとする．閉路はis_a階層の概念に合わない．

[定義3．12] is_a階層
フレームF_1, F_2がF_1 is_a F_2の関係にあるとき，F_2からF_1に属性が継承され，F_1からF_2に元が遡及する（属性の継承と元の遡及）．
[例3．12] is_a階層
is_a階層なしの状態
EMP:{[a:S, b:{m, c, s}], [a:Y, b:{s}], [a:M, b:{k, s}]}
MNG:{[a:W, c:{1, 2}], [a:P, c:{4}]}
"MNG is_a EMP"とすると，属性の継承と元の遡及が起きる．
EMP:{[a:S, b:{m, c, s}], [a:Y, b:{s}], [a:M, b:{k, s}],
[a:W, b:?], [a:P, b:?]}
MNG:{[a:W, b:?, c:{1, 2}], [a:P, b:?, c:{4}]}

元の遡及によりフレーム代表束が拡大する．この拡大をis_a拡大と呼ぶ．
[定義3．13] is_a拡大
フレームF, F_1, …, F_kの間に
F_1 is_a F, …, F_k is_a F
のis_a関係があるとき，is_a拡大後のFをF^+とおくと
$F^+ = F \cup is\_a(F_1, F) \cup \cdots \cup is\_a(F_k, F)$
ただし，is_a(F_1, F)はF_1をFの属性群に射影する演算で，is_a射影と呼ぶ．
[例3．13] is_a拡大

F_1 is_a F , F_2 is_a F , F_3 is_a F

F : {[a: {1}]}

F_1: {[a: {2}, b: {11}], [a: {3}, b: {13}], [a: {4}, b: {14}]}

F_2: {[a: {3}, c:21], [a: {5}, c:22]}

F_3: {[a: {6}, d: {31, 32}]}

is_a射影： is_a(F_1, F) = {[a: {2, 3, 4}]}

is_a(F_2, F) = {[a: {3, 5}]}

is_a(F_3, F) = {[a: {6}]}

is_a拡大： F^+ = {[a: {1, 2, 3, 4, 5, 6}]}

is_a射影は次の性質を持つ.

［定理3．5］

is_a($F_1 \cup F_2$, F) = is_a(F_1, F) ∪ is_a(F_2, F)

is_a($F_1 \cap F_2$, F) ≦ is_a(F_1, F) ∩ is_a(F_2, F)

is_a拡大は再帰的に実行する．このとき，is_a階層のリーフから順に上に向かって計算する．

［例3．14］is_a拡大の計算

F_1 is_a F, F_2 is_a F, F_3 is_a F_2, F_4 is_a F_2

このとき

F_2^+ = F_2 ∪ is_a(F_3, F_2) ∪ is_a(F_4, F_2)

F^+ = F ∪ is_a(F_1, F) ∪ is_a(F_2^+, F)

代入して展開すると

= F ∪ is_a(F_1,F) ∪ is_a(F_2,F) ∪ is_a(F_3,F) ∪ is_a(F_4,F)

［定理3．6］フレーム代表束をis_a拡大した集合もフレーム代表束とすることができる．

以降では，is_a拡大後のフレーム代表束を考える．

4. フレーム代数

ｉｓ＿ａ拡大を終えたフレーム代表束の上に２種の検索系を考える．関係代数，関係論理を拡張したフレーム代数，フレーム論理である．フレーム代数，フレーム論理はフレーム代表束を台とする二つの代数系と考えることができる．

［定理4．1］フレーム代数，フレーム論理の演算の間には全単射が存在する．この意味において，フレーム代数とフレーム論理は同類である．

［定理4．2］フレーム束の上限，下限はフレーム代数の演算系と両立する．

［定理4．3］同値関係～はフレーム代数の演算系と両立する．

定理4．2，4．3により，束の演算の他に検索系を含めても，フレーム代表束が保持できる．

関係代数は深さ(ﾈｽﾄ)のない正規関係を操作する演算体系であるが，深さを持つ複合オブジェクトに対しては無力である．以降にて関係代数を拡張するのであるが，深さと集合の扱いが要点となる．

