

Mysql 1

Index

1. MySQL Workbench을 이용한 MySQL 서버 접속

2. 샘플 데이터 추가

3. 데이터 베이스 모델링

4. SQL문의 종류 : DML DDL DCL

5. SELECT FROM

6. MySQL의 주석

7. OPERATORS

8. WHERE

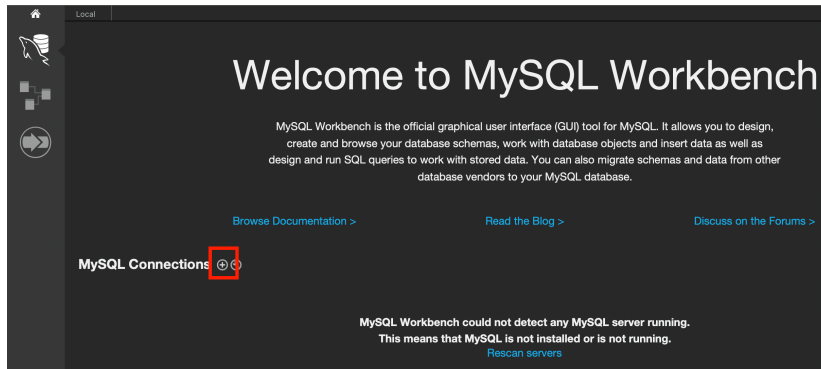
9. ORDER BY

10. LIMIT

1. MySQL Workbench을 이용한 MySQL 서버 접속

Workbench를 이용해 Local PC에 있는 MySQL 서버에 접속합니다.

Workbench 를 실행한후 + 버튼을 클릭해 MySQL 서버에 접속하는 세션 정보를 입력합니다.



1 ~ 6번 순서로 본인 설정에 맞게 값을 입력합니다.

1 : 세션(접속)정보의 이름

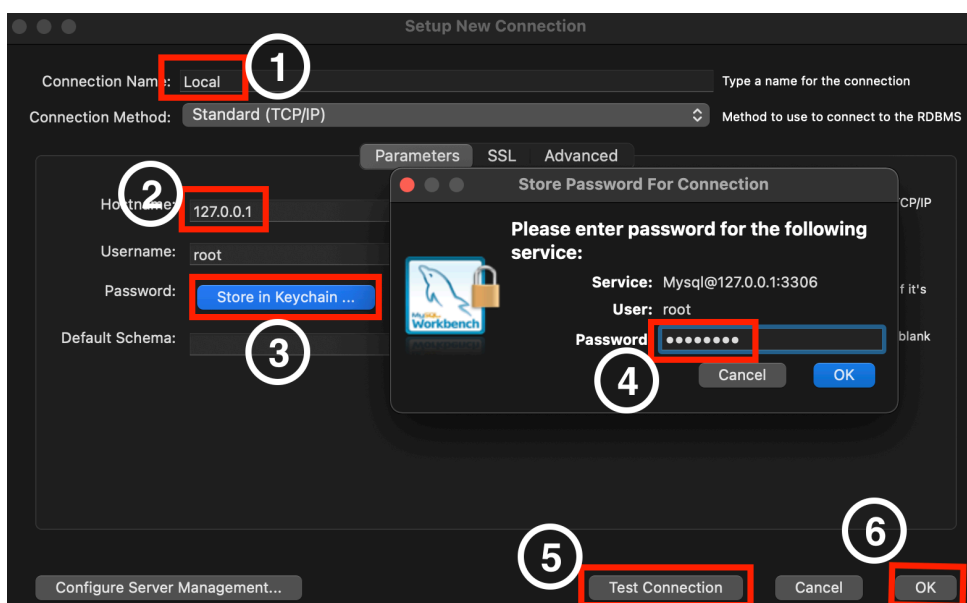
2 : 접속하고자 하는 MySQL의 IP 주소 (Local PC는 127.0.0.1 입력)

