

# 5장. MariaDB(MySQL) 사용법과 필수 쿼리

컴퓨터정보과  
김진숙

# 이 장에서 공부할 것은...

- JSP 프로그래밍을 하는 데 필수적인 수준의 MariaDB(MySQL) 사용법
- 학습내용
  - MariaDB 개요
  - MariaDB 설치와 클라이언트 사용
  - 사용자 계정 생성
  - 데이터베이스 관련 명령어
  - 테이블 관련 명령어
  - 데이터 조작 명령어
  - 배치 파일(batch file)을 통한 명령 실행

# 이 단원이 끝나면 알게 되는 것

1. 테이블, 레코드, 필드의 의미가 무엇인지 설명하시오(그러서 표시).
2. MariaDB 접속을 위해 사전에 알고 있어야 하는 중요 정보는 무엇인가?
3. 새로운 계정을 생성하는 쿼리를 적어보시오. 이때 사용자 아이디는 kim, 비밀번호는 abcd, 이 계정을 위한 전용 데이터베이스는 kimdb로 한다.
4. "sample"이라는 이름의 데이터베이스를 만드는 쿼리와 삭제하는 쿼리를 적어 보시오.
5. "addr"이라는 이름의 테이블이 있다고 할 때, 이 테이블의 구조를 조회하는 쿼리와, 테이블을 삭제하는 쿼리를 적어 보시오.
6. 레코드 추가, 조회, 수정, 삭제 쿼리의 사용 형식을 정리해 보시오.
7. HeidiSQL에서 "데이터베이스를 SQL로 내보내기"를 하면, 어떤 형태의 파일로 백업되는가?

# 5.1 MariaDB 개요

## ■ 관계형 데이터베이스

### ■ 데이터를 **테이블** 형태로 저장하는 데이터베이스

- 테이블의 한 줄 : 레코드(record) 또는 로우(row; 말 그대로 행이라는 뜻)
- 세로 열 : 필드(field) 또는 컬럼(column; 열이라는 뜻)

table

num	name	kor	eng	math
1	홍길동	50	60	70
2	이순신	65	75	85
3	강감찬	60	80	70

record (row)

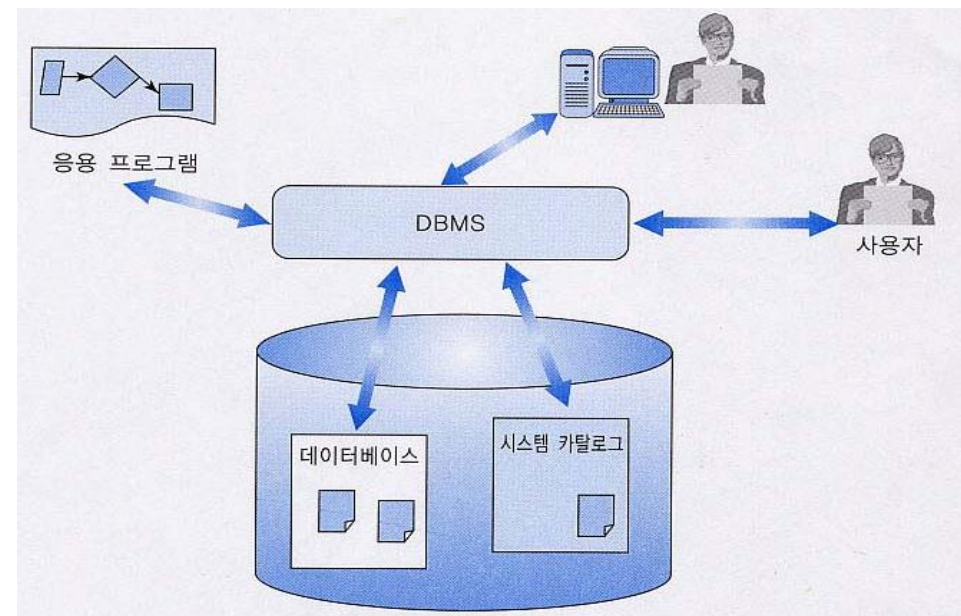
field (column)

## ■ 데이터베이스

- 관련 있는 테이블들의 집합
- 하나의 컴퓨터에 여러 개의 데이터베이스가 있을 수 있다.

