

# データベース設計論

## 第4回 操作体系(概要編)

---

2015/11/3

# リレーショナルデータベースモデルの操作体系

- **関係論理** (第一階述語論理に基づく)

- $P(t)$ を述語論理とする時, それを $P(t)$ が真となるものの集合 $\{t \mid P(t)\}$ を求める

$P(t) \equiv$  “ $t$ はAB型である”  
 $\{t \mid P(t)\}$  : AB型の人集合

- 非手続的言語
- **SQL**のベースとなる操作体系

- **関係代数**

- 集合に対する演算の組合せで必要な集合を求める

- 手続き的言語
- 関係論理と等価
- 問合せ実行プランの生成に必要な体系

$A - B, A \cup B, A \cap B, A \times B, \neg A,$   
 $\sigma_C A, \pi_a A, \delta A, A \bowtie B$

# 本日の内容

- 基本的な問合せに関して
  - 関係論理 & SQL
  - 関係代数
- の対応関係を具体的な例を使って演習しながら一通り解説します

# と、その前に...

- 論理設計  
（ER図からリレーションスキーマを作る）  
をDBWorksViewerの例を使って  
説明しましょう

# DBWorksViewerとは...

- データベース設計論のグループワーク課題を提出し、お互いの提出作品を閲覧したりコメントしたりできるDBアプリケーション

The screenshot displays the DBWorksViewer web application. The browser address bar shows the URL `localhost/DB2013/groupwork/work_list.php`. The page title is "データベース設計論 グループワーク課題共有システム". The user is logged in as "渡辺知恵美さん" with a [LOGOUT] link. The navigation menu includes HOME, 課題の提出, 作品を見る, 他のチーム, and PROFILE.