3. 패스워드 입력창 띄우기

4. 패스워드 입력

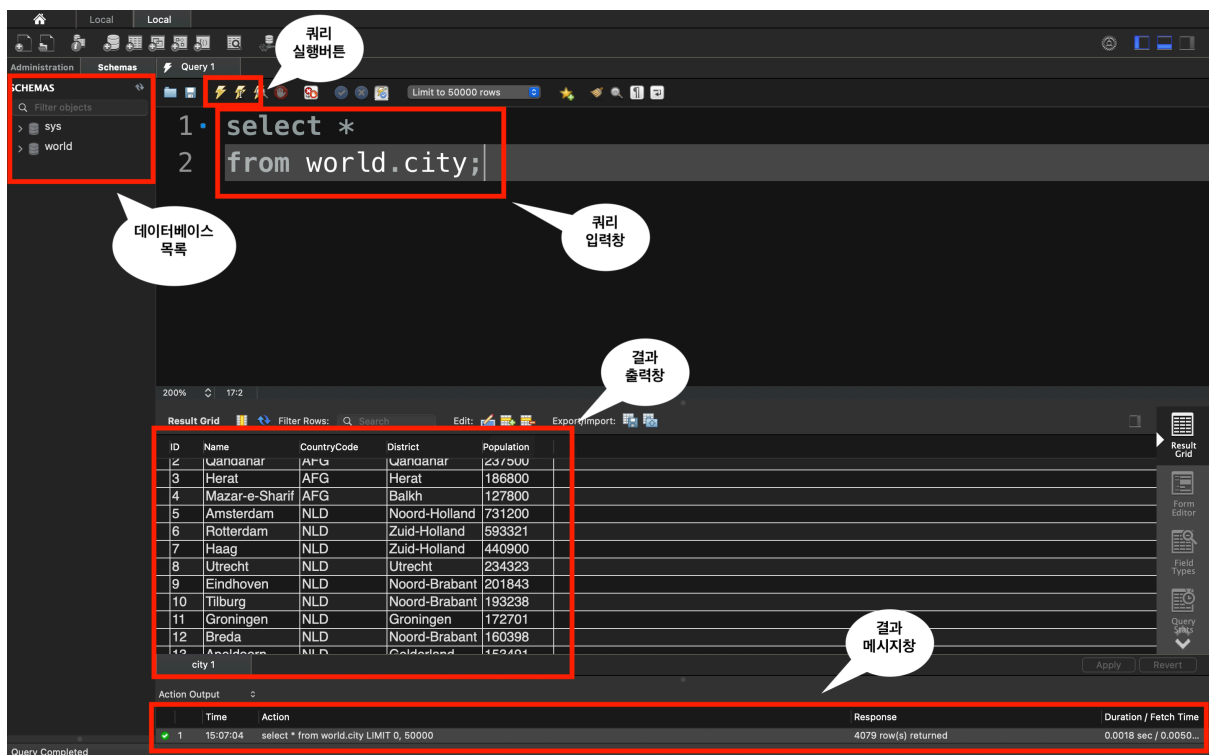
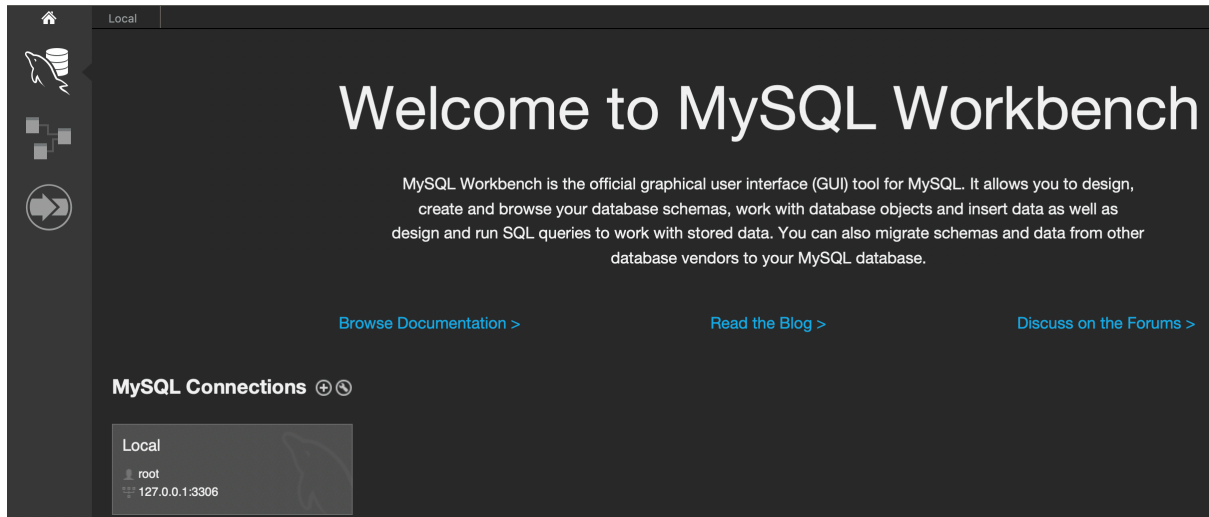
5. 접속 정보로 MySQL 서버에 접속하기