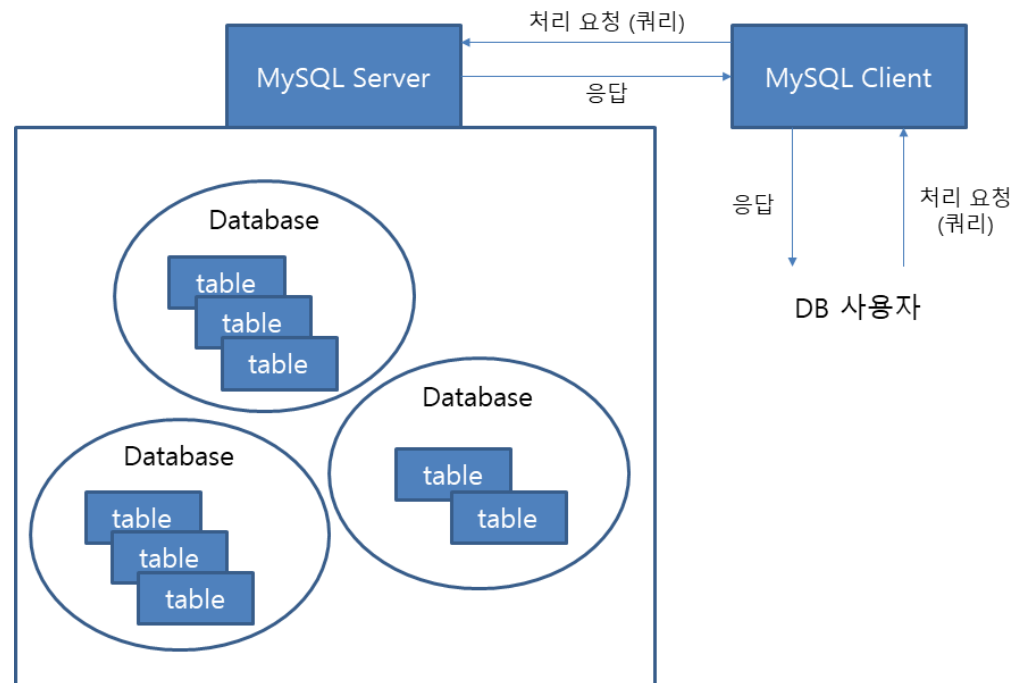
# 5.1 MariaDB 개요

- DBMS(DataBase Management System; 데이터베이스관리시스템)
  - 데이터베이스들을 서로 문제가 생기지 않게 관리해주는 소프트웨어
  - 데이터베이스에 관련된 동작들을 할 때에는
    - 사용자가 직접 데이터베이스를 접근하여 조작하지 않음
    - 모두 DBMS에게 요청하고, DBMS가 이 요청을 받아들여 실제 동작을 수행
  - MariaDB(MySQL), MS-Access, MS-SQL Server, Oracle 등은 DBMS의 이름
  - MariaDB는 MySQL에 호환되는 완전 무료 DB



# 5.1 MariaDB 개요

- MariaDB의 구조 및 동작
  - MariaDB 서버 : DBMS
  - MariaDB 클라이언트 : 사용자가 MariaDB 서버에게 줄 명령을 전달하거나, 서버로부터의 응답을 화면에서 볼 수 있게 해주는 프로그램



## 5.2 MariaDB 설치와 클라이언트 사용

- MariaDB는 <http://mariadb.com/downloads/> 에서 다운로드

The screenshot shows the MariaDB download page. A red box highlights the 'MariaDB Platform' tab in the top navigation bar. Another red box highlights the 'Version' dropdown menu, which is set to '10.4.6-GA'. A third red box highlights the 'Download' button at the bottom right. The page content includes a description of MariaDB Server as a lightweight but powerful, innovative but mature, and 100% open source database. It also lists features like Oracle Database compatibility, temporal tables, transparent sharding, and modern SQL. The download information at the bottom shows the URL <https://downloads.mariadb.com/MariaDB/mariadb-10.4.6/winx64-packages/mariadb-10.4.6-winx64.msi> and the file size of 49.38 MB.

MariaDB Platform apt/yum Connectors Tools

MariaDB Server MariaDB ColumnStore MariaDB MaxScale Developer Docker Image

MariaDB Server: Lightweight but powerful, innovative but mature, and 100% open source

MariaDB Server is the foundation of MariaDB Platform, an enterprise database platform for hybrid transactional/analytical workloads. It features Oracle Database compatibility (e.g., sequences and PL/SQL), temporal tables, transparent sharding, instant schema changes, point-in-time rollback and modern SQL (i.e., common table expressions, window functions, JSON functions and table value constructors).

Version 10.4.6-GA

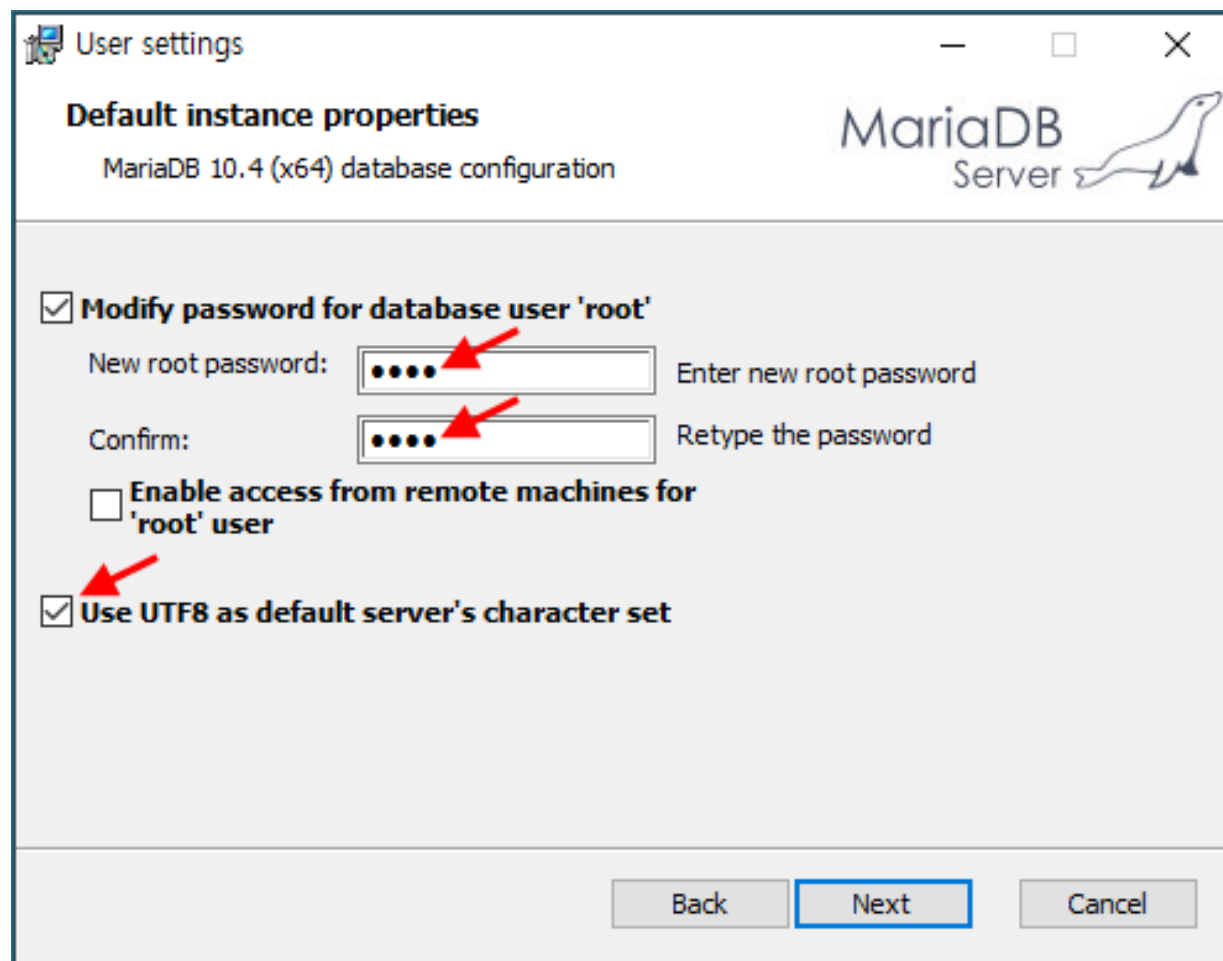
OS MS Windows (64-bit)

