

SQL (2)

データベース論 I 第7回

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

作成者 末次文雄 ©

課題6の解答例→テーブル定義

```
CREATE DATABASE 学科;  
CREATE TABLE 学科TBL  
  (学科番号 INT(7) NOT NULL UNIQUE,  
   学科名称 NCHAR(10),  
   主任      NCHAR(10)                );  
CREATE TABLE 学生TBL  
  (学籍番号 CHAR(7) NOT NULL UNIQUE,  
   年次      INT(1),  
   入学年度  INT(4),  
   氏名      NCHAR(10)                );  
CREATE TABLE 所属TBL  
  (学科コード INT(7) NOT NULL,  
   年度        INT(4),  
   学生コード CHAR(7),                );
```

課題6の解答例→SELECT文

```
SELECT  学科TBL.学科名称,  
        学生TBL.学籍番号,学生TBL.氏名  
FROM    学科TBL,学生TBL,所属TBL  
WHERE   学生TBL.入学年度=1999  
        AND 所属TBL.年度=2002  
        AND 学科TBL.学科番号=所属TBL.学科コード  
        AND 学生TBL.学籍番号=所属TBL.学生コード  
ORDER  BY 所属TBL.学科コード,所属TBL.学生コード ;
```

結合(JOIN)

学科名称	学籍番号	氏名
電子工学	A101000	Aさん
電子工学	A101200	Bさん
情報工学	A101010	Cさん
~	~	~

目次

1. データ定義（詳細編）
2. データ操作（詳細編）
3. データ操作（副問合せ）（サブクエリ）
4. 埋め込みSQL
5. カーソルの使用
6. レポート課題
7. 参考書ほか

1. データ定義（詳細編）

- 1. 1 テーブル制約
- 1. 2 表の定義変更
- 1. 3 INDEXの作成

1.1 テーブル制約

- テーブル定義時に各種のデータ制約を設定することが出来る。
 - ①列(データ項目)のNOT NULL値
 - ②列内でのデータの一貫性
 - ③テーブル内の主キー
 - ④外部キー制約
 - ⑤列(データ項目)のデータ値

上記によって、テーブル単位での「**データの一貫性**」を保証できる。

1. 1 (続き) テーブル制約

①列(データ項目)のNOT NULL値

- ・この列には、値が入っていないなければならない。

(値が入っていないとエラーになる。)

```
CREATE TABLE テーブル名  
    (データ項目名 データ型(長さ) NOT NULL);
```

②列内でのデータの一意性

- ・この列では、値の重複が許されない。

(値が重複すれば、エラーになる。INDEXが作られる)

```
CREATE TABLE テーブル名  
    (データ項目名 データ型(長さ) NOT NULL  
    UNIQUE);
```

1. 1 (続き) テーブル制約

③ テーブル内の主キー

- この列が主キーであることを示す。

(通常他のテーブルの外部キーに対応しており、外部キー側でREFERENCES制約があれば、この行は削除できない。)

(INDEXが作成される。)

```
CREATE TABLE テーブル名  
  (データ項目名 データ型(長さ) NOT NULL  
   CONSTRAINT 識別名 PRIMARY KEY);
```


1. 1 (続き) テーブル制約

④外部キー制約

- この列が外部キーであることを示す。

(他のテーブルの主キーが既に存在していなければ、外部キー側で行の作成が出来ない。)

```
CREATE TABLE テーブル名  
  (データ項目名 データ型(長さ) NOT NULL  
   CONSTRAINT 識別名 FOREIGN KEY  
   REFERENCES 主キーのあるテーブル名(列名));
```

1. 1 (続き) テーブル制約

⑤列(データ項目)のデータ値

- この列の値を制約する条件を記述する。

(この条件に違反する行は、存在出来ない。)

```
CREATE TABLE テーブル名  
  (データ項目名A データ型(長さ),  
   データ項目名B データ型(長さ),  
   データ項目名C データ型(長さ),  
   CHECK (条件文));
```