以降の定義で以下の記号を使用する．

Ｉ：スキーマインスタンス

$e, e_1, e_2, \cdots$：フレーム代数式

$A, A_1, A_2, \cdots$：属性

表記を簡単にするため，属性は常にフレーム毎に先頭から，#1, #2, …と命名する．また，正規関係では属性は表面にのみ一層あるが，フレームでは表面だけでなくネストした奥にもある．表面にあるものを表面属性，ネストの奥にあるものを内部属性と呼ぶ．ネストを一段深く入るには組オブジェクトの記号[]を用いる．

4．1 射影

関係代数の射影は正規関係から指定された属性を取り出して正規関係を作るものであった．フレーム代数の射影はフレームから，その属性構造を全く変えずに，指定された属性を抜き出してフレームを作るものと拡張する．属性の指定が深さを配慮したものとなる．

［定義4．1］射影

形式　e[A]

意味　(e[A])(I) = {t[A] | t ∈ e(I)}

［例4．1］射影

F: {[#1:a, #2: {0, 1},
#3: {[#1:c, #2: {p, q}], [#1:d, #2: {p, r}]}],
[#1:b, #2: {1, 2, 3},
#3: {[#1:c, #2: {p, t}], [#1:e, #2: {q, s}]}]}

F = F[#1, #2, #3] = F[#1, #2, #3[#1, #2]]

F[#2] = {[#1: {0, 1}], [#1: {1, 2, 3}]} = {[#1: {0, 1, 2, 3}]}

F[#3[#2]] = {[#1: {[#1: {p, q}], [#1: {p, r}]}],
[#1: {[#1: {p, t}], [#1: {q, s}]}]}

= {[#1: {[#1: {p, q, r}]}], [#1: {[#1: {p, q, s, t}]}]}

= {[#1: {[#1: {p, q, r}], [#1: {p, q, s, t}]}]}

= {[#1: {[#1: {p, q, r, s, t}]}]}

4．2 選択

関係代数の選択は正規関係から選択条件を満足する組を選び正規関係を作る．フレーム代数の選択はフレームから選択条件を満足する組オブジェクトを選びフレームを作る．

内部属性を選択の条件に指定することができる．その属性の属性値が条件を満足した場合，その属性値を持つ組オブジェクトだけでなく，それを含む組オブジェクトが芋ずる式に選択対象となる．

［定義4．2］選択

形式　$e[F_s]$

ただし，F_sは選択条件式で以下の単位式を∧, ∨, ￢で結んだ論理式．なお，(), []も必要に応じて使用できる．

〈単位式の形式〉

(1) A型属性 θ 原子オブジェクト
（θは=, ≠, <, ≦, >, ≧）

(2) ∃(A型属性のみから成る論理式)
￢∃(A型属性のみから成る論理式)

(3) 原子オブジェクト ∈ S型属性
原子オブジェクト ∉ S型属性

(4) S型属性 = {}　　S型属性 ≠ {}

意味　$(e[F_s])(I) = \{t \mid t \in e(I) \wedge t[F_s]\}$

［例4．2］選択

F: {[#1:c, #2: {[#1:0, #2: {p}], [#1:2, #2: {p, r}]}],
[#1:d, #2: {[#1:2, #2: {q}], [#1:3, #2: {}]}]}

F[#2[#1=0]] = {[#1:c, #2: {[#1:0, #2: {p}]}]}

F[#2[∃(#1=0)]]
= {[#1:c, #2: {[#1:0, #2: {p}], [#1:2, #2: {p, r}]}]}

F[#2[q∈#2]] = {[#1:d, #2: {[#1:2, #2: {q}]}]}

F[#2[#1=2∧r∈#2]] = {[#1:c, #2: {[#1:2, #2: {p, r}]}]}

F[#2[#2={}]] = {[#1:d, #2: {[#1:3, #2: {}]}]}

4．3 結合

関係代数の結合は２つの正規関係から組と組をつないで正規関係を作る．フレーム代数の結合は２つのフレームから組オブジェクト

をつないでフレームを作る．
　内部属性を結合の条件の片方に指定することができる．その属性の属性値が条件を満足した場合，その属性値を持つ組オブジェクトだけでなく，それを含む組オブジェクトが芋ずる式に結合対象となる．他方は表面属性とする．双方を内部属性とすると，組オブジェクトの連鎖の方法に問題が生じるので避ける．
　結合条件には，集合オブジェクト用の判定機能を追加する．

