

MariaDB

Index

1. 개요

- MariaDB

2. 데이터베이스 설치

- 서버
- 설치

3. SQL 1

- 개요
- DDL, DCL

4. SQL 2

- DML

Overview

개요



Maria DB

- 오픈소스 기반 RDBMS
- MySQL과 동일한 소스코드 기반
- Oracle 소유의 불확실한 MySQL 라이선스 상태에 반발하여 만들어짐
- 마리아는 주요 개발자 몬티 와이드니어스의 둘째 딸 이름

발표일 : 2009년 1월 22일

```
[kousekip@ako-kaede-mirai]-(10:37pm~:~03/30)~$  
[kousekip@ako-kaede-mirai]~$ sudo mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 6  
Server version: 10.5.9-MariaDB Arch Linux  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> status  
  
mysql Ver 15.1 Distrib 10.5.9-MariaDB, for Linux (x86_64) using readline 5.1  
  
Connection id:          6  
Current database:        
Current user:          root@localhost  
SSL:                   Not in use  
Current pager:         stdout  
Using outfile:         ''  
Using delimiter:       ;  
Server:                MariaDB  
Server version:        10.5.9-MariaDB Arch Linux  
Protocol version:      10  
Connection:            Localhost via UNIX socket  
Server characterset:    utf8mb4  
Db characterset:        utf8mb4  
Client characterset:    utf8  
Conn. characterset:     utf8  
UNIX socket:           /run/mysqld/mysqld.sock  
Uptime:                1 min 44 sec  
  
Threads: 1  Questions: 12  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.115  
  
MariaDB [(none)]> █
```

Installation

설치

서버

- 정보나 서비스를 네트워크를 통해 클라이언트에게 제공하는 컴퓨터 시스템



데이터베이스 서버

| 이름 | 전화번호 |
|-----|---------------|
| 홍길동 | 010-3242-5931 |
| 이동진 | 010-3943-1992 |
| 박철우 | 010-6123-4453 |

데이터베이스

| 이름 | 구입일 | 상품 | 수량 |
|-----|------------|----|----|
| 홍길동 | 2022-06-11 | 우유 | 1 |
| 홍길동 | 2022-06-13 | 식빵 | 2 |
| 이동진 | 2022-06-20 | 치즈 | 1 |
| 박철우 | 2022-06-10 | 소금 | 1 |
| 박철우 | 2022-06-12 | 우유 | 3 |



MariaDB 설치

- <https://mariadb.org/>
- Download

- 버전, OS, 아키텍처, 패키지 설정 후 다운로드

Download | Documentation | Contribute | Server Fest | Events | Sponsor | Blog | Planet MariaDB | About

Latest releases 10.9.1 (RC) 10.8.3, 10.7.4, 10.6.8, 10.5.16, 10.4.25, 10.3.35, 10.2.44.

MariaDB Server: The open source relational database

[Download](#) [Sponsor](#)

MariaDB replication using containers

In this blog we are going to demonstrate how to replicate a MariaDB database that runs in a Docker container (let's call it primary) to one or more MariaDB servers that run in a Docker container (let's call them replicas) using binary logging, a method that creates binary log files and an index that contains the record of all changes to the database (both data and structure). You can find an overview of how replication works [here](#) and you can find how to setup replication [here](#). An example can be found in this GitHub script...

[Read more](#)

MariaDB Foundation YouTube

CEO Fireside Chat with Xiang Peng (Sean), Director, RDS, Open Source

Seppo Jaakola - MariaDB Galera Cluster

Jeff Hardy - An Economic Model of Cyber Protection - 2022

Serg Fest

Download | Documentation | Contribute | Server Fest | Events | Sponsor | Blog | Planet MariaDB | About

Latest releases 10.9.1 (RC) 10.8.3, 10.7.4, 10.6.8, 10.5.16, 10.4.25, 10.3.35, 10.2.44.

Download MariaDB Server

MariaDB Server is one of the world's most popular open source relational databases and is available in the standard repositories of all major Linux distributions. Look for the package mariadb-server using the package manager of your operating system. Alternatively you can use the following resources:

[Download MariaDB Server REST API](#)

[Release Schedule](#)

[Reporting Bugs](#)

[MariaDB Server Statistics](#)

[Download MariaDB Server Documentation](#)

[View all releases for:](#)

- MariaDB Server
- MariaDB Galera
- Connector/C
- Connector/J
- Connector/ODBC
- Connector/Python
- Connector/Node.js

MariaDB Server **MariaDB Repositories** **Connectors**

MariaDB Server Version

MariaDB Server 10.6.8

Display older releases: ☐

Operating System

Windows

Architecture

x86_64

Package Type

MSI Package

[Download](#) **Mirror**

Yongbok.net - South Korea

Release date: 2022-05-21
File name: mariadb-10.6.8-winx64.msi
File size: 64.4 MB
• [Download galera-26.4.11](#)

[Display signature and checksums](#)

