3.7 関係データベース

3.7.1 データベースとは

データベースとは，コンピュータ上で大量に蓄積されたデータを構造をもたせた形式で記録しておくことにより，効率的な利用ができるようにしたものである．データベースを使う際には，データの検索 (特定の条件を満たすデータを探し出すこと)，データの更新 (追加や削除など)，データの加工 (2 つのデータベースを合併したり，既存データベースのうちの一部分を抜き出して新規データベースを作ったりすることなど) などの操作/処理が必要である*.

● **データを記録するための構造**　データベースを実現するために考案されている「データを記録するための構造」(データ構造という) としては,

・階層型　　・ネットワーク型　　・リレーショナル型

という 3 つの方法がある．階層型データモデルではデータを階層に類別し，データベース全体を階層間の上下関係として表わす．ネットワーク型データモデルは，類別されたどのデータ同士にも自由につながりを設定できるようにしたものであり，データベース全体はデータを頂点とする有向グラフ (＝ネットワーク) になる．これらに対し，リレーショナルデータモデル (関係データモデル) は，E.F.Codd (1970) がデータベースの数学的モデルの 1 つとして提案したものであり，荒っぽくいうと，データの間の関係を表として記録する方式である†.
リレーショナルデータモデルは,「表」という分かりやすい概念に基づいていただけでなく，厳密に数学的に形式化されたモデルであったために理論的な研究が活発に行なわれ，その結果，今日のデータベースのほとんどはリレーショナル方式に基づいている．リレーショナルデータモデルに基づいたデータベースを**関係データベース**(リレーショナルデータベース) と呼ぶ‡.

---

*データベースとその利用者の間を仲介し，このような処理や管理を行うプログラムのことを**データベース管理システム**(DBMS) という．database management system.

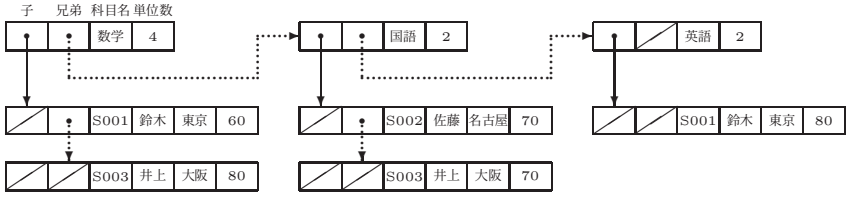
†後ほど明らかになるが，表は n 項関係に他ならない (n は適当な正整数) ので，数学的には，リレーショナルデータモデルはネットワーク型データモデル (＝有向グラフ＝ 2 項関係) を特別の場合として含む，より一般的なモデルである．

‡関係データベース relational database (relation ＝関係).

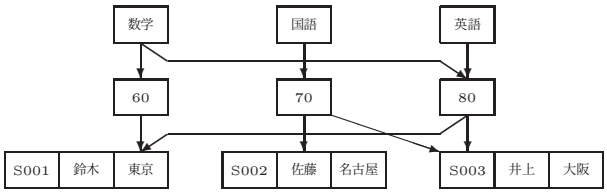
〔例 3.20〕 3 つのデータモデルの比較

階層型とネットワーク型では，ポインタによってデータが関連付けられる．ポインタとは，次のデータがどこにあるかを指し示す情報のことであり，図では矢印で表わす．階層型では子データ (ポインタが指す所) は親データ (ポインタが出る所) を 1 つしか持てないのに対し，ネットワーク型では複数の親を持つことができるという違いがある．以下に，3 つのデータモデルの違いを，科目・学生・履修状況に関するデータベースを例として示す．

（1） 階層型



（2） ネットワーク型



（3） リレーショナル型

科目 [科目名, 単位数]

科目名	単位数
数学	4
国語	2
英語	2

学生 [学生番号, 氏名, 住所]

学生番号	氏名	住所
S001	鈴木	東京
S002	佐藤	名古屋
S003	井上	大阪

履修状況 [科目名, 学籍番号, 得点]

