

関係モデル

データベース論 I 第4回

URL <http://homepage3.nifty.com/suetsuguf/>

作成者 末次文雄 ©

目次

1. 関係モデルの特徴
2. 用語の説明
3. データの正規化
4. 関係の集合演算
5. 関係の論理式
6. レポート課題
7. 参考書ほか

1. 関係モデルの特徴

1970年にアメリカの E.F.コッド博士がアメリカ計算機学会誌に発表した論文

「A Relational Model of Data for Large Shared Data Banks」
により初めて世の中に紹介された。

この論文で次の事項に関する明快なアイデアを提案した。

- ・ **関係データベース**
- ・ **正規形**
- ・ **関係代数と関係論理**

1. (続き)

- Relational data model
- 1987年、データベース言語のSQLが国際標準
- 明解な**数学的基盤**の上に成り立つモデル
- フラットな表形式という非常に**単純**な表現形式
- モデリング時に、**データ構造が固定されない**ので
データ独立性が高い
(表同士の関連(データ構造)を規定する定義が無い)
- データ利用時に、集合論に基づいた関係操作により
動的に自由にデータの結合や分解が行える

2. 用語の説明

- ・ リレーショナルモデルでは、
 - ファイル（関係、リレーション）
 - レコード（組、タプル）
 - フィールド（属性）
 - ・ リレーショナルデータベースでは、
 - ファイル（表、テーブル）
 - レコード（行、ロー）
 - フィールド（列、カラム）
- ー 通常、表が複数集まって、意味のあるDBとなる

2. (続き)

- **関係**とは、
データ項目間の関係を言う
(例:社員番号－氏名－所属－業務)
- **タプル**とは？
関係するデータの一組を言う
- **属性**とは？
データ項目のことである

2. (続き)

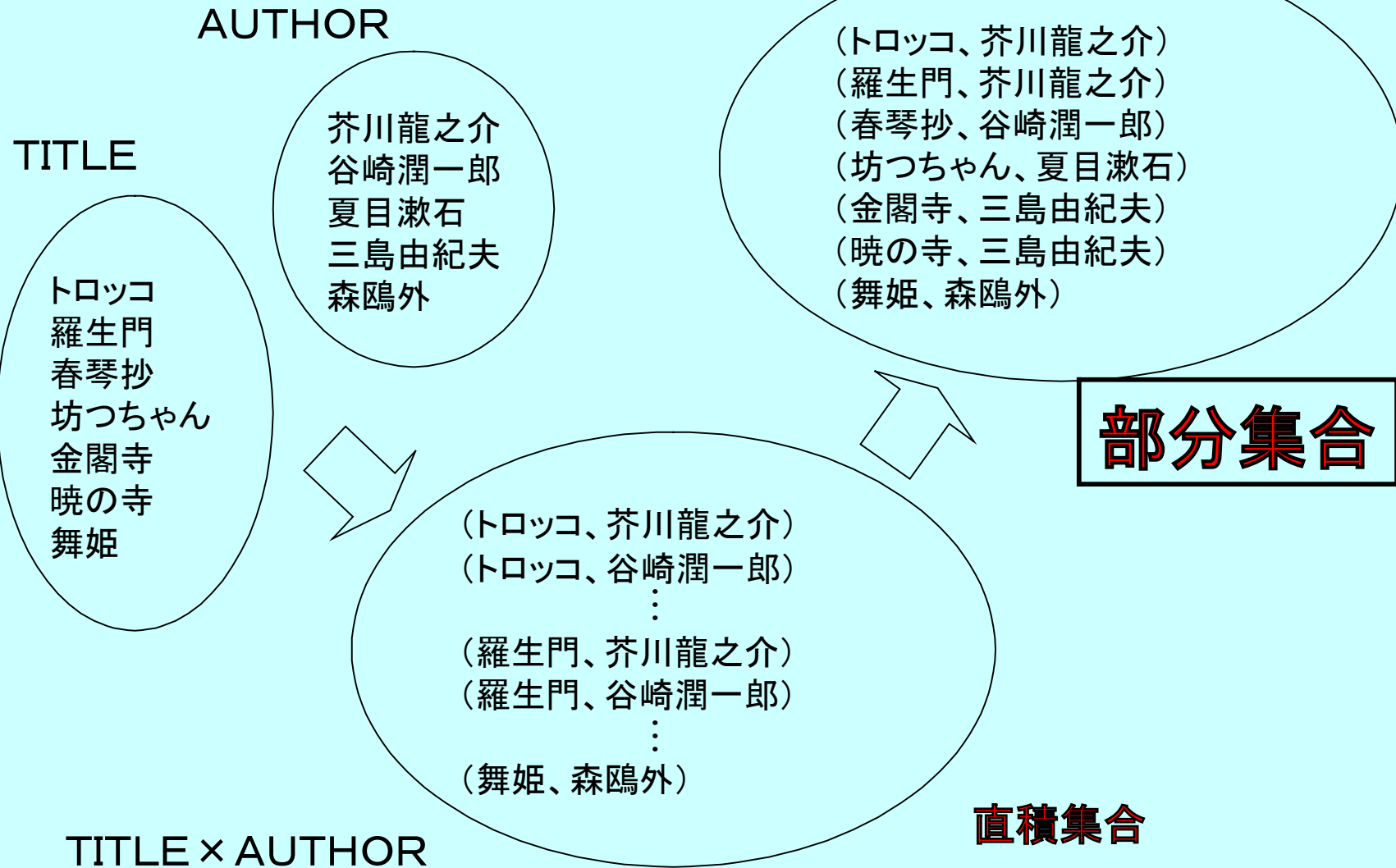
- **集合**とは？ (set)
 - 互いに識別できる‘もの’の集まり。
 - この集まりの中から、任意に二つの‘もの’
を取出したとき、同じか違うか判別できる。
 - ある‘もの’を持ってきて、それがその集合に
入るか入らないかを判別できること。
- 集合の論理表現——— ‘1から5までの整数の例’
 - 外延的表現(集合を構成する**要素の列挙**)
集合 $A = \{1, 2, 3, 4, 5\}$
 - 内包的表現(集合を構成する**要素の条件**)
集合 $A = \{x \mid 1 \leq x \leq 5 \wedge x \in \text{integer}\}$

