

Mysql 1

Index

1. 데이터 베이스의 개요
2. Mysql 설치 및 설정
3. 샘플 데이터 추가
4. Mysql Workbench 사용법
5. 데이터 베이스 모델링
6. SQL문의 종류 : DML DDL DCL
7. SELECT FROM
8. WHERE, IN, LIKE
9. ORDER BY
10. LIMIT
11. GROUP BY, HAVING
12. CREATE USE ALTER DROP
13. DATA TYPE
14. Constraint : 제약조건
15. INSERT
16. UPDATE SET
17. DELETE TRUNCATE
18. Functions 1 (CONCAT, CEIL, ROUND, TRUNCATE, DATE_FORMAT)
19. Functions 2 (IF, IFNULL, CASE)
20. JOIN
21. UNION
22. Sub Query
23. VIEW
24. INDEX

1. 데이터 베이스의 개요

DB : Database

데이터를 통합하여 관리하는 데이터의 집합

DBMS : Database Management System

데이터 베이스를 관리하는 미들웨어 시스템, 데이터 베이스 관리 시스템

RDBMS : Relational Database Management System

Oracle, Mysql, Postgresql, Sqlite

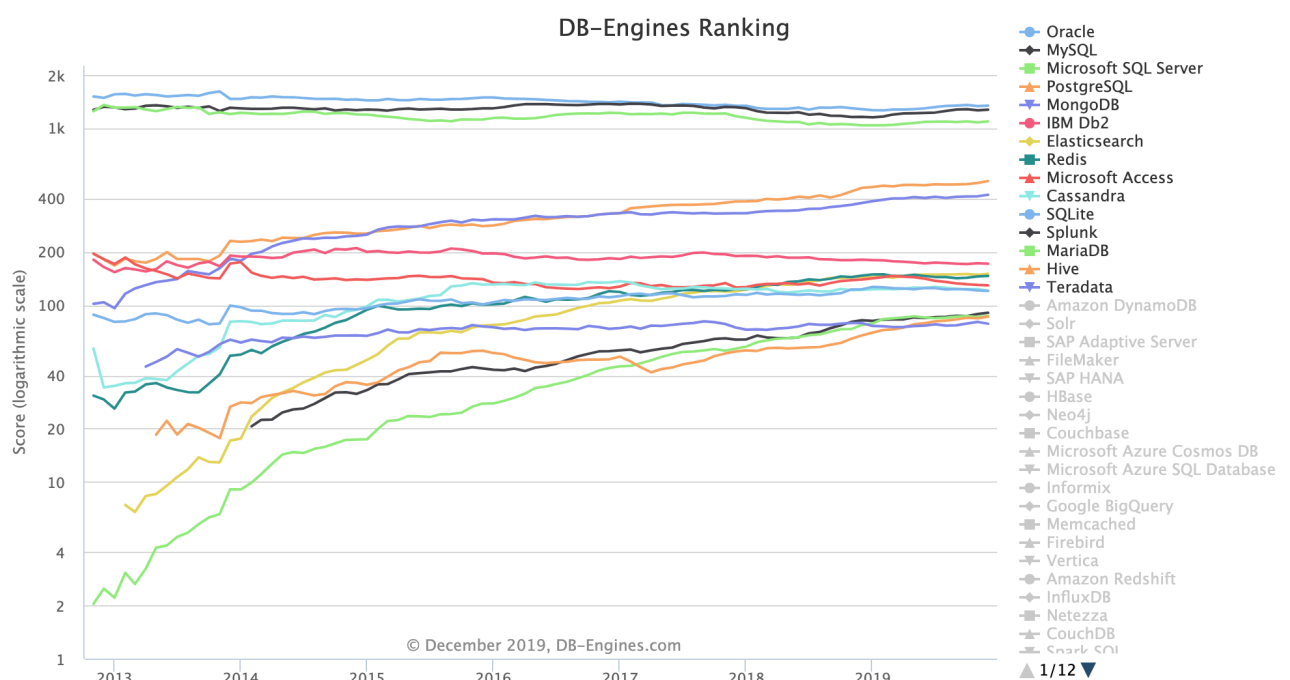
데이터의 테이블 사이에 키값으로 관계를 가지고 있는 데이터 베이스

NoSQL

Mongodb, Hbase, Cassandra

데이터 테이블 사이의 관계가 없이 데이터를 저장하는 데이터 베이스

데이터 사이의 관계가 없으므로 복잡성이 작고 많은 데이터의 저장이 가능



MySQL의 특징

MySQL은 오픈소스이며 다중 사용자와 다중 스레드 지원

다양한 운영체제에 다양한 프로그래밍 언어 지원

표준 SQL을 사용

작고 강력하며 가격이 저렴

Oracle pricing table - <http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>