REST API

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

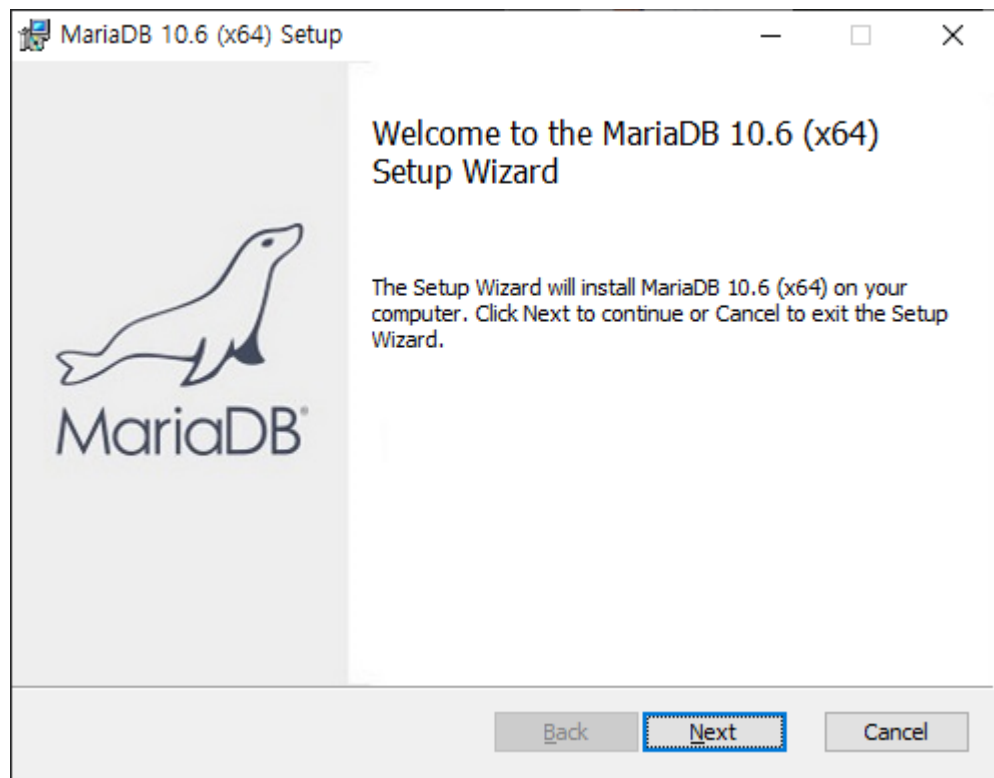
☒ Necessary ☒ Preferences ☒ Statistics ☐ Marketing [Show details](#) [OK](#)

This website uses cookies

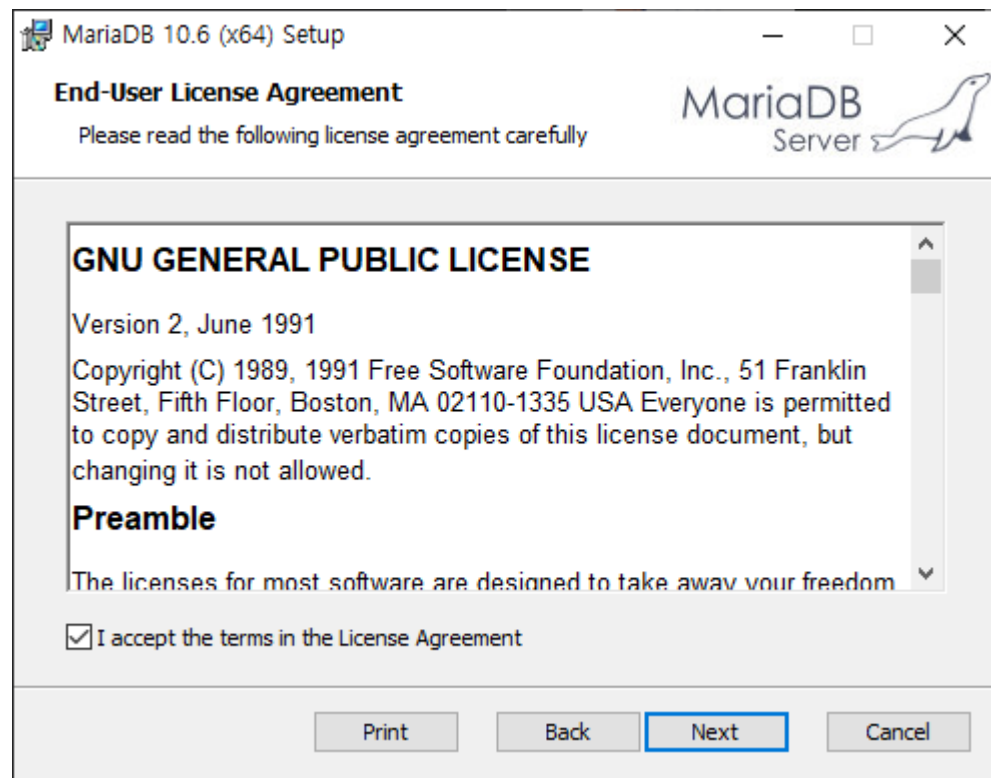
We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

☒ Necessary ☒ Preferences ☒ Statistics ☐ Marketing [Show details](#) [OK](#)