Release Notes Show All Files

MariaDB Server <https://downloads.mariadb.com/MariaDB/mariadb-10.4.6/winx64-packages/mariadb-10.4.6-winx64.msi> 49.38 MB Download

## 5.2 MariaDB 설치와 클라이언트 사용

- Msi 설치파일을 더블클릭하면 설치 시작됨
  - 비번은 모두 **1111** 로 합시다!!





## 5.2 MariaDB 설치와 클라이언트 사용

- 자바 커넥터 파일도 다운로드 받아 보관해 둬
  - mariadb-java-client-2.7.5.jar

The screenshot shows the MariaDB Connectors download page. The 'Connectors' tab is selected in the top navigation bar. Below it, the 'Data Access' category is chosen. The main heading reads 'Lightweight, advanced connectors for high-performance data access and data streaming.' The 'Product' dropdown menu is set to 'Java 8 connector'. The description states: 'MariaDB Connector/J is a lightweight JDBC driver (Type 4) for building Java applications on top of MariaDB Platform, complete with built-in connection pooling and encrypted connections via TLS/SSL. MariaDB Connector/J is LGPL. This version (2.4.x) of MariaDB Connector/J is for Java 8.' There are links for 'Release Notes' and 'Show All Files'. At the bottom, a table lists the download for 'Java 8 connector' with the URL 'https://downloads.mariadb.com/Connectors/java/connector-java-2.4.2/mariadb-java-client-2.4.2.jar', a size of '573.93 KB', and a prominent 'Download' button.

Product	Download Link	Size	Action
Java 8 connector	<a href="https://downloads.mariadb.com/Connectors/java/connector-java-2.4.2/mariadb-java-client-2.4.2.jar">https://downloads.mariadb.com/Connectors/java/connector-java-2.4.2/mariadb-java-client-2.4.2.jar</a>	573.93 KB	Download

## 5.2 MariaDB 설치와 클라이언트 사용

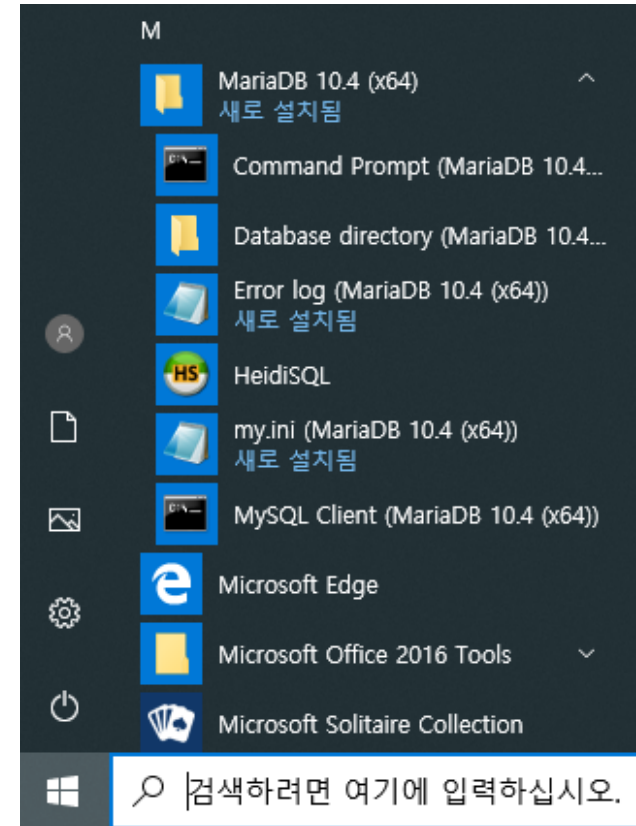
### ■ MySQL 클라이언트

- 명령행 방식 클라이언트

- 명령 행 방식이라 쿼리를 입력하거나 수정할 때에 불편함

- GUI(Graphic User Interface) 방식의 MySQL 클라이언트 : HeidiSQL

- 대부분의 데이터베이스 조작을 메뉴 선택만으로 할 수 있으니 수업에서는 항상 쿼리를 직접 입력해서 데이터베이스를 조작할 것
  - 응용 프로그램에서는 오직 쿼리로만 데이터베이스를 조작해야 하기 때문