History

1995 - MySQL AB사에 의해 첫 버전 발표

2008 - 썬마이크로시스템이 MySQL AB 인수, 5.1 버전

2009 - 오라클이 썬마이크로시스템 인수

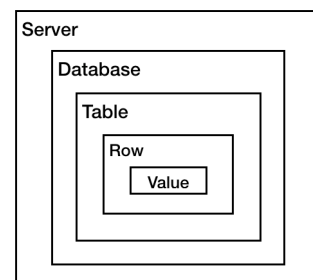
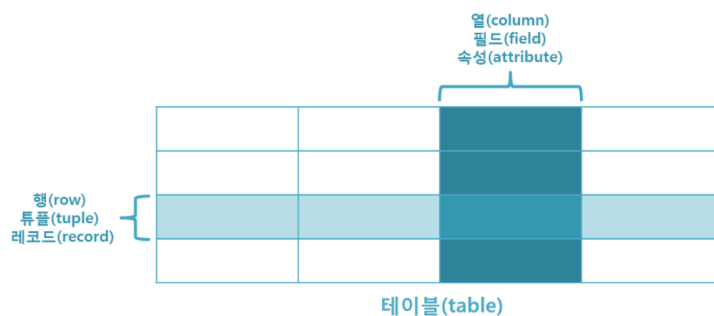
2018 - MySQL 8.0 버전 발표

License

MySQL을 포함하는 하드웨어나 소프트웨어 기타 장비를 판매하는 경우 라이선스 필요

배포시 소스를 공개하면 무료이지만 소스공개를 원하지 않는 경우 사용라이선스 필요

서비스에 이용하는건 무료로 가능



2. Mysql 설치 및 설정

AWS EC2 인스턴스에 Ubuntu OS에 MySQL 5.7.x 버전 설치

EC2 인스턴스 생성

- t2.micro
- Ubuntu 18.04 버전
- 보안그룹에서 3306 포트 추가

EC2 인스턴스에 접속

pem 파일 400 권한으로 변경

\$ ssh -i ~/.ssh/rada.pem ubuntu@15.164.231.87

apt-get 업데이트

\$ sudo apt-get update -y

\$ sudo apt-get upgrade -y

MySQL Server 설치

\$ sudo apt-get install -y mysql-server mysql-client

MySQL secure 설정

\$ sudo mysql_secure_installation

Would you like to setup VALIDATE PASSWORD plugin? N

New password: rada

Re-enter new password: rada

Remove anonymous users? Y

Disallow root login remotely? N

Remove test database and access to it? Y

Reload privilege tables now? Y

MySQL 패스워드 설정

\$ sudo mysql

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password  
BY 'rada';
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

```
mysql> exit
```

설정한 패스워드를 입력하여 접속

\$ mysql -u root -p

Enter password: rada

외부접속 설정

\$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

bind-address를 127.0.0.1을 0.0.0.0 으로 변경

```
-----  
bind-address      = 0.0.0.0  
-----
```

외부접속 패스워드 설정

```
mysql> grant all privileges on *.* to root@'%' identified by 'rada';
```

서버 시작 종료 상태 확인

\$ sudo systemctl start mysql.service

\$ sudo systemctl stop mysql.service

\$ sudo systemctl restart mysql.service

\$ sudo systemctl status mysql.service

설정 후 서버 재시작으로 설정 내용 적용

\$ sudo systemctl restart mysql.service

password 변경 : rada로 패스워드를 변경하는 경우

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'rada';

3. 샘플 데이터 추가

World, Sakila 데이터 베이스 추가

아래의 링크에서 world와 sakila 데이터 베이스 다운로드

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

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;
```

```
sql> create database sakila;
```

```
sql> create database employees;
```

```
sql> quit
```

데이터 베이스에 데이터 추가

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

