

SQL

Structured Query Language その 2

複数テーブルの問い合わせ
データ定義と変更

§3.3 複数テーブルの問い合わせ

- 副問い合わせ(subquery)
- 結合(join)
- 和(union), 差(except), 共通部分(intersect)

前回に加えて, 今回扱う表

enrollment studentnum	classname	positionnum
100	BD445	1
150	BA200	1
200	BD445	2
200	CS250	1
300	CS150	1
400	BA200	2
400	BF410	1
400	CS250	2
450	BA200	3

class name	btime	room
BA200	M-F9	SC110
BD445	MWF3	SC213
BF410	MWF8	SC213
CS150	MWF3	EA304
CS250	MWF12	EB210

2つのテーブルを使う例

BD445の講義に登録した学生の名前を知りたい.

'BD445'に登録した学生のSIDは,

```
SELECT studentnum FROM enrollment  
WHERE classname='BD445'
```

studentnum

100

200

SIDが 100, 200の学生の名前は

```
SELECT name FROM student  
WHERE sid IN [100, 200]
```

name

Jones

Baker

副問い合わせ

BD445に登録した学生

```
SELECT name FROM student
WHERE sid IN
  (SELECT studentnum
   FROM enrollment
   WHERE classname='BD445')
```

name

Jones

Baker

sidとstudentnumは同一ドメイン.

3つのテーブルを使う例

月水金の3時から始まる講義に登録した学生の名前

```
SELECT student.name  
FROM student  
WHERE student.sid IN
```

```
(SELECT enrollment.studentnum  
FROM enrollment  
WHERE enrollment.classname IN
```

```
(SELECT class.name  
FROM class  
WHERE class.btime='MWF3'))
```

SQLによる結合(Join)

```
SELECT student.sid, student.name,  
       enrollment.classname  
FROM student, enrollment  
WHERE student.sid=enrollment.studentnum
```

sid	name	classname
100	Jones	BD445
150	Parks	BA200
200	Baker	BD445
200	Baker	CS250
300	Baker	CS150
400	Rye	BA200
400	Rye	BF410
400	Rye	CS250
450	Jones	BA200

結合と WHERE の条件

結合以外の目的で WHERE を指定しても良い.

```
SELECT student.sid, enrollment.classname  
FROM student, enrollment  
WHERE student.sid=enrollment.studentnum  
      AND student.name='Rye'  
      AND enrollment.positionnum=1
```

sid	classname
400	BF410

3つのテーブルの結合

```
SELECT student.sid, class.name, enrollment.positionnum
FROM   student, enrollment, class
WHERE  student.sid = enrollment.studentnum
      AND enrollment.classname = class.name
      AND student.name = 'Baker'
```

sid	name	positionnum
200	BD445	2
300	CS150	1
200	CS250	1

同じ意味の問い合わせ 1

```
SELECT name FROM student
WHERE sid IN
  (SELECT studentnum FROM enrollment
   WHERE classname = 'BD445')
```

```
SELECT student.name
FROM student,enrollment
WHERE student.sid = enrollment.studentnum
AND enrollment.classname = 'BD445'
```

同じ意味の問い合わせ 2

```
SELECT student.name
FROM student
WHERE student.sid IN
  (SELECT enrollment.studentnum
   FROM enrollment
   WHERE enrollment.classname IN
     (SELECT class.name
      FROM class
      WHERE class.btime='MWF3'))
```

```
SELECT student.name
FROM student, enrollment, class
WHERE student.sid = enrollment.studentnum
  AND enrollment.classname = class.name
  AND class.btime = 'MWF3'
```

結合でのみ可能な問い合わせ

```
SELECT DISTINCT enrollment.classname, student.gradelevel  
FROM enrollment, student  
WHERE enrollment.studentnum = student.sid  
AND student.gradelevel <> 'GR'
```

classname	gradelevel
BA200	FR
BA200	SN
BA200	SO
BF410	FR
CS150	SN
CS250	FR

求める結果が2つのテーブルからの値である.

条件に EXISTS を用いる

SELECT ... WHERE EXISTS 副問い合わせ

- 副問い合わせの結果にタプルが存在すれば条件は真
- 相関サブクエリとともに使われる.
 - 評価が複数回行われる.

```
SELECT * FROM a WHERE EXISTS  
(SELECT * FROM b WHERE a.n = b.n)
```

副問い合わせでのみ可能な問い合わせ 1

EXISTS: 複数クラスに登録している学生

```
SELECT DISTINCT studentnum
FROM enrollment AS a
WHERE EXISTS
  (SELECT * FROM enrollment AS b
   WHERE a.studentnum = b.studentnum
    AND a.classname <> b.classname)
```

studentnum

200

400

副問い合わせでのみ可能な問い合わせ 2

NOT EXISTS: 受講していない学生

```
SELECT a.name  
FROM student AS a  
WHERE NOT EXISTS  
  (SELECT * FROM enrollment AS b  
   WHERE b.studentnum = a.sid)
```

name

Glass

Russell

副問い合わせと結合の比較

- どちらの方法でも表現できる場合がある.
- 結果が2つ以上のテーブルの属性を含むのは, 結合のみ.
- **EXISTS, NOT EXISTS**を持つ副問い合わせは, 結合では表現できない.

