MySQL

0.Install

homepage

홈페이지

→ DOWNLOADS → MySQL Community (GPL) Downloads

→ MySQL Community Server → OS 선택 → 컴퓨터 사양에 맞는 파일 선택

→ Downloads → No thanks, just start my download.

0.Install


```
docker run --name mysql -e MYSQL_ROOT_PASSWORD=1234 -p 3306:3306 -d mysql

docker start mysql

docker exec -it mysql bash

mysql -u root -p
```

0.Install

homebrew

```
brew install mysql

brew services start mysql

mysql_secure_installation
* 비밀번호 관련 설정

mysql -u root -p
```

1.DataBase

여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합 -wiki

RDBMS (Relational DataBase Management System)

- 관계형 데이터 베이스

- 정형 데이터들의 관계를 정의

NoSQL (Not only SQL)

- 비 정형 데이터 저장

- 빅데이터에 용이 (분산 저장 등)

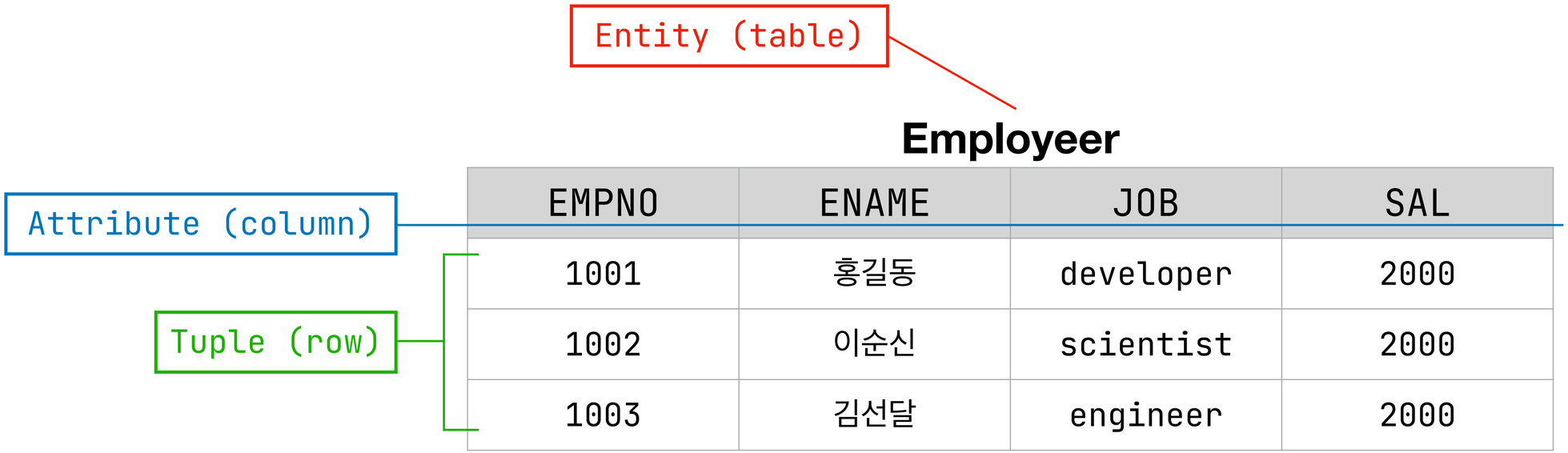
1.DataBase

RDB

Entity 데이터베이스에 저장하려고 하는 현실상의 개념 / 객체

Attribute Entity의 속성

Tuple Entity의 값



EMPNO	ENAME	JOB	SAL
1001	홍길동	developer	2000
1002	이순신	scientist	2000
1003	김선달	engineer	2000

2.SQL

Structured Query Language : 구조화 된 질의 언어

DDL (Data Definition Language) 데이터 정의 언어
DB 스키마 정의, 조작

DML (data Manipulation Language) 데이터 조작 언어
Data 조작

DCL (Data Control Language) 데이터 제어 언어
Data 제어

2.SQL

데이터 정의 언어 (Data Definition Language)

CREATE	데이터베이스, 테이블, 뷰(view), 프로시저(procedure) 등을 생성
ALTER	데이터베이스, 테이블, 뷰, 프로시저 등을 수정
DROP	데이터베이스, 테이블, 뷰, 프로시저 등을 삭제

* View : 다른 테이블/뷰 에 있는 데이터를 보여주기 위해 사용 (수정 불가)
  Procedure : 특정 작업에 필요한 Query들을 함수처럼 사용