```
$ mysql -u root -p sakila < sakila-schema.sql
```

```
$ mysql -u root -p sakila < sakila-data.sql
```

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

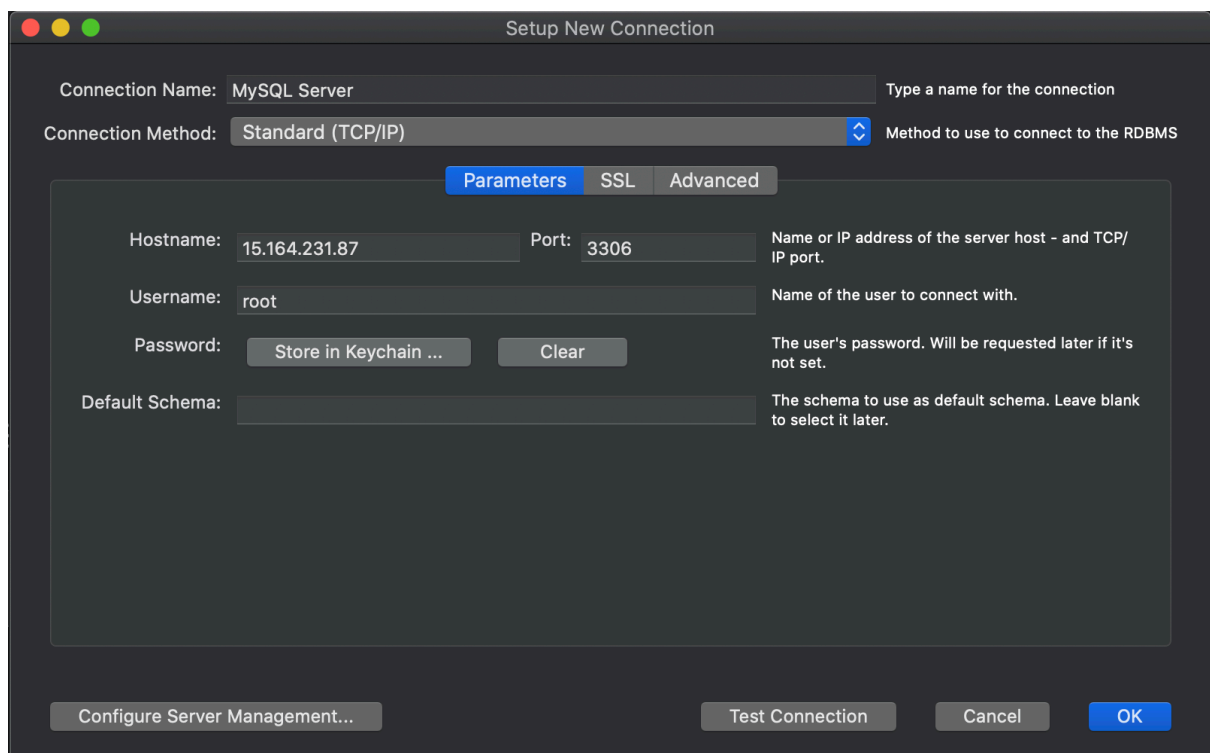
4. Mysql Workbench 사용법

Mysql Management Tool

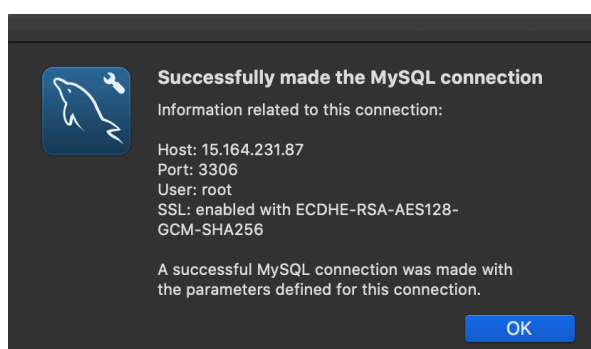
Workbench 다운로드 및 설치

<https://dev.mysql.com/downloads/workbench/>

Workbench를 이용한 접속



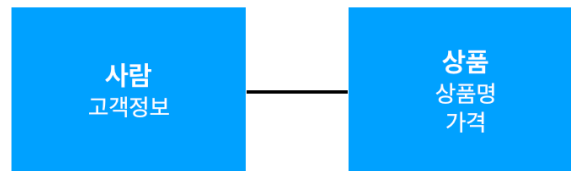
Test Connection



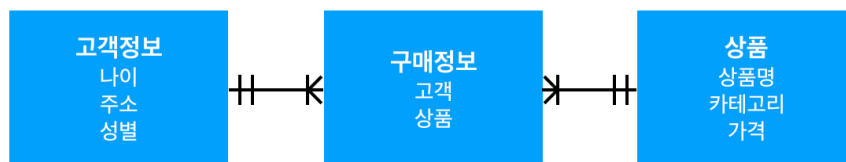
5. 데이터 베이스 모델링

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.

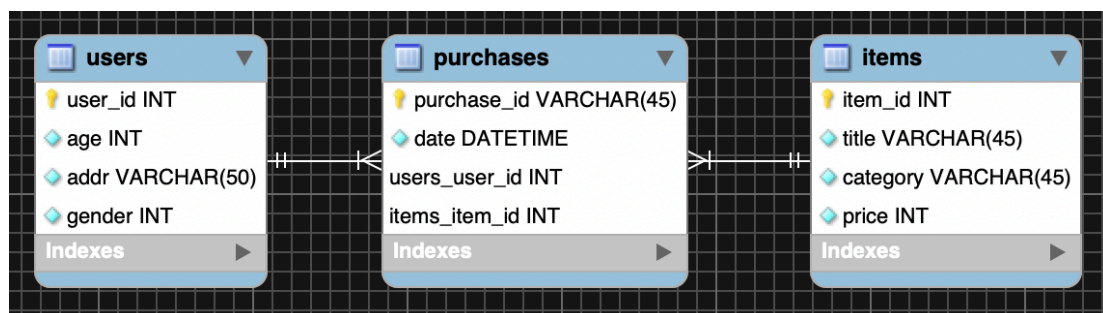
개념적 모델링 : 업무분석해서 핵심 데이터의 집합을 정의하는 과정



논리적 모델링 : 개념적 모델링을 상세화 하는 과정



물리적 모델링 : 논리적 모델링을 DBMS에 추가하기 위해 구체화 되는과정



실습

EER 다이어그램 생성

File - New Model 선택

이름 설정

EER(Diagram) - ADD Diagram

테이블 추가

컬럼추가

테이블 관계 연결

모델링 파일을 실제 데이터 베이스에 연결

File - Save Model 선택 후 저장 (test.mwb)

EER 다이어그램 파일을 적용하기

File - Open Model

Database - Forward Engineer

MySQL Server 연결정보를 선택하고 Continue 선택

실행 쿼리에서 VISIABE 제거 후 실행 (버전에 따른 syntax 문제)

데이터 베이스를 EER 다이어그램으로 변경

Database - Reverse Engineer

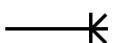
관계선 의미

실선 : 식별관계 : 부모가 있어야 자식이 생성됨

점선 : 비식별관계 : 부모가 없어도 자식이 생성됨

1:N - 없을수 있음 

1:1 - 1:1로 존재 

1:N - 1개는 존재 

6. 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

7. SELECT FROM

데이터를 검색할때 사용되는 문법

기본 포맷

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

FROM <table_name>

전체 컬럼 데이터 조회