条件の例→→CHECK (データ項目名B >= 10000)

1. 1 (続き) テーブル制約

補足: 複数の列を連結した制約文

- ・テーブル定義の終わりにまとめて記述

```
CREATE TABLE テーブル名
```

```
(列A データ型(長さ) NOT NULL,
```

```
列B データ型(長さ) NOT NULL,
```

```
列C データ型(長さ) NOT NULL,
```

```
UNIQUE (列A, 列B),
```

```
PRIMARY KEY (列A, 列B),
```

```
CONSTRAINT 識別名 PRIMARY KEY (列A, 列B),
```

```
CONSTRAINT 識別名 FOREIGN KEY
```

```
REFERENCES 主キーのあるテーブル名(列名),
```

```
CONSTRAINT 識別名 FOREIGN KEY
```

```
REFERENCES 主キーのあるテーブル名(列名);
```

1. 2 表の定義変更

- 作成済みのテーブル定義を変更、削除する

テーブルの削除 (DATABASE、VIEW、INDEXも同様である)

DROP TABLE テーブル名;

テーブルの変更

ALTER TABLE テーブル名 **ADD** 追加列名 列の属性;

ALTER TABLE テーブル名 **MODIFY** 変更列名 新属性;

ALTER TABLE テーブル名 **DROP COLUMN** 列名;

ALTER TABLE テーブル名 **ADD CONSTRAINT** 制約文;

ALTER TABLE テーブル名 **DROP CONSTRAINT** 制約文;

1. 3 INDEXの作成

INDEXの作成

```
CREATE [UNIQUE] INDEX インデックス名  
ON テーブル名 (列名 [, 列名.....]);
```

UNIQUE指定時は、重複があればエラー。

INDEXの削除

```
DROP INDEX インデックス名
```

2. データ操作（詳細編）

2. 1 取だし条件

2. 2 結合 (JOIN)

2. 3 和と差 (UNION、MINUS)

2. 4 グループ化

2. 5 一括挿入 (INSERT)

2. 6 演算子と関数

2. 1 取だし条件

① 取だし表、列、組 (=行) の指定

```
SELECT 列1, 列2, 列3, .....
```

←列の指定(射影演算)

```
FROM 表A
```

←表の指定

```
WHERE 条件式 ;
```

←組 (=行) の取だし条件

```
SELECT *      --注 ホスト言語では使用しない習慣が良い  
FROM 表A  
WHERE 列1 > 1000;
```

```
SELECT DISTINCT 入学年度    ←重複を排除する  
FROM 学生TBL  
WHERE 入学年度 >= 1990;
```

2.1 (続き) 取だし条件

① (続き) AND OR

```
SELECT 学科コード, 学籍コード  
FROM 所属TBL  
WHERE 学科コード=112200  
      AND  
      年度=1998 ;
```

←AND条件

```
SELECT 学科コード, 学籍コード  
FROM 所属TBL  
WHERE 年度=1995  
      OR  
      年度=1998 ;
```

←OR条件

2. 2 結合 (JOIN)

① テーブル (表) の内部結合 --- 等 (自然) 結合

SELECT

FROM 表A INNER JOIN 表B

ON 表A. 列1 = 表B. 列2 ;

または NATURAL JOIN

(SQL-92、SQL Server、ACCESS...)

表は3つ以上でも可能

または、

SELECT

FROM 表A, 表B

WHERE 表A. 列1 = 表B. 列2 ;

(ORACLE、SQL Server...)

2. 2 結合 (JOIN)

② 内部結合に条件を追加

```
SELECT .....  
FROM 表A INNER JOIN 表B  
ON 表A. 列1 = 表B. 列2  
WHERE 表A. 列1 >= 1000 ;
```

(SQL-92、SQL Server、ACCESS...)