6. 세션(접속)정보 저장하기



세션(접속)정보를 저장하면 아래와 같이 접속 정보가 저장되어 있는 버튼이 생성됩니다.

생성된 버튼을 더블클릭하면 MySQL Workbench를 이용하여 MySQL 서버에 접속하게 됩니다. 접속 후에 해당 MySQL 서버에 Query문을 실행할수 있습니다.



2. 샘플 데이터 추가

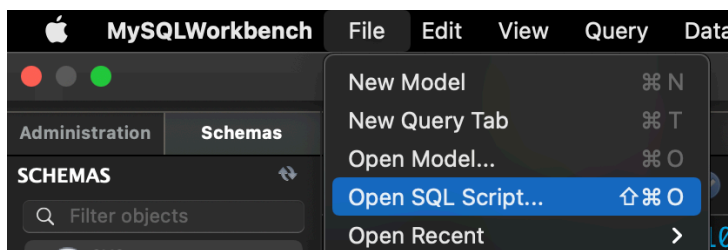
World, Sakila 데이터 베이스 추가 합니다.

아래의 링크에서 world와 sakila 데이터 베이스 다운로드 후 압축을 해제 합니다.

다운 로드 링크 : <https://dev.mysql.com/doc/index-other.html>

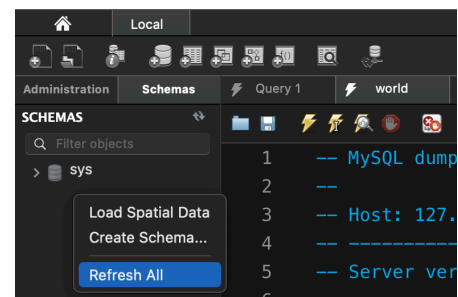
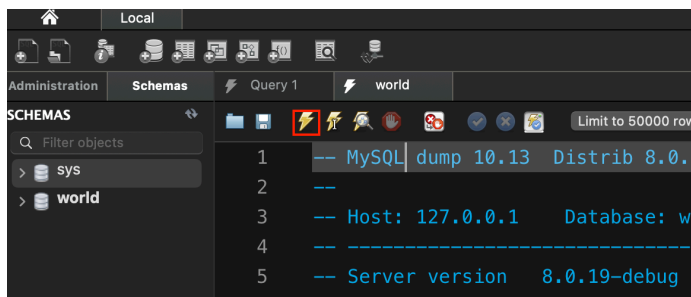
주로 World, Sakila 데이터를 이용하여 SQL을 실습합니다.

파일 메뉴에서 Open SQL Script를 선택 후 world.sql 파일을 선택



world.sql 파일에 있는 Query문이 로드되면 쿼리 실행 아이콘을 클릭합니다.

쿼리가 실행되고 좌측 레이아웃에서 오른쪽 마우스 클릭 후 Refresh All 메뉴를 선택하면 World 데이터 베이스가 나타납니다.



Sakila의 경우 스키마 파일과 데이터 파일이 분리되어 있습니다.