2. (続き)

- 関係モデルでの集合とは？
 - **タプルの集まり**を言う
(relational set)
- 一つの集合からある部分を取出した時、それも集合である。(**部分集合**)
- 関係(リレーション)は、複数の集合の **直積集合の部分集合**である。

図書館データベースの例

関係 (relation)



関係(リレーショナル)データベースの構造

関係(リレーション)

従業員

定義域名、
属性(アトリビュート)

定義域の数(次数)

関係の組(タ
プル)

組の数(濃度)

定義域(ドメイン) : 属性値のとりうる範囲

社員番号	氏名	所属	業務
110	吉田茂	10	社長
320	中曽根康弘	20	営業
150	岸信介	30	技師
220	佐藤栄作	30	設計
140	池田隼人	40	事務
260	田中角栄	10	設計

関係演算 (Viewの事例)

T1 従業員

社員番号	氏名	所属
110	吉田茂	10
320	中曽根康弘	20
150	岸信介	30
220	佐藤栄作	20
140	池田隼人	40
260	田中角栄	10

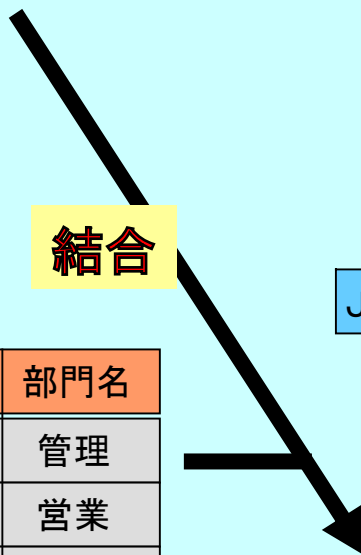
ST

社員番号	氏名	所属
110	吉田茂	10
260	田中角栄	10

選択



結合



射影



PT

氏名
吉田茂
中曽根康弘
岸信介
佐藤栄作
池田隼人
田中角栄

T2 部門

部門番号	部門名
10	管理
20	営業
30	開発
40	製造

JT

社員番号	氏名	部門名
110	吉田茂	管理
320	中曽根康弘	営業
150	岸信介	開発
220	佐藤栄作	営業
140	池田隼人	営業
260	田中角栄	管理