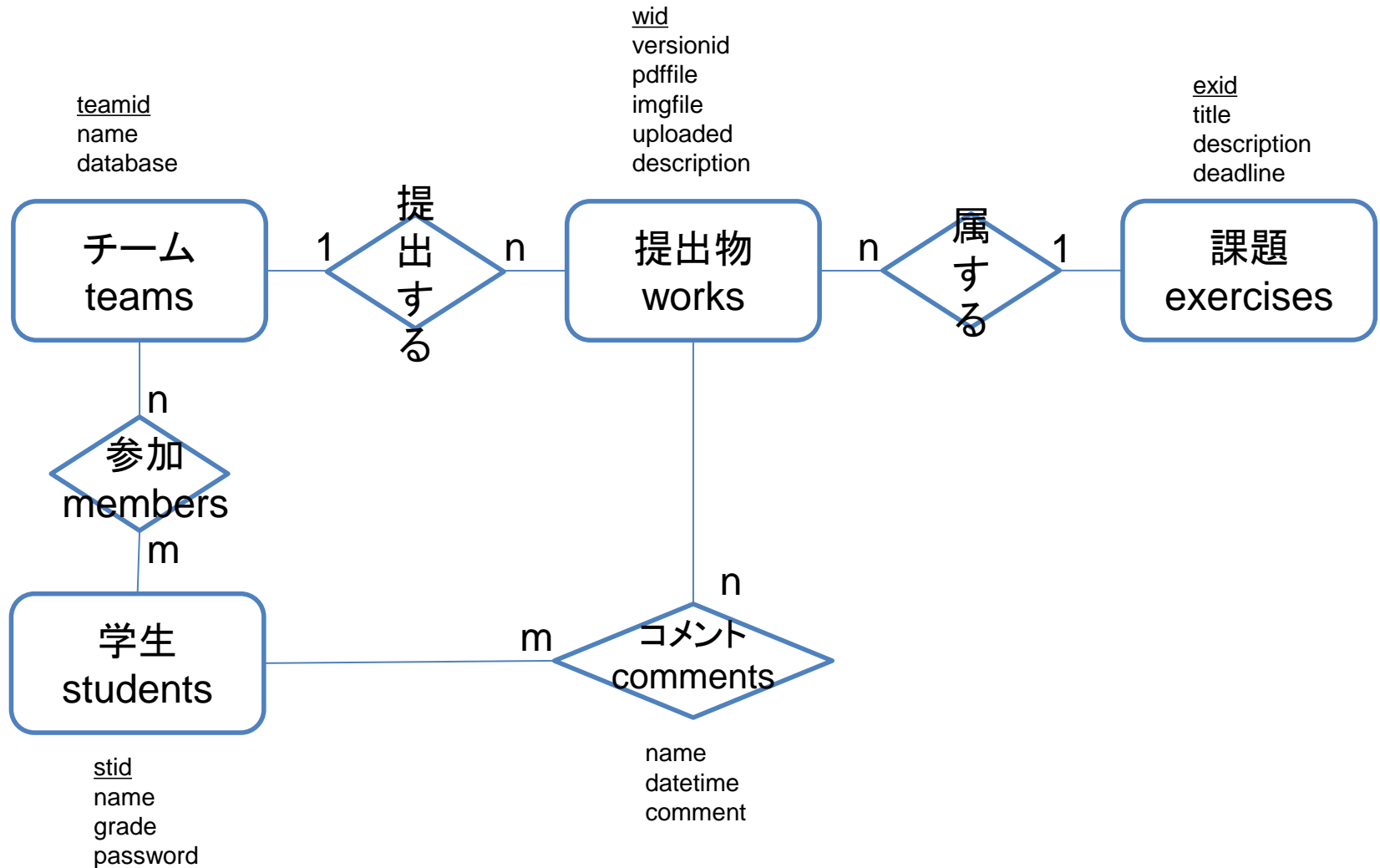
**課題リスト**

No.	課題名	締切	提出件数
1	ER図の設計	2013/10/22	2
2	リレーションスキーマの設計	2013/11/10	0

**提出されたレポート(課題番号: 1)**

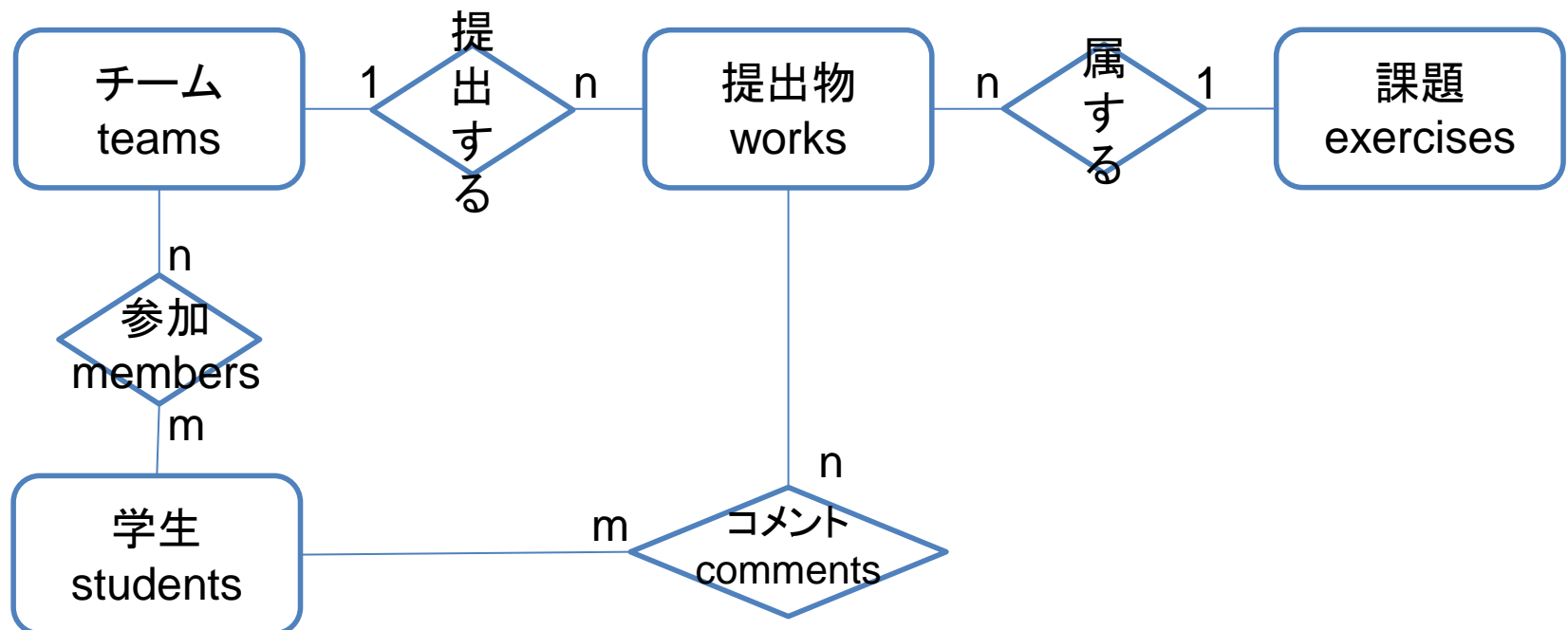
チーム名 データベース名 提出日	概要(図)	概要(文)
きんぎょ DBWorksViewer 2013/10/25 [PDFファイル] [詳細]		データベース概要 グループワークの課題を提出して、お互いにチェックしたり、いいね!したり、コメントしたりできるようなシステム。一つの課題を何度も提出しなおしたり、コメントを返すことができる。要求仕様 グループワークに登録する学生・教員はユーザ登録をする。チームも登録する。1チームのメンバー数は1名から3名まで。教員は課題を登録する。チームは教員の出した◆◆(以下略)
	回転寿司の注文データベース ER図設計案1	概要 回転寿司の注文を管理するデータベース