스키마 파일로 데이터 베이스의 테이블 구조를 먼저 만든 후에 데이터를 입력할수 있으므로 스키마 파일(sakila-schema.sql)을 위와 같이 실행한 후에 데이터 파일(sakila-data.sql) 실행합니다.

서버에서 데이터베이스의 데이터 추가

서버로 sql 파일을 전송

```
$ scp -i ~/.ssh/rada.pem ~/Desktop/sql/* ubuntu@15.164.231.87:~/
```

데이터 베이스 생성

```
$ mysql -u root -p
```

```
sql> create database world;
```

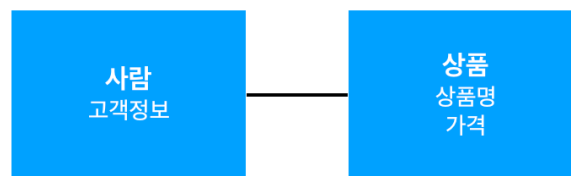
데이터 베이스에 데이터 추가

```
$ mysql -u root -p world < world.sql
```

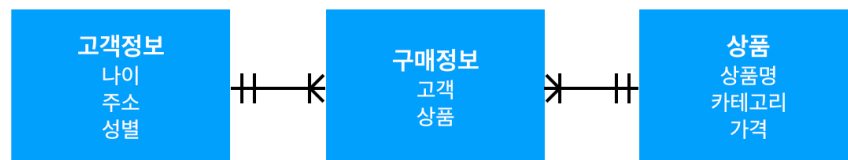
3. 데이터 베이스 모델링

데이터 베이스 모델링은 데이터 베이스에서의 테이블 구조를 미리 계획해서 작성하는 작업입니다. RDBMS는 테이블간에 유기적으로 연결되어 있기 때문에 모델링을 잘하는것이 중요합니다. 기본적으로 개념적 모델링, 논리적 모델링, 물리적 모델링 절차로 설계됩니다.

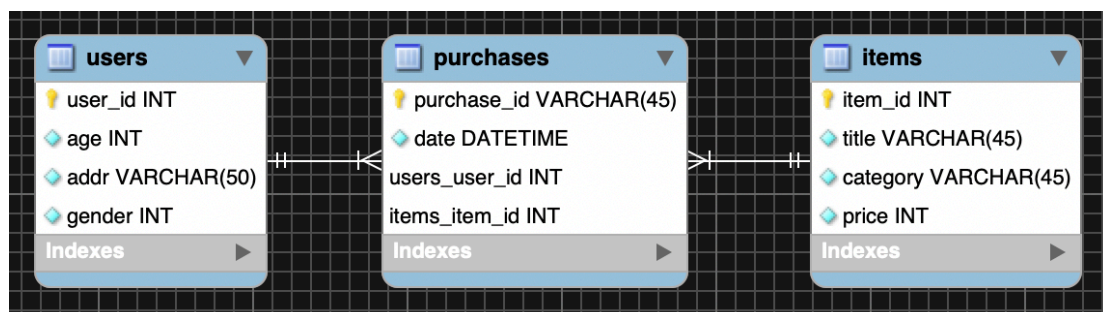
개념적 모델링 : 업무분석해서 핵심 데이터의 집합을 정의하는 과정



논리적 모델링 : 개념적 모델링을 상세화 하는 과정



물리적 모델링 : 논리적 모델링을 DBMS에 추가하기 위해 구체화 되는과정



실습

EER(Enhanced Entity-Relationship) 다이어그램 생성

1. File - New Model 선택
2. 이름 설정
3. EER(Diagram) - ADD Diagram
4. 테이블 추가
5. 컬럼추가
6. 테이블 관계 연결
7. File - Save Model 선택 후 저장 (test.mwb)

EER 다이어그램 파일을 적용하기

1. File - Open Model
2. Database - Forward Engineer
3. MySQL Server 연결정보를 선택하고 Continue 선택
4. 실행 쿼리에서 VISIABE 제거 후 실행 (버전에 따른 syntax 문제)

데이터 베이스를 EER 다이어그램으로 변경

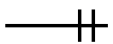
1. Database - Reverse Engineer
2. 세션 선택
3. 데이터 베이스 선택