3. データの正規化

帳票、画面といった一組の「情報の製品」から適正な単位のデータの組として「データの部品」を作成する作業である。

単純化して言えば、データのダブリを取り除いて冗長さを無くすことである。

- ・メリット: データの更新、追加、削除が弾力的になり、データ間の整合を保ちやすい。

①非正規形

現行の帳票、画面のままで、データベース作成すると、冗長。

社員		組織 コード	組織 名	管理 者名	スキル コード	スキル 名	経験 年数	スキル レベル
社員 番号	社員 氏名							

↑
キー

(例) 社員スキル管理表

繰り返し項目

②第1正規形

- ・これ以上分解できないデータ項目のみで構成。
- ・繰り返しのデータ項目を持たせない。
- ・直積空間で、必ず単独の点として識別できる。

社員＝社員番号＋社員氏名＋組織コード＋
組織名＋管理者名

保有スキル＝社員番号＋スキルコード＋スキル名＋
経験年数＋スキルレベル

スキルコードに従属

③第2正規形

- ・キーの一部分に従属しているデータ項目を分離。
- ・キー以外の全てのデータ項目がキーに従属(完全従属)

社員 = 社員番号 + 社員氏名 + 組織コード +
組織名 + 管理者名 ← 組織コードに従属(推移的従属)

保有スキル = 社員番号 + スキルコード + 経験年数
+ スキルレベル

スキル = スキルコード + スキル名

④第3正規形

主キー

外部キー

- ・キー以外のデータ項目間の従属関係を分離。
- ・推移的従属データ項目の分離

社員＝社員番号＋社員氏名＋組織コード

組織＝組織コード＋組織名＋管理者名

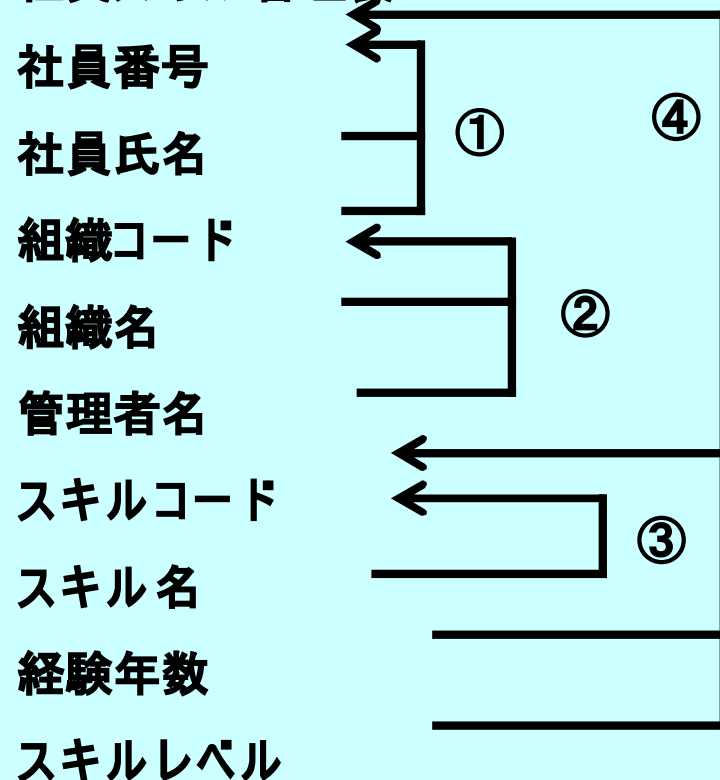
保有スキル＝社員番号＋スキルコード＋経験年数
＋スキルレベル

スキル＝スキルコード＋スキル名

⑤ データ正規化の簡便な方法

- ・データ項目間の従属性に着目し、
- ・全てのデータ項目間の従属関係を洗出す
- ・**従属関係にあるデータ項目のグループを**独立させる。

社員スキル管理表＝



①社員＝社員番号＋社員氏名＋組織コード

②組織＝組織コード＋組織名＋管理者名

③スキル＝スキルコード＋スキル名＋

④保有スキル＝社員番号＋スキルコード＋
経験年数＋スキルレベル

4. 関係の集合演算

- この章の目的
集合論を修得することではなくて、
関係データベースの利用(**データ操作**)では、
どのようなデータ操作が可能であるのかを
理解する。
- データ操作言語(SQLの内、**DML**)は、
関係の集合演算式に基づいて開発された。