または、

```
SELECT .....  
FROM 表A, 表B  
WHERE 表A. 列1 = 表B. 列2  
AND  
表A. 列1 >= 1000 ;
```

(ORACLE、SQL Server...)

2. 2 結合 (JOIN)

③ 外部結合 (条件適合以外も取出す)

```
SELECT .....
```

```
FROM 表A LEFT OUTER JOIN 表B  
ON 表A.列1 = 表B.列2;
```

反対は RIGHT

(SQL-92、SQL Server、ACCESS...)

または、

```
SELECT .....
```

```
FROM 表A, 表B(+)  
WHERE 表A.列1 = 表B.列2;
```

反対は 表A(+)

(ORACLE...)

2.3 和と差

① 和集合 (UNION)

```
SELECT 表A. 列1,  
       FROM 表A  
       WHERE 検索条件
```

```
UNION [ALL]  
SELECT 表B. 列2  
       FROM 表B  
       WHERE 検索条件;
```

ALL指定は重複を排除しない意味

② 差集合 (MINUSまたはEXCEPT)

省略 (書式は、UNIONと同様。ただしALL指定は無い。)

2.4 グループ化

①列の値でグループを作り集計をする。

```
SELECT 列1, 集計関数([* | ALL | DISTINCT] 列2)
      FROM 表A
      GROUP BY 列1;
```

```
SELECT 科目コード, COUNT(*), AVG(成績)
      FROM 表A
      GROUP BY 科目コード ;
```

2.4 (続き) グループ化

②グループ化したデータに条件設定。

```
SELECT 列1, 集計関数([* | ALL | DISTINCT] 列2)
      FROM 表A
      GROUP BY 列1
      HAVING 集計関数の検索条件 ;
```

```
SELECT 科目コード, COUNT(*), AVG(成績)
      FROM 表A
      GROUP BY 科目コード
      HAVING COUNT(*) > 10 ;
```

2.5 一括挿入 (INSERT)

```
INSERT INTO 商品価格TBL 1行追加  
VALUES (10030,'商品名BB',1500,  
          'Y商事',1200)
```

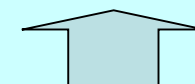
```
INSERT INTO 商品価格TBL 1行追加  
(商品コード、商品名、価格、仕入先、単価)  
VALUES (10030,'商品名BB',1500,  
          'Y商事',1200)
```

```
INSERT INTO 商品価格TBL 一括挿入  
SELECT 列1, 列2, 列3, 列4, 列5  
FROM 別の表  
WHERE 検索条件
```

SELECT *
にすれば、
バックアップが可能

商品価格TBL

商品コード	商品名	価格	仕入先	仕入単価



別の表

2.6 演算子と関数

①演算子の種類

算術演算子

+
-
*
/
%

比較演算子

=
>
>=
<
<=
<>
!=

ビット演算子

&
^
~

論理演算子

AND
BETWEEN
IN
NOT
OR
ALL
ANY
EXISTS
SOME

集合演算子

INTERSECT
MINUS
UNION
JOIN

その他演算子

+
-
IS NULL
CASE
LIKE

2.6 (続き) 関数

②関数の種類

集計関数

AVG	VAR
COUNT	VARIANCE
GROUPING	VARP
MAX	
MIN	
STDDEV	
STDEV	
STDEVP	
SUM	

文字列関数

ASCII	LOWER	SAPACE
CHAR	LPAD	STR
CHARINDEX	LTRIM	STUFF
CHR	NCHAR	SUBSTR
CONCAT	REPLACE	SUBSTRING
INSTR	REPLICATE	TRIM
LEFT	REVERS	UNICODE
LEN	RIGHT	UPPER
LENGTH	RPAD	
LENGTHB	RTRIM	