科目名	学籍番号	得点
数学	S001	60
数学	S003	80
国語	S002	70
国語	S003	70
英語	S001	80

3.7.2 関 係 代 数

● データベース操作言語 問い合わせ (検索) を行なったりデータの更新をしたりするためにデータベースを操作する際には，データベースとの間で交わす遣り取りを記述するための言語が必要であるが，そういった言語はどのようなデータベースモデルを使っているかによって仕様が大きく異なったものとなる．荒っぽくいうと，データベース操作言語とは，データベースに対してどのような操作 (＝演算) ができるかということであり，数学的にいうと，データの集合 (どのような構造をもった集合であるか) とその上でどのような (複数個の) 演算が定義されているかということ (すなわち，一種の代数系) である．リレーショナルデータモデルに対しては，関係代数，タップル関係論理，ドメイン関係論理という 3 つの代数的特徴づけが知られており，これらは一方から他方へ同値変換できるという意味で数学的に等価である．こういったデータモデルに基づいて実際のデータベース操作言語とそれに基づく DBMS が作られる[§]．以下では，関係代数についてだけ述べる．

● 関係表 関係代数とは，以下のように定義される「関係表」の集合とその上のいくつかの操作演算の集まりのことである．まず，その基本となる関係表とは何かを定義しよう．

$A_1, A_2, \ldots A_n$ を有限集合 (データの集合) とする．直積 $A_1 \times A_2 \times \cdots \times A_n$ の部分集合

$$R \subseteqq A_1 \times A_2 \times \cdots \times A_n \qquad (\text{または, } R[A_1, A_2, \ldots, A_n] \text{ と表わす})$$

のことを n 項関係とか，関係表 (または，テーブル) という．n を R の**次数**という．R の元

$$(a_1, a_2, \ldots, a_n) \in R$$

は，$a_1 \in A_1$, $a_2 \in A_2$, …, $a_n \in A_n$ が R の関係にある 一組のデータである

[§]例えば，エスキューエル SQL (Structured Query Language) は関係代数に基づいて設計された関係データベース操作用の言語である．関係データベースに関するアプリケーションの開発者は，関係データベースと相互に遣り取りするための操作を SQL を使って記述する．今日では，SQL は関係データベース操作用言語として事実上の標準言語になっている．例えば，マイクロソフト社のデータベース管理システムである Access も SQL に準拠している．

ことを表わす．$(a_1, a_2, \ldots, a_n)$ を**タップル**¶と呼ぶ．$A_1, A_2, \ldots, A_n$ それぞれは，データの集合であると同時にその集合の名称であり，その集合に属すデータが持つ**属性**をも表わす‖．

関係表は，次のように「表」として表わすとわかりやすい．

$R[A_1, A_2, \ldots, A_n]$

A_1	A_2	$\cdots$	A_n
a_{11}	a_{12}	$\cdots$	a_{1n}
$\cdots$	$\cdots$	$\cdots$	$\cdots$
a_{k1}	a_{k2}	$\cdots$	a_{kn}

R が $A_1, A_2, \ldots, A_n$ を属性とする関係表であることを表わす．$a_{1i}, a_{2i}, \ldots a_{ki}$ は属性 A_i を持つデータである．

〔例 3.21〕 学生情報の関係表

学生情報 [学籍番号, 氏名, 性別, 学部, 学科, 入学年度]

学籍番号	氏名	性別	学部	学科	入学年度
03A056	山田太郎	男	理学	数学	2003
03C003	桜井美香	女	文学	日文	2003
02B023	朝倉　南	女	法学	法律	2002
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
01A005	伊藤　靖	男	理学	数学	2001

R の次数は 6 で,「学籍番号」「氏名」「性別」「学部」「学科」「入学年度」という 6 つの属性を持つ．例えば,

(03A056, 山田太郎, 男, 理学, 数学, 2003)

はタップルである (個数を付けて「6-タップル」ともいう)． □

この例では属性「学籍番号」はタップルが違えば必ず異なっている．このように, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ が 2 つの任意の n-タップルである