(補足1) 集合の記号

- 集合記号
 - $\subset$ 集合の包含 $A \subset B$
 - $\cup$ 集合の和 $A \cup B$
 - $-$ 集合の差 $A - B$
 - $\cap$ 集合の積 $A \cap B$
 - $\times$ 集合の直積 $A \times B$
- 論理記号
 - $\wedge$ (and、および)
 - $\vee$ (or、または)
 - $\neg$ (not、否定、～でない)
 - $\forall$ (全称記号、**全ての、任意の**、any)、
 - $\exists$ (存在記号、～が存在する、～が有る、exist)
 - $\in$ (集合の要素、elemen)

(補足2) 集合の表現

- 表現法1 $\{ \underline{n} \mid \underline{n \text{ は自然数}} \}$ $\{ n \mid n=1, 2, 3\cdots \}$

集合の要素

集合の要素になる 為の 条件

- 表現法2 $R_n = \{ \underline{n} \mid n=1, 2, 3\cdots \}$

左辺は、集合の要素の簡略表現

- 表現法3 $R_1(t)$ 関数型 集合の要素 t が 集合 R_1 に属する

(補足3) 集合の表現法の例示

“ $A \cup B$ は、集合A、集合Bそれぞれの要素 x 、 y からなる和集合である。”

$$A \cup B = \{ (x,y) \mid x \in A \vee y \in B \}$$

4. (関係の集合演算)の構成

4. 1 一般的な集合演算

- ①和(集合演算)、Union
- ②差(集合演算)、Difference
- ③積(集合演算)、Intersection
- ④直積(集合演算)、Cartesian Product

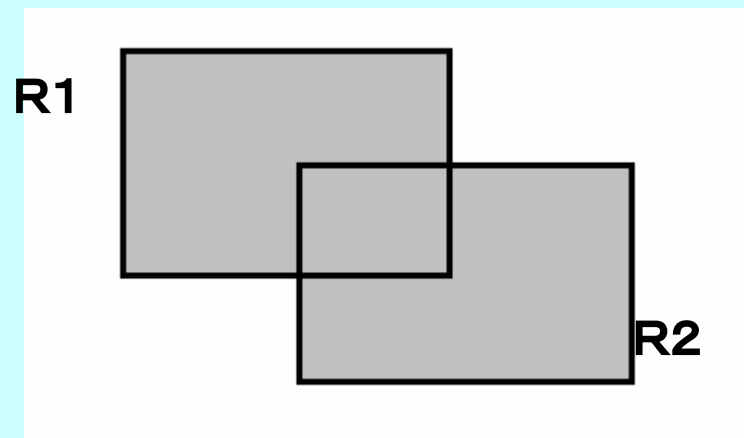
4. 2 関係に特有な集合演算(関係演算)

- ⑤選択(演算)、Selection
- ⑥射影(演算)、Projection
- ⑦結合(演算)、Join

4. 1 一般的な集合演算

①和(集合演算)、Union

- 二つの集合から、全ての要素を集めたもの。
- 表で言えば、二つの表の行を重複を無くして、全て集めたもの。



(続き)①和(集合演算)、Union

$$R_1 \cup R_2 = \{t \mid t \in R_1 \vee t \in R_2\}$$

R_1 、 R_2 は関係
 t は組(タプル)

