

Chapter 3

MySQL 기초

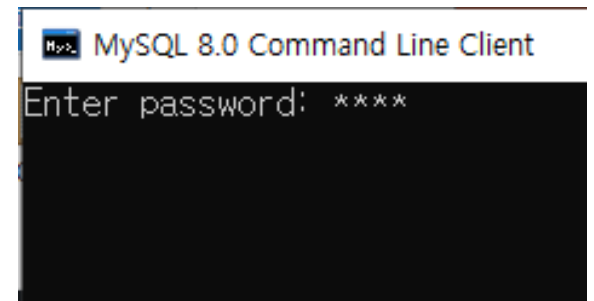
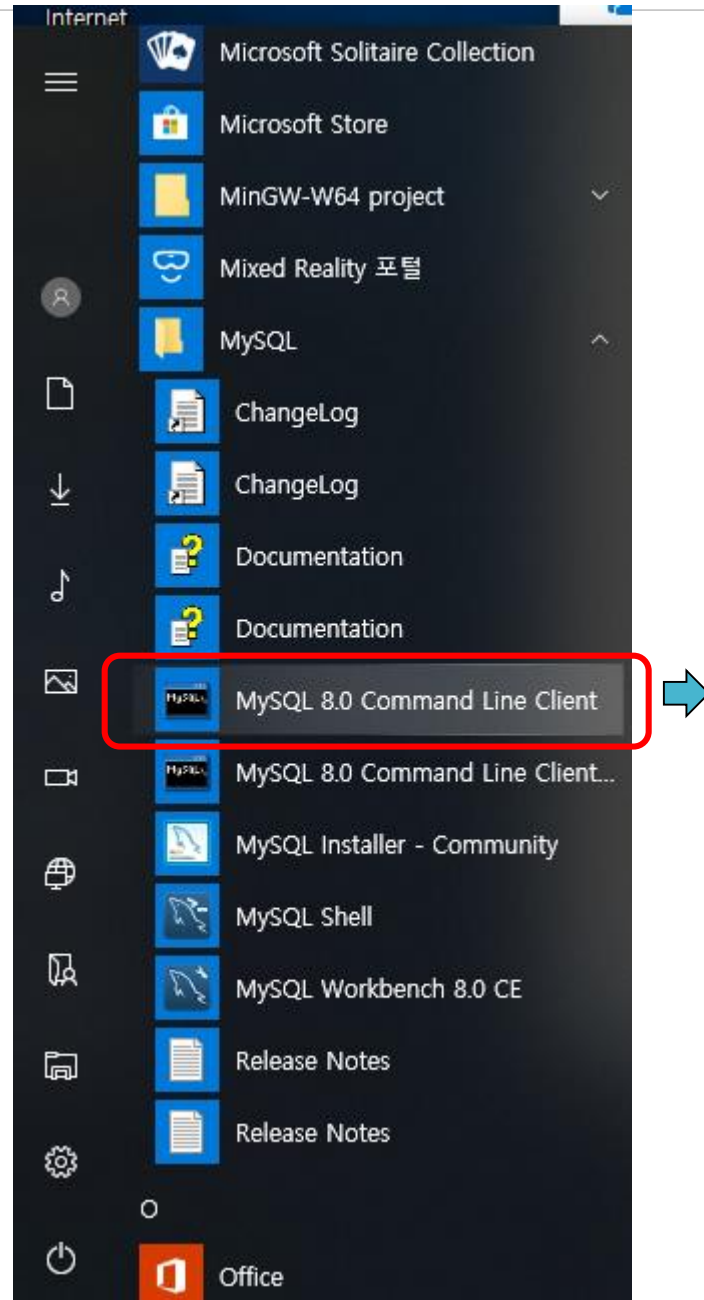
오 세 종

1. MySQL 특징

- SQL에 기반을 둔 관계형 DBMS 중 하나
- Oracle, IBM, Infomix 등의 데이터베이스는 고가이지만, MySQL 데이터베이스는 무료
- Oracle 에 인수된 이후에 Maria DB 등장
- 리눅스, 유닉스, 윈도우 등 거의 모든 운영체제에서 사용가능
- 처리 속도가 상당히 빠르고 대용량에 데이터도 처리 용이
- 설치 방법이 쉽고 초보자도 익히기 쉬움
- 보안성이 우수

2. MySQL – 콘솔에서 사용하기

- 콘솔 열기



2. MySQL – 콘솔에서 사용하기

마우스 오른쪽 버튼 클릭
(글씨크기, 바탕색등 조절)



```
MySQL 8.0 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 140
Server version: 8.0.15 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

2. MySQL – 콘솔에서 사용하기

- 설치된 데이터베이스 보기

```
mysql> show databases ;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.00 sec)

mysql>
```

2. MySQL – 콘솔에서 사용하기

- world 데이터베이스 내용보기

```
mysql> use world ;  
Database changed  
mysql> show tables ;  
+-----+  
| Tables_in_world |  
+-----+  
| city             |  
| country          |  
| countrylanguage |  
+-----+  
3 rows in set (0.01 sec)  
  
mysql>
```

2. MySQL – 콘솔에서 사용하기

- city 테이블 구조 보기

```
mysql> desc city ;
```

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Name	char(35)	NO			
CountryCode	char(3)	NO	MUL		
District	char(20)	NO			
Population	int(11)	NO		0	

5 rows in set (0.01 sec)

```
mysql>
```

나머지 2개의 테이블 구조도 확인해 보자

2. MySQL – 콘솔에서 사용하기

- city 테이블의 내용 보기 (앞쪽 10개만)

```
mysql> select * from city limit 10;
```

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238

```
10 rows in set (0.00 sec)
```

```
mysql>
```


2. MySQL – 콘솔에서 사용하기

- Seoul 에 대한 정보를 찾아보기

```
mysql> select * from city where Name='Seoul';
```

ID	Name	CountryCode	District	Population
2331	Seoul	KOR	Seoul	9981619

1 row in set (0.00 sec)

```
mysql>
```

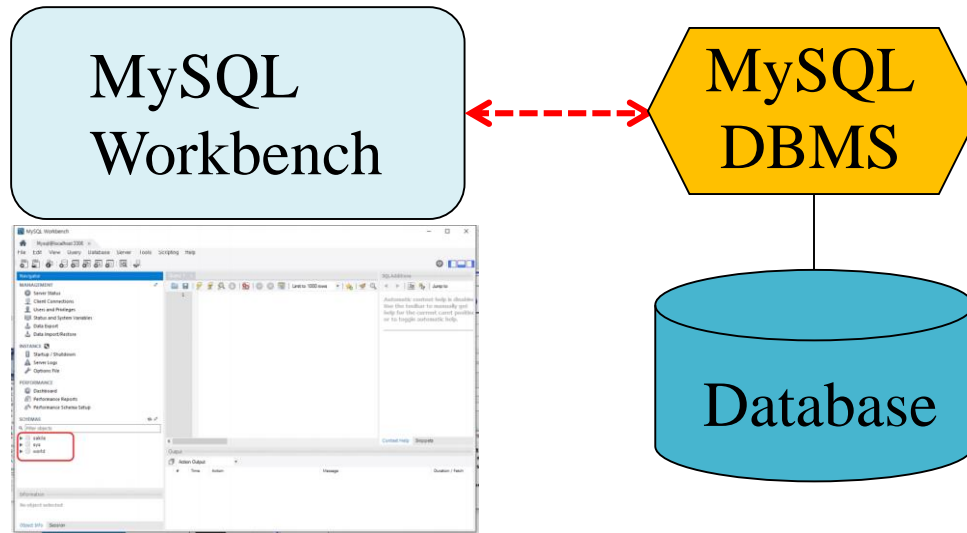
2. MySQL – 콘솔에서 사용하기

- mysql 종료하기