# DBWorksViewerのER図



# 手順1: 実体をリレーションスキーマにする

- teams(teamid, name, database)
- works(wid, versionid, pdf file, img file, uploaded, description)
- exercises(exid, title, description, deadline)
- students(stid, name, grade, password)

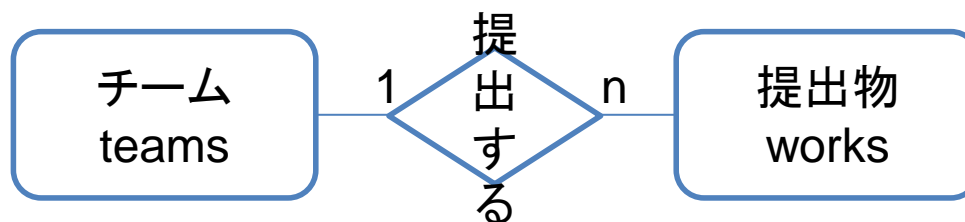


## 手順2: 1:nの関係の場合の対処

- n側の実体に対するリレーションスキーマに1側の主キーを外部キーとして追加
- 関連に属性がついていたら、それも1側の実体に追加

teams(teamid, name, database)

works(wid, versionid, pdf, img, uploaded, description, teamid)





## 手順2: 1:nの関係の対処

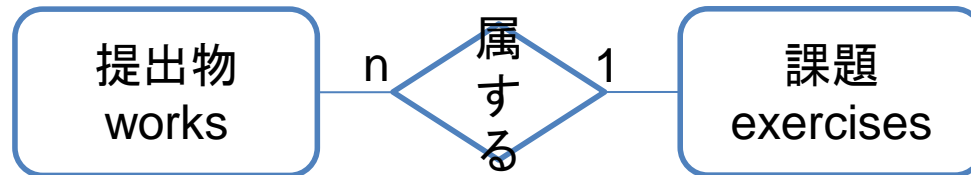
(1:1の場合も同じ対処ができる)

- n側の実体に対するリレーションスキーマに1側の主キーを外部キーとして追加
- 関連に属性がついていたら、それも1側の実体に追加

※これは作業前です。どうなるか自分で変更してみましょう。

exercises(exid, title, description, deadline)

works(wid, versionid, pdf, img, uploaded, description, teamid)



## 手順3: n:m関係の対処

- 関連に対するリレーションスキーマを作る
- 二つの実体の主キーを追加し, これらを外部キーとする

members(stid, teamid)