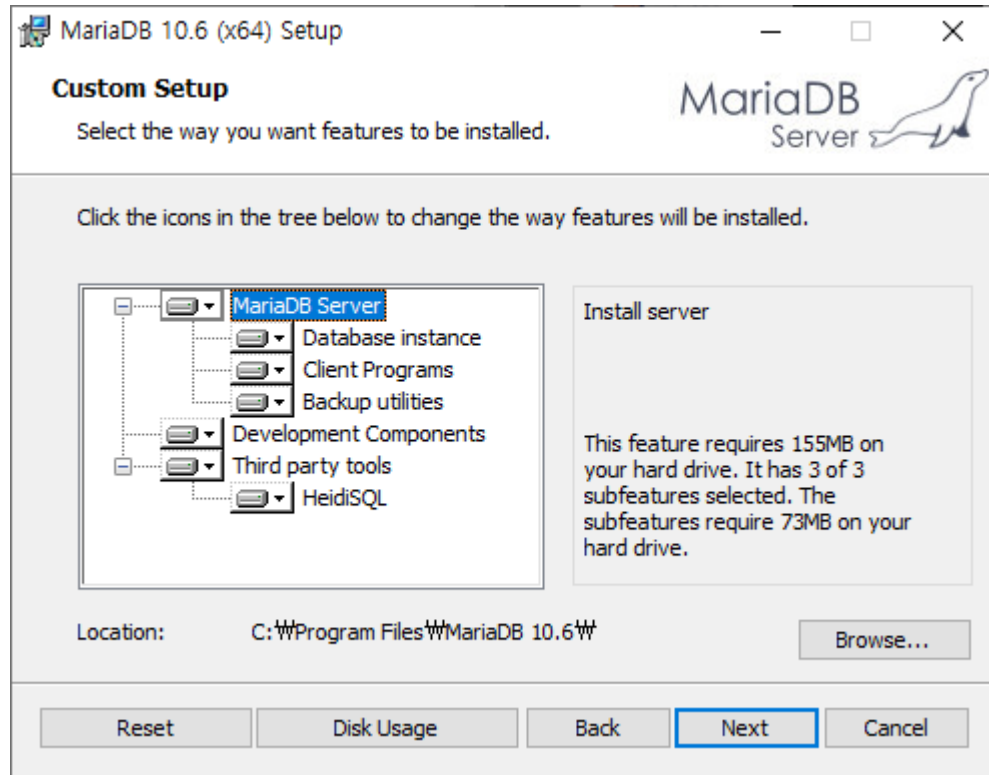
- msi 파일 실행



- 라이선스 동의

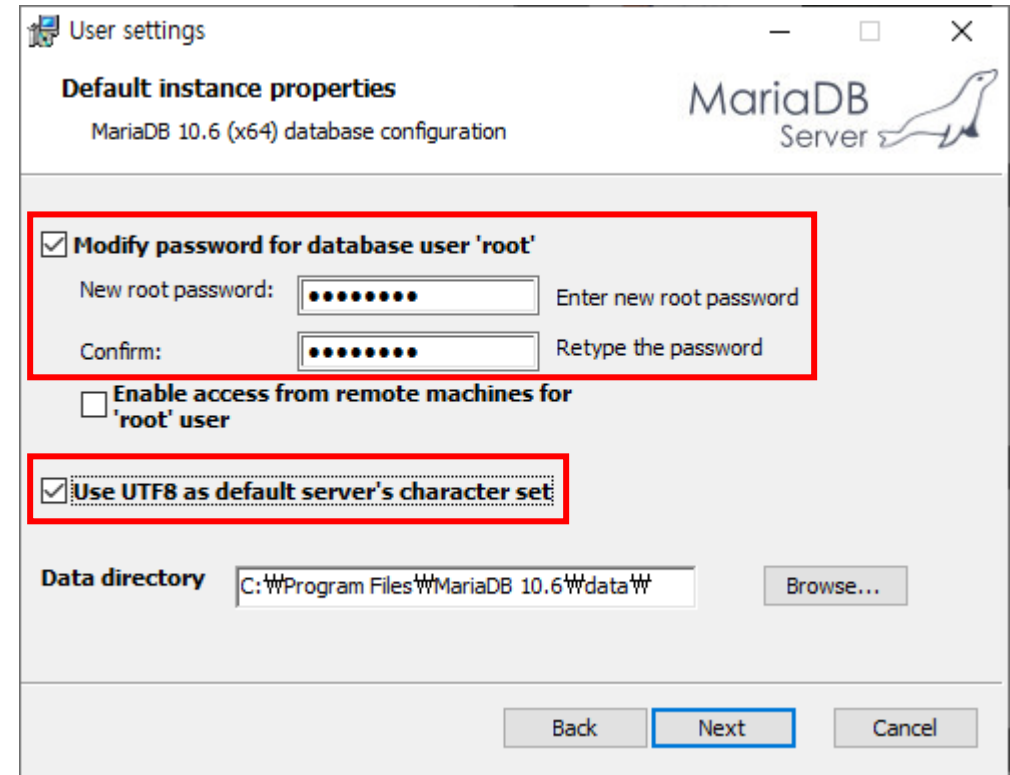


- 기본 경로에 설치

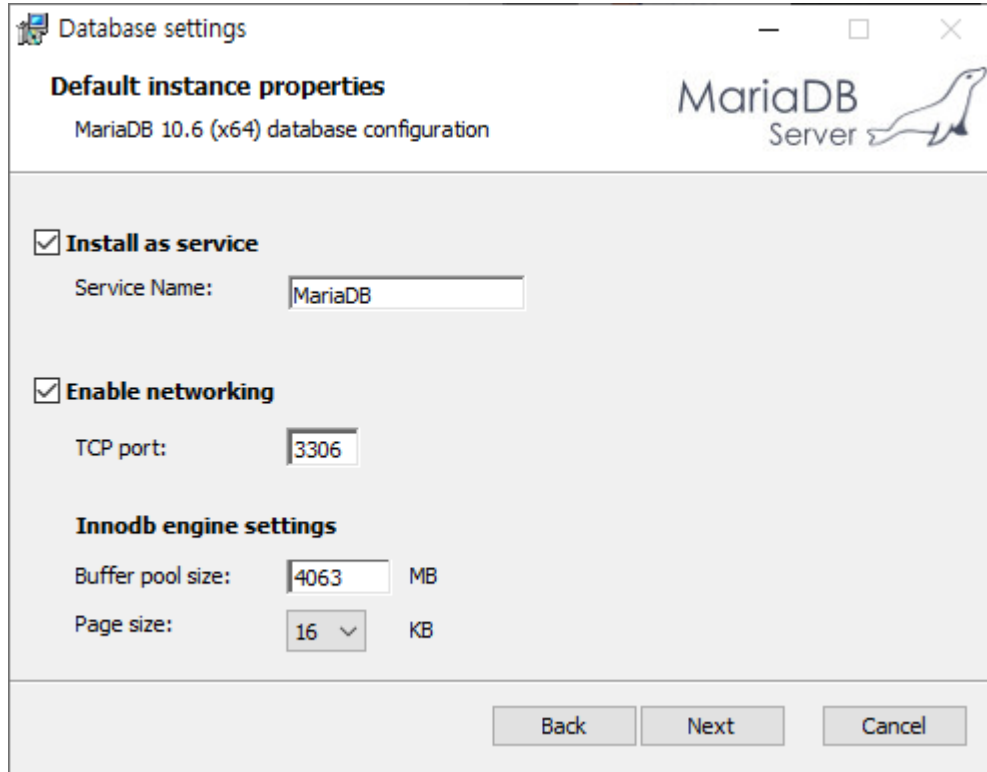


- 환경설정

1. root 계정 password 변경
2. 원격 시스템에서 root 계정 접근 설정
3. UTF-8을 기본 문자열셋으로 설정



- 데이터베이스 설정



Database settings

Default instance properties