```
mysql> quit
```

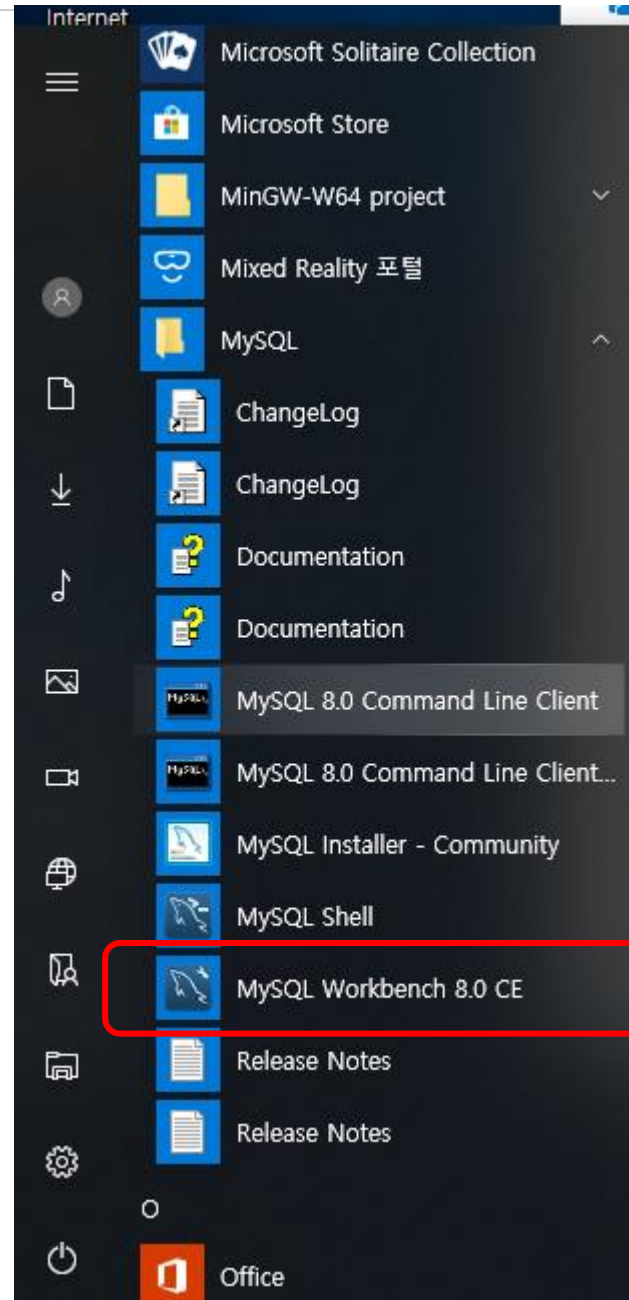
3. MySQL Workbench

- mysql 작업을 편하게 할 수 있도록 도와주는 보조 SW
- Console 에서 할 수 있는 모든 작업을 Window 환경에서 진행할 수 있다.



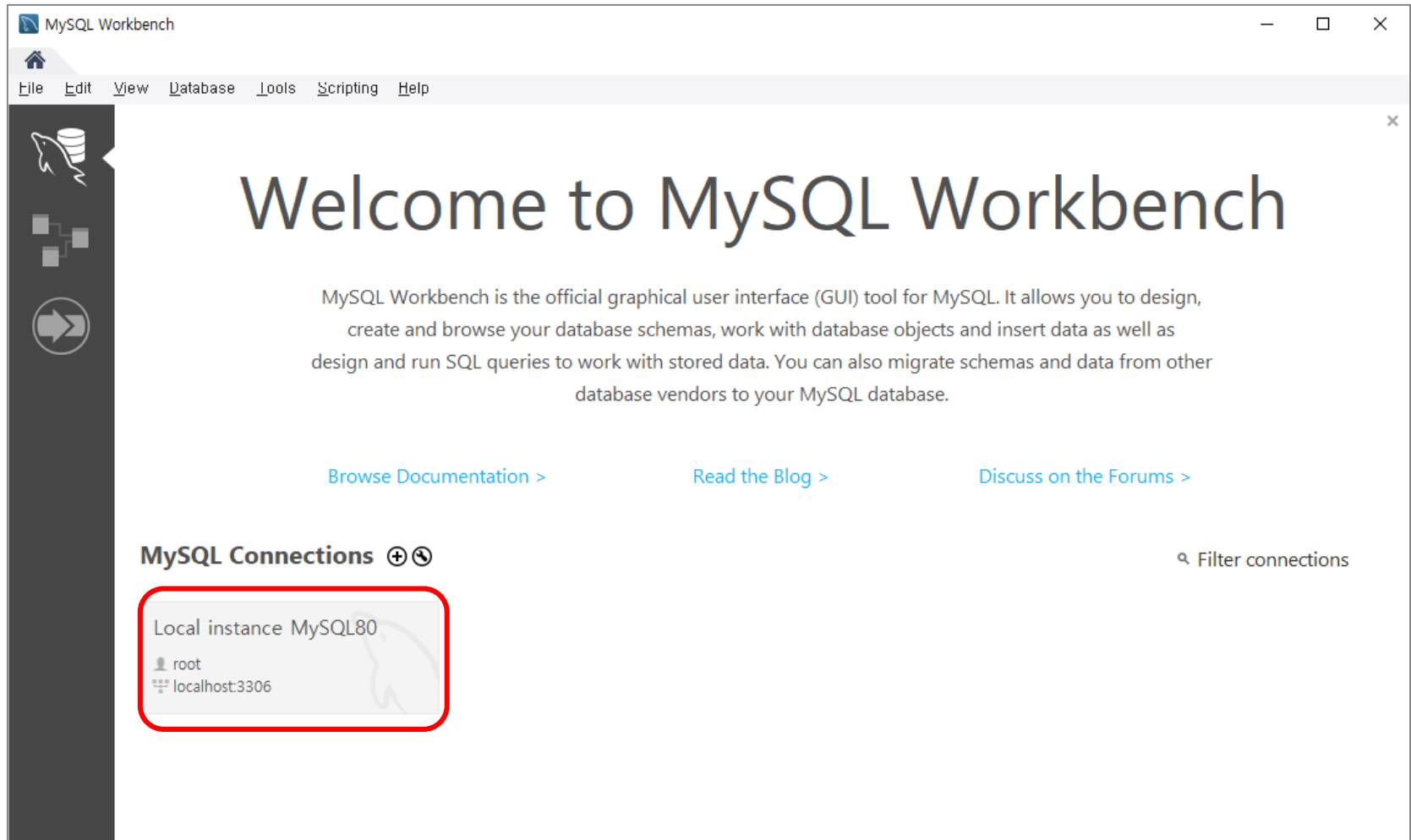
3. MySQL Workbench

- 시작



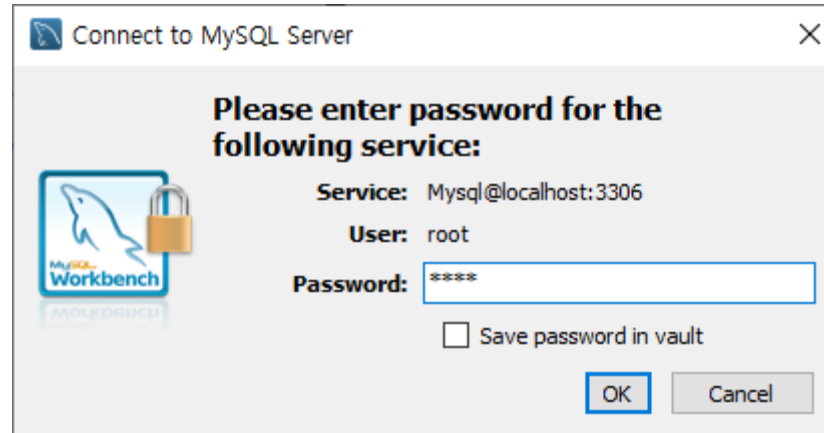
3. MySQL Workbench

- Open connection

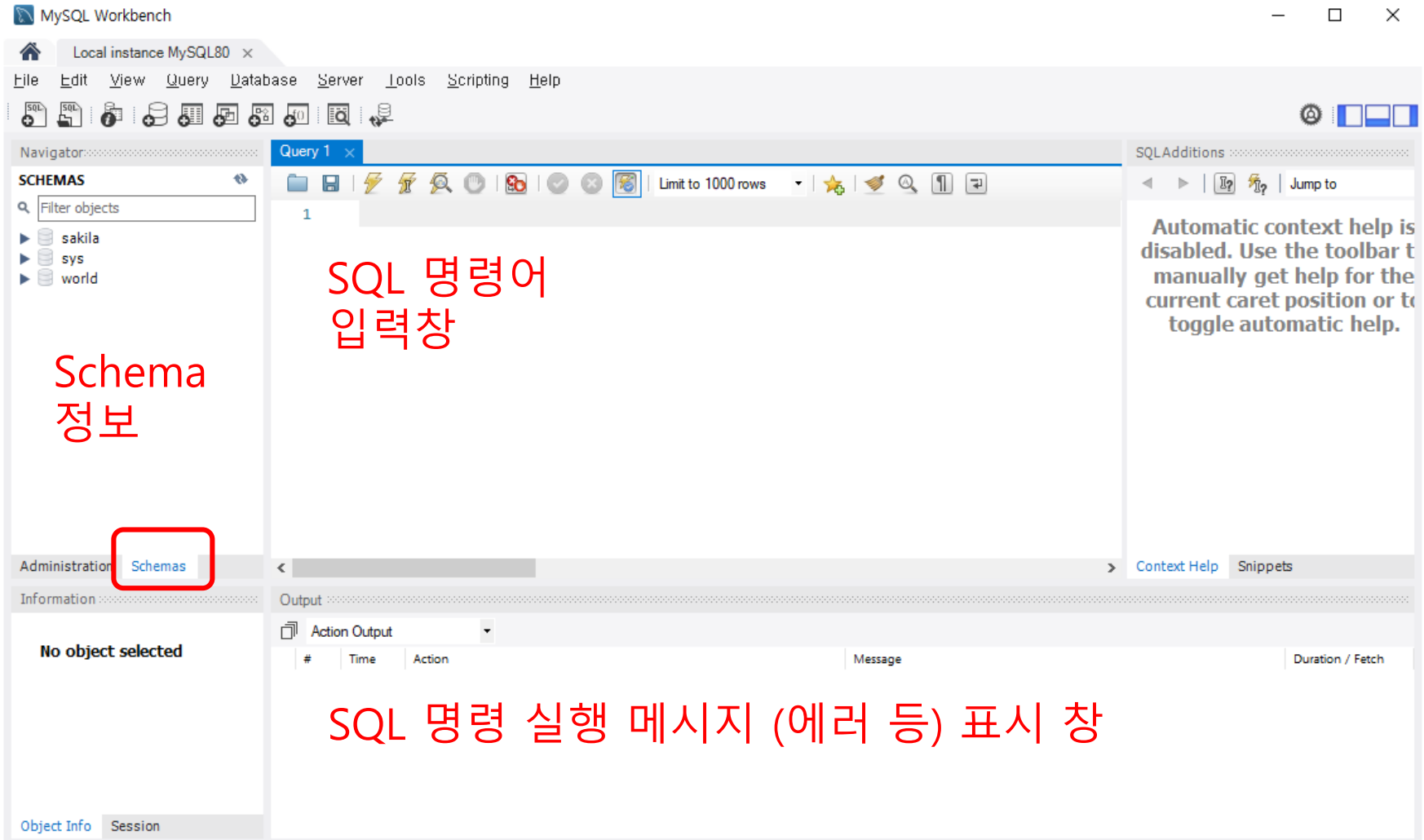


3. MySQL Workbench

- 로그인

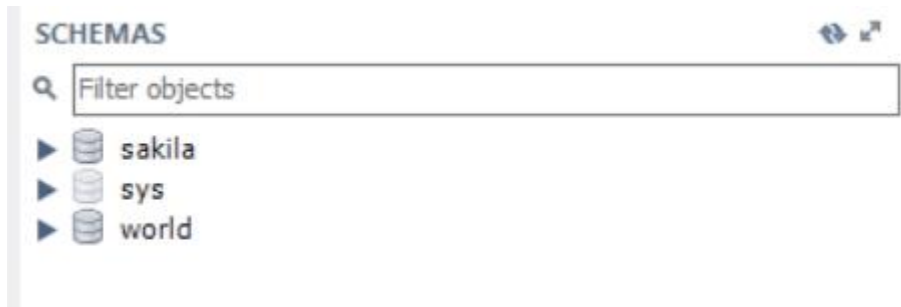


3. MySQL Workbench



3. MySQL Workbench

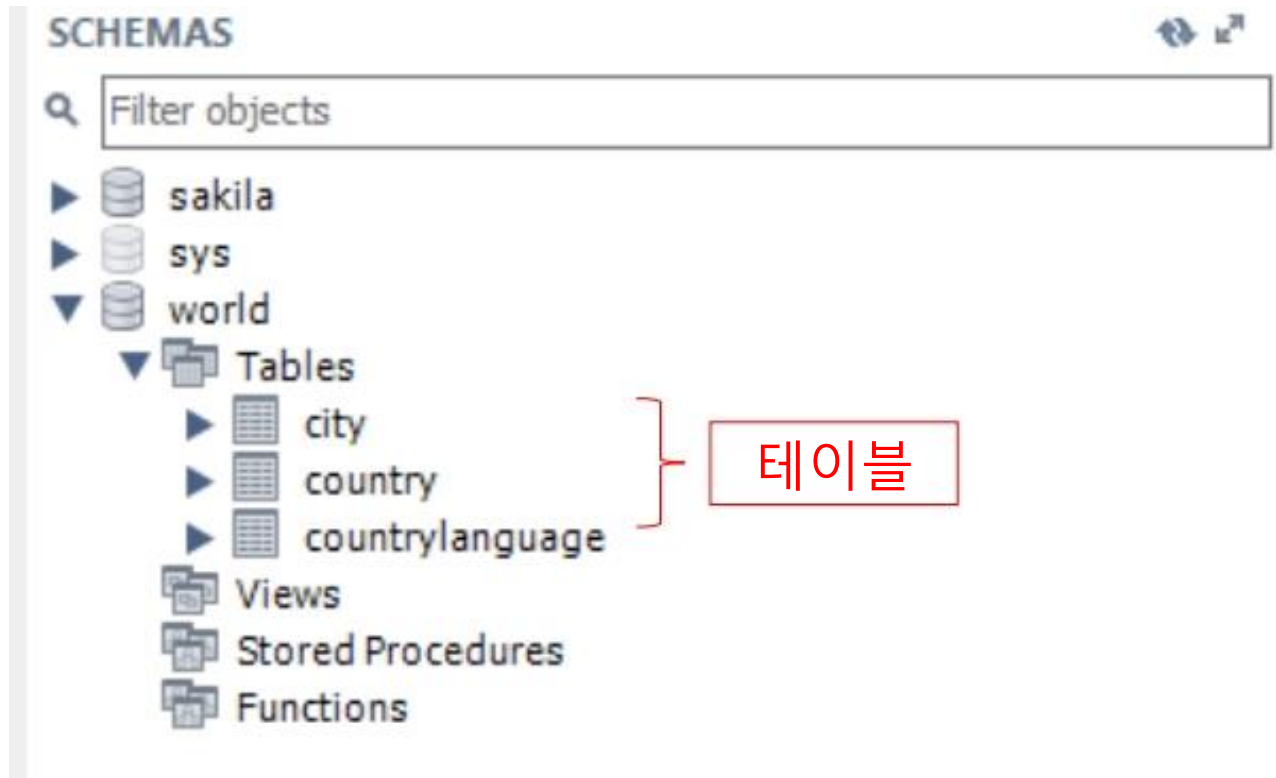
- 초기에 설치되어 있는 데이터베이스



- sakila : 영화 관련 데이터베이스
- sys : 시스템 데이터베이스 (수정하지 않는다)
- world : 도시,국가,언어 데이터베이스

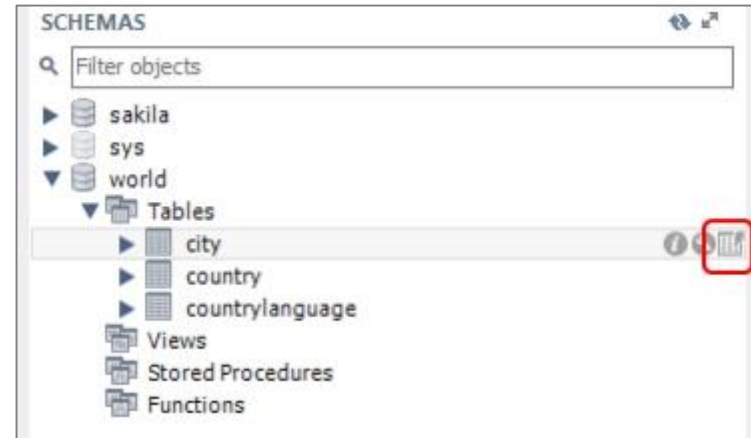
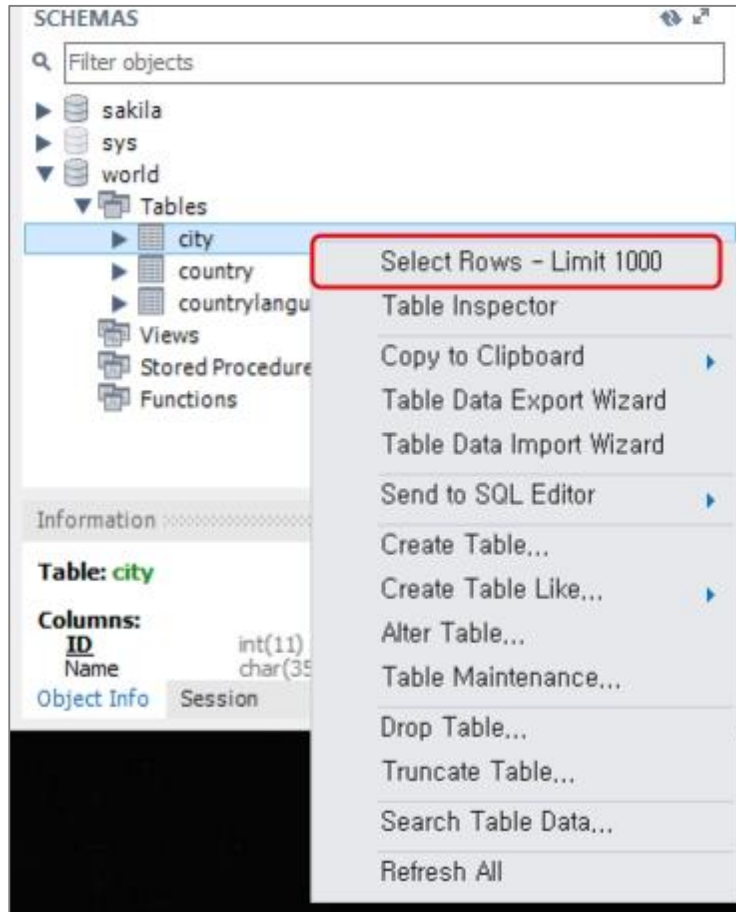
3. MySQL Workbench

- World 데이터베이스 살펴보기



3. MySQL Workbench

- World 데이터베이스 살펴보기
 - city 테이블 내용 보기 (두가지 방법)



3. MySQL Workbench

The screenshot displays the MySQL Workbench interface with the following components:

- Navigator:** Shows the database structure with 'world' selected, containing tables 'city', 'country', and 'countrylanguage'.
- Query Editor:** Contains the query `SELECT * FROM world.city;`. A toolbar above the editor includes icons for query execution and a 'Limit to 1000 rows' dropdown.