［定義4．3］組オブジェクトの連鎖
　組オブジェクトt_1，t_2が与えられたとき，その連鎖$t_1 \circ t_2$は連鎖の糊しろとなる属性を持つ組オブジェクトを単純に合体して1つの組オブジェクトとすることにより実現する．
［例4．3］組オブジェクトの連鎖
下線が連鎖の糊しろとなる属性
t_1:[#1:a, #2:{[#1:c, #2:4]}]
t_2:[#1:c, #2:{20, 22}]
$t_1 \circ t_2$:[#1:a, #2:{[#1:c, #2:4, #3:c, #4:{20, 22}]}]

［定義4．4］結合
形式　e_1[#m op #n]e_2
　ただし，#m op #nは結合条件式で以下の単位式
　(1) A型属性 θ A型属性
　(2) S型属性 θ' S型属性　F型属性 θ' F型属性
　(3) S型属性 ∩ S型属性　F型属性 ∩ F型属性
　ここで，θ'は=, ≠, ≦, ≧
意味　(e_1[#m op #n]e_2)(I)
　　＝{t∘s | t∈e_1(I)∧s∈e_2(I)∧cond}
　ただし，condは以下
　t[#m] op s[#n]　(1), (2)のとき
　t[#m]=s[#n]∧t[#m]≠{}∧t[#m]≧(t'[#m]∩s'[#n])∧
　t'∈e_1(I)∧s'∈e_2(I)　(3)のとき
［例4．4］結合
F:{[#1:a, #2:{[#1:c, #2:p]}], [#1:b, #2:{[#1:e, #2:s]}]}
G:{[#1:c, #2:10], [#1:d, #2:20]}
F[#2[#1]=#1]G＝{[#1:a, #2:{[#1:c, #2:p, #3:c, #4:10]}]}
H:{[#1:60, #2:{0}], [#1:61, #2:{1, 3}]}
I:{[#1:a, #2:{0, 1}], [#1:b, #2:{1, 2, 3}]}
H[#2≦#2]I＝{[#1:60, #2:{0}, #3:a, #4:{0, 1}],
　[#1:61, #2:{1, 3}, #3:b, #4:{1, 2, 3}]}
H[#2∩#2]I＝{[#1:60, #2:{0}, #3:a, #4:{0}],
　[#1:61, #2:{1}, #3:a, #4:{1}]
　[#1:61, #2:{1, 3}, #3:b, #4:{1, 3}]}

4.4 準結合

　関係代数の準結合は結合を一方の正規関係の選択に利用する．フレーム代数の準結合も同様とする．準結合はあくまで選択的に働く．準結合には，肯定準結合と否定準結合がある．そのほかは結合とまったく同じである．

［定義4．5］準結合
形式　肯定準結合　e_1[∃;#m op #n]e_2
　　　否定準結合　e_1[¬∃;#m op #n]e_2
　ただし，#m op #nは結合条件式
意味　(e_1[∃;#m op #n]e_2)(I)
　　＝{t | t∈e_1(I)∧{s | s∈e_2(I)∧cond}≠{}}
　　(e_1[¬∃;#m op #n]e_2)(I)
　　＝{t | t∈e_1(I)∧{s | s∈e_2(I)∧cond}＝{}}
　ただし，condは定義4．4と同じ

［例4．5］準結合
F:{[#1:a, #2:{[#1:c, #2:p]}], [#1:b, #2:{[#1:e, #2:s]}]}
G:{[#1:c, #2:10], [#1:d, #2:20]}
F[#2[#1]=#1]G＝{[#1:a, #2:{[#1:c, #2:p]}]}
H:{[#1:60, #2:{0}], [#1:61, #2:{1, 3}], [#1:62, #2:{2}]}
I:{[#1:a, #2:{0, 1}]}
H[∃;#2≦#2]I＝{[#1:60, #2:{0}]}
H[¬∃;#2≦#2]I＝{[#1:61, #2:{1, 3}], [#1:62, #2:{2}]}
H[∃;#2∩#2]I＝{[#1:60, #2:{0}], [#1:61, #2:{1, 3}]}
H[¬∃;#2∩#2]I＝{[#1:62, #2:{2}]}
(注) 結合の射影が準結合ではない！
F[∃;φ]G≠(F[φ]G)[att(F)]