2.SQL


```
CREATE DATABASE 데이터베이스;
```

데이터베이스 생성

```
CREATE TABLE 테이블 (
  컬럼 DATA_TYPE(SIZE),
  ...,
  CONSTRAINT 제약조건 제약조건 (컬럼)
);
```

테이블 생성

```
CREATE VIEW 뷰 AS SELECT ~;
```

뷰 생성

* auto_increment : 숫자 자동 증가 옵션 (sequence)

2.SQL


```
ALTER TABLE 테이블 ADD | DROP | MODIFY 컬럼 DATA_TYPE
```

ADD 컬럼 추가

DROP 컬럼 삭제

MODIFY 컬럼 수정

DROP DATABASE 데이터베이스

- 데이터베이스 삭제

DROP TABLE 테이블

- 테이블 삭제

* 데이터베이스 삭제 시, 해당 데이터베이스 안에 있는 모든 데이터(테이블, 뷰, ...)도 삭제됨!

2.SQL

데이터 조작 언어 (Data Manipulation Language)

SELECT	데이터 읽기
INSERT	데이터 삽입
UPDATE	데이터 수정
DELETE	데이터 삭제

2.SQL

SELECT 컬럼 FROM 테이블;

- 테이블 에서 컬럼 조회

SELECT 컬럼 FROM 테이블 WHERE 조건;

- 테이블 에서 조건에 맞는 컬럼 조회

* 컬럼의 값 연산 (산술, 비교 등) : >, >=, <, <=, =, !=(<>), BETWEEN, IN, NOT IN, ANY, ...
  조건의 연결 : AND, OR

2.SQL

SELECT 컬럼 FROM 테이블 ORDER BY 컬럼;

- 테이블 에서 컬럼의 값으로 정렬 후 컬럼 조회 (ASC : 오름차순 / DESC : 내림차순)

SELECT 컬럼 FROM 테이블 GROUP BY 컬럼;

- 테이블 에서 컬럼의 값으로 그룹화 시킨 후 컬럼 조회

SELECT 컬럼 FROM 테이블 WHERE|HAVING 조건 GROUP BY 컬럼;

- 그룹화 후 집계함수에 조건을 줄 땐 WHERE 대신 HAVING 사용

2.SQL


```
INSERT INTO 테이블 VALUES(모든 컬럼의 값);
```

- 테이블의 모든 컬럼에 값 삽입

```
INSERT INTO 테이블(컬럼, ...) VALUES(명시된 컬럼의 값);
```

- 테이블의 특정 컬럼에 값 삽입

2.SQL


```
UPDATE 테이블 SET 컬럼 = 값, 컬럼 = 값, ...;
```

- 해당 컬럼의 모든 값 수정

```
UPDATE 테이블 SET 컬럼 = 값, 컬럼 = 값, ... WHERE 조건;
```

- 조건에 맞는 컬럼의 값 수정

2.SQL


```
DELETE FROM 테이블;
```

- 테이블의 모든 값 삭제

```
DELETE FROM 테이블 WHERE 조건;
```

- 조건에 맞는 값 삭제

2.SQL

데이터 제어 언어 (Data Control Language)

COMMIT	데이터, 트랜잭션 저장
ROLLBACK	데이터, 트랜잭션 취소 (가장 마지막 COMMIT으로 되돌아감)
GRANT	DB 권한 부여
REVOKE	DB 권한 삭제

* TCL (Transaction Controll Language) : COMMIT, ROLLBACK

2.SQL

키 (key) : 식별자(Identifier) → 튜플을 식별(검색, 정렬 등) 할 때 사용

슈퍼키 (Super key / Composite key)	복합키 라고도 하며, 유일성을 만족하는 속성
후보키 (Candidate key)	유일성과 최소성을 만족하는 속성
기본키 (Primary key)	후보키 중 선택된 키
대체키 (Alternate key)	후보키 중 선택되지 않은 나머지 키
외래키 (Foreign key)	다른 테이블의 컬럼이나 기본키를 참조

2.SQL

정규화 (Normalization) : 데이터들의 관계를 구조화시켜 테이블을 분리

1 NF	하나의 컬럼엔 하나의 값
2 NF	기본키에 종속적이지 않은 컬럼 분리
3 NF	기본키를 제외한 나머지 컬럼들 간의 종속 불가
BCNF	기본키가 여러 개 존재할 경우, 복합키로
4 NF	다치종속 제거
5 NF	조인이 후보키를 통해서만 성립되도록