- Result Grid:** Displays the query results in a table format with columns: ID, Name, CountryCode, District, and Population. The first five rows are visible.
- Table Information:** A sidebar on the left shows the structure of the 'city' table, including columns: ID (int(11), AI PK), Name (char(3)), CountryCode (char(3)), District (char(2)), and Population (int(11)).
- Action Output:** A table at the bottom showing the execution of the query, including the time taken and the number of rows returned.
- SQLAdditions:** A sidebar on the right with a 'Jump to' dropdown and a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200

#	Time	Action	Message	Duration / Fetch
1	13:11:24	SELECT * FROM world.city	1000 row(s) returned	0.000 sec / 0.000 sec
2	13:12:23	SELECT * FROM world.city LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.000 sec

3. MySQL Workbench

The screenshot shows the MySQL Workbench interface with the following components and annotations:

- (1)** Points to the **Navigator** sidebar on the left, which contains sections for **MANAGEMENT** (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), **INSTANCE** (Startup / Shutdown, Server Logs, Options File), **PERFORMANCE** (Dashboard, Performance Reports, Performance Schema Setup), and **SCHEMAS** (Filter objects, sakila, sys, world, Tables, city, country, country/language, Views, Stored Procedures, Functions).
- (2)** Points to the **Output** tab at the bottom, showing the **Action Output** table with columns: Time, Action, Message, and Duration / Fetch.
- (3)** Points to the **SQLAdditions** sidebar on the right, which contains a text area for automatic context help and a **Jump to** button.

The central **Query Editor** shows a query: `SELECT * FROM world.city;` The **Result Grid** displays the following data:

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238
11	Groningen	NLD	Groningen	172701
12	Breda	NLD	Noord-Brabant	160398
13	Apeldoorn	NLD	Gelderland	153491

The **Output** tab shows the following action output:

Time	Action	Message	Duration / Fetch
3 15:07:10	SELECT * FROM world.country/language LIMIT 0, 1...	984 row(s) returned	0.000 sec / 0.000 sec
4 15:22:07	SELECT * FROM world.city LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec

3. MySQL Workbench

- 네덜란드의 도시들만 보기

sql 실행 버튼

The screenshot shows the MySQL Workbench interface. At the top, a tab labeled 'city' is active. Below it, a SQL query is entered in the editor:

```
1 • SELECT * FROM world.city
2 Where CountryCode = 'NLD' ;
```

A red box highlights the lightning bolt icon in the toolbar, which is the SQL execution button. Below the query editor, the 'Result Grid' is displayed, showing a table of cities in the Netherlands:

	ID	Name	CountryCode	District	Population
▶	5	Amsterdam	NLD	Noord-Holland	731200
	6	Rotterdam	NLD	Zuid-Holland	593321
	7	Haag	NLD	Zuid-Holland	440900
	8	Utrecht	NLD	Utrecht	234323
	9	Eindhoven	NLD	Noord-Brabant	201843

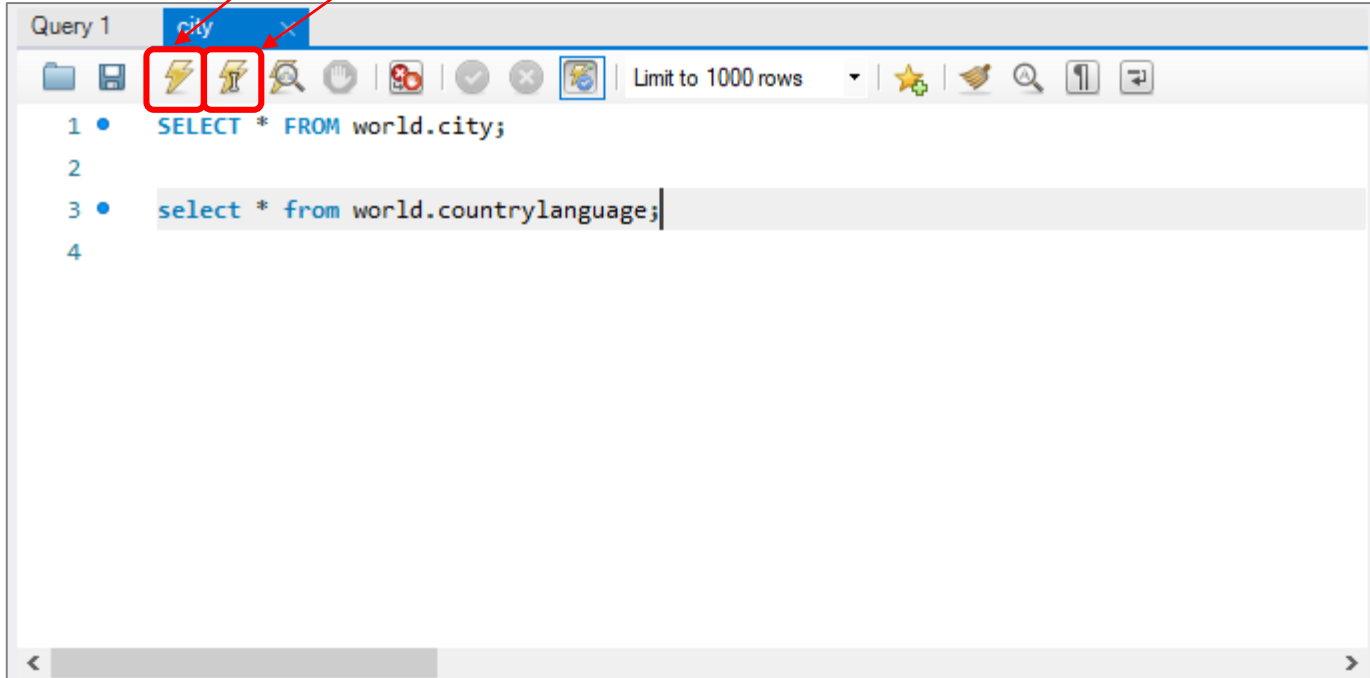
Below the result grid, the 'Output' tab is selected, showing the 'Action Output' log:

#	Time	Action	Message
✓ 1	13:11:24	SELECT * FROM world.city LIMIT 0, 1000	1000 row(s) returned
✓ 2	13:12:23	SELECT * FROM world.city LIMIT 0, 1000	1000 row(s) returned
✓ 3	13:18:54	SELECT * FROM world.city Where CountryCode = 'NLD' LIMIT 0, 10...	28 row(s) returned

3. MySQL Workbench

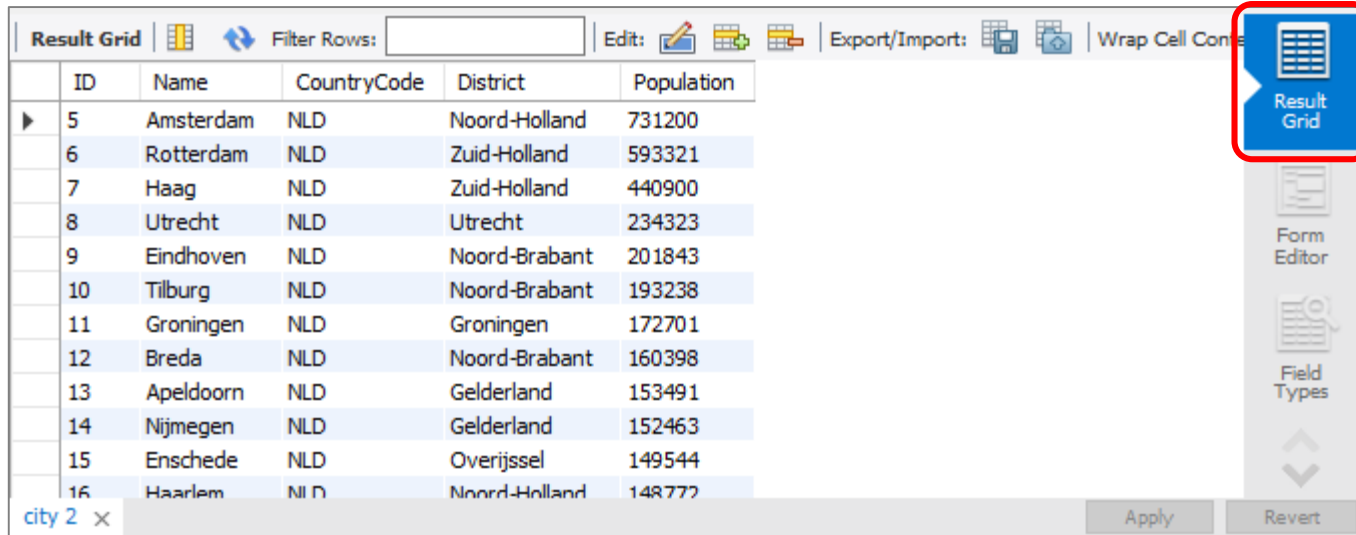
- SQL 명령어가 여러 개인 경우 선택하여 실행하기