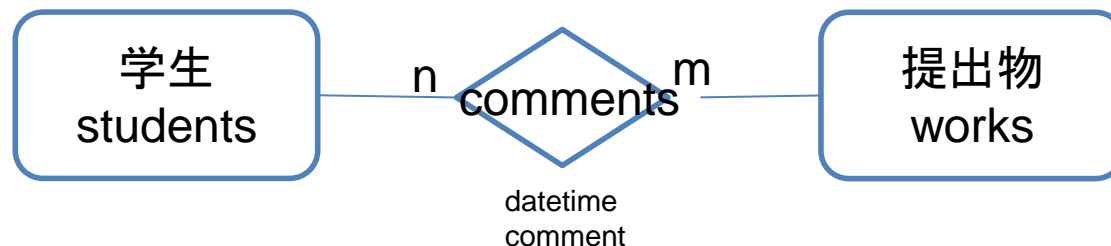


## 手順3: n:m関係の対処

- 関連に対するリレーションスキーマを作る
- 二つの実体の主キーを追加し, これらを外部キーとする

※ commentsに対するリレーションスキーマを作ってみましょう

comments(stid, wid, name, datetime, comment)



# 以下のリレーションスキーマができました (以降, 講義や課題でこの例を使います)

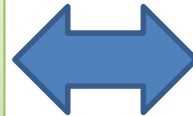
- teams(teamid, name, database)
- works(wid, teamid, exid, versionid, pdffile, imgfile, uploaded, description)
  - teamidはteamsの外部キー
  - exidはexercisesの外部キー
- exercises(exid, title, description, deadline)
- students(stid, name, grade, password)
- members(stid, teamid)
  - stidはstudentsへの外部キー
  - teamidはteamsへの外部キー
- comments(stid, wid, name, datetime, comments)
  - stidはstudentsへの外部キー
  - widはworksへの外部キー

# 操作体系の関係

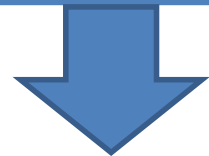
関係完備  
relational complete

RCで書いた式は  
RAでも書くことができる  
RAで書いた式は  
RCでも書くことができる

関係論理  
relational calculus  
(RC)



関係代数  
relational algebra  
(RA)



関係完備

SQL

# 関係論理 (Relational Calculus : RC)

- 問合せ(query) 結果の満たすべき性質を第一階述語論理式で記述する
- 二つの流派がある
  - タプル関係論理  
(tuple relational calculus)
  - ドメイン関係論理  
(domain relational calculus)
- この講義ではタプル関係論理のみ扱います

# タプル関係論理

- 問合せ (query) は次の形式を持つ

$$\{t | t \in P(t)\}$$

- $P$ は論理式
  - $P(t)$ を満たすタプル $t$ を求める
- 次のアトムは論理式である

$s \in R$	タプル $s$ はリレーション $R$ に含まれる
$s.A\theta t.B$	タプル $s$ の属性 $A$ の値と タプル $t$ の属性 $B$ の値は $\theta$ の関係である ( $\theta \in \{=, >, <, <>, \geq, \leq\}$ )
$s.A\theta C$	タプル $s$ の属性 $A$ の値と 定数 $C$ は $\theta$ の関係である

# タプル関係論理

- $P$ が論理式ならば,  $\neg P$ は論理式である

- $P_1, P_2$ が論理式ならば,

$$P_1 \wedge P_2, P_1 \vee P_2, P_1 \rightarrow P_2$$

は論理式である

- $P(s)$ が自由変数 $s$ を含む論理式ならば,

$$(\forall s)(P(s)),$$

$$(\exists s)(P(s))$$

は論理式である



# 関係論理式の例

- シンプルな問合せ

- 例1) nameがchiemiであるstudentsのタプル

$$\{t \mid t \in students \wedge t.name = 'chiemi'\}$$

students(stid, name, grade, password)

stid	name	grade	password
g001	chiemi	3	xlskejrs;l
g002	aya	2	lakjwr
g003	takako	3	xlkjwerlkj

# 関係論理式の例

- 例2) studentsのname一覧を求める

$$\{t | (\exists s)(s \in students \wedge s.name = t.name)\}$$

- 例3) std='g001'である学生の名前を求める

$$\{t | (\exists s)(s \in students \wedge s.std = 'g001' \wedge t.name = s.name)\}$$

students(std, name, grade, password)

std	name	grade	password
g001	chiemi	3	xlskejrs;l
g002	aya	2	lakjwr
g003	takako	3	xlkjwerlkj