¶tuple: 「組」の意．この一組のデータを 1 つの基本データとして扱い，**レコード**(record) と呼ぶこともある．

‖attribute: 「特質」の意．A_i のことをドメイン (データベースに記録される「もの」の集合の意, domain) と呼ぶことも，フィールド (field = レコードの構成要素の意) と呼ぶこともある．

とき，$\mathbf{a} \neq \mathbf{b}$ ならば $a_i \neq b_i$ であるような属性 A_i のことを $R[A_1, A_2, \ldots, A_n]$ の**キー**という．すなわち，キーとは，タップルを一意的に定める属性のことである．1つの属性だけではタップルが一意的に定まらないが複数の属性を考慮すると一意的に定まる場合，そのような属性の組をキーと呼ぶこともある．すなわち，属性の組 $(A_{i1}, A_{i2}, \ldots, A_{ik})$ が $R[A_1, A_2, \ldots, A_n]$ のキーであるとは，次が成り立つことである：

$$\mathbf{a} \neq \mathbf{b} \implies (a_{i1} \neq b_{i1}) \vee (a_{i2} \neq b_{i2}) \vee \cdots \vee (a_{in} \neq b_{in}).$$

● **関係代数の演算**　関係代数は，関係表の集合と，その上のいくつかの演算の組として定義される．

（a）　**合併演算**

これは，属性名のリストが一致している2つの関係表を合併して新しい関係表を求める演算である．すなわち，$R[A_1, \ldots, A_n]$ と $S[A_1, \ldots, A_n]$ とに対してその和集合

$$(R \cup S)[A_1, \ldots, A_n] := \{x \mid x \in R \text{ または } x \in S\}$$

を得る操作を合併という．R と S は $A_1 \times A_2 \times \cdots \times A_n$ の部分集合として扱っていることに注意されたい．

〔例 3.22〕　2つの関係表を合併する

$R[A, B, C]$

A	B	C
a	b	c
d	a	f
c	b	d

$\cup$

$S[A, B, C]$

A	B	C
b	g	a
d	a	f

$=$

$(R \cup S)[A, B, C]$

A	B	C
a	b	c
d	a	f
c	b	d
b	g	a

(d, a, f) のように同じ要素が2個以上あるとき，$R \cup S$ を集合とする限りはこの例のように要素 (d, a, f) は1つだけが残るが，実用的には，同じ要素が何個でも登録できるように考慮した集合 (多重集合という) を考えることが多い．また，集合として考える限りはタップルの出現順序は任意であるが，実際のシステムではある種の順序が付けられる．　□

（b） **差演算**

属性名一覧が一致している 2 つの関係表の一方に入っているが他方に入っていないようなタップルを求めて新しい関係表をつくる演算．すなわち，$R[A_1, \ldots, A_n]$ と $S[A_1, \ldots, A_n]$ とに対して，その差集合

$$(R - S)[A_1, \ldots, A_n] := \{x \mid x \in R \text{ かつ } x \notin S\}$$

を得る操作を差演算という．

〔例 3.23〕 2 つの関係表の差を求める

$R[A, B, C]$

A	B	C
a	b	c
d	a	f
c	b	d

$-$

$S[A, B, C]$

A	B	C
b	g	a
d	a	f

$=$

$(R - S)[A, B, C]$

A	B	C
a	b	c
c	b	d

□

（c） **直積演算**

2 つの関係表のそれぞれのタップルの組み合わせすべてを新しいタップルとする関係表を作る演算．すなわち，$R[A_1, \ldots, A_m]$ と $S[B_1, \ldots, B_n]$ とに対して，その直積集合

$$(R \times S)[A_1, \ldots, A_m, B_1, \ldots, B_n] := \{\, (x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n) \mid (x_1, x_2, \ldots, x_m) \in R \text{ かつ } (y_1, y_2, \ldots, y_n) \in S\}$$