명령창에 있는 모든 명령어 실행
현재 커서 위치의 명령어 실행



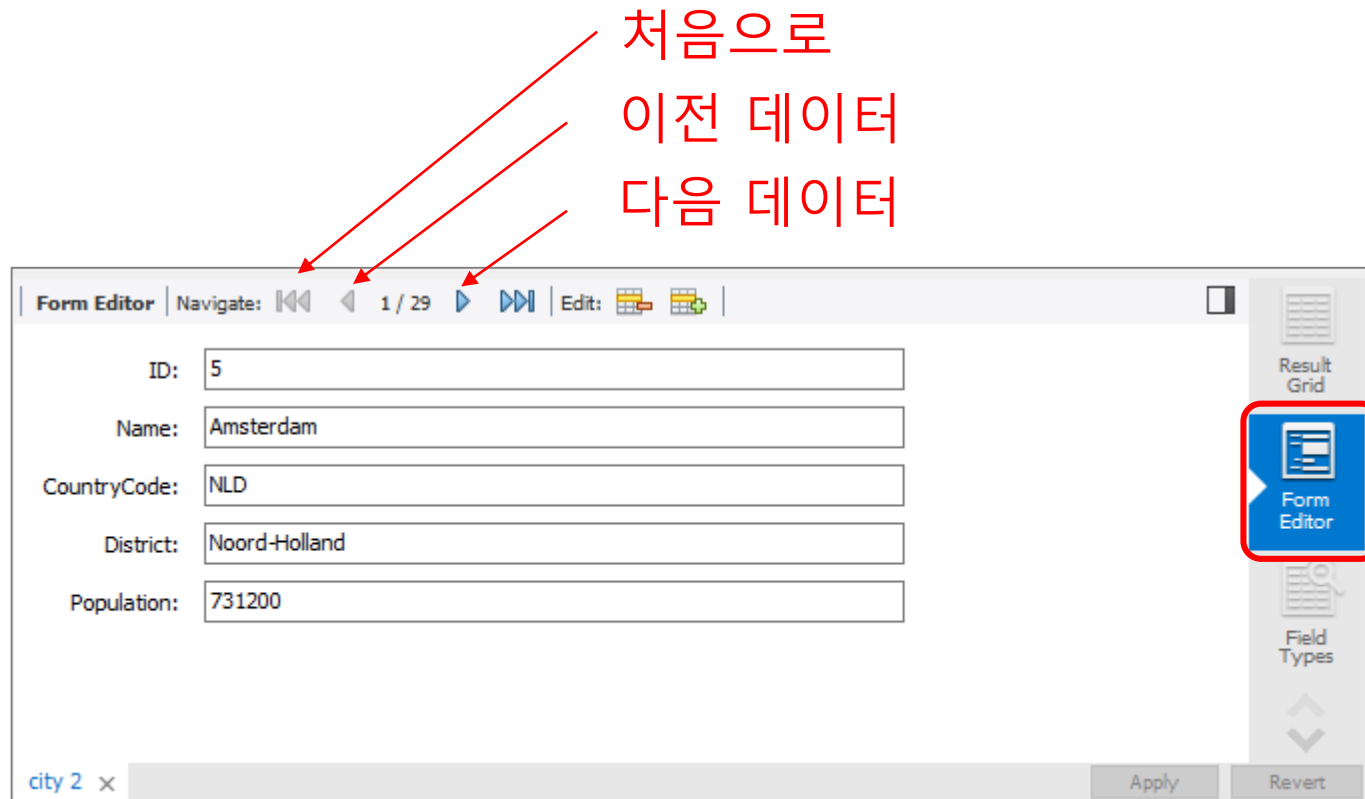
3. MySQL Workbench

- 질의 결과를 격자 형태로 보기 (default)



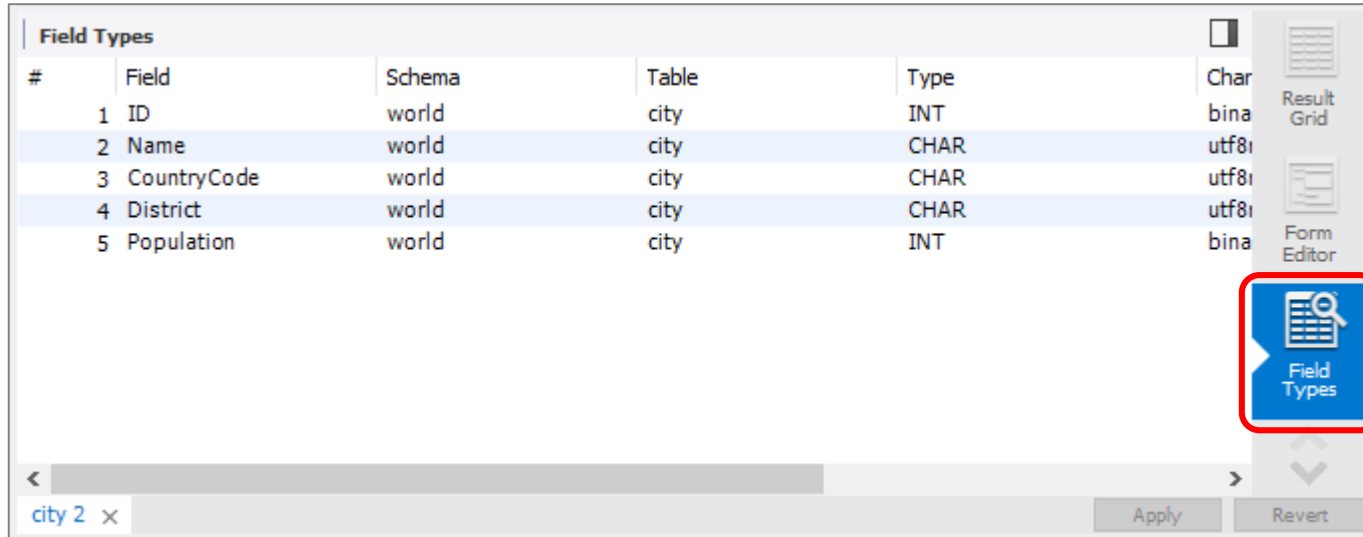
3. MySQL Workbench

- 질의 결과를 폼 형태로 보기



3. MySQL Workbench

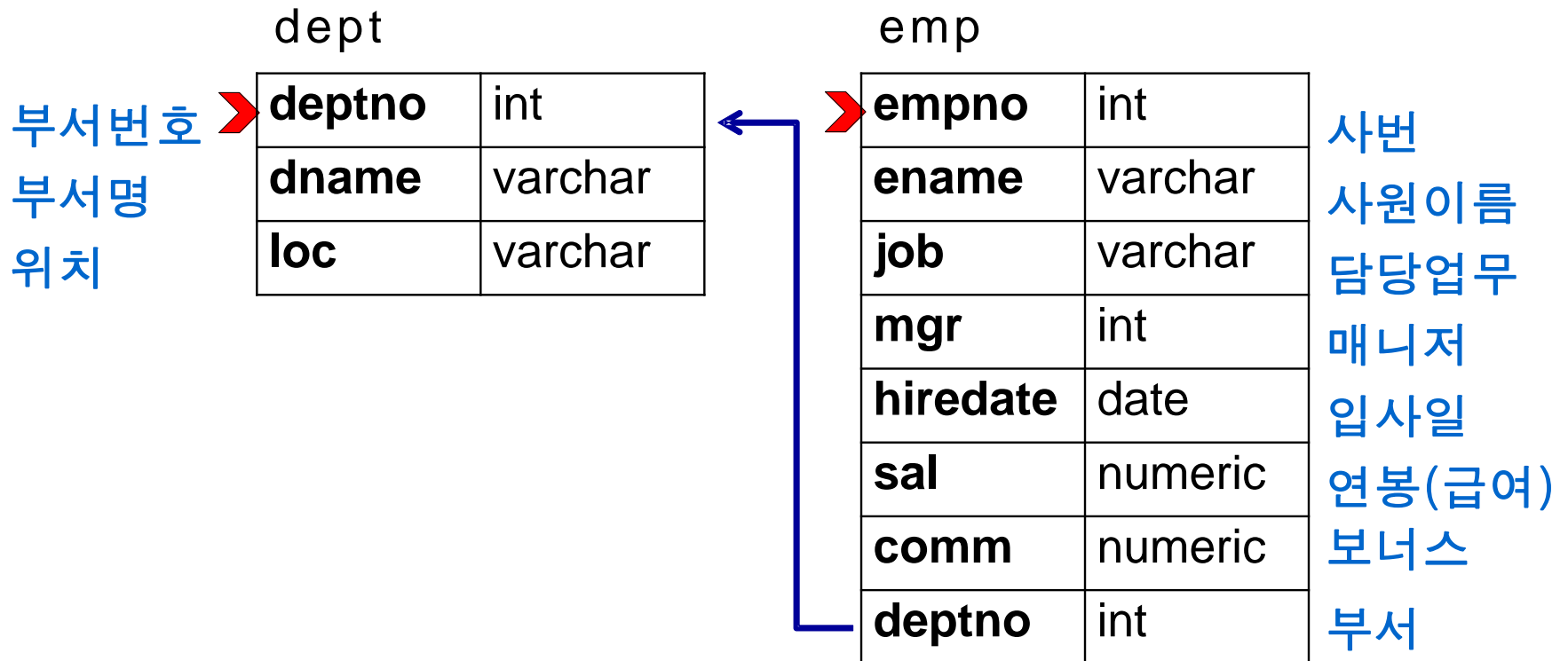
- 테이블 구조 정보 보기



Field = Column

4. 실습용 데이터베이스 생성하기

- 부서-사원 데이터베이스 (mydb)



4. 실습용 데이터베이스 생성하기

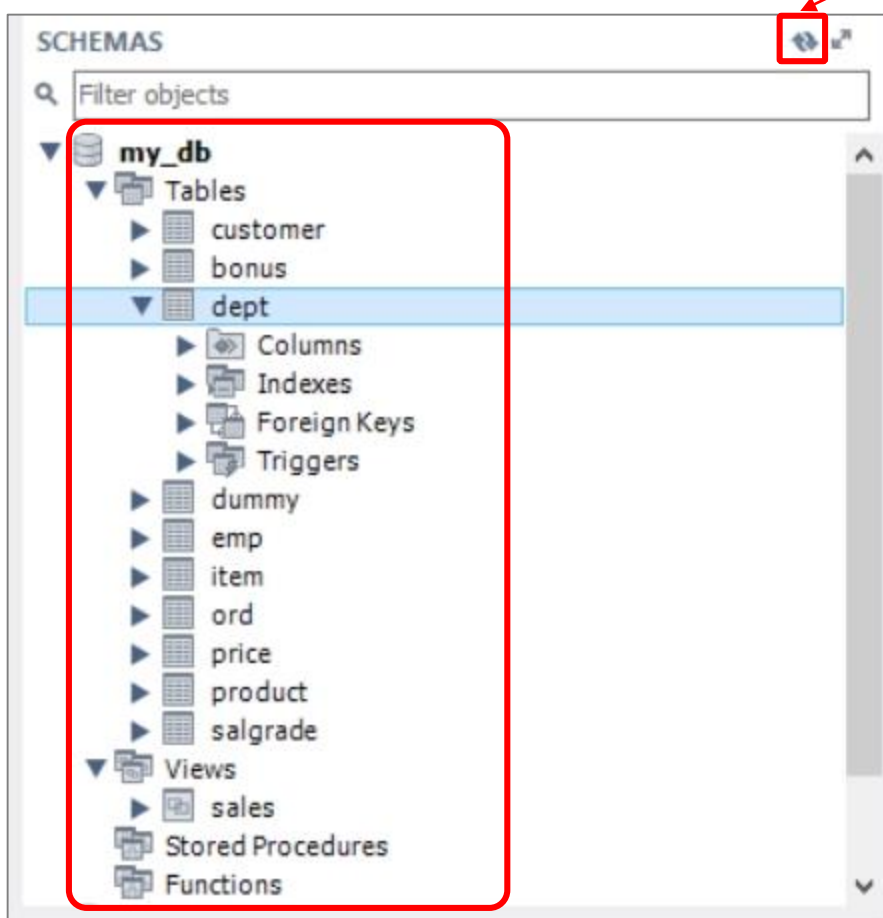
- create_table.sql 의 내용을 SQL 명령어 실행창에 붙여 넣기
한 후 실행