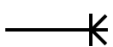
EER 다이어그램의 관계선 의미

실선 : 식별관계 : 부모가 있어야 자식이 생성됨 : 예) 비회원 허용 X

점선 : 비식별관계 : 부모가 없어도 자식이 생성됨 : 예) 비회원 허용 O

1:N - 없을수 있음 

1:1 - 1:1로 존재 

1:N - 1개는 존재 

4. SQL문의 종류 : DML DDL DCL

SQL문의 종류

DML

Data Manipulation Language

데이터 조작어

데이터 검색, 삽입, 수정, 삭제등에 사용

SELECT, INSERT, UPDATE, DELETE

트랜잭션이 발생하는 SQL문

DDL

Data Definition Language

데이터 정의어

데이터 베이스, 테이블, 뷰, 인덱스등의 데이터 베이스 개체를 생성, 삭제, 변경에 사용

CREATE, DROP, ALTER, TRUNCATE

실행 즉시 DB에 적용

DCL

Data Control Language

데이터 제어어

사용자의 권한을 부여하거나 빼앗을때 사용

GRANT, REVOKE, DENY

5. SELECT FROM

데이터를 검색할때 사용되는 문법입니다.

기본 포맷

SELECT <column_name_1>, <column_name_2>, ...

FROM <database_name>.<table_name>;

또는 FROM <table_name>;

전체 컬럼 데이터 조회

```
sql> SELECT *
```

```
    FROM world.country;
```

code, name 세개의 컬럼 데이터 조회

* SQL은 일반적으로 대소문자를 구분하지 않습니다.

* 주로 예약어(SELECT, FROM)는 대문자, 사용자 정의어(code, name, population)는 소문자를 사용합니다.

```
sql> SELECT code, name
```

```
    FROM world.country;
```

데이터베이스 선택 : FROM 절에 world. 을 사용할 필요가 없습니다.

```
sql> USE world;
```

```
sql> SELECT *
```

```
FROM country;
```

alias : 컬럼의 이름을 변경할수 있습니다.

```
sql> SELECT code AS country_code, name AS country_name
```

```
FROM country;
```

데이터베이스, 테이블, 컬럼 리스트 확인

```
sql> SHOW DATABASES;
```

```
sql> SHOW TABLES;
```

```
sql> DESC city;
```

6. MySQL의 주석

-- : 한줄주석, -- 를 한 후에 반드시 한칸을 띄어서 사용해야 합니다.

/* */ : 블록주석

-- 국가코드와 국가 이름을 출력합니다.

```
/* SELECT code, population
```

```
FROM country; */
```

```
SELECT code, name
```

```
FROM country;
```

7. OPERATORS

SQL Query에서는 세가지 연산자의 사용이 가능합니다.

연산자는 SELECT 절에서 숫자나 논리값들을 연산할때 사용이 가능하고 컬럼과 컬럼을 연산하는 경우에도 사용이 가능합니다.

산술연산자

+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
%	나머지
DIV	몫

world 데이터 베이스에서 각 국가의 국가코드, 국가이름, 인구밀도(1m²당 인구수)를 출력하는 쿼리를 작성하세요.

world 데이터 베이스에서 각 국가의 국가코드, 국가이름, 1인당 GNP를 출력하는 쿼리를 작성하세요.

비교연산자

결과가 True 이면 0을 출력, 결과가 False 이면 1을 출력

=	같다
!=	같지 않다
>	크다
<	작다
>=	크거나 같다
<=	작거나 같다

world 데이터 베이스에서 국가코드, 국가이름, 대륙, asia 대륙이면 1(True)을 출력하는 쿼리를 작성하세요.

world 데이터 베이스에서 국가코드, 국가이름, 독립년도, 독립년도가 1900년 이후에 독립했으면 1(True)을 출력하는 쿼리를 작성하세요.

논리연산자

True와 False의 논리 값을 연산

AND	둘다 True 이면 True
OR	둘다 False 이면 False

world 데이터 베이스에서 국가코드, 국가이름, 인구수, 기대수명, 인구수가 5천만 이상이고, 기대수명이 70세 이상이면 1(True)을 출력하는 쿼리를 작성하세요.

