



Chapter10

데이터베이스 설치 및 기본 SQL문



개요

1

학습목표

- ✓ 데이터베이스에 대한 기본적인 이해를 통해서 필요한 자료를 검색하고, 저장하고, 수정 및 삭제하는 방법을 익힌다.

2

학습내용

- ✓ 데이터베이스란?
- ✓ **MySQL** 질의문을 이용한 회원테이블 데이터베이스 작성하기
- ✓ 기본적인 **SQL**문 **create, select, insert, update, delete** 실습



데이터베이스란?

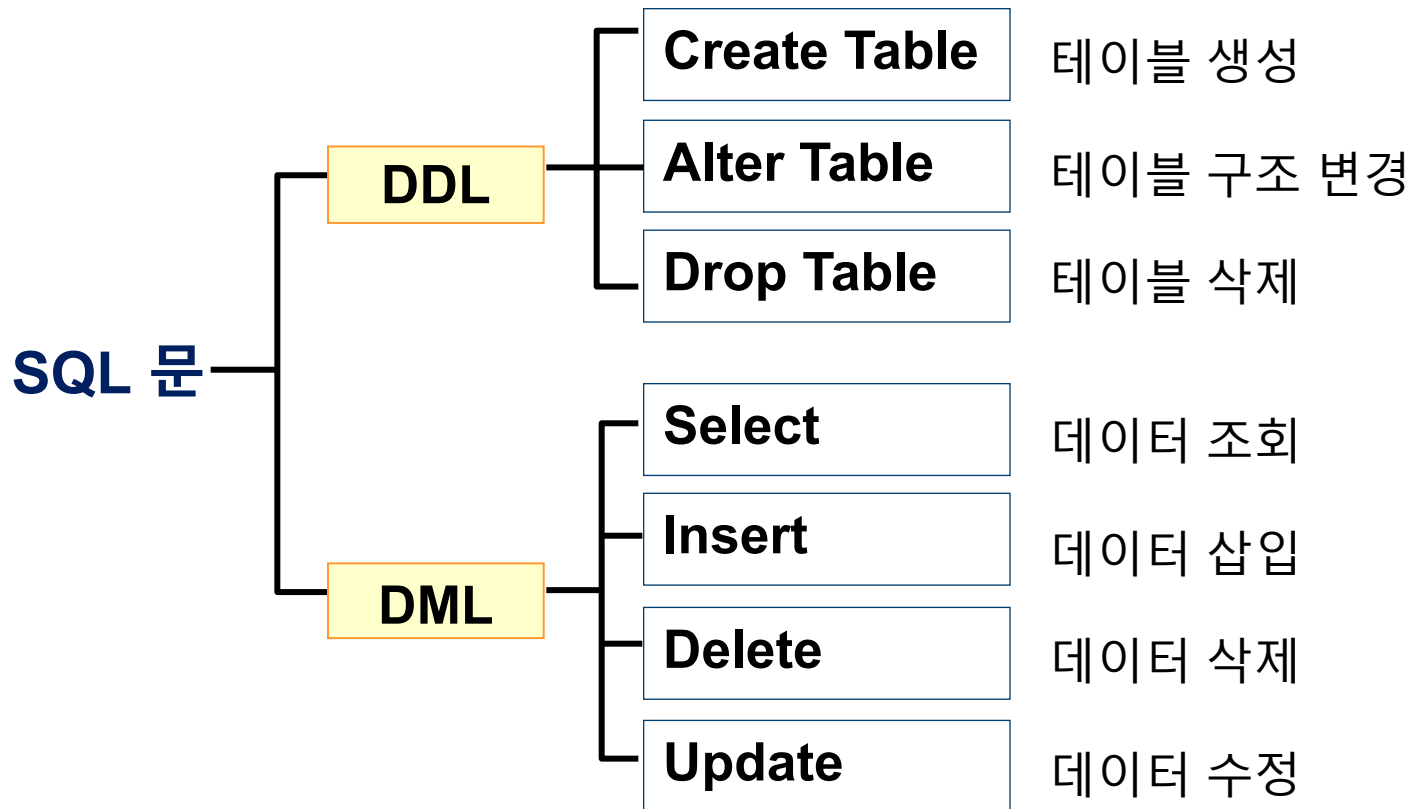
❖ 데이터베이스 & DBMS

- 데이터베이스
 - 많은 자료들을 특정한 규칙에 맞게 대용량의 저장장치에 보관하여 필요한 업무에 사용하는 자료의 저장 창고
- DBMS(DataBase Management System)
 - 저장된 데이터를 읽고, 변경, 추가, 삭제