を得る操作を直積演算という．

〔例 3.24〕 2 つの関係表の直積を求める

$R[A, B]$

A	B
a	b
d	a
c	b

$\times$

$S[A, C, D]$

A	C	D
1	3	5
6	2	8

$=$

$(R \times S)[A, B, A, C, D]$

A	B	A	C	D
a	b	1	3	5
a	b	6	2	8
d	a	1	3	5
d	a	6	2	8
c	b	1	3	5
c	b	6	2	8

□

（d） **射影演算**

射影演算は，1 つの関係表から特定の属性だけを抜き出して新しい関係表を作る演算である．すなわち，$R[A_1, \ldots, A_m]$ と，属性番号 $1 \leqq i_1, \ldots, i_k \leqq m$ に対して，R の第 i_1 成分，…， 第 i_k 成分の射影

$$\pi_{i_1,\ldots,i_k}(R)[A_{i_1}, A_{i_2}, \ldots, A_{i_k}] := \{\, (x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \mid (x_1, x_2, \ldots, x_m) \in R\}$$

を得る操作を射影演算という**．属性の添字の代わりに，属性名そのものを書いてもよい：

$$\pi_{A_{i_1},A_{i_2},\ldots,A_{i_k}}(R)[A_{i_1}, A_{i_2}, \ldots, A_{i_k}]$$

〔例 3.25〕 関係表の射影を求める

$S[A, C, D]$

A	C	D
b	g	a
d	a	f

$\xrightarrow{\pi_{1,3}}$

$\pi_{1,3}(S)[A, D]$

A	D
b	a
d	f

$R[A, B]$

A	B
a	b
d	a
c	b

$\xrightarrow{\pi_{A,A,B}}$

$\pi_{A,A,B}(R)[A, A, B]$

A	A	B
a	a	b
d	d	a
c	c	b

□

（e） **選択演算**

選択演算は，1 つの関係表から，ある特定の条件を満たすようなタップルだけを抜き出して新しい関係表を作る演算である．「条件」としては，属性の間の関係として記述できるようなものだけを考える．R の元 (エンティティ*とも呼ばれる) を定数と考えよう．そのような定数は「数」や「文字」や「文字列」などである．「属性名を表わす番号」と区別するために，エンティティは ‘ ’ で囲んで表わすことにする．このような定数や「属性名を表わす番号」(属性名そのものでもよい) に比較演算子 $<$, $\leqq$, $=$, $\neq$, $>$, $\geqq$ を適用した形の論理式，あるいは，そういった論理式を論理演算子 $\wedge$ (論理積：かつ)，$\vee$ (論理和：または)，

**射影：projection．π は p に相当するギリシャ文字である．

*entity: 「実際のもの」「実体」の意．

¬ (論理否定：··· でない) で結んでできる論理式を ρ としよう (**条件式**と呼ぶことにする†)．$R[A_1,\ldots,A_m]$ と条件式 ρ (ただし，ρ の中に現れる属性番号は 1 以上 m 以下であること) に対し，R のタップルのうち，条件 ρ を満たすものだけを抜き出した集合

$$\sigma_\rho(R)[A_1,\ldots,A_m] := \{\,(x_1,\ldots,x_m) \in R \mid (x_1,\ldots,x_m) \text{ は } \rho \text{ を満たす}\,\}$$

を求める演算を選択という‡．$(x_1,\ldots,x_m)$ が ρ を満たすとは，論理式 ρ の中に現れるそれぞれの属性番号 i_k に x_{i_k} を代入して得られる論理式が成り立つことをいう．

〔例 3.26〕　関係表に選択演算を適用して新しい関係表を求める

$R[A,B,X,Y,Z]$

A	B	X	Y	Z
a	b	2	3	2
d	b	3	5	6
d	b	1	3	7
d	c	5	7	5
c	a	3	8	2
b	b	7	6	5

のとき

$\sigma_{3=5}(R)[A,B,X,Y,Z]$

A	B	X	Y	Z
a	b	2	3	2
d	c	5	7	5

$R_{(A='d')\wedge(X\ \ '2')}[A,B,X,Y,Z]$

A	B	X	Y	Z
d	b	3	5	2
d	c	5	7	5

□

以上(a)〜(e)の５つの演算があれば，他のどんな関係表操作も記述することができる．以下では，その他の関係表操作のうち，頻繁に使われる重要なものいくつかを(a)〜(e)を使って表わしてみよう．