* 역정규화 (반정규화) : 필요할 경우 정규화를 거꾸로 진행

2.SQL

데이터의 무결성 : 데이터의 정확성과 일관성 유지

제 약 조 건	설 명
NOT NULL	해당 컬럼에 NULL 입력 불가
UNIQUE	해당 컬럼에 중복되는 값 불가
PRIMARY KEY	각 행을 유일하게 식별
FOREIGN KEY	다른 테이블의 컬럼 값 참조
CHECK	해당 컬럼에 특정 값만 허용

3.MYSQL

DOCUMENTATION → MySQL Server → DataTypes

Numeric	숫자	정수, 고정 소수점, 부동 소수점, 비트
Date & Timedata	날짜	날짜, 날짜 및 시간
String	문자	고정 길이, 가변 길이, ...
Spatial	공간	공간 데이터 (점, 선, 면, ...)
Json	json	'{"key": "value"}'

...

3.MYSQL

numeric

데이터 형식	바이트 수	설 명
bit		1 ~ 64bit 표현. b'0000' 형식
tinyint	1	정수 (-128 ~ 127)
bool tinyint(1)		0이면 False, 이외의 숫자는 True
smallint	2	정수 (-32,768 ~ 32767)
mediumint	3	정수 (-8388608 ~ 8388607)
int integer	4	정수 (약 -21억 ~ 21억)
bigint	8	정수 (약 -900경 ~ 900경)
dec decimal		고정 소수점 (-10^38+1 ~ 10^38+1)
float		부동 소수점 -3.40E+38 ~ -1.17E-38
double		부동 소수점 -1.22E-308 ~ 1.79E+308

* unsigned : 양수만 표현

3.MYSQL

데이터 형식	바이트 수	설 명
date	3	YYYY-MM-DD
datetime	5	YYYY-MM-DD HH:MM:SS + @
timestamp	4	YYYY-MM-DD HH:MM:SS + @
time	3	HH:MM:SS + @
year	1	YYYY

* time, datetime, timestamp → 밀리초 단위는 2자리당 1바이트 추가

* datetime / timestamp는 거의 같음 (time_zone 저장 방식이 다름)

3.MYSQL

데이터 형식	바이트 수	설 명
char	1 ~ 255	고정 길이 문자열 값
varchar	1 ~ 65535	가변 길이 문자열 값
binary	1 ~ 255	고정 길이 이진 데이터 값
varbinary	1 ~ 255	가변 길이 이진 데이터 값
tinyblob	1 ~ 255	255 크기의 blob 데이터
tinytext	1 ~ 255	255 크기의 text 데이터
blob	1 ~ 65535	blob 데이터
text	1 ~ 65535	text 데이터
mediumblob	1 ~ 16777215	16777215 크기의 blob 데이터
mediumtext	1 ~ 16777215	16777215 크기의 text 데이터
longblob	1 ~ 약 42억	4GB 크기의 blob 데이터
longtext	1 ~ 약 42억	4GB 크기의 text 데이터
enum	1, 2	문자열을 숫자와 맵핑 (열거형 데이터)
set	1,2,3,4,8	문자열을 숫자와 맵핑 (하나의 컬럼에 두 개 이상의 데이터 저장 가능)

3.MYSQL

BLOB : Binary Large OBject

blob, text : 초기값을 가질 수 없음

char, varchar, text : charset 가능

한글의 크기 : utf8mb4 (utf8) ⇒ 3byte
utf16 ⇒ 2byte

3.MYSQL

데이터 형식	설 명
point	점
linestring	선
polygon	다각형
multipoint	여러 개의 점
multilinestring	여러 개의 선
multipolygon	여러 개의 다각형
geometrycollection	여러 개의 geometry 객체

* geometry : 공간 데이터 형식 (점, 선, 다각형 등의 공간 데이터 개체 저장 및 조작)

3.MYSQL

서로 관계가 있는 테이블을 하나로 연결

INNER JOIN

NATURAL JOIN

CROSS JOIN

OUTER JOIN

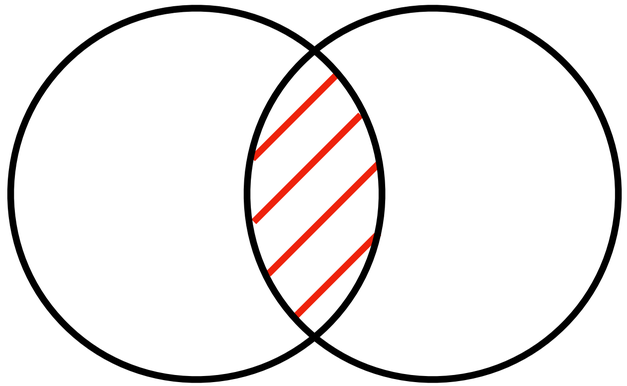
...