 - 종류 : Oracle, MS-SQL, My-SQL, PostgreSQL, Informix 등
- SQL(Structured Query Language)
 - 약자로 표준화된 데이터베이스 질의어



데이터베이스란?

❖ SQL문 구성





데이터베이스란?

❖ MySQL 설치

- <https://dev.mysql.com/downloads/windows/installer/5.7.html>
Download from Oracle eDelivery 다운로드 설치

❖ 데이터베이스 외부 관리 툴 설치하기

- <https://www.heidisql.com>
Heidisql 10.1 MySQL Developer – Freeware 다운로드 설치

❖ 데이터베이스 접속하기

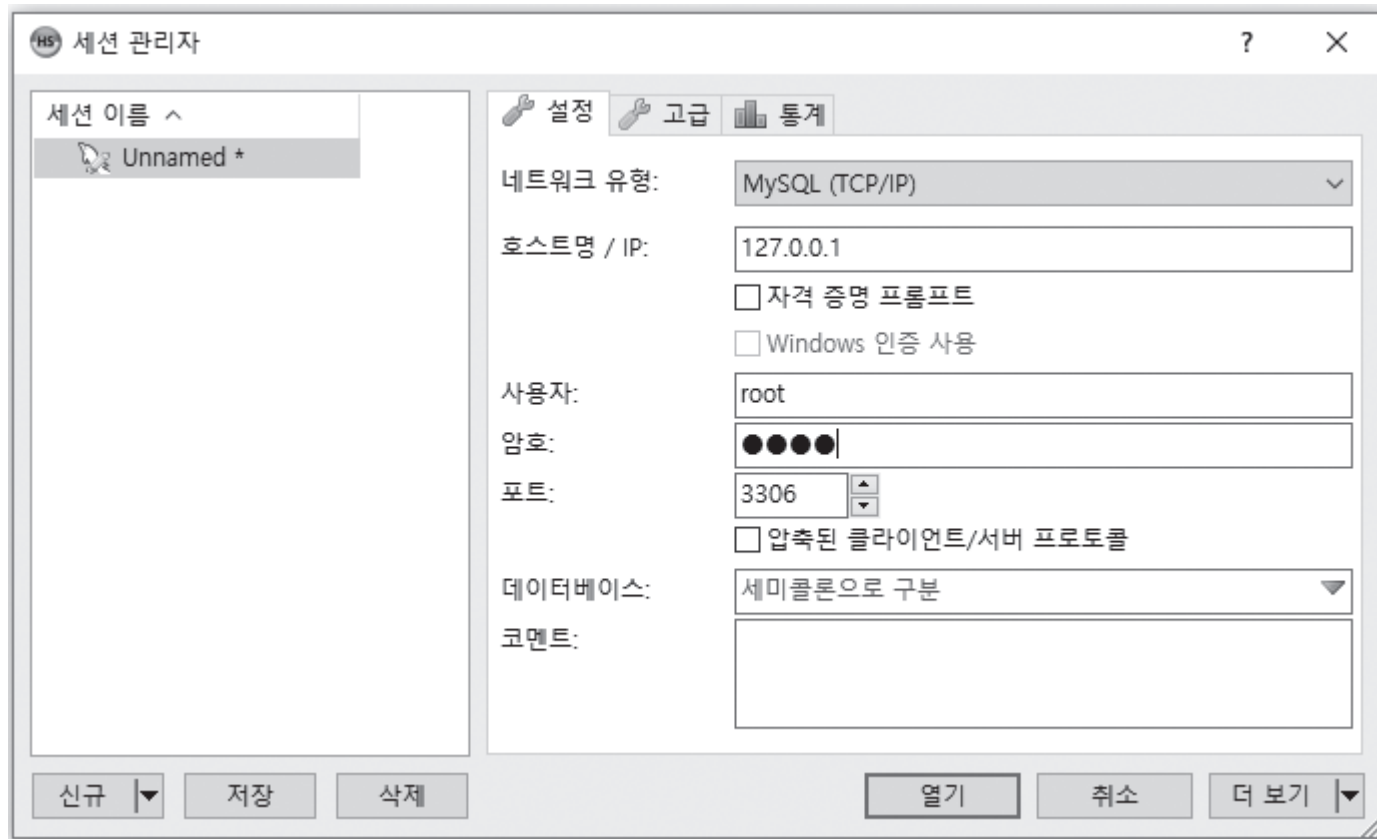
- 터미널을 통한 접속
 - cmd창에서 C:\mysql\bin>mysql
 - Heidiql for MySQL에서의 접속