§3.4 スキーマ定義機能

特定の利用者に所有されるデータベースの一部を定義.

例 :

```
CREATE SCHEMA 受注 AUTHORIZATION 岡本  
CREATE TABLE 商品表
```

.....

```
CREATE VIEW
```

実表 student の定義

```
CREATE TABLE student
( sid          INTEGER NOT NULL,
  name          CHAR(35) NOT NULL,
  major         CHAR(30),
  gradelevel    CHAR(2),
  age           INTEGER,
  PRIMARY KEY (sid),
  CHECK (sid BETWEEN 100 AND 500)
)
```

実表 class の定義

```
CREATE TABLE class
( name CHAR(5) NOT NULL,
  btime CHAR(10),
  room CHAR(5),
  PRIMARY KEY (name)
)
```

実表 enrollment の定義

```
CREATE TABLE enrollment
(studentnum    INTEGER NOT NULL,
 classname    CHAR(5) NOT NULL,
 positionnum   INTEGER,
 PRIMARY KEY (studentnum,classname),
 FOREIGN KEY (studentnum )
             REFERENCES student,
 FOREIGN KEY (classname )
             REFERENCES class
)
```

実表定義における列定義

■ 列定義

- 列 データ型 [NOT NULL][規定定義]

■ 規定定義

- DEFAULT NULL
- DEFAULT 定数

データ型の例

型	説明
BOOLEAN	true/false の論理値
CHAR	文字
CHAR(n)	固定長文字列
VARCHAR(n)	可変長文字列
DATE	日付
REAL	単精度浮動小数点数*
DOUBLE PRECISION	倍精度浮動小数点数*
SMALLINT	符号つき 2 バイト整数*
INT, INTEGER	符号つき 4 バイト整数*

注) *の大きさは処理系依存

実表定義における主な制約

■ 主キ一定義：

- PRIMARY KEY (列X)

■ 候補キ一定義：

- UNIQUE (列X)

■ 非ナル値制約：

- NOT NULL

■ 検査条件：

- CHECK(条件)

■ 外部キ一定義：

- FOREIGN KEY 列X REFERENCES 表Y

NULL の検査

■ NULL は = や <> で調べられない

- ×SELECT * FROM A WHERE B = NULL;
- ×SELECT * FROM A WHERE B <> NULL;

■ IS や IS NOTを使う

- SELECT * FROM A WHERE B IS NULL;
- SELECT * FROM A WHERE B IS NOT NULL;

データの挿入

- 表に 1 つの行（タプル）を挿入

```
INSERT INTO enrollment  
VALUES(400,'BD445',44)
```

- 指定した列（属性）のみ値を設定(他はNULL)

```
INSERT INTO enrollment  
(studentnum, classname)  
VALUES(400,'BD445')
```

テーブルからテーブルへ

```
INSERT INTO junior VALUES  
  ( SELECT sid,name,major  
    FROM student  
    WHERE gradelevel='JR' )
```

データの削除

```
DELETE FROM student  
WHERE student.sid = 100
```

enrollmentとの参照整合性制約に注意.

削除の順番を考慮すべき例

```
DELETE FROM enrollment  
WHERE enrollment.studentnum IN  
  ( SELECT student.sid  
    FROM student  
    WHERE student.major = 'Accounting')
```

```
DELETE FROM student  
WHERE student.major = 'Accounting'
```

順番を逆にすると問題が生じる.

データの更新

```
UPDATE enrollment  
SET    positionnum = 44  
WHERE  studentnum = 400  
AND    classname = 'BA200'
```

```
UPDATE enrollment  
SET    positionnum = positionnum+1  
WHERE  classname = 'BF410'
```

大量のデータ更新の例

クラス名が BD445からBD567へ変更になった

```
UPDATE enrollment  
SET  classname='BD567'  
WHERE classname='BD445'
```

```
UPDATE class  
SET  classname='BD567'  
WHERE classname='BD445'
```

更新の順序を考慮する必要がある.

テーブルの削除

DROP TABLE enrollment

DROP TABLE student

DROP TABLE class

削除する順番に注意.