## 二つのリレーションを使った論理式の例

- 例4) name='chiemi'である学生の  
コメント時刻とコメント内容を求める

$$\{u | (\exists s)(\exists t)(s \in students \wedge t \in comments \\ \wedge s.name = 'chiemi' \wedge s.stid = t.stid \\ \wedge u.datetime = t.datetime \wedge u.comment = t.comment)\}$$

stid	name	grade	password
g001	chiemi	3	xlskeirs:l
g002	aya		
g003	takaki		

stid	wid	datetime	comment
g001	w01	10/23 8:20	nice!
g001	w02	10/24 9:00	I can't read...
g002	w01	10/31 8:12	good job

# SQL

- 関係論理に基づいたデータベース問合せ言語
- ISO国際標準で規格化されている

```
SELECT 属性名, 属性名, ...  
FROM <リレーション名>, <リレーション名>  
WHERE <検索条件>
```

# 関係論理式とSQL文の対応関係

- 例3)  $stdid = 'g001'$ である学生の名前を求める

リレーション名

$$\{t | (\exists s) (s \in students \wedge s.stdid = 'g001' \wedge t.name = s.name)\}$$

検索条件                      出力する属性



```
SELECT s.name
FROM students s
WHERE s.stdid = 'g001'
```

# 関係論理式とSQL文の対応関係

- 例4) name='chiemi'である学生の  
コメント時刻とコメント内容を求める

$$\{u | (\exists s)(\exists t)(s \in students \wedge t \in comments \\ \wedge s.name = 'chiemi' \wedge s.stid = t.stid \\ \wedge u.datetime = t.datetime \wedge u.comment = t.comment)\}$$


```
SELECT t.datetime, t.comment
FROM students s, comments t
WHERE s.name = 'chiemi'
and t.stid = s.stid
```