Hedysql로 Mysql 접속하기

❖ HeidiSQL for MySQL에서의 접속

- 사용자명과 암호를 입력한 뒤 접속을 클릭



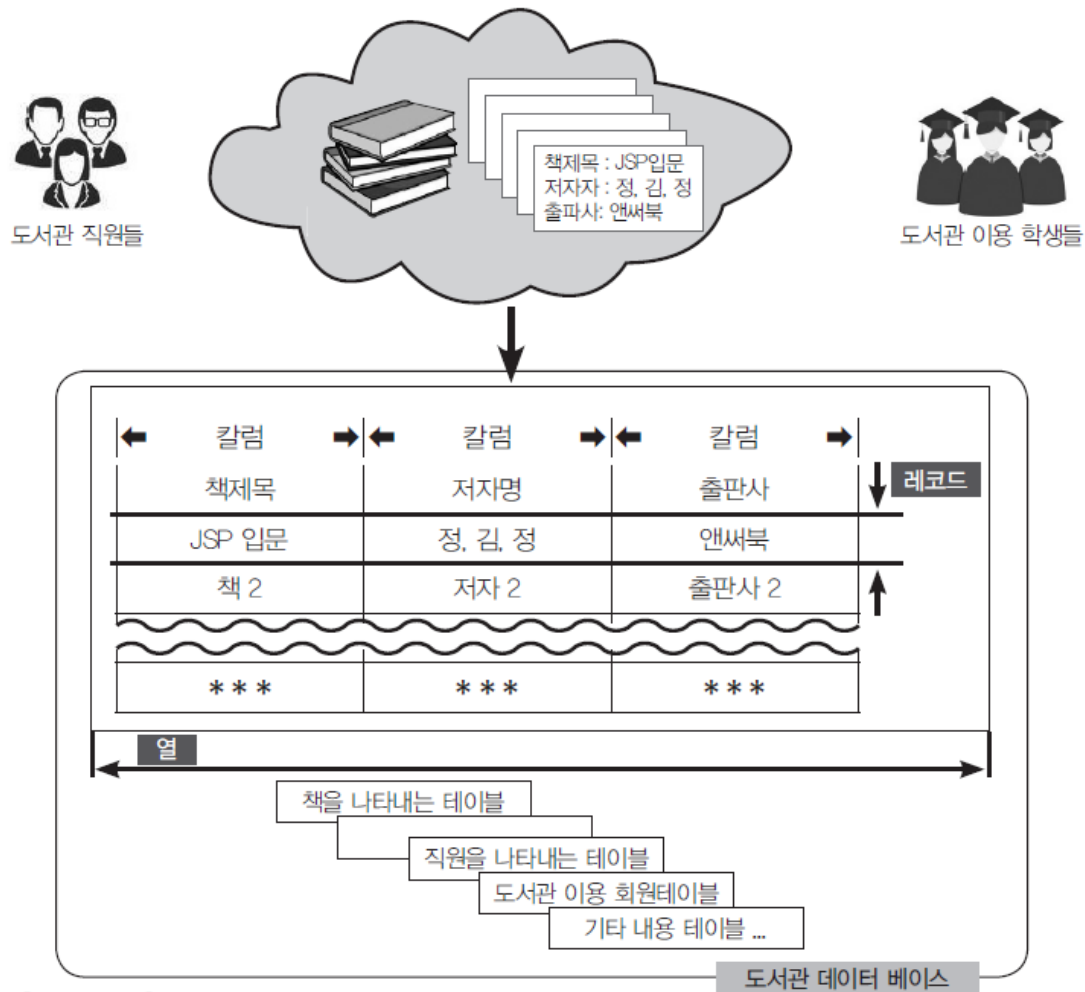


❖ 데이터베이스 용어 정리

- 데이터베이스
 - 다수의 테이블이 모여있는 공간
- 테이블
 - 다수의 속성과 레코드가 모여있는 공간
- 열, 속성(Attribute)
 - 하나의 행(가로)를 이루고 있는 다수의 속성
 - 테이블에서 각각의 세로로 구분되는 부분
- 레코드
 - 테이블에서 행(가로)에 해당되는 부분
 - 여러 속성값의 집합
- Primary Key(기본키)
 - 레코드를 구분할 수 있도록 유일한 값을 가지는 속성



❖ 데이터베이스의 구성 '도서관'



▲ [그림 10-15] 칼럼, 레코드, 테이블, 데이터베이스 개념도



❖ MySQL 데이터형

| | 데이터형 | 저장공간 크기 | 설명 및 특징 |
|-----|-----------|------------|------------|
| 숫자형 | INT(size) | 4 bytes | 숫자형 칼럼(정수) |
| | FLOAT | 4 bytes | 숫자형 칼럼(실수) |
| | DOUBLE | 8 byte | 숫자형 칼럼(실수) |
| | REAL | 8 byte | 숫자형 칼럼(실수) |
| 날짜형 | DATETIME | 8 byte | 날짜형 칼럼 |
| | DATE | 3 byte | 날짜형 칼럼 |