2.6 (続き) 関数

日付関数

ADD_MONTHS
DATEDIFF
DATENAME
DATEPART
DAY
GETDATE
GETUTCDATE
MONTH
MONTH_BETWEEN
SYSDATE
YEAR

変換関数

CAST
CONVERT
DECODE
ISNULL
NVL
TO_CHAR
TO_DATE
TO_NUMBER

2.6 (続き) 関数

算術関数

ABS	FLOOR	SIGN
ACOS	LN	SIN
ASIN	LOG	SQUARE
ATAN	LOG10	SORT
CEIL	MOD	TAN
CEILING	PI	TRUNK
COS	POWER	
COT	RADIANS	
DEGREES	RAND	
EXP	ROUND	

3. データ操作（副問合せ）

・問合せの中で、更に問合せを行う。

① 結果が一つのケース

・複数あればエラー

『売上数の平均を下回っている商品IDは？』

```
SELECT AVG(売上数) FROM 売上TBL;  
答えは、5になる。  
SELECT 商品ID          FROM 売上TBL  
       WHERE 売上数<5      ;
```



```
SELECT 商品ID          FROM 売上TBL  
       WHERE 売上数<  
             (SELECT AVG(売上数)  
              FROM 売上TBL) ;
```

例: 売上TBL

商品ID	売上数
A1	2
A3	8
C2	4
D1	5
D3	6

3. (続き) 副問合せ

② 結果が複数のケース

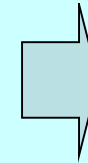
例: 売上TBL

商品ID	売上数
A1	2
A3	8
C2	4
D1	5
D3	6

商品TBL

商品ID	商品名	カテゴリー
A1	あ	飲料
A2	い	飲料
A3	う	飲料
C2	え	弁当
D1	お	菓子
D3	か	菓子

『カテゴリーが「飲料」の売上を作成する。』



商品ID	売上数
A1	2
A3	8

```
SELECT 商品ID,売上数
FROM 売上TBL
WHERE 商品ID IN (
    SELECT 商品ID FROM 商品TBL
    WHERE カテゴリー = '飲料');
```

「いずれかと等しい」の意味

3. (続き) 副問合せ

③ 相関副問合せ(繰返し)

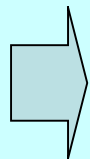
例: 売上TBL (U)

商品ID	売上数
A1	2
A3	8
C2	4
D1	5
D3	6

商品TBL (S)

商品ID	商品名	カテゴリー
A1	あ	飲料
A2	い	飲料
A3	う	飲料
C2	え	弁当
D1	お	菓子
D3	か	菓子

『カテゴリーが「飲料」の売上を作成する。』



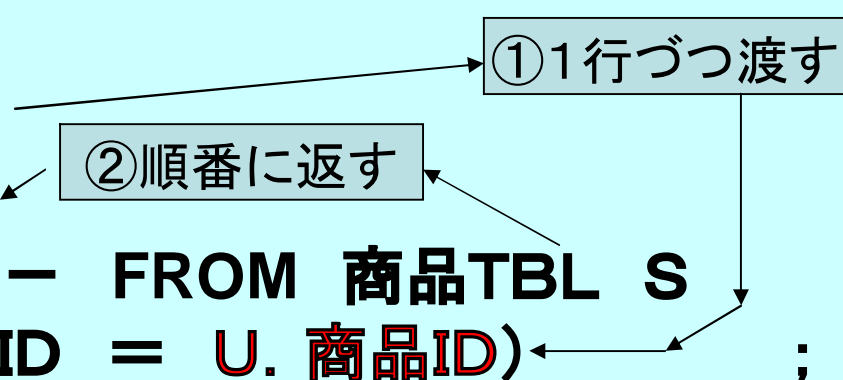
商品ID	売上数
A1	2
A3	8

外側のSQL文から、値を、副問合せ文に渡す方法。

```
SELECT 商品ID,売上数
FROM 売上TBL U
WHERE '飲料' = (
    SELECT カテゴリー FROM 商品TBL S
    WHERE S.商品ID = U.商品ID) ;
```

