

関係データベース言語 SQL（その2）

Hiroshima Institute of Technology

1

授業計画

- 第1回 ガイダンス・データベースの基本概念
- 第2回 データモデル
- 第3回 関係代数
- 第4回 データベース設計
- 第5回 リレーションの正規化
- 第6回 中間まとめ
- 第7回 関係データベース言語 (SQL1)
- 第8回 関係データベース言語 (SQL2)
- 第9回 計算機実習
- 第10回 データの検索機構 MySQL実習
- 第11回 トランザクション管理 MySQL実習
- 第12回 障害回復 MySQL実習
- 第13回 分散データベース MySQL実習
- 第14回 期末まとめ
- 第15回 応用技術と将来動向 MySQL実習

7. 関係データベース言語SQL(その2)

講義内容

- 関係演算子
- 論理演算子
- その他の演算子
- 副問合せ
- 埋め込み型SQL

Hiroshima Institute of Technology

3

7. 関係データベース言語SQL(その2)

1. 関係演算子

- 関係演算子は、左辺と右辺の値を比較する演算子

| 演算子 | 説 明 |
|-----|---------------|
| = | 左辺と右辺の値が等しい |
| > | 左辺が右辺の値より大きい |
| < | 左辺が右辺の値より小さい |
| >= | 左辺が右辺の値以上 |
| <= | 左辺が右辺の値以下 |
| ◇ | 左辺と右辺の値が等しくない |

Hiroshima Institute of Technology

4

7. 関係データベース言語SQL(その2)

1. 関係演算子

SELECT * FROM 試験成績 WHERE 点数 = '80'

SELECT * FROM 試験成績 WHERE 点数 > '60'

SELECT * FROM 試験成績 WHERE 点数 >= '60'

SELECT * FROM 試験成績 WHERE 点数 <> '60'

7. 関係データベース言語SQL(その2)

2. 論理演算子

- 論理演算子は, 左辺と右辺の値を比較する演算子

| 演算子 | 説明 |
|-----|-----|
| AND | 論理積 |
| OR | 論理和 |
| NOT | 否定 |

7. 関係データベース言語SQL(その2)

2. 論理演算子

```
SELECT * FROM 商品  
WHERE 商品名 = 'ボールペン' AND 単価 = 100
```

```
SELECT * FROM 商品  
WHERE 商品名 = '定規' OR 単価 < 150
```

```
SELECT * FROM 商品  
WHERE NOT 商品名 = '定規'
```

7. 関係データベース言語SQL(その2)

3. その他の演算子

| 演算子 | 説 明 |
|----------------|------------|
| BETWEEN | ～以上～以下 |
| IN | いずれかの値と等しい |
| IS NULL | NULL判定 |
| LIKE | 部分一致検索 |

7. 関係データベース言語SQL(その2)

3. その他の演算子

■ BETWEEN演算子

BETWEEN 演算子は、指定した式の値が範囲内に収まっているかを調べる
WHERE <式> [NOT] BETWEEN <数値式1> AND <数値式2>

■ IN演算子

IN演算子は、指定した式の値に副問合せ結果のリスト、あるいは指定した値と一致するかを調べる
WHERE <式> [NOT] IN (<副問合せ> | <値1> [<値2>...])

■ IS NULL演算子

IS NULL演算子は、= 演算子や<> 演算子では判定できない NULL を判定する
<式> IS [NOT] NULL

7. 関係データベース言語SQL(その2)

3. その他の演算子

■ LIKE演算子

LIKE演算子は、指定した文字列データのパターンマッチングを行う
WHERE <文字列式> LIKE '<文字列式>' [ESCAPE <エスケープ文字>...]

(例1)「都道府県表」より、都道府県名が「山」で始まるデータを取り出す。
SELECT * FROM 都道府県 WHERE 都道府県名 LIKE '山%'

(例2)「都道府県表」より、都道府県名に「山」が含まれるデータを取り出す。
SELECT * FROM 都道府県 WHERE 都道府県名 LIKE '%山%'

(例3)「都道府県表」より、都道府県名が3文字で真中が「山」であるデータを取り出す。
SELECT * FROM 都道府県 WHERE 都道府県名 LIKE '_山_'

7. 関係データベース言語SQL(その2)

4. 集合関数

| 集合関数 | 説 明 |
|--------------|---------|
| SUM | 合計値を求める |
| AVG | 平均値を求める |
| MAX | 最大値を求める |
| MIN | 最小値を求める |
| COUNT | 件数を求める |