4.5 除算

　フレーム代数の除算は関係代数の除算を含む自然な拡張である．

［定義4．6］除算
形式　e_1[A_1/A_2]e_2
　ただし，A_1は表面属性であること
意味
　(e_1[A_1/A_2]e_2)(I)＝{t[～A_1] | t∈e_1(I)∧{r[A_2] |
　r∈e_2(I)}≦{s[A_1] | s∈e_1(I)∧s[～A_1]=t[～A_1]}}
　ただし，～A_1はA_1以外の属性
［例4．6］除算
F:{[#1:a, #2:{0, 1}], [#1:b, #2:{1, 2, 3}], [#1:c, #2:{2, 4}]}
G:{[#1:{1, 2}]}
F[#2/#1]G＝{[#1:b]}

［補助定理］除算と準結合の関係
A_1, A_2がS型のとき
　F[A_1/A_2]G＝F[∃;A_1≧A_2]G
A_1, A_2がA型のとき
　F[A_1/A_2]G＝(F{A_1}[∃;A_1≧A_2]G{A_2}){A_1}$^{-1}$

4.6 集合演算

　集合演算は和集合，積集合，差集合からなる．フレーム代数の和集合，積集合はおのおのフレーム束の上限，下限にほかならない．それらは関係代数の和集合，積集合の自然な拡張であることは3．3節で述べた．フレーム代数の差集合も定義できる．

［定義4．7］集合演算
形式　和集合 e_1[+]e_2　積集合 e_1[*]e_2　差集合 e_1[-]e_2
意味　(e_1[+]e_2)(I)＝e_1(I)∪e_2(I)
　　(e_1[*]e_2)(I)＝e_1(I)∩e_2(I)
　　(e_1[-]e_2)(I)＝{t | t∈e_1(I)∧s≦t∧s∈e_1(I)∧s∉e_2(I)}
［例4．7］差集合
F:{[#1:a, #2:{1, 2, 3}], [#1:b, #2:{3, 4}]}
G:{[#1:a, #2:{1}], [#1:b, #2:{3}]}
F[-]G＝{[#1:a, #2:{2, 3}], [#1:b, #2:{4}]}

［補助定理］任意のフレームF_1，F_2について
　(F_1[-]F_2)∪(F_1∩F_2)＝F_1

4.7 集約演算

　集約演算は中核的な関係代数には含まれないが，検索の高度化に伴い必要となる．
　グループ化するもの2種類，グループ化しないもの2種類の計4

種類の集約演算が考えられる．グループ分けするフレームの中からグループを作り出す内集約演算，グループ分けするフレームとは別のフレームから作り出す外集約演算，グループ化しないスカラ集約演算とS型属性を単純に集約する集合集約演算である．

複合オブジェクトにおけるグループ化をあまり一般的にすると複雑になりすぎて制御できなくなる．そこで，同一フレームに閉じたグループ化のみを考える．それは正規関係におけるグループ化の単純な拡張である．

［定義4．9］集約演算

形式　内集約演算　$e[A;agg]$

外集約演算　$e_1[A_1/A_2;agg]e_2$

スカラ集約演算　$e[agg]$

集合集約演算　$e[A,sgg]$

ただし，A, A_1, A_2はおのおの同一フレームの表面属性

agg=count, xxx［A型属性番号］,

sgg=count［S/F型属性番号］, xxx［S/F型属性番号］

xxx=avg, sum, max, min

agg, sggの属性はA, A_1と同一フレームの表面属性

意味

$(e[A;agg])(I)=\{t[A]\circ f \mid t\in e(I)\wedge$
$\quad f=agg(\{s \mid s\in e(I)\wedge s[A]=t[A]\})\}$

$(e_1[A_1/A_2;agg]e_2)(I)=\{t[A_2]\circ f \mid \underline{t\in e_2(I)}\wedge$
$\quad f=agg(\{s \mid s\in e_1(I)\wedge s[A_1]=t[A_2]\})\}$