```
1 CREATE DATABASE my_db;
2 USE my_db;
3
4 CREATE TABLE DEPT (
5     DEPTNO          DECIMAL(2) NOT NULL,
6     DNAME           VARCHAR(14),
7     LOC             VARCHAR(13),
8     CONSTRAINT DEPT_PRIMARY_KEY PRIMARY KEY (DEPTNO));
9
10 INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW YORK');
11 INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');
12 INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');
13 INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');
14
15 CREATE TABLE EMP (
16     EMPNO           DECIMAL(4) NOT NULL,
17     ENAME           VARCHAR(10),
18     JOB             VARCHAR(9),
19     MGR             DECIMAL(4),
20     HIREDATE        DATE,
21     SAL             DECIMAL(7,2),
22     COMM            DECIMAL(7,2),
23     DEPTNO          DECIMAL(2) NOT NULL,
24     CONSTRAINT EMP_SELF_KEY FOREIGN KEY (MGR) REFERENCES EMP(EMPNO),
25     CONSTRAINT EMP_FOREIGN_KEY FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO),
26     CONSTRAINT EMP_PRIMARY_KEY PRIMARY KEY (EMPNO));
27
28 INSERT INTO EMP VALUES (7839, 'KING', 'PRESIDENT', NULL, '81-11-17', 5000, NULL, 10);
29 INSERT INTO EMP VALUES (7698, 'BLAKE', 'MANAGER', 7839, '81-5-1', 2850, NULL, 30);
```

4. 실습용 데이터베이스 생성하기

- my_db 가 정상적으로 생성되었는지 확인

새로고침



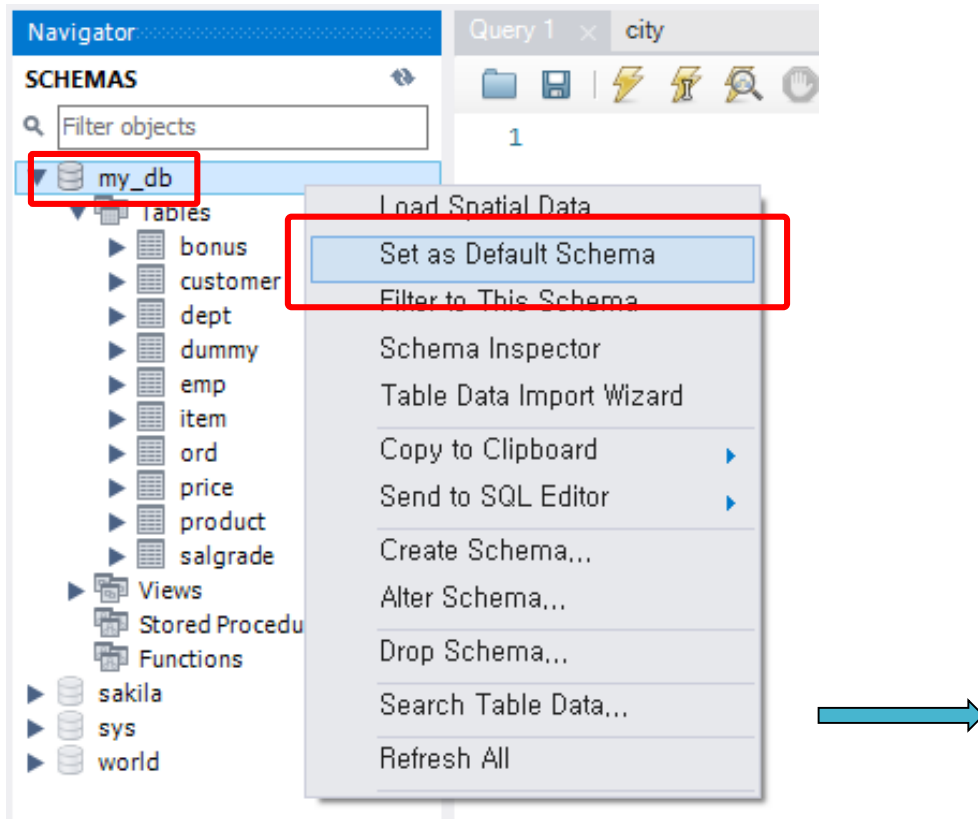
The Schemas window displays the database structure for my_db. The 'Tables' folder is expanded, showing a list of tables including customer, bonus, dept, dummy, emp, item, ord, price, product, and salgrade. The 'dept' table is selected. A red box highlights the entire database structure, and a red arrow points to the refresh button in the top right corner of the Schemas window.

Result Grid

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
NULL	NULL	NULL

4. 실습용 데이터베이스 생성하기

- 작업 데이터베이스 설정하기

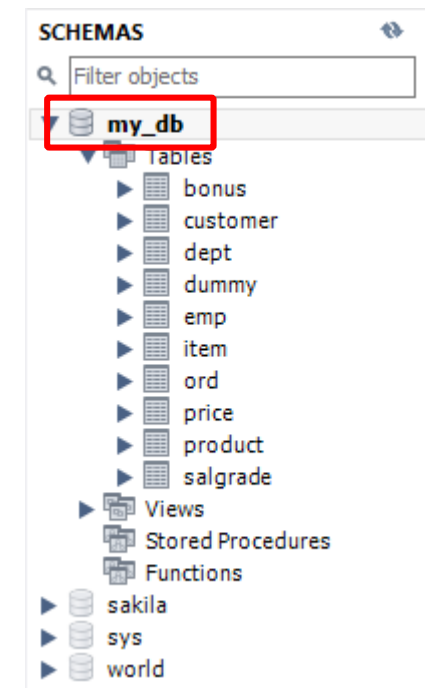


설정 전

```
select * from my_db.emp ;
```

설정 후

```
select * from emp ;
```



Note

- SQL에서는 명령어, 테이블이름, 컬럼 이름 등에 대해 대소문자를 구별하지 않는다

```
SELECT * FROM world.city;
```

```
select * from world.COUNTRYLANGUAGE;
```

- 한 명령어를 여러줄에 걸쳐 작성해도 된다

```
1 • select *  
2   from emp  
3   where empno = 7369 ;  
4
```

- 명령어와 명령어는 ; 으로 구분한다