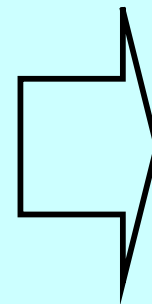
または

R1

名前	年齢
Aさん	19
Bさん	18
Cさん	20

R2

名前	年齢
Cさん	20
Yさん	21

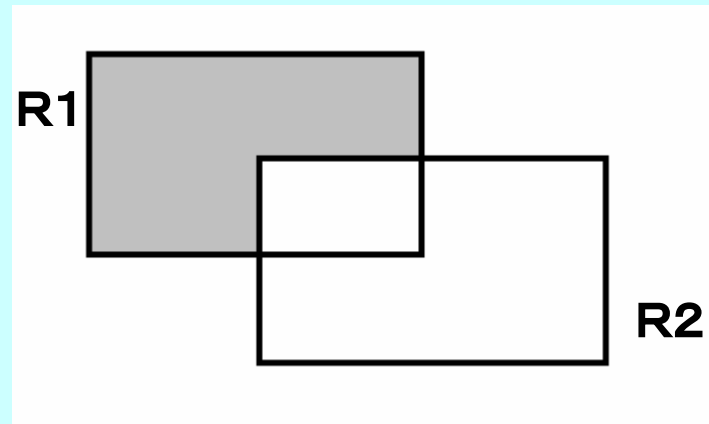


R1 ∪ R2

名前	年齢
Aさん	19
Bさん	18
Cさん	20
Yさん	21

② 差(集合演算)、Difference

- 基準の集合から、他の集合の要素を差し引いたもの
- 表で言えば、表から行の単位で引き算をする。



(続き) 差 (集合演算)、Difference

$$R_1 - R_2 = \{t \mid t \in R_1 \wedge \neg(t \in R_2)\}$$

かつ

~でない (否定)

R1

名前	年齢
Aさん	19
Bさん	18
Cさん	20

R2

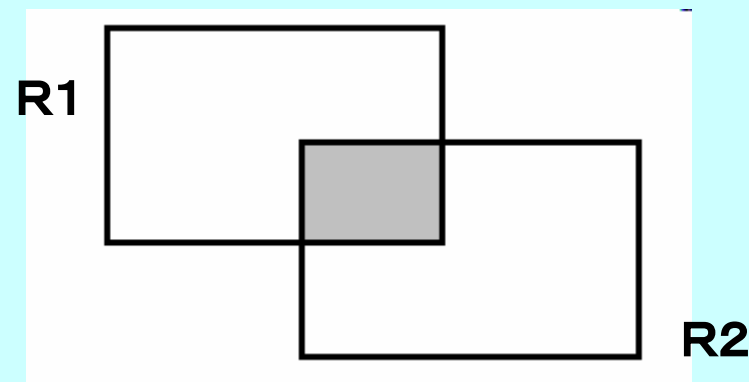
名前	年齢
Cさん	20
Yさん	21

R1 - R2

名前	年齢
Aさん	19
Bさん	18

③積（集合演算）、Intersection

- 二つの集合から、両方の集合に含まれる要素を集めたもの。
- 表で言えば、二つの表から共通の行を取出す。



(続き) ③積(集合演算)、Intersection

$$R_1 \cap R_2 = \{t \mid t \in R_1 \wedge t \in R_2\}$$

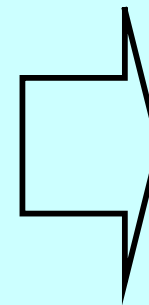
かつ

R1

名前	年齢
Aさん	19
Bさん	18
Cさん	20

R2

名前	年齢
Cさん	20
Yさん	21

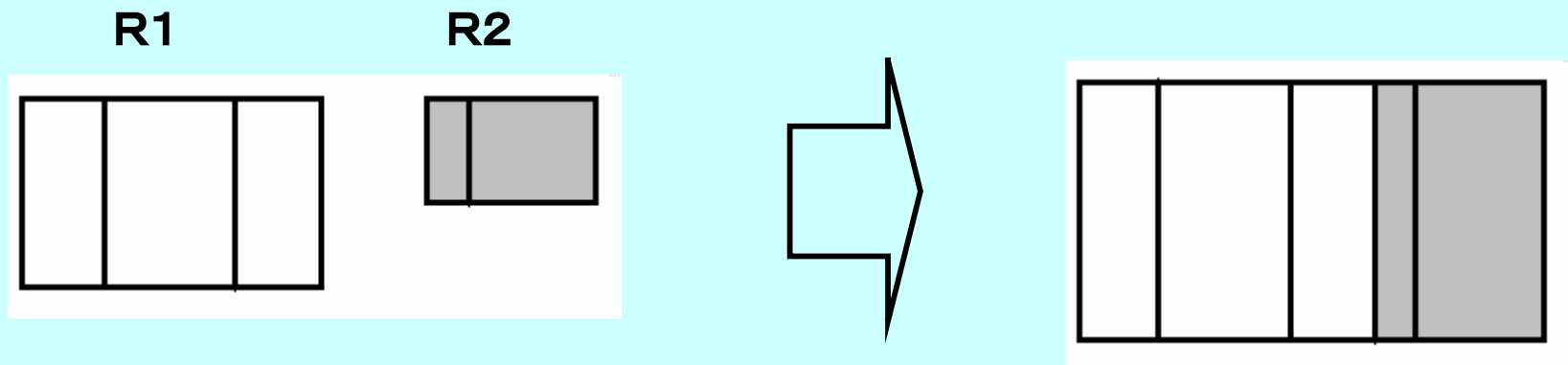


R1 ∩ R2

名前	年齢
Cさん	20

④直積（集合演算）、Cartesian Product

- 二つの集合の要素について、その要素同士の全ての組合せを求めたもの。
- 表で言えば、二つの表の行について、全ての組合せをもとめたもの。



(続き)