$(e[agg])(I)=agg(e(I))$

$(e[A,sgg])(I)=\{t[A]\circ f \mid t\in e(I)\wedge f=sgg(t)\}$

［例4．9］集約演算

$F:\{[\#1:a,\#2:\{1,4\},\#3:\{[\#1:p,\#2:\{1,4\}],[\#1:q,\#2:\{2\}]\}],$
$\quad[\#1:b,\#2:\{5,7,9\},$
$\quad\ \#3:\{[\#1:p,\#2:\{3\}],[\#1:q,\#2:\{2,5,8\}]\}]\}$

集合集約演算

$F[\#1,count[\#2],max[\#2]]$
$=\{[\#1:a,\underline{\#2:}2,\underline{\#3:}4],[\#1:b,\underline{\#2:}3,\underline{\#3:}9]\}$

$F[\#1,\#3[\#1,sum[\#2],count[\#2]]]$
$=\{[\#1:a,\#2:\{[\#1:p,\underline{\#2:}5,\underline{\#3:}2],[\#1:q,\underline{\#2:}2,\underline{\#3:}1]\}],$
$\quad[\#1:b,\#2:\{[\#1:p,\underline{\#2:}3,\underline{\#3:}1],[\#1:q,\underline{\#2:}15,\underline{\#3:}3]\}]\}$

4.8 ネスト化・逆ネスト化

複合オブジェクト特有の演算として，ネスト化(nest)，逆ネスト化(unnest)がある．これらは関係代数には存在しない．ネスト化にはA型からS型へのS変換とF型へのF変換がある．逆ネスト化はそれらの逆である．従来の研究では両者を明確に区別していない．

また，S型，F型に変換された後，代表元への変形は同値関係を利用して行われる．型の変換さえやれば，後はモデルが自動的にグループ化相当の処理を行う．

［定義4．10］ネスト化・逆ネスト化

形式　S変換　$e\{A\}$　　逆S変換　$e\{A\}^{-1}$

F変換　$e\{[A_1,\cdots,A_k]\}$　　逆F変換　$e\{[A]\}^{-1}$

ただし，F変換のときのみ複数属性でもよい．複数属性は同一組オブジェクトの属性とする．他は1属性とする．

意味

$e\{A\}(I)=e(A\rightarrow\{A\})(I)$
$=\{g\circ[\#n:\{a\}] \mid g\in e[\sim A](I)\wedge[\#1:a]\in e[\sim A=g][A](I)\}$

$e\{A\}^{-1}(I)=e(\{A\}\rightarrow A)(I)$
$=\{g\circ[\#n:a] \mid g\in e[\sim A](I)\wedge[\#1:\{a\}]\in e[\sim A=g][A](I)\}$

$e\{[A_1,\cdots,A_k]\}=e(A_1,\cdots,A_k\rightarrow\{[A_1,\cdots,A_k]\})(I)$
$=\{g\circ[\#n:\{[\#1:a_1,\cdots,\#k:a_k]\}] \mid g\in e[\sim(A_1,\cdots,A_k)](I)\wedge$
$\quad[\#1:a_1,\cdots,\#k:a_k]\in e[\sim(A_1,\cdots,A_k)=g][A](I)\}$

$e\{[A]\}^{-1}=e(\{[A]\}\rightarrow A_1,\cdots,A_k)(I)$
$=\{g\circ[\#n:a_1,\cdots,\#n+k:a_k] \mid g\in e[\sim A](I)\wedge$
$\quad[\#1:\{[\#1:a_1,\cdots,\#k:a_k]\}]\in e[\sim A=g][A](I)\}$

［例4．10］ネスト化・逆ネスト化

$F:\{[\#1:A,\#2:1,\#3:x],[\#1:A,\#2:1,\#3:z],$
$\quad[\#1:B,\#2:2,\#3:y],[\#1:B,\#2:2,\#3:v],$
$\quad[\#1:B,\#2:2,\#3:p],[\#1:C,\#2:3,\#3:w]\}$

$F\{\#3\}=\{[\#1:A,\#2:1,\#3:\{x,z\}],$
$\quad[\#1:B,\#2:2,\#3:\{y,v,p\}],$
$\quad[\#1:C,\#2:3,\#3:\{w\}]\}$

$F\{[\#2,\#3]\}$
$=\{[\#1:A,\#2:\{[\#1:1,\#2:x],[\#1:1,\#2:z]\}],$
$\quad[\#1:B,\#2:\{[\#1:2,\#2:y],[\#1:2,\#2:v],[\#1:2,\#2:p]\}],$
$\quad[\#1:C,\#2:\{[\#1:3,\#2:w]\}]\}$