MariaDB 10.6 (x64) database configuration

MariaDB Server

☒ **Install as service**

Service Name:

☒ **Enable networking**

TCP port:

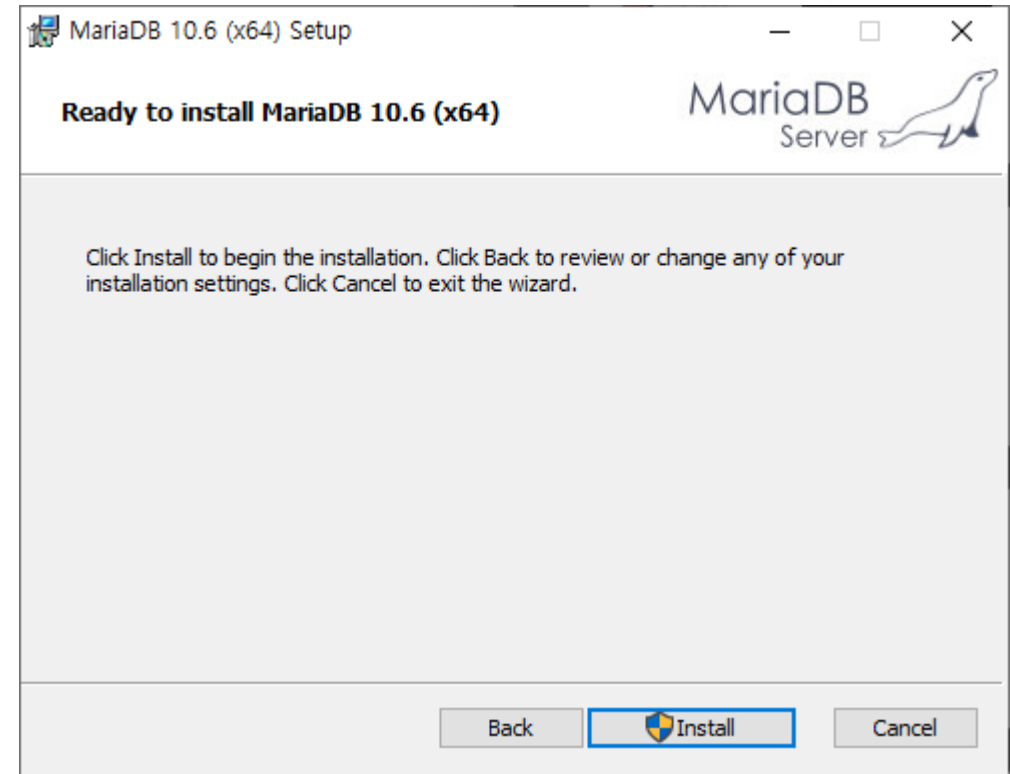
InnoDB engine settings

Buffer pool size: MB

Page size: KB

Back Next Cancel

- 설치



MariaDB 10.6 (x64) Setup

Ready to install MariaDB 10.6 (x64)