---

†条件式は，正確には次のように定義される：

1) エンティティを ‘ ’ でくくったもの，および属性名 (属性番号でもよいが，属性名と属性番号は混在させない) を定数という．

2) a,b がそれぞれ定数なら，$a<b$, $a\quad b$, $a=b$, $a\neq b$, $a>b$, $a\quad b$ はそれぞれ条件式である．

3) α,β がそれぞれ条件式なら，$\neg\alpha$, $\alpha\wedge\beta$, $\alpha\vee\beta$ はそれぞれ条件式である．

4) 1) 〜 3) で定まるものだけが条件式である．

〔注〕実際の関係データベース操作言語 (例えば，SQL) では，選択演算は単なる比較演算だけでなく，計算機能が組み込めるように拡張されている．

‡選択：selection．σ は s に相当するギリシャ文字である．

（f） **共通集合演算**

2つの関係表に同時に入っているようなタップルだけを求めて新しい関係表を作る．すなわち，$R[A_1,\ldots,A_n]$ と $S[A_1,\ldots,A_n]$ とに対して，その共通集合

$$(R\cap S)[A_1,\ldots,A_n] := \{x \mid x\in R \text{ かつ } x\in S\}$$

を得る操作を共通集合演算という．$R\cap S$ は次のように表わすことができる：

$$R\cap S = R-(R-S)$$

〔例 3.27〕 2つの関係表の共通部分を求める

$R[A,B,C]$

A	B	C
a	b	c
d	a	f
c	b	d

$\cap$

$S[A,B,C]$

A	B	C
b	g	a
d	a	f

$=$

$(R\cap S)[A,B,C]$

A	B	C
d	a	f

□

（g） **商演算**

商演算は，直積演算とは逆に，1つの関係表のタップルの末尾が別の関係表のタップルであるとき，その末尾を削除したタップルを求める演算である．すなわち，関係表 $R[A_1,\ldots,A_m,B_1,\ldots,B_n]$ の末尾の属性 $B_1,\ldots,B_n$ が $S[B_1,\ldots,B_n]$ の属性 $B_1,\ldots,B_n$ と等しいとき，

$$(R\div S)[A_1,\ldots,A_m] := \{\,(x_1,\ldots,x_m) \mid \text{すべての } (y_1,\ldots,y_n)\in S \text{ に対して } (x_1,\ldots,x_m,y_1,\ldots,y_n)\in R\}$$

を，R を S で割った商という．$R\div S$ は次のように表わすことができる：

$$R\div S = \pi_{1,2,\ldots,m}(R) - \pi_{1,2,\ldots,m}(\pi_{1,2,\ldots,m}(R)\times S - R)$$

〔例 3.28〕 R を S で割った商を求める

$R[A,B,C]$

A	B	C
a	b	2
a	c	1
c	b	2

$\div$

$S[C]$

C
2

$=$

$(R\div S)[A,B]$

A	B
a	b
c	b

□

（h） 結合演算

結合は，2つの関係表が与えられたとき，それぞれの関係表ごとに属性を指定し，その属性同士の間で特定の条件が成り立つようなタップルだけを選ぶ演算である．このようなやり方で2つの関係表を結合して新しい関係表を作ることは，よく行われる重要な演算である．例えば，学生データベース内の住所情報に関する関係表と成績情報に関する関係表とを合併して住所と成績の両方を含む関係表を作る際には，同じ学生の住所と成績を一緒にしなければならない (このとき，同一人物であるかどうかを特定するためには，同姓同名の場合もあるかもしれないので,「名前」が等しいかどうかではなく,「キーとなるような属性」が等しいかどうかで判断しなければならない)．$R[A_1,\ldots,A_m]$ と $S[B_1,\ldots,B_n]$ を関係表，$1 \leqq i \leqq m$ を R の属性番号 (属性名 A_i でもよい)，$1 \leqq j \leqq n$ を S の属性番号 (属性名 B_j でもよい)，θ を比較演算とするとき，