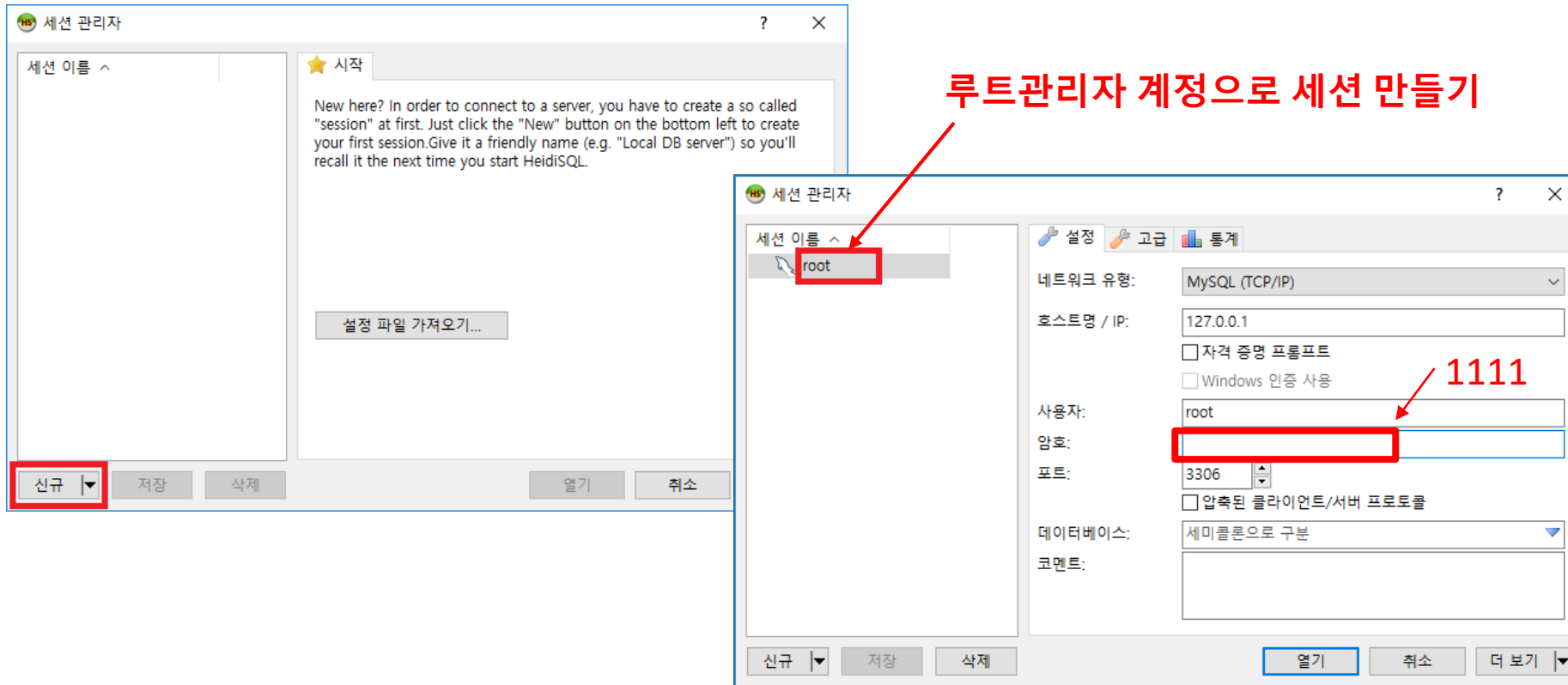


GUI 방식  
클라이언트

명령행 방식  
클라이언트

## 5.2 MariaDB 설치와 클라이언트 사용

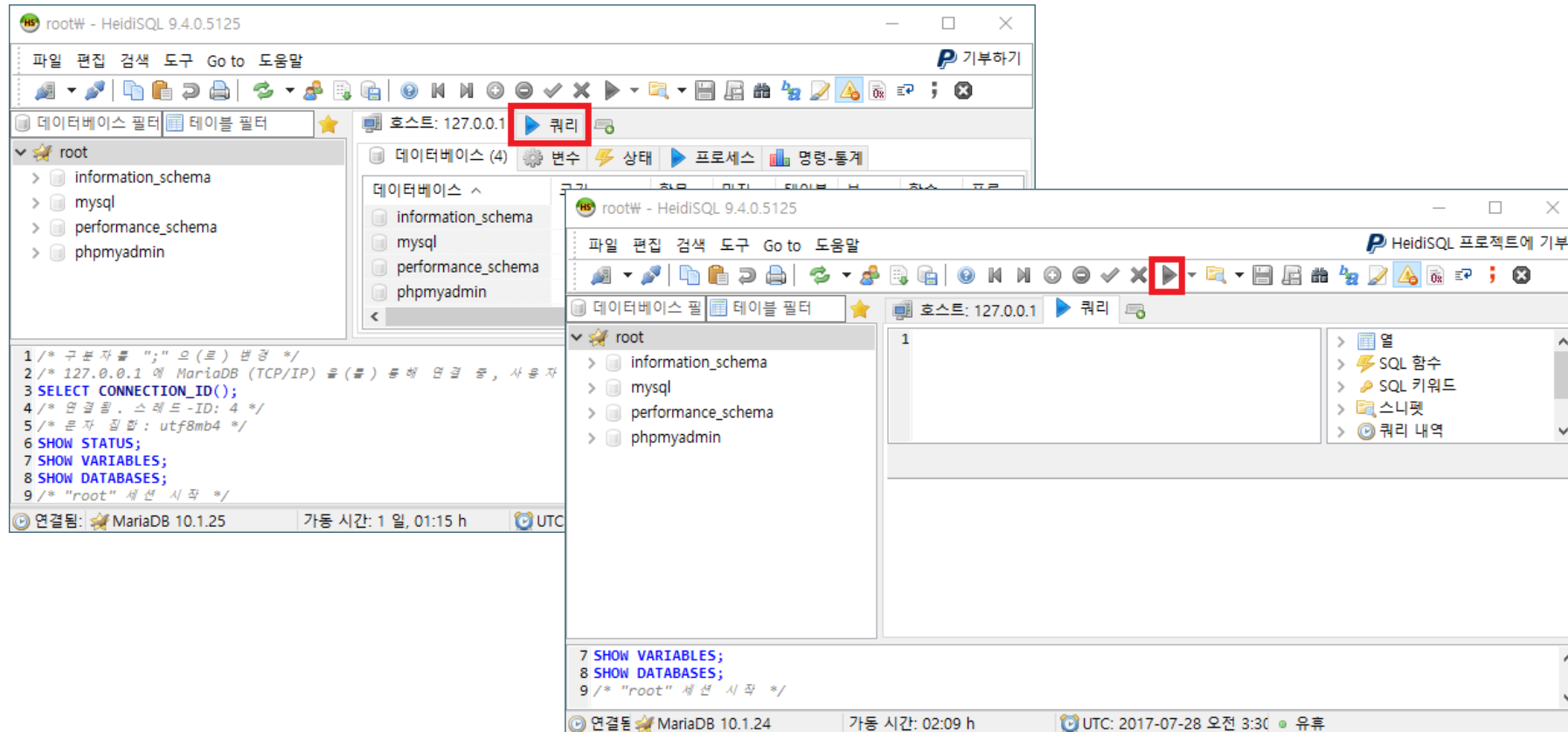
- 계정 정보 입력 및 저장
  - HeidiSQL을 실행하면 세션 관리자 화면이 먼저 나옴
  - 세션 관리자 : MySQL에 접속할 계정 정보 관리자