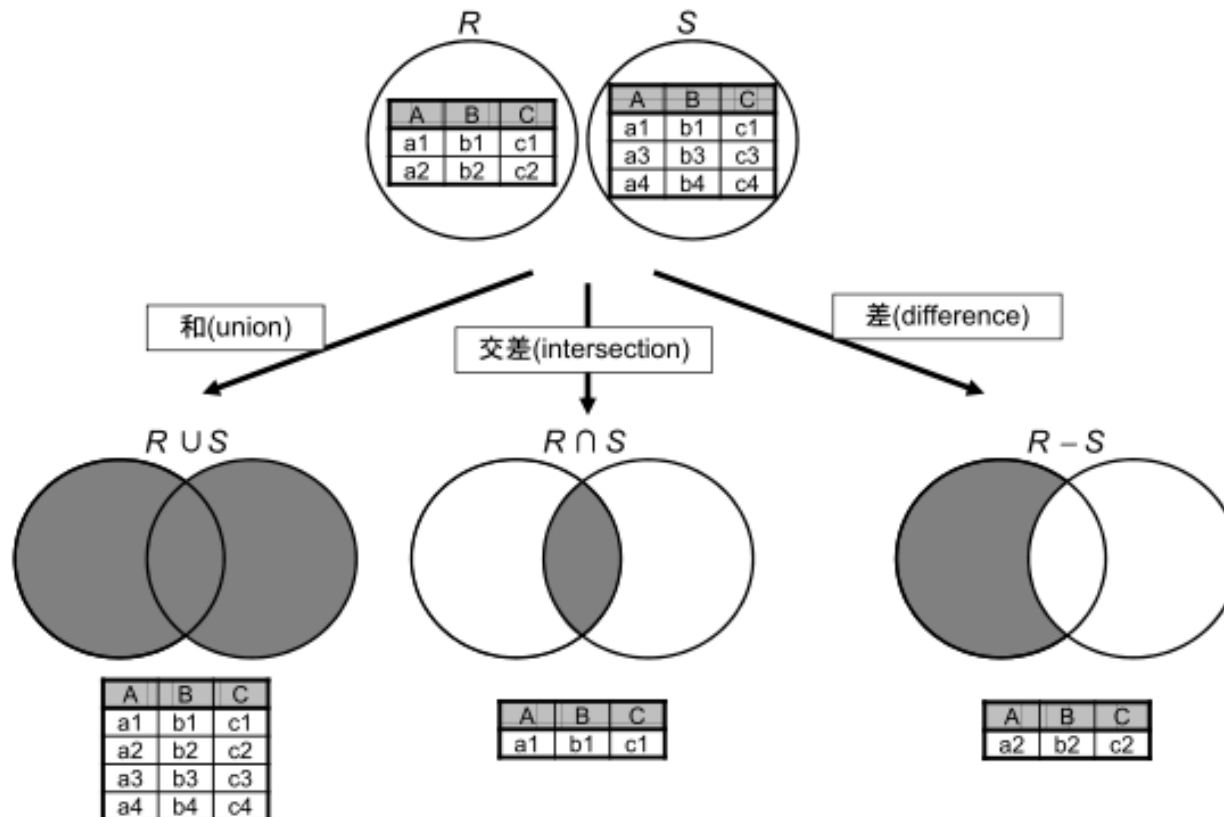
# 関係代数

- リレーションを対象にした演算の組合せで問合せ (query) を表す
- 演算子
  - 和 ( $A \cup B$ ), 差 ( $A - B$ ), 交差 ( $A \cap B$ )
  - 直積 ( $A \times B$ )
  - 射影 ( $\pi_L(R)$ ), 選択 ( $\sigma_C(R)$ )
  - 結合 ( $A \bowtie_C B$ )
  - 商 ( $A \div B$ )

関係代数のために  
導入された演算子

# 関係代数の演算子

- 和, 差, 交差





# 射影(projection) $\pi_L(R)$

- $R_2 = \pi_L(R_1)$ 
  - $L$ は $R_1$ から選んだ属性のリスト
  - $R_2$ は $R_1$ の各タプルの $L$ にある属性を指定された順番で抜き出したもの
- 例2) studentsのname一覧を求める

$\pi_{name}(students)$

students(stid, name, grade, password)

stid	name	grade	password
g001	chiemi	3	xlskejrs;l
g002	aya	2	lakjwr
g003	takako	3	xlkjerlkj

# 射影(projection) $\pi_L(R)$

- 射影演算は以下の関係論理式とSQL文で表すことができる

- 関係論理式

$$L = \{l_1, \dots, l_n\} \text{として}$$

$$\{t | s \in R \wedge t.l_1 = s.l_1 \wedge \dots \wedge t.l_n = s.l_n\}$$

- SQL文

SELECT  $l_1, \dots, l_n$  FROM R

# 選択 (selection) $\sigma_C(R)$

- $R_2 = \sigma_C(R_1)$ 
  - $C$ は $R_1$ の属性を参照する条件
  - $R_2$ は条件 $C$ を満たすような $R_1$ のタプルすべて
- 例1) nameがchiemiであるstudentsのタプル

$\sigma_{name='chiemi'}(students)$

students(stid, name, grade, password)

stid	name	grade	password
g001	chiemi	3	xlskejrs;l
g002	aya	2	lakjwr
g003	takako	3	xlkjwerlkj

# 選択 (selection ) $\sigma_C(R)$

- 選択演算は以下の関係論理式とSQL文で表すことができる

- 関係論理式

$C$ は $R$ に対する論理式 $C(R)$ とする  
 $\{t | t \in R \wedge C(R)\}$

- SQL文

```
SELECT *  
FROM R  
WHERE C
```

該当するタプルの全ての属性をもとめたいときには「\*」と書きます。

# 演算子の組合せ

- 関係代数の演算子の出力はリレーションなので出力結果に対して演算を適用できる

- 例)  $R_2 = \sigma_{grade=3}(students)$

stid	name	grade	password
g001	chiemi	3	xlskejrs;l
g003	takako	3	xlkjwerlkj

$$R_3 = \pi_{name}(R_2)$$

name
chiemi
takako

上記の処理をまとめて書ける

$$R_3 = \pi_{name}(\sigma_{grade=3}(students))$$

# 演算子の組合せ

- 例3)  $stdid='g001'$ である学生の名前を求める

$$\pi_{name}(\sigma_{stdid='g001'}students)$$

対応する関係論理式とSQL文(再掲)

$$\{t | (\exists s)(s \in students \wedge s.stdid = 'g001' \wedge t.name = s.name)\}$$

```
SELECT s.name  
FROM students s  
WHERE s.stdid = 'g001'
```