④直積(集合演算)、Cartesian Product

$$R_1 \times R_2 = \{(t_1, t_2) \mid t_1 \in R_1 \wedge t_2 \in R_2\}$$

タプルの連結

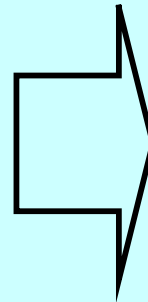
かつ

R1

名前	年齢
Aさん	19
Bさん	18
Cさん	20

R2

住所
広島
山口



R1 × R2

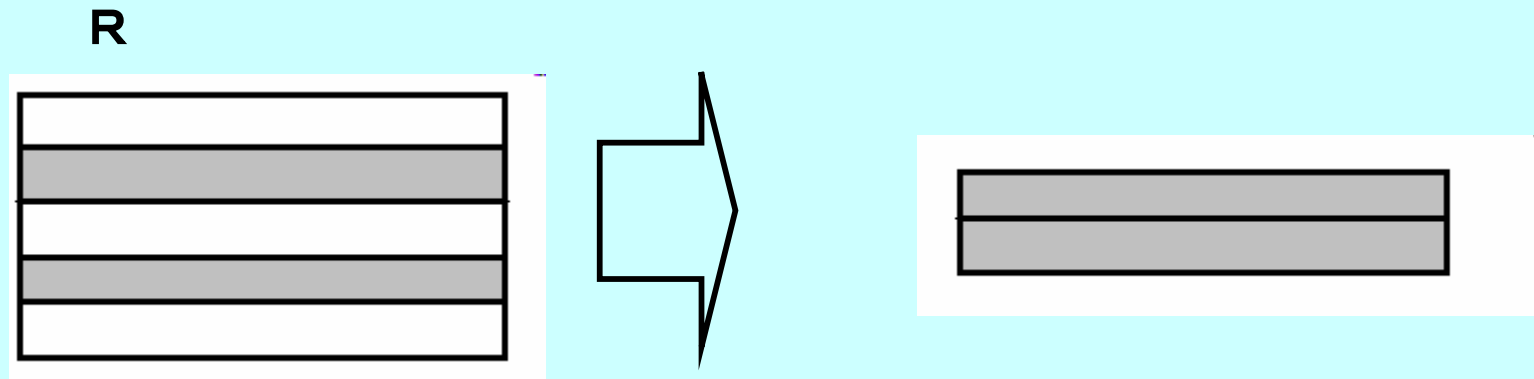
名前	年齢	住所
Aさん	19	広島
Bさん	18	広島
Cさん	20	広島
Aさん	19	山口
Bさん	18	山口
Cさん	20	山口

4. 2 関係に特有な集合演算(関係演算)

⑤選択(演算)、Selection

- ・集合から、指定した条件に合ったものを選択する。
- ・表で言えば、指定した条件を満足する行を取出す。
- ・条件には、比較演算子を使う。

(= ≠ > < ≥ ≤)



(続き) ⑤選択(演算)、Selection

$$R[A \theta B] = \{t \mid t \in R \wedge (t[A] \theta t[B])\}$$

かつ

常数も可

A、Bは属性集合
 θ は比較演算子
t は組(タプル)

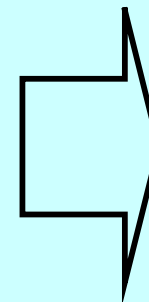
R

名前	年齢
Aさん	19
Bさん	18
Cさん	20
Xさん	19
Yさん	21

条件

年齢 $\geq$ 20

R[年齢 $\geq$ 20]