MariaDB Server

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Back Install Cancel

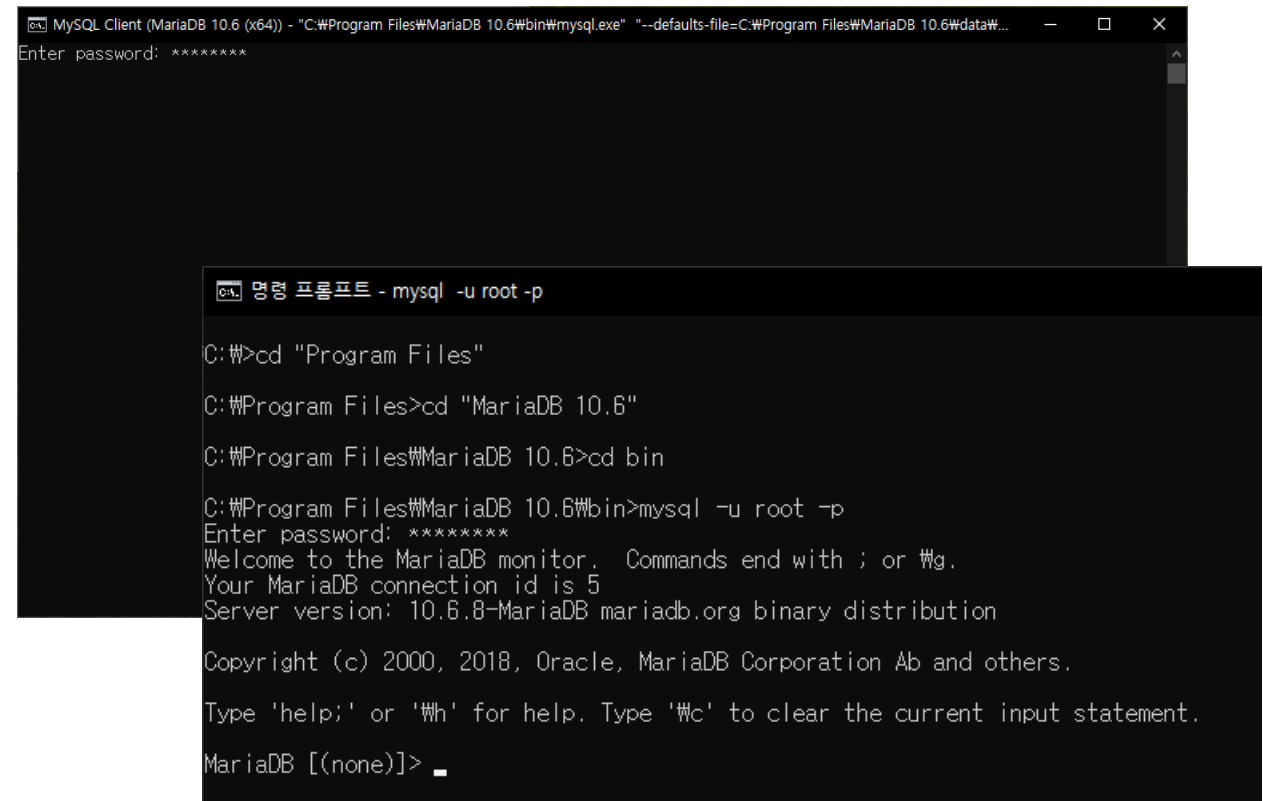
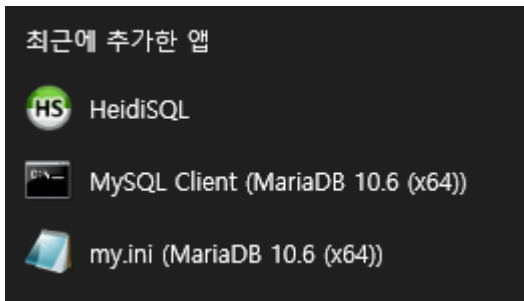
MariaDB 접속방법

- MySQL Client 실행

- 설치 시 설정한 비밀번호를 입력

cf) CMD에서 접속 방법

MariaDB설치 경로 -> bin 폴더에서 `mysql -u root -p`
비밀번호 입력 후 접속



- Database 확인
- SHOW DATABASES;

```
명령 프롬프트 - mysql -u root -p

C:\Program Files\MariaDB 10.6\bin>mysql -u root -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.6.8-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sys        |
+-----+
4 rows in set (0.000 sec)
```

- mariadb 종료
- exit;

```
선택 MySQL Client (MariaDB 10.6 (x64))

1 row in set (0.000 sec)

MariaDB [(none)]> exit;
Bye

C:\Program Files\MariaDB 10.6\bin>
```

SQL 1

SQL 1

SQL (Structured Query Language)

- RDBMS의 데이터를 관리하기 위해 사용되는 언어

| 명령어 종류 | 명령어 | 설명 |
|---|----------|------------------------------|
| 데이터 정의어 DDL (Data Definition Language) | CREATE | 데이터베이스 구조생성 (ex 테이블, 데이터베이스) |
| | ALTER | 데이터베이스 구조변경 |
| | DROP | 데이터베이스 구조삭제 |
| | RENAME | 데이터베이스 이름변경 |
| 데이터 조작어 DML (Data Manipulation Language) | SELECT | 데이터베이스에 있는 데이터를 조회하거나 검색 |
| | INSERT | 데이터베이스에 데이터를 입력 |
| | UPDATE | 데이터베이스에 데이터를 수정 |
| | DELETE | 데이터베이스에 데이터를 삭제 |
| 데이터 제어어 DCL (Data Control Language) | GRANT | 데이터베이스 접근 권한 생성 |
| | REVOKE | 데이터베이스 접근 권한 삭제 |
| 트랜잭션 제어어 TCL(Transaction Control Language) | COMMIT | 트랜잭션을 정상 처리 확정, 영구 저장 |
| | ROLLBACK | 트랜잭션을 취소, 변경 이전 상태로 롤백 |

DDL (Data Definition Language) & DCL (Data Control Language)

- 데이터베이스 생성
- CREATE DATABASE {DB_NAME} CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

```
MariaDB [(none)]> CREATE DATABASE study CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.009 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| study |
| sys |
+-----+
5 rows in set (0.000 sec)

MariaDB [(none)]> █
```

- 데이터베이스 사용
- use {DB_NAME};

- 사용자 생성

- CREATE USER {USER_NAME}@{HOST_NAME} IDENTIFIED BY '{PASSWORD}';



아이피 - 해당 아이피에서만 접속 가능

Localhost - 로컬에서만 접속가능

% - 외부에서 접속가능

```
MariaDB [(none)]> CREATE USER sosin@localhost IDENTIFIED BY 'password';  
Query OK, 0 rows affected (0.011 sec)
```

- 권한 부여

- GRANT ALL PRIVILEGES ON {DB_NAME}.* TO {USER_NAME}@{HOST_NAME};

```
MySQL Client (MariaDB 10.6 (x64)) - mysql -u root -p  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON study.* TO sosin@localhost;  
Query OK, 0 rows affected (0.013 sec)  
  
MariaDB [(none)]> _
```