```
sql> SELECT *
```

```
FROM world.country
```

code, name 세개의 컬럼 데이터 조회

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

```
FROM world.country
```

데이터 베이스 선택 : FROM 절에 world. 을 사용할 필요가 없다.

```
sql> USE world;
```

```
sql> SELECT *
```

```
FROM country
```

alias : 컬럼의 이름을 변경할수 있습니다.

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

```
FROM country
```

데이터 베이스, 테이블, 컬럼 리스트 확인

```
sql> SHOW DATABASES;
```

```
sql> SHOW TABLES;
```

```
sql> DESC city;
```

8. WHERE

특정 조건을 주어 데이터를 검색하는데 사용되는 문법

비교연산

인구가 1억이 넘는 국가를 출력

```
SELECT *
```

```
FROM country
```

```
WHERE Population >= 100000000
```

논리 연산 : AND, OR

인구가 7000만에서 1억인 국가를 출력

```
SELECT *
```

```
FROM country
```

```
WHERE Population >= 70000000 AND Population <= 100000000
```

```
# 범위 연산 : BETWEEN
```

```
# 인구가 7000만에서 1억인 국가를 출력
```

```
SELECT *
```

```
FROM country
```

```
WHERE Population BETWEEN 70000000 AND 100000000
```

```
# 아시아와 아프리카 대륙의 국가 데이터를 출력
```

```
SELECT *
```

```
FROM country
```

```
WHERE Continent = "Asia" OR Continent = "Africa"
```

```
# 특정 조건을 포함 : IN, NOT IN
```

```
# 아시아와 아프리카 대륙의 국가 데이터를 출력
```

```
SELECT *
```

```
FROM country
```

```
WHERE Continent IN ("Asia", "Africa")
```

```
# 아시아와 아프리카 대륙의 국가가 아닌 데이터를 출력
```

```
SELECT *
```

```
FROM country
```

```
WHERE Continent NOT IN ("Asia", "Africa")
```

```
# 아시아와 아프리카 대륙의 국가가 아닌 데이터를 출력 (논리연산 사용)
```

```
SELECT *
```

```
FROM country
```

```
WHERE Continent != "Asia" AND Continent != "Africa"
```

```
# 특정 문자열이 포함된 데이터 출력 : LIKE
```

```
# 정부형태에 Republic이 포함된 데이터 출력
```

```
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
```

```
# 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

SELECT *

FROM country

ORDER BY population DESC

LIMIT 5, 3