単一のリレーションに関する関係論理式は  
選択演算と射影演算の組合せで表すことができる

# $\theta$ -結合(theta-join) $R_1 \bowtie_C R_2$

- 二つのリレーション  $R_1, R_2$  の各タプルのうち、条件  $C$  を満たす組合せを求める

- 例  $S = \pi_{std, name} students$      $C = \pi_{std, comment} comments$

std	name		std	comment
g001	chiemi	→	g001	nice!
g002	aya	→	g001	I can't read...
g003	takako	→	g002	good job

$$S \bowtie_{S.std=C.std} C$$

S.std	S.name	C.std	C.comment
g001	chiemi	g001	nice!
g001	chiemi	g001	I can't read...
g002	aya	g002	good job

## $\theta$ -結合(theta-join) $R_1 \bowtie_C R_2$

- 例4) name='chiemi'である学生のコメント時刻とコメント内容を求める
  1.  $R_1$ : name='chiemi'である学生
    - $R_1 = \sigma_{name='chiemi'} students$
  2.  $R_2$ :  $R_1$ に対応するcommentsのタプルを求める
    - $R_2 = R_1 \bowtie_{students.stid=comments.stid} comments$
  3.  $R_3$ :  $R_2$ からdatetimeとcommentsを射影する
    - $R_3 = \pi_{datetime, comments} R_2$

$$\pi_{datetime, comments}((\sigma_{name='chiemi'} students) \bowtie_{students.stid=comments.stid} comments)$$



# 関係論理→関係代数→実行プラン

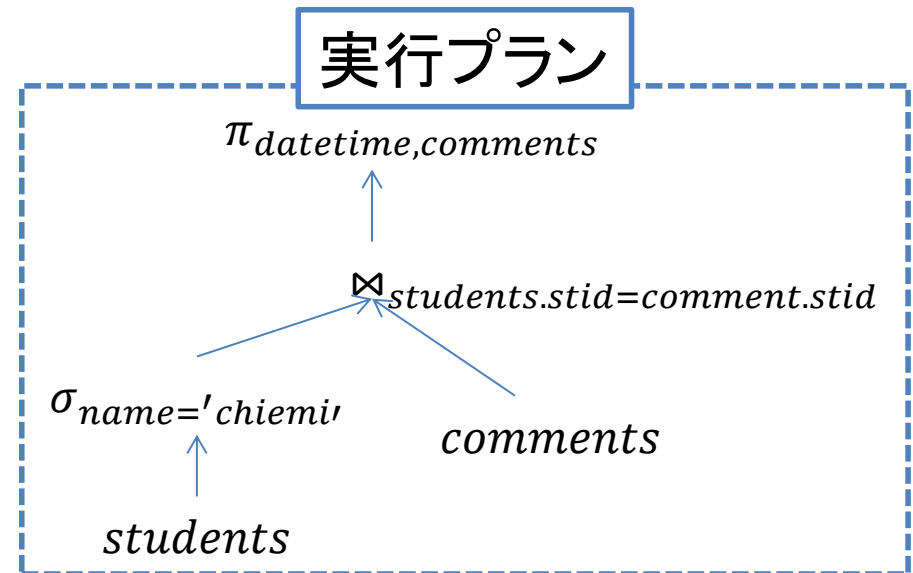
- 利用者が指定する問合せは非手続的

```
SELECT t.datetime,t.comment
FROM students s, comments t
WHERE s.name = 'chiemi'
and t.stid = s.stid
```

- DBMSはそれと等価な関係代数式を求める

```
 $\pi_{datetime,comments}(\sigma_{name='chiemi',students} \bowtie_{students.stid=comments.stid} comments)$ 
```

- 関係代数式から実行プランを求め、最適なプランに書き換えて実行する



※DBMSは各演算子のための実行プログラムをいくつか用意しており、最適なプログラムを選ぶ