| | TIMESTAMP | 4 byte | 날짜형 칼럼 |
| 문자형 | CHAR | 1~255까지 저장 | 문자형 칼럼 |
| | VARCHAR | 1~255까지 저장 | 문자형 칼럼 |
| | BOLB | 최대길이 65536 | 문자형 칼럼 |



❖ 회원테이블 만들기

| 회원테이블 |
|------------|
| 회원아이디 |
| 패스워드 |
| 이름 |
| 주민등록번호앞자리 |
| 주민등록번호 뒷자리 |
| 이메일 |
| 전화번호 |
| 우편번호 |
| 주소 |
| 직업 |

| tblRegister |
|----------------------|
| id: VARCHAR(20) |
| pwd: VARCHAR(20) |
| name: VARCHAR(20) |
| num1: CHAR(6) |
| num2: CHAR(7) |
| email: VARCHAR(30) |
| phone: VARCHAR(30) |
| zipcode: CHAR(5) |
| address: VARCHAR(30) |
| job: VARCHAR(30) |



❖ 데이터베이스 생성

- 데이터베이스 생성

```
CREATE DATABASE [DATABASE_NAME]
```

설명 : [DATABASE_NAME]의 부분에 만들어질 데이터베이스 이름을 적습니다.

예 CREATE DATABASE myDB

❖ 데이터베이스 사용

```
USE [DATABASE_NAME]
```

설명 : [DATABASE_NAME] 사용할 데이터베이스 이름을 입력합니다.