## 5.2 MariaDB 설치와 클라이언트 사용

### ■ 쿼리 실행

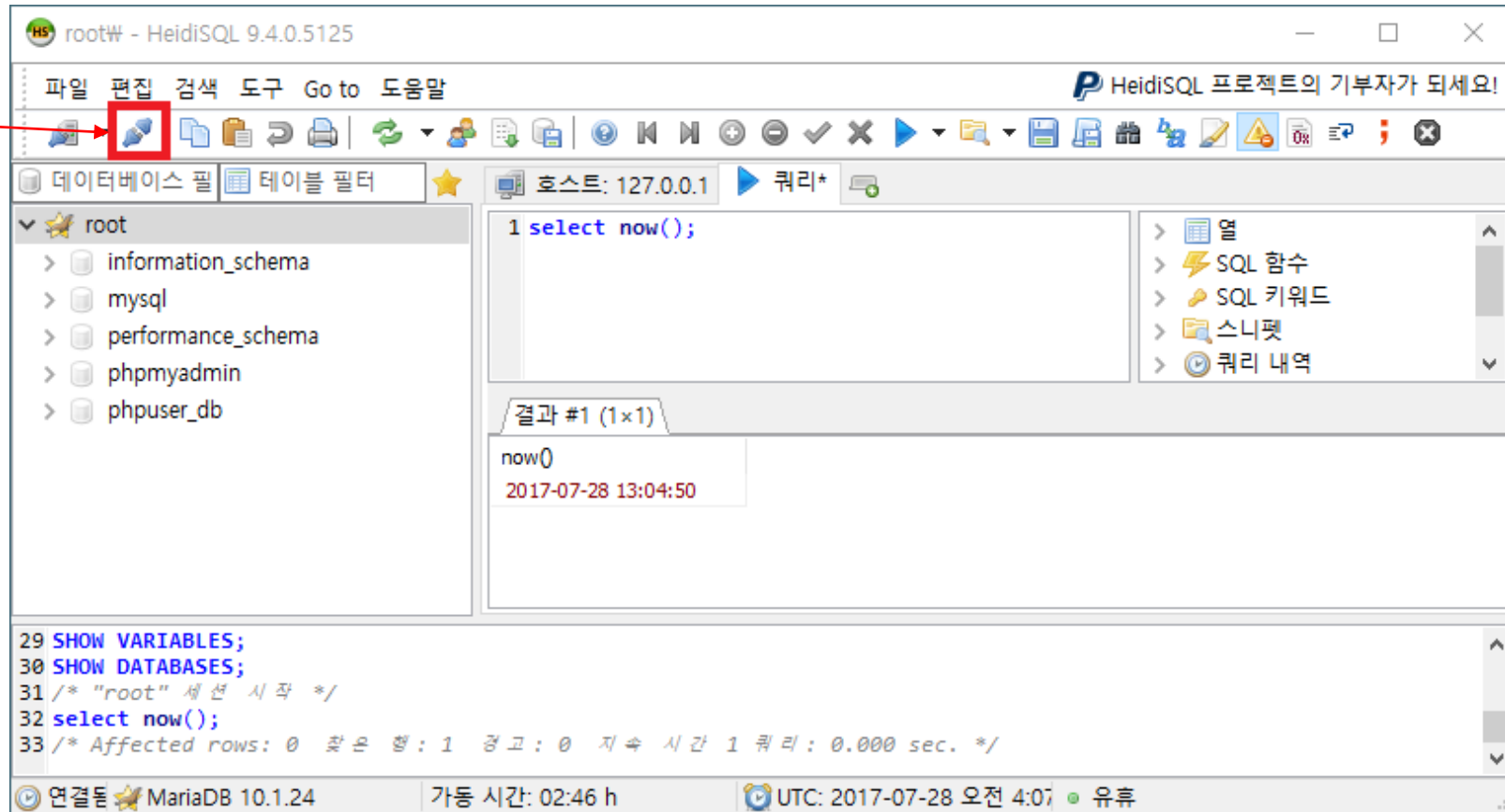
- 쿼리 창에 쿼리를 입력하고 실행 버튼(▶) 클릭



## 5.2 MariaDB 설치와 클라이언트 사용

### ■ 쿼리 실행 (계속)

DB 연결 해제



# [참고] 데이터베이스에 따라 다른 스키마 개념

- 오라클
  - 단일 사용자에게 소유된 테이블이나 객체를 의미
- MariaDB
  - 데이터베이스와 동일한 의미

## 5.3 사용자 계정 생성

- root 계정
  - DB 관리자 계정이므로 권한이 막강
  - 평소 시에는 일반 사용자 계정을 사용하는 것이 안전
- 사용자 계정 생성 쿼리
  - 예문의 gildong 대신 각자의 영문 이름 사용

```
create database DB명;  
grant all privileges on DB명.* to ID@localhost identified by '비밀번호' with grant option;  
flush privileges;
```

```
create database gildongdb;  
grant all privileges on gildongdb.* to gildong@localhost identified by '1111' with grant option;  
flush privileges;
```

