

# DDL – TABLE을 만들자

웹 데이터 베이스

# TABLE

- 자료가 저장되는 공간
  - 문자 자료의 경우 DB 생성시 지정한 Character Set 대로 저장
  - Table 생성시 Table의 구조를 결정짓는 열 속성 지정
  - 열(Column, Attribute)은 이름과 자료형을 갖는다.
    - 자료형 : <http://dev.mysql.com/doc/refman/5.1/en/data-types.html>
- TABLE 생성
  - <http://dev.mysql.com/doc/refman/5.1/en/create-table.html>
  - CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name  
(create\_definition, ... )  
[table\_options]  
[partition\_options]  
[select\_statement];
- TABLE 제거
  - DROP TABLE [IF EXISTS] tbl\_name;

# TABLE : 예를 통한 생성문

- CREATE TEMPORARY TABLE tmp\_tab (  
    myid                int        NOT NULL  
);
  - 임시 테이블 tmp\_tab을 만든다.
  - 테이블의 열은 myid라는 이름을 가지며 저장될 자료의 형태는 정수형 자료(INT)이고 반드시 값이 있어야 한다(NOT NULL).
  - TEMPORARY TABLE은 현재 접속에서만 유지되는 TABLE로 현재 접속을 종료하면 사라지는 TABLE이다.
- CREATE TABLE dept\_tab (  
    dept\_id                int        NOT NULL        PRIMARY KEY,  
    deptKName            varchar(255)  NOT NULL,  
    deptENAME            varchar(255)  NOT NULL  
);
  - 테이블 dept\_tab을 만든다.
  - dept\_id는 주키이다.
  - > **desc** dept\_tab

```
C:\APM_Setup\Server\MySQL5\bin\mysql.exe
mysql> CREATE TABLE dept_tab (
  -> dept_id          int          NOT NULL          PRIMARY KEY,
  -> deptKName       varchar(255)  NOT NULL,
  -> deptENAME       varchar(255)  NOT NULL
  -> );
Query OK, 0 rows affected (0.07 sec)

mysql> desc dept_tab;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dept_id    | int(11)       | NO   | PRI | NULL    |       |
| deptKName  | varchar(255) | NO   |     | NULL    |       |
| deptENAME  | varchar(255) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)

mysql>
```



```
C:\APM_Setup#Server#MySQL5#bin#mysql.exe
mysql> CREATE TABLE student_tab (
  -> myid      int          NOT NULL      AUTO_INCREMENT PRIMARY KEY,
  -> mySNO     char(8)       NOT NULL      UNIQUE KEY,
  -> myName    varchar(50)   NOT NULL      DEFAULT '없음',
  -> myDept    int          NOT NULL      COMMENT 'dept_tab의 dept_id를 참조한
  [.]' REFERENCES dept_tab(dept_id)
  -> );
Query OK, 0 rows affected (0.03 sec)

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   |     | 없음    |                |
| myDept | int(11)       | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

# TABLE : 자료형

- 자료형
  - 지정 형태 : 자료형이름(정의)
- 문자열 자료형
  - 고정 길이 문자열과 가변길이 문자열
    - 고정길이 문자열
      - 지정한 크기만큼 용량 할당
      - 지정한 크기보다 적게 문자열이 입력되더라도 실제 차지하는 양은 지정한 크기와 동일
      - 검색에 있어 상대적으로 빠르다.
      - CHAR : 뒤에 붙는 공백문자는 무시된다.
    - 가변길이 문자열
      - 지정한 크기는 최대 크기
      - 지정한 크기보다 적게 문자열이 입력될 경우 실제 차지하는 용량은 문자열의 길이만큼만 차지함
      - 용량을 효율적으로 관리할 수 있다.
      - VARCHAR, TEXT 등

# TABLE : 자료형

- 문자열 자료형

Value	CHAR(4)	Storage Required	VARCHAR(4)	Storage Required
"	' '	4 bytes	"	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

- > **CREATE TABLE vc (v VARCHAR(4), c CHAR(4));**
- > **INSERT INTO vc VALUES ('ab ', 'ab');**
- > **SELECT \* FROM vs;**
- > **INSERT INTO vc VALUES ('ab ', 'ab ');**
- > **SELECT CONCAT('(', v, ')'), CONCAT('(', c, '))' FROM vc;**



# TABLE : 자료형

- 문자열 자료형

- 열거형 자료 : ENUM

- Domain을 한정하여 해당 Domain의 값만 입력될 수 있도록 한다.
    - Ex

- > CREATE TABLE enumTest ( name ENUM('small', 'medium', 'large') );

- > INSERT INTO enumTest VALUES ('small');

- > INSERT INTO enumTest VALUES ('medium-large');

- > SELECT \* FROM enumTest;

- 집합 자료 : SET

- 한정된 Domain들의 부분집합이 값이 되도록 한다(최대 64개의 원소로 Domain을 구성할 수 있다)
    - 중복될 경우 생략된다.

- > CREATE TABLE setTest ( sd SET('a', 'b', 'c', 'd') );

- > INSERT INTO setTest VALUES ('a, d'), ('d, a'), ('a, d, d '), ('a, d, d');

- > SELECT \* FROM setTest;

# TABLE : 자료형

- 숫자 자료형
  - 정수형
    - 정수자료형 [UNSIGNED] [ZEROFILL]

Type	Storage (Bytes)	Minimum Value (Signed/Unsigned)	Maximum Value Signed/Unsigned)
<a href="#"><u>TINYINT</u></a>	1	-128	127
		0	255
<a href="#"><u>SMALLINT</u></a>	2	-32768	32767
		0	65535
<a href="#"><u>MEDIUMINT</u></a>	3	-8388608	8388607
		0	16777215
<a href="#"><u>INT</u></a>	4	-2147483648	2147483647
		0	4294967295
<a href="#"><u>BIGINT</u></a>	8	-9223372036854775808	9223372036854775807
		0	1844674407370955161

# TABLE : 자료형

- 숫자 자료형
  - 실수형
    - 부동 소수점형 [(M, D)] [UNSIGNED] [ZEROFILL]
      - FLOAT
        - 4Byte, 단일 정밀도
          - $-3.402823466E+38$  에서  $-1.175494351E-38$ , 0 이고, 그리고  $1.175494351E-38$  에서  $3.402823466E+38$
        - M : 전체 자리, D : 소수 이하 자리수 지정
      - DOUBLE
        - 8Byte, 이중 정밀도
          - $1.7976931348623157E+308$  에서  $-2.2250738585072014E-308$ , 0, 그리고  $2.2250738585072014E-308$  에서  $1.7976931348623157E+308$  까지
    - 고정 소수점형
      - DECIMAL (M, D)
      - 65자리까지 표현 가능

# TABLE : 자료형

```
> CREATE TEMPORARY TABLE numericTest (  
-> col1 int zerofill,  
-> col2 int unsigned,  
-> col3 float(5, 2),  
-> col4 decimal(5, 2)  
-> );  
> INSERT INTO numericTest VALUES (1, -1, 5.2, 5.2);  
> INSERT INTO numericTest VALUES (1, 12, 5.222, 5.222);  
> SELECT * FROM enumTest;  
> show warnings;
```

# 다음 시간에는 ...

- SQL 에 대해 학습합니다.
  - TABLE 나머지에 대해 이야기 하고 TABLE 만드는 과정을 정리합니다.