❖ 회원테이블 생성

▪ <source/ch10/table.sql>

```
01 : CREATE TABLE tblRegister(  
02 :     id          VARCHAR(20) NOT NULL,  
03 :     pwd          VARCHAR(20) NOT NULL,  
04 :     name         CHAR(6) NULL,  
05 :     num1         CHAR(6) NULL,  
06 :     num2         CHAR(7) NULL,  
07 :     email        VARCHAR(30) NULL,  
08 :     phone        VARCHAR(30) NULL,  
09 :     zipcode      CHAR(5) NULL,  
10 :     address      VARCHAR(60) NULL,  
11 :     job          VARCHAR(30) NULL  
12 : );
```



❖ 생성된 테이블 관리하기

- 테이블 보기 명령

```
SHOW TABLE;
```

- 테이블속성 보기 명령

```
DESC [TABLE_NAME]
```

설명 : [TABLE_NAME]에 원하는 테이블이름을 입력합니다.

- 테이블구조 변경 명령(특정 칼럼에 키를 부여할 때)

```
ALTER TABLE [TABLE_NAME]
```

```
ADD PRIMARY KEY (COL_NAME) ;
```

설명 : [TABLE_NAME] 구조를 변경할 테이블명을 입력합니다.

COL_NAME 구조를 변경할 칼럼명을 입력합니다.



❖ 데이터를 조회하는 명령

- 데이터 조회 명령

```
SELECT * FROM [TABLE_NAME]
```

설명 : '*' 은 모든 칼럼을 불러온다는 뜻입니다. 필요한 칼럼만 불러올 경우에는 칼럼명을 입력하고 중간에 ','를 사용해서 구분하면 됩니다.

[TABLE_NAME]에 조회대상 테이블이름을 입력합니다.

예 SELECT ID, PASSWD, JOB, FROM tblRegister

```
001  SELECT * FROM tblRegister;
```



❖ 데이터를 입력하는 명령

```
INSERT INTO [TABLE_NAME] (COL_NAME1, COL_NAME2...)
```

```
VALUES (INPUT_VALUE1, INPUT_VALUE2...)
```

설명 : [TABLE_NAME] 데이터를 입력할 대상 테이블이름을 입력합니다.

[COL_NAMEn] 데이터가 입력될 칼럼이름을 입력합니다.

[INPUT_VALUEn] 실제 데이터를 입력합니다.

```
001  INSERT INTO
002      tblRegister(ID, PASSWD, NAME, NUM1, NUM2, EMAIL, PHONE,
003                  ZIPCODE, ADDRESS, JOB)
004  VALUES( 'vbass' , '11111111' , '김준형' , '1234567' , '1234567' ,
005          'vbass@naver.com' , '011-111-1111' , '111-111' , '하와이' ,
006          '프로그래머' );
```



❖ 데이터를 변경하는 명령

```
UPDATE [TABLE_NAME] SET [COL_NAME1] = [VALUE1], ..... WHERE [조건];
```

설명 : [TABLE_NAME] 데이터를 바꿀 칼럼을 가지고 있는 테이블 명을 입력합니다.

[COL_NAMEn]은 데이터를 바꿀 칼럼 명을 지정합니다.

[VALUEn]은 실제 변경할 데이터를 입력합니다.

[조건] WHERE문 다음에 나오는 조건은 데이터를 바꿀 조건을 지정하는 것입니다. 해당 조건이 맞는 행만 값이 바뀌게 됩니다. 만약 WHERE문이 주어지지 않은 경우라면 테이블의 모든 행에 있는 칼럼 값이 바뀌게 됩니다.

```
001  UPDATE tblRegister SET PWD = '4321'
002  WHERE ID='rorod';
```



❖ 데이터를 삭제하는 명령

```
DELETE FROM [TABLE_NAME] WHERE [조건];
```

설명 : [TABLE_NAME] 데이터를 삭제할 필요가 있는 테이블 명을 입력합니다.

[조건] WHERE절에 있는 조건문은 이 조건에 만족하는 행에 대해서 데이터를 삭제하기 위한 조건입니다. 만약 WHERE문이 주어지지 않는다면 테이블에 입력되어 있는 모든 행들이 삭제가 됩니다.

```
001 DELETE FROM tblRegister WHERE ID='vbass' ;
```




Thank You !