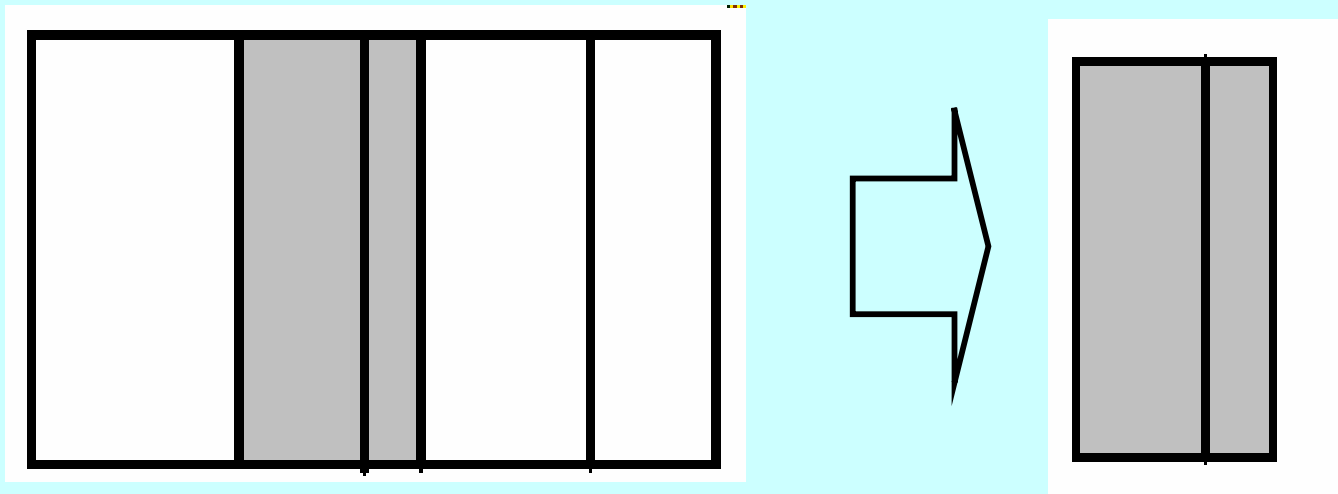


名前	年齢
Cさん	20
Yさん	21

⑥射影(演算)、Projection

- 集合から、ある属性を取出す。
- 表で言えば、ある列をとりだす。

R



(続き) ⑥射影(演算)、Projection

$$R[A] = \{ t[A] \mid t \in R \}$$

A は属性

θ は比較演算子

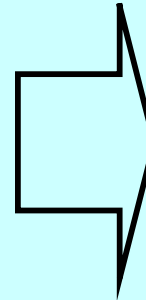
t は組(タプル)

R

名前	年齢	住所
Aさん	19	広島
Bさん	18	山口
Cさん	20	岡山
Xさん	19	広島
Yさん	21	山口

属性

[住所]



R[住所]