- 권한 조회

- SHOW GRANTS FOR {USER_NAME}@{HOST_NAME}

- 계정 삭제

- DROP USER {USER_NAME}@{HOST_NAME}

- 권한 삭제

- REVOKE [권한1, 권한2 ...] ON {DB_NAME}.{TABLE_NAME} FROM {USER_NAME}

- 모든 권한 삭제

- REVOKE ALL ON *.* FROM {USER_NAME}@{HOST_NAME}

- 유저 및 권한 조회

- SELECT GRANTEE, PRIVILEGE_TYPE, IS_GRANTABLE FROM INFORMATION_SCHEMA.USER_PRIVILEGES;

- 테이블 생성

```
- CREATE TABLE {TABLE_NAME} (  
    column1 data_type,  
    column2 data_type,  
    column3 data_type,  
);
```

```
- CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

- 테이블 생성

```
- CREATE TABLE {TABLE_NAME} (  
    column1 data_type NOT NULL AUTO_INCREMENT,  
    column2 data_type,  
    column3 data_type,  
    PRIMARY KEY(column1)  
);
```

```
- CREATE TABLE Persons (  
    PersonID int NOT NULL AUTO_INCREMENT,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255),  
    PRIMARY KEY(PersonID)  
);
```

- 데이터 타입 (숫자)

| TYPE | 사용되는 바이트 | 최소값 (signed/unsigned) | 최대값 (signed/unsigned) |
|----------------|----------|-------------------------------|---|
| TINYINT | 1 | -128 0 | 127 255 |
| SMALLINT | 2 | -32768 0 | 32767 65535 |
| MEDIUMINT | 3 | -8388608 0 | 8388607 16777215 |
| INT or INTEGER | 4 | -2147483648 0 | 2147483647 4294967295 |
| BIGINT | 8 | -9223372036854775808 0 | 9223372036854775807 18446744073709551615 |
| FLOAT | 4 | -3.40E+45 (no unsigned) | 3.40E+45 (no unsigned) |
| DUBLE or REAL | 8 | -1.7976E+320 (no unsigned) | 1.7976E+320 (no unsigned) |

- 데이터 타입 (문자)

| TYPE | 표현범위 | 설명 |
|---------------|---------------------|----------------|
| CHAR(N) | 정확히 n (<=255) | 무조건 크기만큼 메모리사용 |
| VARCHAR(N) | 최대 n (<=65535) | 안에 데이터만큼 메모리사용 |
| TINYTEXT(N) | 최대 n (<=255) | 문자열 |
| TEXT(N) | 최대 n (<=65535) | 문자열 |
| MEDIUMTEXT(N) | 최대 n (<=16777215) | 문자열 |
| LONGTEXT(N) | 최대 n (<=4294967295) | 문자열 |

- 데이터 타입 (바이너리)

| TYPE | 표현범위 | 예제 |
|---------------|----------------------------|------|
| TINYBLOB(N) | 최대 n (≤ 255) | 바이너리 |
| BLOB(N) | 최대 n (≤ 65535) | 바이너리 |
| MEDIUMBLOB(N) | 최대 n (≤ 16777215) | 바이너리 |
| LOB(N) | 최대 n (≤ 4294967295) | 바이너리 |

- 데이터 타입 (날짜)

| TYPE | 표현범위 | 예제 |
|-----------|--|--|
| DATE | 1000-01-01 ~ 9999-12-31 | YYYY-MM-DD |
| DATETIME | 1000-01-01 00:00:00 ~ 9999-12-31 23:59:59 | YYYY-MM-DD HH:MM:SS 문자로 저장 |
| TIMESTAMP | 1970-01-01 ~ 2037-01-01 임의시간 |  YYYY-MM-DD HH:MM:SS 숫자로 저장, 타임존 |
| TIME | -838:59:59 ~ 838:59:59 | |