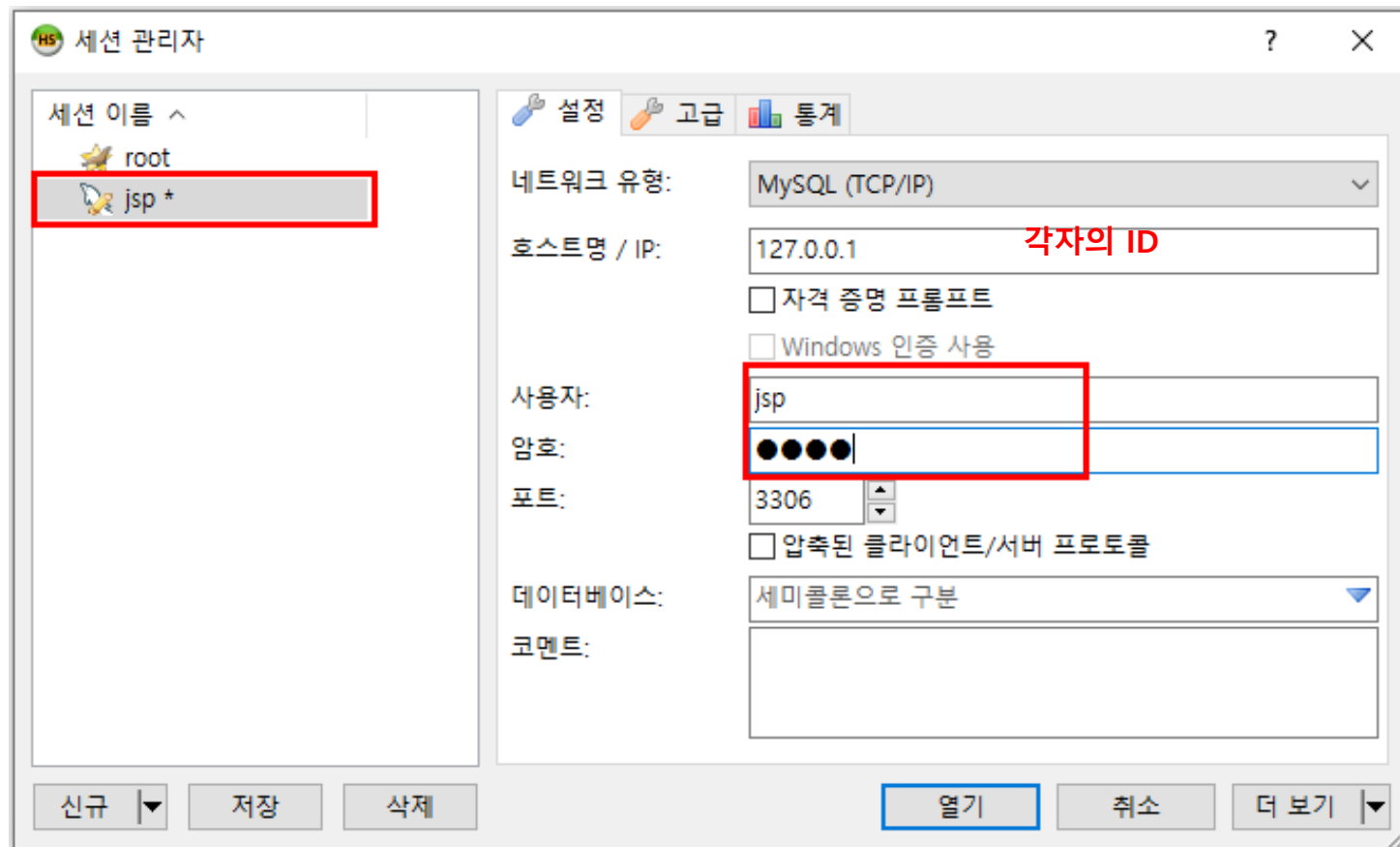
모든 호스트 의미

↓

```
grant all privileges on gildongdb.* to gildong@'%' identified by '1111' with grant option;
```