$$\begin{aligned}&(R \bowtie_{i\,\theta\,j} S)[A_1,\ldots,A_m,B_1,\ldots,B_n]\\ &\qquad := \{\,(x_1,\ldots,x_m,y_1,\ldots,y_n) \in R \times S \mid x_i\,\theta\,y_j\}\end{aligned}$$

を R と S の θ による結合という．$R \bowtie_{i\,\theta\,j} S$ は

$$R \bowtie_{i\,\theta\,j} S = \sigma_{i\,\theta\,(m+j)}(R \times S)$$

と表わすことができる[§]．

〔例 3.29〕 関係表の条件付き結合

$R[A,B]$

A	B
3	1
6	2

$S[A,C,D]$

A	C	D
1	2	4
3	5	7
2	8	3

のとき，

$(R \bowtie_{A<C} S)[A,B,A,C,D]$

A	B	A	C	D
3	1	6	5	7
3	1	2	8	3
6	2	2	8	3

$(R \bowtie_{(1=1)\vee(1+2<3)} S)[A,B,A,C,D]$

A	B	A	C	D
3	1	3	5	7
6	2	6	5	7

□

[§]実際の関係データベースでは，比較演算 θ は 2 項演算に限らず多数の属性の間の関係式を表わすことが許されている．

（i） 自然結合演算

結合の中でもとくに重要なのが自然結合である．自然結合演算では,「関係表 R の属性と関係表 S の属性のうち,「属性が同じところ (すなわち，属性名が等しいところ) は値も等しくなければならない」(そのような属性の対がいくつあろうとも，そのすべてについてそうでなければならない) という条件を満たすように結合する．すなわち，$R[A_1,\ldots,A_m]$ と $S[B_1,\ldots,B_n]$ を関係表とし，$1 \leqq i_1, i_2, \ldots, i_k \leqq m$ を R の属性番号，$1 \leqq j_1, j_2, \ldots, j_k \leqq n$ を S の属性番号とする．もし，$A_{i_1} = B_{j_1}$，$A_{i_2} = B_{j_2}$，$\ldots$，$A_{i_k} = B_{j_k}$ (つまり，R と S の対応する k 組の属性が等しい) であるなら，

$$(R \bowtie S)[A_{p_1},\ldots,A_{p_q}] := \{(x_{p_1},\ldots,x_{p_q}) \mid (x_1,\ldots,x_{m+n}) \in R \times S,\ x_{i_1} = x_{m+j_1},\ \ldots,\ x_{i_k} = x_{m+j_k}\}$$

により定義される関係表を R と S の自然結合という．ただし，$p_1,\ldots,p_q$ は $1,\ldots,m,m+1,\ldots,m+n$ から $m+j_1,\ldots,m+j_k$ を除いたものであり，$q = m+n-k$ である．$R \bowtie S$ は

$$R \bowtie S = \pi_{p_1,\ldots,p_q}(\sigma_{(i_1=m+j_1)\wedge(i_2=m+j_2)\wedge\cdots\wedge(i_k=m+j_k)}(R \times S))$$

と表わすことができる．

〔例 3.30〕 関係表の自然な結合

$R[A,B,C]$

A	B	C
a	p	1
b	r	2
a	q	3
c	q	5
a	r	1

$\bowtie$

$S[A,C,D]$

A	C	D
a	2	4
b	2	7
a	1	3
c	6	8

$=$

$(R \bowtie S)[A,B,C,D]$

A	B	C	D
a	p	1	3
a	r	1	3
b	r	2	7

□

関係データベースを操作するどんな演算も，上述の基本的演算を組み合わせれば記述することができるが，現実に用いられている関係データベース操作言語では，もっと使いやすいように様々な操作・演算が用意されている．この節で学んで欲しかったのは，きちんとした理論的バックグラウンドがあることが現実にも良いシステムを生み出すということである．