| YEAR | 901~2155 | |

- 테이블 삭제
- DROP TABLE {TABLE_NAME};

- 테이블 초기화
- TRUNCATE TABLE {TABLE_NAME};

- 테이블 수정
- ALTER TABLE {TABLE_NAME}

ADD column_name datatype;

- ALTER TABLE Customers

ADD Email varchar(255);

실습

Q. 아래 테이블을 두 개 만들어주세요

테이블명: Students

StudentID - 학번 (빈값 허용안함, 자동증가)

Name - 이름

Age - 나이

Address - 주소

테이블명: Grades

StudentID - 학번

Math - 수학점수

English - 영어점수

Science - 과학점수

SQL 2

SQL 2

DML (Data Manipulation Language)

- 데이터를 조작하는 언어

- 데이터 추가

- INSERT INTO {TABLE_NAME} (column1, column2, ...)
VALUES (value1, value2, ...);

- 데이터 삭제

- DELETE FROM {TABLE_NAME} WHERE condition;

- 데이터 수정

- UPDATE {TABLE_NAME} SET column1=value1, column2=value2, ...
WHERE condition;

실습

Q. 각 테이블에 데이터를 입력해주세요.

테이블명: Students

Name: 홍길동, Age: 30, Address: 인천

Name: 이연걸, Age: 60, Address: 서울

Name: 이몽룡, Age: 42, Address: 대전

Name: 성춘향, Age: 27, Address: 경기

테이블명: Grades

StudentID: 홍길동 StudentID, Math: 90, English: 80, Science: 50

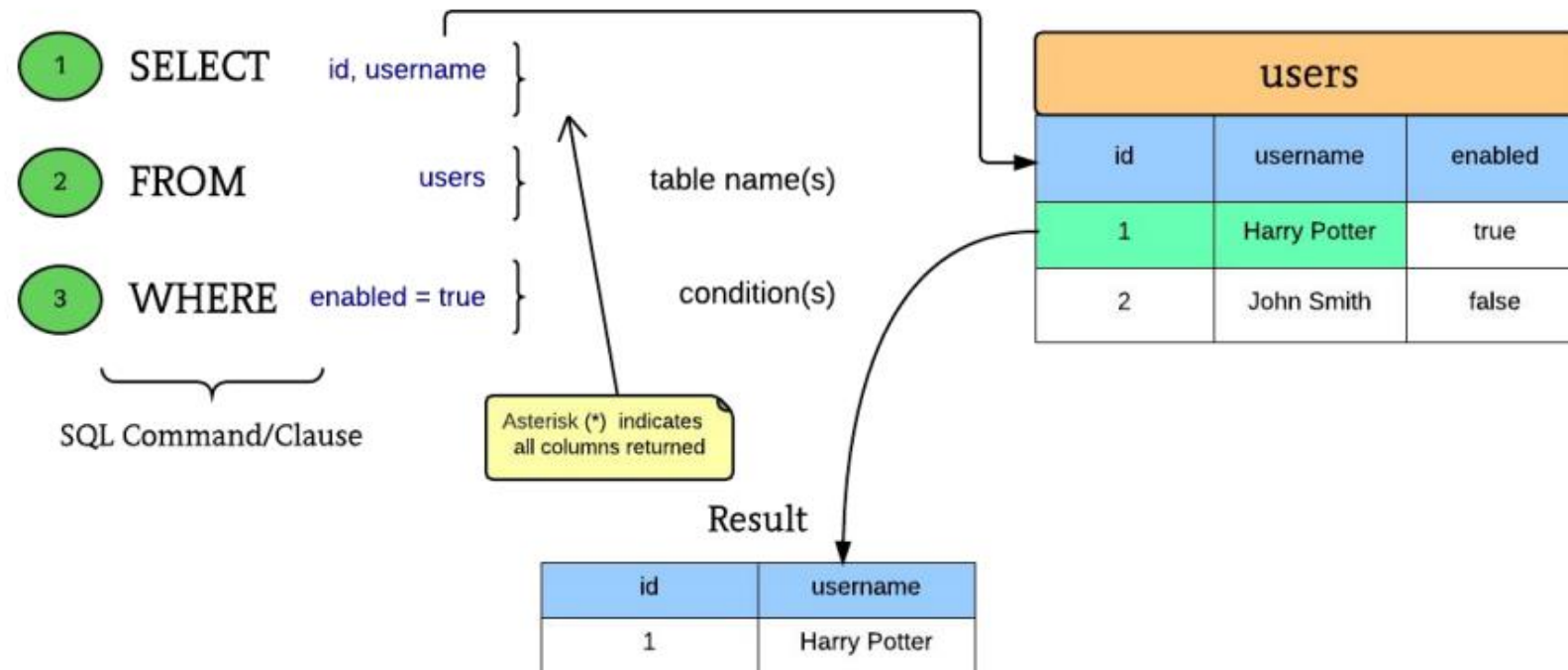
StudentID: 이연걸 StudentID, Math: 69, English: 76, Science: 65

StudentID: 이몽룡 StudentID, Math: 98, English: 87, Science: 97

StudentID: 성춘향 StudentID, Math: 87, English: 67, Science: 79

- 데이터 검색
- SELECT

Anatomy of the SELECT Statement



- 칼럼지정 조회
- SELECT column1, column2, ...
FROM {TABLE_NAME};

- *(asterisk)는 전체 칼럼
- SELECT * FROM {TABLE_NAME};

- 지정칼럼의 유일한 값 조회
- SELECT DISTINCT columns1, columns2, ...
FROM {TABLE_NAME};

Q. 고객(Customers)의 이름과 국가를 조회

Table: Customers

Q. 고객(Customers)의 정보 전체 조회

Table: Customers

Q. 고객(Customers)의 국가(Country) 목록 조회 (중복 x)

Table: Customers

- WHERE: 조건에 맞는 데이터 조회
- SELECT column1, column2, ...
FROM {TABLE_NAME}
WHERE condition;

Q. 국가(Country)가 France인 고객(Customers) 조회

Table: Customers

- pattern에 맞는 데이터 조회
- SELECT column1, column2, ...
FROM {TABLE_NAME}
WHERE columnN LIKE pattern;

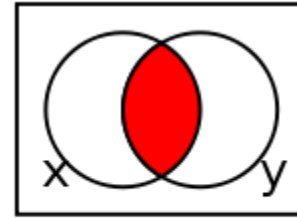
고객(Customers) 테이블에서

Q. 이름(ContactName) 이 'Mar'로 시작하는 직원 조회

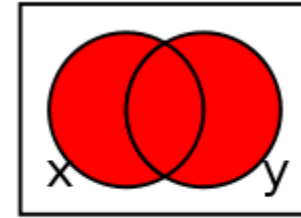
Q. 이름(ContactName) 이 'et'로 끝나는 직원 조회

Table: Customers

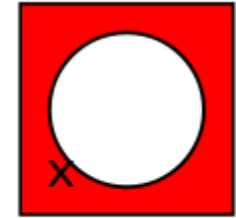
- AND, OR, NOT



AND



OR



NOT

- AND 조건

```
- SELECT column1, column2, ...  
FROM {TABLE_NAME}  
WHERE condition1 AND condition2;
```

- OR 조건

```
- SELECT column1, column2, ...  
FROM {TABLE_NAME}  
WHERE condition1 OR condition2;
```

- NOT 조건