## 5.3 사용자 계정 생성

- 새로 만든 계정 정보 저장
  - 각자의 이름으로 세션을 만들고 각자의 자신의 이름으로 만든 사용자 ID와 암호로 열기

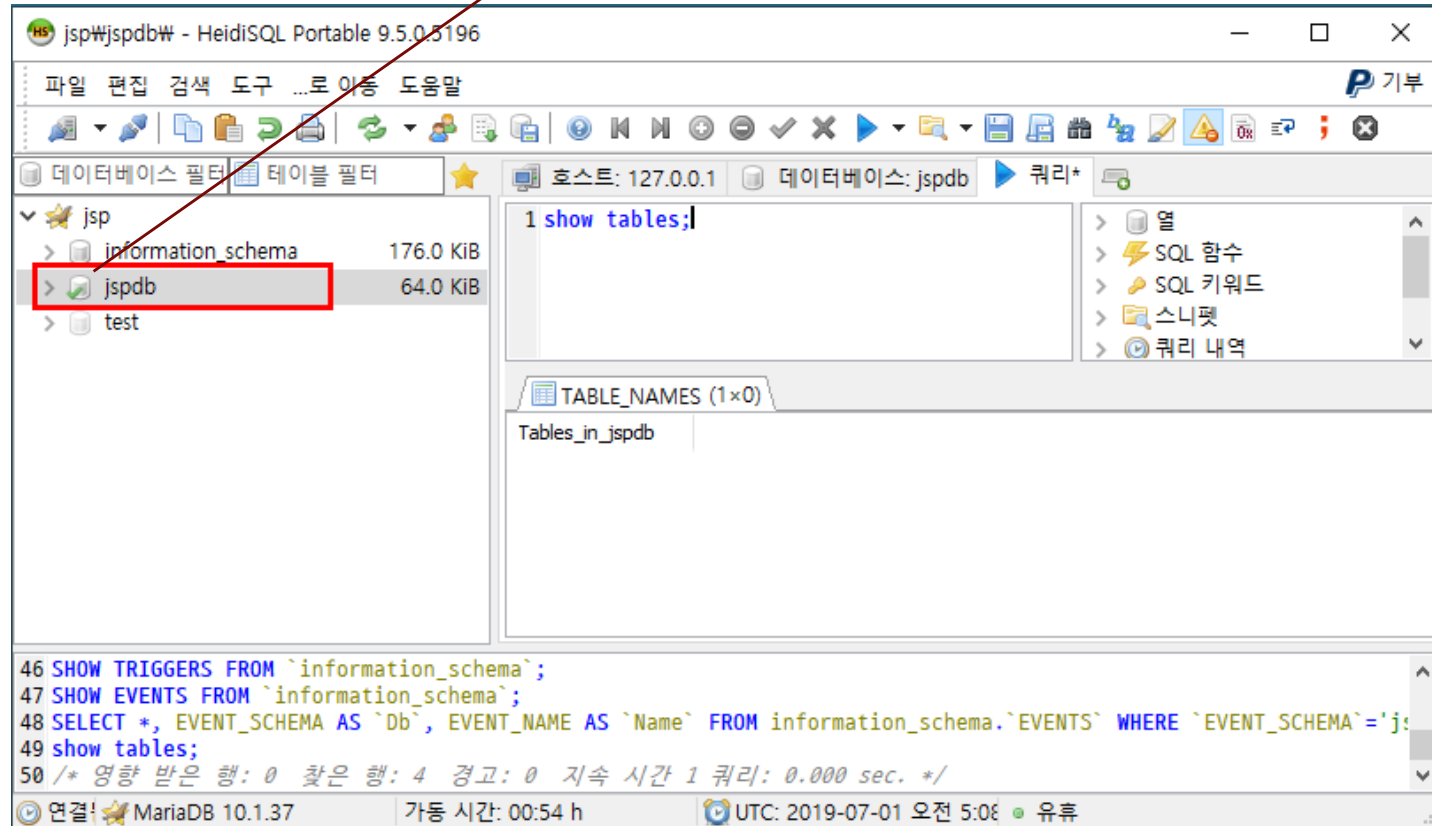




## 5.3 사용자 계정 생성

- 접속 및 사용
  - 쿼리 실행

선택된 DB는 체크 표시가 있음



# [참고]

- 사용자 관련 명령어(**root** 권한 필수 : root 세션에서 실행해야 함)

구분	명령어
사용자 확인	<code>select host, user from mysql.user;</code>
사용자 추가	<code>create user 'userid'@'%' identified by 'password';</code> <code>create user 'userid'@'localhost' identified by 'password';</code>
사용자 삭제	<code>drop user 'userid'@'%';</code>
권한 부여	<code>grant all privileges on useriddb.* to 'userid'@'%';</code>
권한 확인	<code>show grants for 'userid'@'%';</code>
권한 삭제	<code>revoke all on useriddb.* from 'userid'@'%';</code>
권한 갱신	<code>flush privileges;</code>

## 5.4 데이터베이스 관련 명령어

구분	명령어
데이터베이스 생성	사용법 : create database DB명; 예 : create database sample1;
데이터베이스 삭제	사용법 : drop database DB명; 예 : drop database sample1;
데이터베이스 목록 보기	show databases;
데이터베이스 선택(사용)	use DB명

- Mariadb 주석 작성의 3가지 방법  
/\* 내용 \*/  
-- 내용  
# 내용

# 5.4 테이블 관련 명령어

구분	명령어	
테이블 생성	<pre>create table 테이블명(     필드명1 자료형 [옵션]     .... );</pre>	
테이블 삭제	<pre>drop table <u>테이블명</u>;</pre>	
테이블 변경 (필드 추가, 필드 삭제, 필드 변경)	<pre>ALTER TABLE 테이블명 ADD (     jbColumn5 INT ); ALTER TABLE 테이블명 DROP jbColumn0 ;</pre>	<pre>ALTER TABLE 테이블명 CHANGE jbColumn0 jbColumn1 VARCHAR(10);</pre>
테이블 목록 보기	<pre>show tables;</pre>	
테이블 구조 보기	<pre>desc 테이블명</pre>	

## 5.5 테이블 관련 명령어

### ■ 테이블 생성

```
create table score (  
    num    int          primary key,  
    name   varchar(20),  
    kor    int,  
    eng    int,  
    math   int  
);
```

## 5.6 데이터 조작 명령어

구분	명령어
데이터 추가	<b>insert into</b> score (num, name, address, tel) <b>values</b> (1, '홍길동', 50, 60, 70);
데이터 조회	<b>select</b> name, kor <b>from</b> score <b>where</b> num=1;
데이터 변경	<b>update</b> score <b>set</b> kor=90 <b>where</b> num=1;
데이터 삭제	<b>delete from</b> score <b>where</b> num=1;

# [실습] DDL

- 다음 테이블을 각자의 DB에 작성하시오.

```
CREATE TABLE MEMBER(  
    ID INT PRIMARY key,  
    PWD VARCHAR(50),  
    NAME VARCHAR(50),  
    GENDER CHAR(2),  
    PHONE CHAR(13),  
    REGDATE DATE  
);  
COMMIT;
```

- 컬럼 추가

```
ALTER TABLE MEMBER  
    ADD(EMAIL VARCHAR(100))  
;
```

# [실습] DML – 각 SQL문을 작성하고 주석을 작성

## 3. 레코드 추가

```
INSERT INTO MEMBER(ID, PWD, NAME, GENDER, PHONE, EMAIL, REGDATE)  
VALUES(1, '11', '홍길동', '남성', '010-1234-5678', 'gildong@naver.com', curdate())
```

```
INSERT INTO MEMBER(ID, PWD, NAME, GENDER, PHONE, EMAIL, REGDATE)  
VALUES(2, '22', '성춘향', '여성', '010-4321-8765', 'chun@naver.com', curdate())
```