● **一貫性制約**　関係データベースでは実世界のデータ (エンティティ) の間の関係を関係表で表わすが，例えば，学生データベースにおいて，

・ある学生の出身地が 2 箇所ある
・ある科目の平均点が最高点より高い
・ある学生の年齢が 2002 歳である

というようなことがあってはデータベースとしての信頼性が落ちてしまう．そこで，データ構造に一定の制約を加え，できるだけ完全無欠に近い状態に近づける努力をする．このような制約を**一貫性制約**と呼ぶ.

● **正規形**　関係データベースの基本要素である関係表はドメイン (データの集合) の直積の部分集合である．もし，ドメイン自身がさらに単純なデータ集合の直積になっていたり，集合であったり，もっと複雑なデータ構造であったりすると，取り扱いが困難になる．そこで，関係データベースを構築する際には，ドメインはできるだけシンプルなものでなければならないとする．すなわち，関係表として記述される基本データ (ドメインの元) がそれ以上分解できないものであるとき，そのような制約を満たしているものを**第 1 正規形**と呼ぶ．その他に，第 2 正規形，第 3 正規形と呼ばれるものもあるが，本書の範囲を超えるので省略する.

記号のまとめ (3.7 節)

$R[A_1, \ldots, A_n]$	$A_1, \ldots, A_n$ を属性とする関係表
$(R \cup S)[A_1, \ldots, A_n]$	R と S の合併
$(R - S)[A_1, \ldots, A_n]$	R と S の差
$(R \times S)[A_1, \ldots, A_n, B_1, \ldots, B_m]$	R と S の直積
$\pi_{i_1,\ldots,i_k}(R)[A_{i_1}, \ldots, A_{i_n}]$	R の射影
$\sigma_\rho(R)[A_1, \ldots, A_n]$	条件 ρ による R の選択
$(R \cap S)[A_1, \ldots, A_n]$	R と S の共通部分
$(R \div S)[A_1, \ldots, A_n]$	R を S で割った商
$(R \bowtie_{i\,\theta\,j} S)[A_1, \ldots, A_n, B_1, \ldots, B_m]$	R と S の条件 θ による結合
$(R \bowtie S)[A_{p_1}, \ldots, A_{p_q}]$	R と S の自然結合

3.7 節 理解度確認問題

問 3.53 X を男性の集合，Y を女性の集合とするとき，

$$R := \{\,(x,y,z) \mid z \text{ は父 } x \text{ と母 } y \text{ の子である}\,\} \subseteq X \times Y \times (X \cup Y)$$

は，“両親と子” を表す 3 項関係である．

（1） $R := \{\,(a,b,x), (a,b,y), (a,c,z), (d,e,u), (d,e,v)\,\}$ とするとき，$\pi_{2,3}(\sigma_{1='a'}(R))$ を求めよ．

（2） 上述の 3 項関係 R に対し，“父子関係” を表す 2 項関係 F，“母子関係” を表す 2 項関係 M，“兄弟関係 (異母兄弟，異父兄弟を含む)” を表す 2 項関係 B_1，“同じ両親をもつ兄弟関係 ”を表す 2 項関係 B_2 それぞれを，関係データベースの基本演算を用いて記述せよ．

（3） 逆に，F, M, B_1, B_2 のうち必要なものを使って R を表わせ．

問 3.54 以下の条件すべてを満たす関係表 $R[A,B,C]$ と $S[B,C]$ をすべて求めよ．

(a) 2 つの関係表に現れるエンティティ(実データ) は整数 $1 \sim 9$ だけである．

(b) R のタップルは 4 個だけ，S のタップルは 2 個だけである．

(c) R のタップルの 1 つは $(1,2,3)$ である．

(d) $\pi_1(\sigma_{A\leqq'2'}(R)) \times S = R$ および $\sigma_{(B\geqq'3')\wedge(C\leqq'2')}(\pi_{B,C}(R)) \cap S = \pi_{2,3}(R)$ が成り立つ．

