

データベース設計論

第5回

演習:リレーションスキーマ作成

講義:関係代数(中級編)

2019/11/5

グループ課題：演習

第1回レポートで提出したERダイアグラムをもとに、リレーションスキーマを作成しよう(15分)

前回の復習(演習 10分+5分)

- 以下の問合せを関係代数およびSQLで書きましょう
 1. アカウント kingjim をフォローしているユーザのアカウント名
(hint : 選択と射影を使う)
 2. アカウント kingjim をフォローしているユーザのアカウント名と名前
(hint : 選択と射影と結合を使う)

【全ての操作体系に共通な注意事項】

バッグ集合とセット集合

- バッグ集合

- 重複を許す集合

- セット集合

- 重複を許さない集合

- 例) $\pi_{grade}(students)$

基本的にはSQLではバッグ集合を返す
・理由: 重複の削除などは重い処理

基本的に関係代数, 関係論理では
セット集合を返す

検索結果のバッグ集合

grade
3
2
3

検索結果のセット集合

grade
3
2

関係代数演算

- 基本演算
 - 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($A \div B$)
- 使うと便利な演算
 - 自然結合 ($A * B$)
 - ネーミング演算 ($\rho_{S(A_1, \dots, A_n)}(R)$)

直積 ($A \times B$)

- 全てのタプルの組合せを求める

$A(\text{stid}, \text{name})$

stid	name
g001	chiemi
g002	aya

$B(\text{stid}, \text{comment})$

stid	comment
g001	nice!
g001	I can't read...

$A \times B$

stid	name	stid	comment
g001	chiemi	g001	nice!
g002	aya	g001	nice!
g001	chiemi	g001	I can't read...
g002	aya	g001	I can't read...

θ -結合($A \bowtie_c B$)は直積と選択で表せる

- $A \bowtie_c B = \sigma_c(A \times B)$
- 例) $A \bowtie_{A.stid=B.stid} B = \sigma_{A.stid=B.stid}(A \times B)$

$A \times B$

stid	name	stid	comment
g001	chiemi	g001	nice!
g002	aya	g001	nice!
g001	chiemi	g001	I can't read...
g002	aya	g001	I can't read...

$A \bowtie_{A.stid=B.stid} B$

stid	name	stid	comment
g001	chiemi	g001	nice!
g001	chiemi	g001	I can't read...

自然結合 $A * B$

- 結合する二つのリレーションに含まれる共通の属性で等結合したもの

$$A * B = \pi_{A.stid, A.name, B.comment}(A \bowtie_{A.stid=B.stid} B)$$

$A(stid, name)$

stid	name
g001	chiemi
g002	aya

$B(stid, comment)$

stid	comment
g001	nice!
g001	I can't read...

$A * B$

stid	name	comment
g001	chiemi	nice!
g001	chiemi	I can't read...

ネーミング演算

- 関係代数演算の結果リレーシヨンのリレーシヨン名や属性名を変更する

$$S(A_1, \dots, A_n) = \rho_{S(A_1, \dots, A_n)} R$$

- 例) リレーシヨンGameを使って
Gameの勝ち負けの対応表を求める

$Game(name, score)$ ↑ 誰が誰に何点勝ってるか $vs(winner, loser, diff)$

name	score
chiemi	67
takako	92
aiko	78

$$G_1 = \rho_{G_1}(Game(name, score))$$

$$G_2 = \rho_{G_2}(Game(name, score))$$

winner	loser	diff
takako	chiemi	25
takako	aiko	14
aiko	chiemi	11

$$\rho_{vs(winner, loser, diff)}(\pi_{G_2.name, G_1.name, G_2.score - G_1.score}(G_2 \bowtie_{G_2.score > G_1.score} G_1))$$

SQLでネーミング演算に対応するもの

- SELECT節の属性とFROM節のリレーション名に別名をつけることができる

関係代数のネーミング演算で使った例(再掲)

Game(name, score)

name	score
chiemi	67
takako	92
aiko	78

$\rho_{G1}(Game(name, score))$

$\rho_{G2}(Game(name, score))$

vs(winner, loser, diff)

winner	loser	diff
takako	chiemi	25
takako	aiko	14
aiko	chiemi	11

$\rho_{vs(winner, loser, diff)}(\pi_{G2.name, G1.name, G2.score - G1.score}(G2 \bowtie_{G2.score > G1.score} G1))$

```
SELECT G2.name as winner, G1.name as loser,  
       G2.score - G1.score as diff  
FROM Game G1, Game G2  
WHERE G2.score > G1.store
```

自己結合

- θ 結合の一種の使い方で、リレーション R を自分自身(R)と結合したもの
 - 名前を結合させるときにはネーミング演算を使って別々のリレーション名をつけ直す

演習2: 下記のツイートのリツイート文一覧を求める関係代数とSQLを求めよ

tweet(id, content, account, datetime,
retweeted_id, replied_id)

id	content	account	date time	retw eete d_id	repl ied_ id
13	みなさんのお給料あるいは冬のボーナスをあてにして、たくさんのシャープ製品が本日発売になります。必要ないのに買えとは思いますが、必要なら候補に入れてください。	SHARP_JP	2019-10-24 11:50:00		

演習3: 下記のツイートをした人がその後にツイートした日時と内容を求める関係代数とSQLを求めよ

id	content	account	date time	retw eete d_id	repl ied_ id
5	もうわからん	RyoNishikido _JP	2019-10-12 03:48:00		

SQLに関するいろいろ

SQL 便利な書き方1

- 問合せの結果をセット集合にしたい場合は `distinct` をつける

```
SELECT distinct grade  
FROM students
```

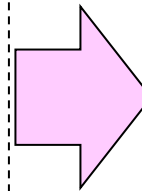
- FROM節に書くリレーションに別名をつけることができる

```
SELECT s2.stid, st.name  
FROM students s1, students s2  
WHERE s1.stid = 'g1120511'  
      and s1.grade = s2.grade;
```

SQL 便利な書き方2

- FROMで複数のリレーションを指定するときは属性名が衝突してはいけない

```
SELECT name, comment  
FROM students s, comments c  
WHERE s.stid = c.stid  
      and s.stid = 'gxx205xx'
```



エラーで実行できない:
nameがstudentsのnameか
commentsのnameか
判別できない



対策

```
SELECT s.name, c.comment  
FROM students s, comments c  
WHERE s.stid = c.stid  
      and s.stid = 'gxx205xx'
```


LIKE構文 (Where節)

- LIKE構文を使って部分一致検索ができる

記号	意味
%	0文字以上の任意の文字にマッチ
_	1文字の任意の文字にマッチ

作成しているデータベース名に「映画」が含まれているチーム名とそのデータベース

```
SELECT t.name, t.database  
FROM teams t  
WHERE database like '%映画%'
```

集合に対する演算 (Where節)

- IN句を使って属性の値がある集合に含まれているタプルを取り出すことができる

教員かその他のアカウント

```
SELECT s.stid, s.name  
FROM students s  
WHERE grade IN (10,20)
```

- BETWEEN句を使って指定した範囲の属性値を持つタプルを取り出すことができる

学籍番号がg1120501からg1120510までの学生

```
SELECT s.stid, s.name  
FROM students s  
WHERE s.stid BETWEEN 'g1120501' AND 'g1120510'
```

文字列の順序関係を使っている