## 4. 레코드 검색

```
SELECT * FROM MEMBER;  
SELECT * FROM MEMBER WHERE ID = 2;
```

## 5. 레코드 수정

```
UPDATE MEMBER SET PWD = '99' WHERE ID=1;
```

## 6. 레코드 삭제

```
DELETE FROM MEMBER WHERE ID=1;
```

## 7. 테이블 삭제

```
DROP TABLE MEMBER;
```



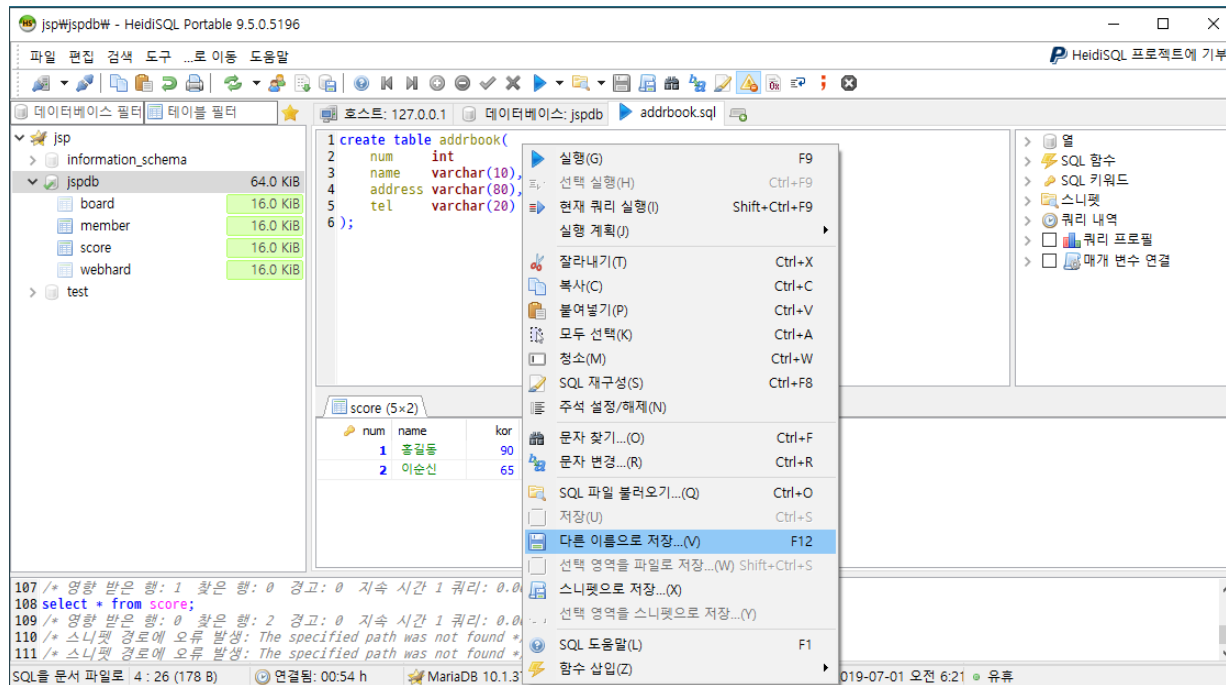
## 5.7 배치 파일(batch file)을 통한 명령 실행

■ 배치(batch) : 여러 명령문들을 한 파일에 저장한 파일

1. 쿼리를 입력한 후 저장해 둠

2. “파일” 메뉴에서 “SQL 파일 실행” → 저장해 둔 파일 선택

3. sql 파일에 저장된 쿼리가 실행됨



## 5.7 배치 파일(batch file)을 통한 명령 실행

- 배치 파일은 데이터베이스 백업에도 사용됨

ch05-backup.sql - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
CREATE TABLE IF NOT EXISTS `member` (  
  `ID` int(11) NOT NULL,  
  `PWD` varchar(50) DEFAULT NULL,  
  `NAME` varchar(50) DEFAULT NULL,  
  `GENDER` char(2) DEFAULT NULL,  
  `PHONE` char(13) DEFAULT NULL,  
  `REGDATE` date DEFAULT NULL,  
  `EMAIL` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-- 테이블 데이터 jinsookdb.member:~1 rows (대략적) 내보내기  
DELETE FROM `member`;  
/*140000 ALTER TABLE `member` DISABLE KEYS */;  
INSERT INTO `member` (`ID`, `PWD`, `NAME`, `GENDER`, `PHONE`, `REGDATE`, `EMAIL`) VALUES  
  (1, '11', '홍길동', '남성', '010-1234-5678', '2022-03-12', 'gildong@naver.com');  
/*140000 ALTER TABLE `member` ENABLE KEYS */;
```

백업 파일도 .sql이므로 [파일]-[SQL 파일실행] 하면 복원됨

# 답변할 수 있겠지요?

1. 테이블, 레코드, 필드의 의미가 무엇인지 설명하시오(그러서 표시).
2. MariaDB 접속을 위해 사전에 알고 있어야 하는 중요 정보는 무엇인가?
3. 새로운 계정을 생성하는 쿼리를 적어보시오. 이때 사용자 아이디는 kim, 비밀번호는 abcd, 이 계정을 위한 전용 데이터베이스는 kimdb로 한다.
4. "sample"이라는 이름의 데이터베이스를 만드는 쿼리와 삭제하는 쿼리를 적어 보시오.
5. "addr"이라는 이름의 테이블이 있다고 할 때, 이 테이블의 구조를 조회하는 쿼리와, 테이블을 삭제하는 쿼리를 적어 보시오.
6. 레코드 추가, 조회, 수정, 삭제 쿼리의 사용 형식을 정리해 보시오.
7. HeidiSQL에서 "데이터베이스를 SQL로 내보내기"를 하면, 어떤 형태의 파일로 백업되는가?