Hiroshima Institute of Technology

11

7. 関係データベース言語SQL(その2)

4. 集合関数

■ SUM関数

SUM関数は、指定した数値列データの合計値を求める

ALL または DISTINCT キーワードを指定できる

ALL キーワードはデフォルト

SUM([ALL | DISTINCT]<数値列>)

戻り値は、数値列の合計値

(使用例) **SELECT SUM(点数) FROM 受験結果**

■ AVG関数

AVG関数は、指定した数値列データの平均値を求める

ALL または DISTINCT キーワードを指定できる

ALL キーワードはデフォルト

AVG([ALL | DISTINCT]<数値列>)

戻り値は、数値列の平均値

(使用例) **SELECT AVG(点数) FROM 受験結果**

Hiroshima Institute of Technology

12

7. 関係データベース言語SQL(その2)

4. 集合関数

■ MAX関数

MAX関数は、指定した数値列データの最大値を求める

MAX(<数値列>)

戻り値は、数値列の最大値

(使用例) **SELECT MAX(点数) FROM 受験結果**

■ MIN関数

MIN関数は、指定した数値列データの最小値を求める

MIN(<数値列>)

戻り値は、数値列の最小値

(使用例) **SELECT MIN(点数) FROM 受験結果**

7. 関係データベース言語SQL(その2)

4. 集合関数

■ COUNT関数

COUNT関数は、指定した列データに NULL 以外の値が何行あるかを求める

ALL または DISTINCT キーワードを指定できる

ALL キーワードはデフォルト

COUNT([ALL | DISTINCT] <列>)

戻り値は、列の行数

(使用例) **SELECT COUNT(*) FROM 受験結果**

7. 関係データベース言語SQL(その2)

5. 副問合せ

- SQL文の中にSQL文を記述し, 内側のSQL文の結果を外側のSQL文で利用する
- 内側のSQL文を副問合せという

(1)関係演算子を使う副問合せ

「試験成績」表から平均以上のデータを取り出すSQL
`SELECT 受験番号, 点数 FROM 試験成績`
`WHERE 点数 >= (SELECT AVG(点数) FROM 試験成績)`

7. 関係データベース言語SQL(その2)

5. 副問合せ

(2)IN演算子を使う副問合せ

「学生」表と「試験成績」表から60点以上のデータを取り出す

```
SELECT 受験番号, 学科, 名前 FROM 学生
WHERE 学生番号
IN ( SELECT 学生番号 FROM 試験成績 WHERE 点数 >= 60 )
```


7. 関係データベース言語SQL(その2)

5. 副問合せ

(3) EXISTS演算子を使う副問合せ

EXISTS 演算子は、指定した式に副問い合わせの結果が存在するかを調べる

「商品」表から1つの店舗で在庫数が30以上ある商品の商品コードと商品名を取り出す

```
SELECT 商品コード, 商品名 FROM 商品
WHERE EXISTS ( SELECT * FROM 在庫 WHERE
在庫数 >= 30 AND 商品コード = 在庫.商品コード )
```

7. 関係データベース言語SQL(その2)

6. 埋め込み型SQL

- 開発言語内に直接SQL文を記述できる機能を**埋め込み型SQL**(embedded SQL)という
- アプリケーションプログラムは、問合せ結果の集合を得るために**カーソル**(cursor)という概念を用いる
- カーソルは、SELECT文などによるデータベース検索による検索実行の結果を1行ずつ取得して処理するために、データベースサーバ側にある結果集合の行取得位置を示す
- カーソルを用いると結果を1行単位で処理することができる
- カーソルはOPEN命令、FETCH命令、そしてCLOSE命令で操作を行う

7. 関係データベース言語SQL(その2)

6. 埋め込み型SQL

- ホスト言語方式の利用形態が規定されている
ADA, C, COBOL, FORTRAN, MUMPS, PASCAL, PL/I, ...
- 「**埋込みSQL**」と「**モジュール言語方式**」
- アプリケーションプログラムからSQLによるデータ操作を行うためにはいくつかの解決しなければならない問題がある
 - ① **SQL記述をソースに埋込む規約が必要**
 - ② 言語プログラムの世界とSQLの世界での**データ引渡し**
 - ③ SQL実行時の**エラーの検出**

7. 関係データベース言語SQL(その2)

まとめ

- 関係演算子
- 論理演算子
- その他の演算子
- 副問合せ
- 埋め込み型SQL