8. WHERE

특정 조건을 주어 데이터를 검색하는데 사용되는 문법입니다.

조건식에 비교연산, 논리연산 등을 사용할수 있습니다.

기본 포맷

SELECT <컬럼이름 1>, <컬럼이름 2>, ...

FROM <테이블 이름>

WHERE <조건식>

비교연산

인구가 1억이 넘는 국가를 출력

```
sql> SELECT *  
      FROM country  
      WHERE Population >= 100000000;
```

AND, OR : 논리 연산

인구가 7000만에서 1억인 국가를 출력

```
sql> SELECT *  
      FROM country  
      WHERE Population >= 70000000 AND Population <= 100000000;
```

BETWEEN

범위 연산시 사용

인구가 7000만에서 1억인 국가를 출력

```
sql> SELECT *  
      FROM country  
      WHERE Population BETWEEN 70000000 AND 100000000;
```

아시아와 아프리카 대륙의 국가 데이터를 출력

```
sql> SELECT *  
      FROM country  
      WHERE Continent = "Asia" OR Continent = "Africa";
```

IN, NOT IN

하나의 컬럼에 여러개의 OR를 사용하는 경우 쿼리를 간단하게 줄여줄수 있습니다.

아시아와 아프리카 대륙의 국가 데이터를 출력

```
sql> SELECT *  
      FROM country  
      WHERE Continent IN ("Asia", "Africa");
```

아시아와 아프리카 대륙의 국가가 아닌 데이터를 출력

```
sql> SELECT *  
      FROM country  
      WHERE Continent NOT IN ("Asia", "Africa");
```

아시아와 아프리카 대륙의 국가가 아닌 데이터를 출력 (논리연산 사용)

```
sql> SELECT *  
      FROM country  
      WHERE Continent != "Asia" AND Continent != "Africa";
```

LIKE

특정 문자열이 포함된 데이터 출력합니다.

country 테이블에서 국가 코드가 Z로 시작되는 데이터를 출력

```
SELECT *  
FROM country  
WHERE code like "Z%";
```

정부형태에 Republic이 포함된 데이터 출력

```
sql> SELECT *
```

```
FROM country
```

```
WHERE GovernmentForm LIKE "%Republic%";
```

9. ORDER BY

특정 컬럼의 값으로 데이터 정렬에 사용되는 문법

오름차순 인구순으로 국가의 리스트를 출력

ASC는 생략이 가능

```
SELECT *
```

```
FROM country
```

```
ORDER BY population ASC
```

내림차순 인구순으로 국가의 리스트를 출력

```
SELECT *
```

```
FROM country
```

```
ORDER BY population DESC
```

ORDER BY에 여러개의 정렬 조건을 사용

국가 코드를 알파벳 순으로 정렬하고 같은 국가 코드를 가지면 인구순으로 내림차순으로 정렬

```
SELECT *
```

```
FROM city
```

```
ORDER BY CountryCode ASC, Population DESC
```

10. LIMIT

LIMIT은 조회하는 데이터의 수를 제한할수 있습니다.

인구가 많은 상위 5개 국가 데이터를 출력

```
SELECT *  
FROM country  
ORDER BY population DESC  
LIMIT 5;
```

인구가 많은 상위 6위 ~ 8위의 3개 국가 데이터를 출력

```
# OFFSET 사용  
SELECT *  
FROM country  
ORDER BY population DESC  
LIMIT 3 OFFSET 5; # 5개 스킵하고 3개를 출력
```

```
SELECT *  
FROM country  
ORDER BY population DESC  
LIMIT 5, 3; # 5개 스킵하고 3개를 출력
```


DISTINCT

출력 데이터에서 중복 데이터를 제거해서 출력하는 예약어

city 테이블에서 도시의 인구수가 100만에서 200만인 도시의 국가 코드를 중복을 제거후 출력

```
SELECT DISTINCT countrycode
```

```
FROM city
```

```
WHERE population BETWEEN 1000000 AND 2000000;
```