［定理4．4］ネスト化・逆ネスト化の性質

(1)ネスト化・逆ネスト化は情報損失しない(lossless)．

$F=F\{A\}\{A\}^{-1}=F\{A\}^{-1}\{A\}$

$F=F\{[A]\}\{[A]\}^{-1}=F\{[A]\}^{-1}\{[A]\}$

(2)ネスト化・逆ネスト化はおのおの可換である．

$F\cdot X_1\cdot X_2=F\cdot X_2\cdot X_1$

ただし，$X_1=\{A_1\},\{[A_1]\},\{A_1\}^{-1},\{[A_1]\}^{-1}$
$\quad X_2=\{A_2\},\{[A_2]\},\{A_2\}^{-1},\{[A_2]\}^{-1}$

5. フレーム論理

5.1 形式

［定義5．1］変数

変数は集合オブジェクトの元を値とする変数である．変数は英小文字(u, v, …)で表記する．

［例5．1］変数

著者(x)において，xは変数である．

［定義5．2］項

(1)原子オブジェクトは項である．

(2)フレームを除く集合オブジェクトは項である．

(3)空は項である．

(4)uを変数，nを自然数とするとき，u[n]は項である．

(5)集約関数は項である．

(6)上記以外は項でない．

［例5．2］項

10　?　{a, b, c}　x[2]　count(文献)

［定義5．3］範囲

(1)アルファαは範囲である．

(2)uを変数，nを自然数とするとき，u[n]は範囲である．

(3)α_1，α_2を範囲とするとき，$\alpha_1\cup\alpha_2$は範囲である．

(4)α_1，α_2を範囲とするとき，$\alpha_1\cap\alpha_2$は範囲である．

(5)α_1，α_2を範囲とするとき，$\alpha_1\cap\neg\alpha_2$は範囲である．

(6)上記以外は範囲でない．

［例5．3］範囲

文献　u[2]

従業員∪アルバイト　従業員∩¬未婚従業員

［定義5．4］範囲式

Rを範囲，uを変数とするとき，R(u)は範囲式である．

［例5．4］範囲式

文献(u)　u[2](v)

(従業員∪アルバイト)(v)　(従業員∩¬未婚従業員)(x)

［定義5．5］論理式
t_1，t_2を項，θを比較演算子$(=, \neq, <, \leq, >, \geq)$，f，$f_1$，$f_2$を論理式，rを範囲式とするとき，
(1) $t_1 \theta t_2$は論理式である．
(2) $t_1 \in t_2$は論理式である．
(3) $f_1 \wedge f_2$，$f_1 \vee f_2$は論理式である．
(4) $\neg$ f は論理式である．
(5) $(\exists r)$ f，$(\forall r)$ f は論理式である．
(6) 上記以外は論理式でない．

［例5．5］論理式
x[1]=2　a∈v[2]　x[1]=2∧y[2]=z[3]
¬(x[1]=2∧y[2]=z[3])
(∃文献(u))(x[1]=u[3]∧u[2]=1988)

［定義5．6］目標リスト
目標リストは以下のどれか（一つ以上）を[]で括ったものである．
(1) 変数[自然数]　または　{変数[自然数]}
(2) 集約関数　または　{集約関数}
(3) アルファ

［例5．6］目標リスト
[u[1], {u[3]}, v[1]]
[u[1], count([v[1]]:R(v):v[1]=u[1])]
[u[1], {[v[2], v[3]]:R(v):v[1]=u[1]}]

［定義5．7］範囲規定
範囲式の集合を範囲規定と呼ぶ．範囲規定では範囲式をカンマで区切って表記する．

［例5．7］範囲規定
文献(u), 従業員(v)

［定義5．8］アルファ
(1) 範囲Rはアルファである．
(2) TLを目標リスト，RFsを範囲規定，WFFを論理式とするとき，以下はアルファである．
{TL:RFs:WFF}
(3) 上記以外はアルファでない．

