

# DML – 자료관리

웹 데이터 베이스

# INSERT : 자료의 입력- Part I

- 만들어져 있는 Table에 자료를 저장한다.
- <http://dev.mysql.com/doc/refman/5.1/en/insert.html>
  - 여기에 있는 기본 형을 바탕으로 새롭게 다음과 같은 형태를 추천한다.

```
INSERT INTO tbl_name [(col_name, ... )]  
VALUES (value, ...), ... , (value);
```

## • 자료 입력

```
> INSERT INTO dept_tab (dept_id, deptKName,  
+> deptENAME) VALUES (1, '통계학과', 'Statistics');  
> INSERT INTO dept_tab VALUES (2, '수학과',  
+> 'Mathematics');  
> INSERT INTO dept_tab VALUES  
+> (3, '컴퓨터공학과', 'Computer Science')  
+> (4, '전자공학과', 'Electronic Science');
```

# INSERT : 자료의 입력- Part I

```
mysql> desc student_tab;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| myid  | int(11)       | NO   | PRI | NULL    | auto_increment |
| mySNO | char(8)       | NO   | UNI | NULL    |                |
| myName | varchar(50)   | NO   | MUL | 없음    |                |
| myDept | int(11)       | NO   | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

- > INSERT INTO student\_tab VALUES (NULL, '12345678', '고길동', 1);
- > SELECT \* FROM student\_tab;
  
- > INSERT INTO student\_tab VALUES ('', '00000001', '둘리', 2);
- > show warnings;
- > SELECT \* FROM student\_tab;

# UPDATE : 자료의 변경

- 기존 데이터의 행을 선택해서 특정 열의 값을 변경한다.
  - 행 선택문이 없으면 전체 테이블의 지정한 열의 값을 하나로 변경한다.
- 행의 선택 (WHERE 문)
  - DB는 기본적으로 행단위로 데이터를 다룬다.
  - WHERE는 특정 조건을 주어 해당 조건에 맞는 행을 선택하도록 한다.
  - 조건들은 AND와 OR 에 의해 결합된다.
- <http://dev.mysql.com/doc/refman/5.1/en/update.html> 을 바탕으로 다음과 같이 사용한다.

```
UPDATE tbl_name SET
col_name1={expr1|DEFAULT}
[, col_name2={expr2|DEFAULT}] ...
[WHERE where_condition]
[ORDER BY ...]
[LIMIT row_count]
```

# UPDATE : 자료의 변경

- 예제

- 학과 코드 3을 2로 잘못 입력하여 학과 코드를 변경하려고 한다.
  - student\_tab 에서 myDept가 2인 행을 찾아 3으로 변경한다.

> UPDATE student\_tab SET myDept = 3 WHERE myDept = 2;

- 컴퓨터 공학과의 이름이 둘리인 학생이 이름을 희동으로 개명하여 바꾸고자 한다.
  - 둘리가 이름인 학생이 여러명 있을 수 있으므로 학과 코드가 3이고 이름이 둘리인 학생의 이름을 희동으로 변경한다.

> UPDATE student\_tab SET myName = '희동이'  
+> WHERE myName = '둘리' AND myDept = 3

# DELETE : 자료 삭제

- 조건에 맞는(WHERE) 행을 선택하고 지우는 데 사용
- <http://dev.mysql.com/doc/refman/5.1/en/delete.html> 을 바탕으로 다음과 같이 사용한다.

```
DELETE FROM tbl_name  
[WHERE where_condition]  
[ORDER BY ...]  
[LIMIT row_count]
```

- 예제
  - 학번이 1001번인 학생의 데이터를 삭제하라.
    - > DELETE FROM student\_tab WHERE myid = 1001;
  - WHERE 이후가 생략되면 해당 테이블의 전체 데이터가 삭제된다.
    - > DELETE FROM student\_tab;

# SELECT : 자료 검색

- 실제 DB에서 가장 많이 사용되는 기능
- 조건에 부합(WHERE) 하는 행을 선택하여 보여준다.
- <http://dev.mysql.com/doc/refman/5.1/en/select.html> 를 참조하여 다음과 같이 사용한다.

```
SELECT select_expr [, select_expr ...]  
  [FROM table_references  
  [WHERE where_condition  
  [GROUP BY {col_name | expr | position}  
    [ASC | DESC], ... [WITH ROLLUP]]  
  [HAVING where_condition  
  [ORDER BY {col_name | expr | position}  
    [ASC | DESC], ...]  
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

# SELECT : 자료 검색

- 예제
  - 원활한 실습을 위해 다음을 입력하자(blog에서 제공하는 자료)

```
INSERT INTO student_tab VALUES
  (NULL, '12345678', '고길동', 1),
  (NULL, '00000001', '둘리', 2),
  (NULL, '00000002', '희동이', 1),
  (NULL, '00000003', '도우너', 3),
  (NULL, '00000004', '또치', 4),
  (NULL, '00000005', '마이콜', 1),
  (NULL, '00000006', '철수', 3),
  (NULL, '00000007', '영희', 2);
```



# SELECT : 자료 검색

- 학과 코드 1번에 해당하는 학생들을 검색하라.

```
SELECT * FROM student_tab WHERE myDept = 1;
```

- 학과 코드 1번에 해당하는 학생들의 학번과 이름을 검색하라.

```
SELECT mySNO, myName FROM student_tab WHERE myDept = 1;
```

- 학과별 학생수를 구하여라. (count(열이름) : 열의 숫자 반환);

```
SELECT myDept, count(myDept) as '명'  
FROM student_tab GROUP BY myDept;
```

- 학과 코드가 1인 학과를 제외한 학과별 학생수를 구하라

```
SELECT myDept, count(myDept) as '명'  
FROM student_tab GROUP BY myDept  
HAVING myDept <> 1;
```

# SELECT : 자료 검색

- 과목코드 순으로 학번과 학생 이름을 출력하시오. (동일 학과 내에서는 학번의 역순)

```
SELECT myDept, mySNO, myName
      FROM student_tab
      ORDER BY myDept ASC, mySNO DESC;
```

- 학번이 가장 빠른 학생의 학번과 이름을 출력하시오.
  - 학번의 오름차순으로 정렬하고 그중 첫번째 행 출력

```
SELECT mySNO, myName FROM student_tab
      ORDER BY mySNO LIMIT 0, 1;
```

# 다음 시간에는 ...

- 두 테이블의 결합에 대해 이야기 해 봅니다.