問 3.55 $R[A,B]$, $S[A,C,D]$ はそれぞれ次のような 2 項関係，3 項関係とする．関係表として表わせ．

$$R := \{(a,x), (b,y), (b,x), (d,z)\}, \quad S := \{(a,1,2), (b,3,2), (c,5,4)\}$$

次の関係表をそれぞれ求めよ：

（1） $T[A,B,C,D] := R[A,B] \bowtie S[A,C,D]$

（2） $U[A,C] := \pi_{1,4}(\sigma_{(2='x')\wedge(4\geqq5)}(R \times S))$

問 3.56 $R[A_1, A_2, \cdots, A_n]$ を関係表とし，X, Y を $\{A_1, A_2, \cdots, A_n\}$ の部分集合とする．R の元であるどの 2 つのタップルについても「X に属する属性がすべて一致していたら Y に属する属性もすべて一致している」が成り立つならば，Y は X に**関数従属**であるといい $X \to Y$ と書く．$\to$ は推移律を満たすことを示せ．

問 3.57　関係データベースについて，次の各問に答えよ．

（1）　次のような学生に関する関係表 S と T がある．

S(成績情報)$[N, S, A, G, C]$

番号 N	氏名 S	代数 A	幾何 G	解析 C
E1J020	西田	89	93	88
E2J002	浅野	77	80	57
E2J054	小川	45	70	20
E2J102	石田	98	65	78
E0J037	森高	85	55	74
E9J078	小泉	60	35	66

T(一般情報)$[S, N, J]$

氏名 S	番号 N	住所 J
西田	E1J020	杉並区
森高	E0J037	新宿区
石田	E2J102	中野区
細川	E2J054	新宿区
西田	E1J098	豊島区
浅野	E0J123	新宿区

（a）　$\sigma_{A \geqq '70'}(S)$，$\pi_S(\sigma_{J='\text{新宿区}'}(T))$，$\pi_{2,1}(S \bowtie_{N=N} T)$，$S \bowtie T$ をそれぞれ求めよ．

（b）　どの科目の成績も 60 点以上である学生の氏名，番号，住所だけをこの順で属性とする関係表を求めるための関係代数の式を示せ．

（2）　$R[A, B]$，$S[A, B]$ のとき，$\sigma_{A=A}(R \times S) \bowtie \sigma_{B=B}(R \times S)$ を簡単にせよ．

問 3.58　次のような学生に関する関係表 S と T がある．

S(成績)$[N, A, G, C, M]$

番号 N	代数 A	幾何 G	解析 C	応数 M
E9J020	80	89	93	88
E0J002	67	77	80	57
E0J054	50	45	70	20
E0J102	86	98	65	78
E1J037	63	85	55	74
E1J078	54	60	35	66

T(その他)$[S, N, J, K]$

氏名 S	番号 N	住所 J	出身地 K
西田	E9J020	杉並区	東京都
森高	E1J037	新宿区	山梨県
石田	E0J102	中野区	京都府
細川	E0J054	新宿区	東京都
小泉	E1J078	豊島区	北海道
浅野	E0J002	新宿区	京都府

次の各問に答えよ．

（1）　$\sigma_{J='\text{新宿区}'}(T)$，$\pi_{A,C,C}(S)$，$(S \times T) \div T$ をそれぞれ求めよ．

（2）　$\sigma_{(A>'60')\wedge(G>'60')\wedge(C>'60')\wedge(M>'60')}(S) \bowtie T$ を求めよ．

（3）　住所が新宿区にあるか出身地が東京都である者で，「応数」の成績が 60 点以上である者だけを抽出し，その氏名と学生番号だけをこの順で属性とする関係表を求めるための関係代数の式を示せ．