［例5．8］アルファ
文献　{[u[1], u[2]]:文献(u):u[3]=1988}

［定義5．9］束縛変数と自由変数
指定された範囲の元を値とする変数を束縛変数と呼ぶ．具体的には
(1) アルファαの範囲規定中の範囲式R(u)で規定された変数uはαの目標リストと論理式の中で束縛されている．
(2) 論理式中の$(\exists r)$ f または$(\forall r)$ f の範囲式 r＝R(u)で規定された変数uは論理式 f の中で束縛されている．
束縛されてない変数を自由変数と呼ぶ，

［例5．9］束縛変数と自由変数
{[u[1], u[2]]:文献(u):u[3]=1988}
において，変数uは目標リスト[u[1], u[2]]と論理式u[3]=1988の中で束縛されている．
∃(x[1](w))(w[2]=y[1]∧w[3]=10)
において，変数wは論理式
(w[2]=y[1]∧w[3]=10)
の中で束縛されている．

［定義5．10］閉アルファと開アルファ
そのアルファのみを取り出したとき，自由変数が皆無のアルファを閉アルファと呼ぶ．それ以外のアルファを開アルファと呼ぶ．閉アルファをフレーム論理式とも呼ぶ．

［例5．10］閉アルファと開アルファ
閉アルファ：
文献
{[u[1], u[2]]:文献(u):u[3]=1988}
{[u[1], count({[v[2], v[3]]:R(v):v[1]=u[1]}):文献(u):u[3]=1988}
（注）最後の例は閉アルファの中に開アルファがある．
開アルファ：
{[v[2], v[3]]:R(v):v[1]=u[1]}
（注）uが自由変数

5．2 解釈

［定義5．11］フレーム論理式の解釈
フレーム論理式
$\{[t_1, \cdots, t_n]:r_1(u_1), \cdots, r_m(u_m):\phi(u_1, \cdots, u_m)\}$
は，スキーマインスタンスIの下で次のように解釈される．
$\{[\#1:t_1(u_1, \cdots, u_m), \cdots, \#n:t_n(u_1, \cdots, u_m)] \mid$
$u_1 \in r_1(I) \wedge \cdots \wedge u_m \in r_m(I) \wedge \phi(u_1, \cdots, u_m)\}$

［例5．11］フレーム論理式の解釈
{[u[1]]:Resort(u):∃u[3](x)(x[1]=tennis∧x[2]<=2)}
このフレーム論理式の解釈は以下
{[u[1]] | u∈Resort∧∃u[3](x)(x[1]=tennis∧x[2]<=2)}
{[u[1]] | u∈Resort∧x∈u[3]∧(x[1]=tennis∧x[2]<=2)}

6．おわりに

複合オブジェクトのためのデータモデルを論じた．扱う対象が複雑になるだけに，関係モデル以上にしっかりとした数学的基盤が望まれる．ここでは束を基盤にして半順序を考え，対象とするオブジェクトを制限することにより，性質の良い複合オブジェクトデータモデルが構築できた．それは関係モデルのスーパーモデルであり，フレーム代数，フレーム論理と呼ぶそれぞれ手続き的，宣言的な検索言語を持つ．また，フレームの間にis_a階層を張ることもできるさらにネスト系操作での情報損失がない．

今後は本モデルの有用性評価，フレーム代数マシン，フレーム論理／フレーム代数変換，フレーム論理のヒューマンインタフェースなどが課題である．

議論していただいたＮＴＴ情報研知基グループのメンバに感謝します．

[参考文献]

[1]Bancilhon, F.:A Calculus for Complex Objects," PODS, pp. 53-59, Mar. 1986.
[2]弥永昌吉, 小平邦彦:現代代数概説Ⅰ, 岩波書店, 1961.
[3]Jaeschke, G, Schek, H.-J.:Remarks on the Algebra of Non First Normal Form Relations," PODS, pp. 124-138, 1982.
[4]中野良平, 木山稔:ﾌﾚｰﾑ論理, 87-AI-50, 中野, 小浜千恵:ｵﾌﾞｼﾞｪｸﾄ指向知識ﾃﾞｰﾀﾍﾞｰｽ, 88-AI-56.
[5]Date, C. J.:An Introduction to Database Systems Vol. 2, Addison-Wesley, 1983.
[6]Klug, A.:Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions, J. ACM, Vol. 29, No. 3, pp. 699-717, 1982.