① 1行ずつ渡す

② 順番に返す



4. 埋め込みSQL

- ホスト言語にSQL文を記述する方式。
- 文法

①実行命令

EXEC SQL SQL文

②使用変数

先頭に「:」を付ける

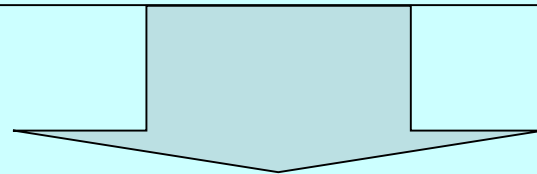
- プリコンパイルが必要

5. カーソルの使用

- 5. 1 カーソルとは？
- 5. 2 カーソルの構文
- 5. 3 実行手順
- 5. 4 使用例

5. 1 カーソルとは？

- SQLは、元来、**非手続き型言語**であり、条件を満たす複数の行を一括して扱う。
- 一方、従来からの言語は、**手続き型言語**であり、1行ずつの処理を行う。



- ホスト言語側で、**埋め込み方式**でSQLを使用する場合には、
- SQLに、データを1行ずつ取出す仕組みが必要。
(**カーソル機能**という。CURSOR)

5.2 カーソルの構文

①カーソル宣言(定義)

```
DECLARE カーソル名 CURSOR FOR  
問合せ式(=SELECT, FROM, WHERE...)  
ORDER BY 列名1, 列名2...
```

②オープン文

```
OPEN カーソル名
```

③フェッチ文

```
FETCH カーソル名 INTO  
パラメータ名1, パラメータ名2, .....,  
[ INDICATOR 標識名 ]
```

データ項目の
入れ場所

④クローズ文

```
CLOSE カーソル名
```

SQLCODE: 整数
SQLSTATE: 文字列(5)

⑤カーソル破棄文

```
DEALLOCATE カーソル名
```

5. 2 (続き)カーソルの構文

⑥更新文

```
DELETE FROM テーブル名  
WHERE CURRENT OF カーソル名
```

値が入っている

```
UPDATE テーブル名  
SET 列名 = パラメータ名  
WHERE CURRENT OF カーソル名
```


5.4 使用例

省略

参照(教科書、161ページ)

6. レポート課題

①次ページの事例について、テーブル定義の変更後に、以下のデータ操作文の具体例を作成してください。

- ・行の挿入、行の更新、行の削除
- ・問合せ文(次ページで指示されたもの)

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. の課題用の事例

②課題のテーブル

学科TBL

学科番号	年度	学科名称	主任教授
------	----	------	------

学生TBL

学籍番号	年次	入学年度	氏名
------	----	------	----

所属TBL

学科コード	年度	学生コード
-------	----	-------

科目TBL

学科番号	科目コード	年度	期	HS区分	科目名称	担当教師
------	-------	----	---	------	------	------

受講者TBL

学科番号	科目コード	年度	受講者コード	評価	粗点
------	-------	----	--------	----	----

③データ検索の事例

- ・「情報」学科、2002年度、「データベース論」科目の評価、粗点の一覧表を作成。
(学籍番号、氏名、入学年度、評価、粗点を出す(ただし入学年度・学籍番号順))
- ・上記の範囲で、受講者総数、平均点を出す。

7. 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)
- 小野哲ほか「まるごと図解、SQLがわかる」(技術評論社)
- 宮坂雅輝「SQLハンドブック」(ソフトバンク社)
- ライアンほか「SQLプログラミング入門」(ソフトバンク)
- 村上毅ほか「MySQL活用ガイド」(秀和システム)
- R. ストーンズほか「PostgreSQL活用テクニック」(インプレス)
- 松崎為裕「データベースの基礎の基礎」(ソフトバンク)
- 山田精一「Oracleのデータベース」(翔泳社)
- <http://www.ann.hi-ho.ne.jp/hirok/sql/index.html>
- <http://www.rfs.jp/sitebuilder/sql/>