3.MYSQL


```
SELECT 컬럼 FROM 테이블A INNER JOIN 테이블B ON 테이블A.컬럼 = 테이블B.컬럼
```

```
SELECT 컬럼 FROM 테이블A INNER JOIN 테이블B USING(컬럼)
```

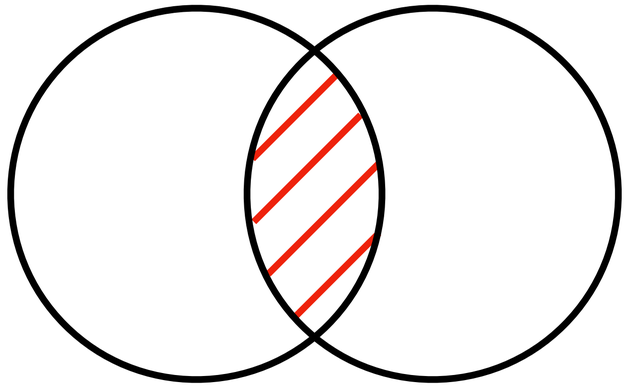
* 테이블A의 컬럼명과 테이블B의 컬럼명이 같으면 USING(컬럼)으로 사용 가능



3.MYSQL

SELECT 컬럼 FROM 테이블A NATURAL JOIN 테이블B

* 테이블A의 컬럼명과 테이블B의 컬럼명이 단 하나만 같은 경우에 사용



3.MYSQL


```
SELECT 컬럼 FROM 테이블A CROSS JOIN 테이블B
```

```
SELECT 컬럼 FROM 테이블A JOIN 테이블B
```

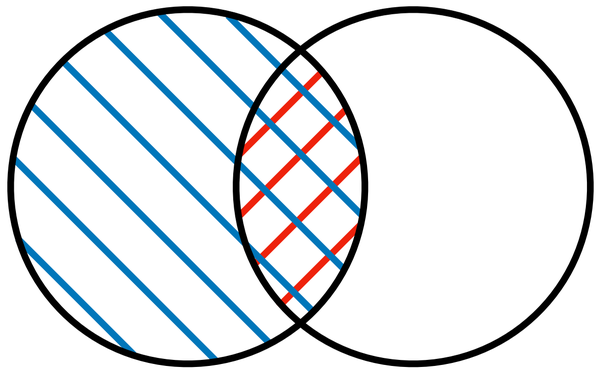
* CARTESIAN PRODUCTS

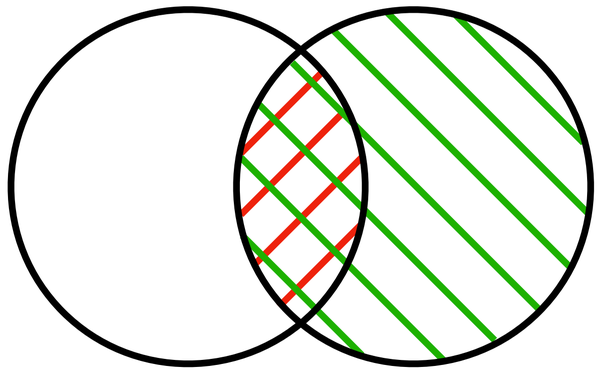
3.MYSQL

outer

```
SELECT 컬럼 FROM 테이블A LEFT JOIN 테이블B ON 테이블A.컬럼 = 테이블B.컬럼
```

```
SELECT 컬럼 FROM 테이블A RIGHT JOIN 테이블B ON 테이블A.컬럼 = 테이블B.컬럼
```





3.MYSQL

QUERY 내부에서 사용되는 SELECT문

중첩된 SELECT (Nested SELECT) 라고 부르기도 한다

SINGLE ROW SUBQUERY	결과가 1개의 값
MULTI ROW SUBQUERY	결과가 여러 개의 값
MULTI COLUMN SUBQUERY	WHERE 조건절에서 여러 개의 컬럼 값 비교
INLINE VIEW	FROM 절에서 사용 (가상 테이블)

3.MYSQL

DOCUMENTATION → Connectors → Connector/Python Developer Guide

```
cd c:\Program Files\MySQL\Connector Python 8.0

pip install mysql-connector-python

import mysql.connector

cnx = mysql.connector.connect(user='root',
                              password='1234',
                              host='127.0.0.1',
                              database='mysql')
cursor = cnx.cursor()
```