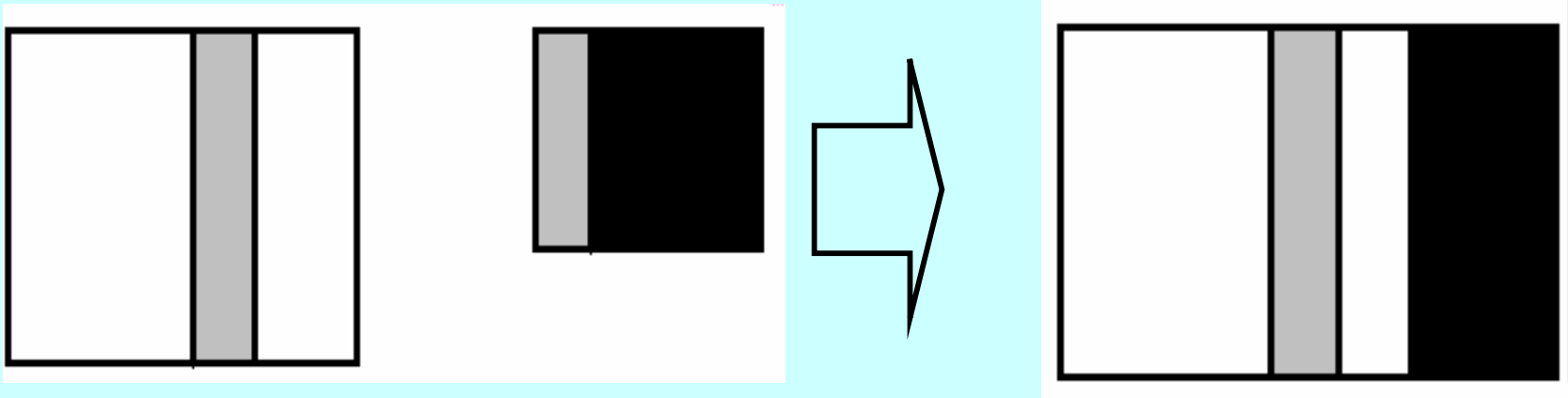
住所
広島
山口
岡山

⑦結合（演算）、Join

- 複数の集合から、指定する属性同士を、指定する条件を満たす組を集めて、新たな集合を作り出す演算。
- 条件には、比較演算子を使う。
- 表で言えば、共通の値を持つ列を介して、新たな表を作り出す。（指定する条件が等号の場合）

R1

R2



(続き) ⑦結合(演算)、Join

$$R_1[A \theta B]R_2 = \{(t_1, t_2) \mid t_1 \in R_1 \wedge t_2 \in R_2 \\ \wedge (t_1[A] \theta t_2[B])\}$$

A、B は属性

θ は比較演算子

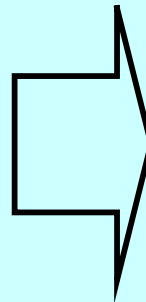
t は組(タプル)

R1

名前	学科コード
Aさん	100
Bさん	200
Cさん	130
Xさん	200
Yさん	100

R2

学科名称	学科番号
電子	100
機械	130
情報	200



R1[学科コード=学科番号]R2

名前	学科コード	学科名称
Aさん	100	電子
Bさん	200	情報
Cさん	130	機械
Xさん	200	情報
Yさん	100	電子

5. 関係の論理式

関係を、1階述語論理で表したものを**関係論理式**という。(First-order Predicate Logic)

(Relational Calculus Expression)

- ・関係集合が満たすべき条件を言い表したものの。
- ・表現ルール

$R_1(t)$ — — — 組 t が 関係 R_1 に属する

5. (続き) 関係の論理式

表現事例:

- ・和演算 $R_1 \cup R_2 = \{t \mid R_1(t) \vee R_2(t)\}$
- ・差演算 $R_1 - R_2 = \{t \mid R_1(t) \wedge \neg R_2(t)\}$

使用法:

- ・選択演算 年齢が20才以上のものの取出し

$$\{t \mid R_1(t) \text{ AND } t[\text{年齢}] \geq 20\}$$

- ・複合演算 学科名称が「情報」に所属している
学生の名前、学科名称の取出し

$$\begin{aligned} \{(\text{名前}, \text{学科名称}) \mid & (\text{名前}) \in R_1 \\ & \wedge (\text{学科名称}) \in R_2 \\ & \wedge (R_2.\text{学科名称} = \text{'情報'}) \\ & \wedge (R_1.\text{学科コード} = R_2.\text{学科番号})\} \end{aligned}$$

6. レポート課題

- ① データ正規化の手順を説明して下さい。
- ② 自分で結合 (Join) 演算の事例を作成して下さい。
(二つの表の結合、取出しの指定条件の指定、
取出す属性の種類限定が含まれること。)
表の作成でも、数式でも可。

- ① レポートの内容レベルは、A4x1 枚程度。
- ② 次回の授業開始時に、提出して下さい。
(ただし、それ以前に提出する場合は、
メールで願います。
アドレス: fwhy6454@mb.infoweb.ne.jp)

6. 参考書ほか

- 大木幹雄「データベース設計の基礎」(日本理工出版会)
- 増永良文「リレーショナル・データベース入門」(サイエンス社)
- 小野哲ほか「まるごと図解、SQLがわかる」(技術評論社)
- 情報処理学会「情報処理ハンドブック」(オーム社)
- 織田敬三「ビジネス・パソコンユーザーのための
ネットワーク対応、データベース構築ガイド」
(電波新聞社)
- 仲田 聡ほか「SEの基礎知識 コンピュータテクノロジー」
(リックテレコム)
- 布川 薫ほか 著「SEの基礎知識、アプリケーション開発技術」
(リックテレコム社、3000円)
- <http://www.ann.hi-ho.ne.jp/hirok/sql/index.html>
- <http://www.rfs.jp/sitebuilder/sql/>