```
- SELECT column1, column2, ...  
FROM {TABLE_NAME}  
WHERE NOT condition;
```

Q. 국가(Country)가 France이고 이름(ContactName)이 'La'로 시작하는 고객(Customers) 조회

Table: Customers

Q. 국가(Country)가 France이거나 이름(ContactName)이 'La'로 시작하는 고객(Customers) 조회

Table: Customers

Q. 국가(Country)가 France가 아닌 고객(Customers) 조회

Table: Customers

- IN, BETWEEN

- IN

```
- SELECT column1, column2, ...  
FROM {TABLE_NAME}  
WHERE column_name IN (value1, value2, ...);
```

- BETWEEN

```
- SELECT column1, column2, ...  
FROM {TABLE_NAME}  
WHERE column_name BETWEEN value1 AND value2;
```

Q. 국가(Country)가 France 혹은 Spain에 사는
고객(Customers) 조회

Table: Customers

Q. 가격(Price)이 15에서 20사이인 상품(Products) 조회

Table: Products

- NULL

- IS NULL

```
- SELECT column1, column2, ...  
FROM {TABLE_NAME}  
WHERE column_name IS NULL;
```

- IS NOT NULL

```
- SELECT column1, column2, ...  
FROM {TABLE_NAME}  
WHERE column_name IS NOT NULL;
```

Q. 우편번호(PostalCode)가 NULL인 고객(Customers) 조회

Table: Customers

Q. 우편번호(PostalCode)가 NULL이 아닌 고객 (Customers) 조회

Table: Customers

- 데이터 정렬과 연산

- ORDER BY 지정 컬럼을 기준으로 오름/내림차순 조회
- SELECT column1, column2, ...
FROM {TABLE_NAME}
ORDER BY column1 ASCIDESC, column2 ASCIDESC, ...;

Q. 고객이름(CustomerName) 오름차순 조회

Table: Customers

Q. 상품가격(Price) 내림차순 조회

Table: Products

- LIMIT
- SELECT column1, column2, ...
FROM {TABLE_NAME}
LIMIT number;

Q. 10명만 조회

Table: Customers

- OFFSET
- SELECT column1, column2, ...
FROM {TABLE_NAME}
LIMIT number OFFSET number;

Q. 그 다음 10명 조회

Table: Customers

- CASE

- 조건에 따라 정해진 값 반환

```
- SELECT *,
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END
FROM {TABLE_NAME};
```

- AS

- 특정 계산된 결과 혹은 칼럼의 이름을 바꿈
- 해당 column 이름은 조건문 등에서 재사용 가능

```
- SELECT *,
CASE
    ...
END AS 'Level'
FROM {TABLE_NAME};
```

Q. 상품가격(Price)이 30 미만이면 '저', 30~50이면 '중', 50 초과는 '고'로 조회

Table: Products

Q. 위 조회한 CASE의 이름을 'Level'으로 바꿔주세요

Table: Products

- 데이터 집계

- COUNT (개수 조회)
- SELECT COUNT(column_name)
FROM {TABLE_NAME}
WHERE condition;

Q. 국가(Country)가 France에 사는
고객(Customers)수 조회

Table: Customers

- AVG (평균 값 계산)
- SELECT AVG(column_name)
FROM {TABLE_NAME}
WHERE condition;

Q. 전체 상품(Products)의 평균 가격(Price) 계산

Table: Products

- SUM (합 계산)
- SELECT SUM(column_name)
FROM {TABLE_NAME}
WHERE condition;

Q. 주문상품 수량(Quantity) 합계 계산

Table: OrderDetails

- MIN (최소값 조회)
- SELECT MIN(column_name)
FROM {TABLE_NAME}
WHERE condition;

Q. 가격(Price) 최소값 조회

Table: Products

- MAX (최대값 조회)
- SELECT MAX(column_name)
FROM {TABLE_NAME}
WHERE condition;

Q. 가격(Price) 최대값 조회

Table: Products

- GROUP BY (열 기준 데이터 그룹화)
- SELECT column_name(s)
FROM {TABLE_NAME}
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);

Q. 국가(Country)별 고객수 조회 (고객수 기준 오름차순)

Table: Customers

Q. 국가(Country)별, 도시(City)별 고객수 조회 (고객수 기준 내림차순)

Table: Customers

- HAVING (집계데이터를 기준으로)
- SELECT column_name(s)
FROM {TABLE_NAME}
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);

Q. 국가(Country)별 고객수를 조회하고 그 중 5명 초과인 국가만 조회 (고객수 기준 내림차순)

Table: Customers

- 서브쿼리

- 쿼리속에 쿼리

```
- SELECT column_name(s),  
FROM (  
    SELECT column_name(s)  
    FROM {TABLE_NAME}  
    WHERE condition  
)  
WHERE condition;
```

```
- SELECT column_name(s), (  
    SELECT SUM(column_name(s))  
    FROM {TABLE_NAME}  
    WHERE condition  
    ) AS name  
FROM {TABLE_NAME}  
WHERE condition;
```

- exists

```
- SELECT column_name(s),  
FROM {TABLE_NAME}  
WHERE exists (  
    SELECT column_name(s)  
    FROM {TABLE_NAME}  
    WHERE condition  
);
```

- 기타

- 주석

--조회문

SELECT * FROM Customers;

- Alias

- SELECT CustomerName AS Cname
FROM Customers;

실습

Q. 직원(Employees)중 이름(LastName)이 'King'인 직원의 이름과 생일(BirthDate)과 노트(Notes)를 조회해주세요.

Table: Employees

Q. 상품(Products)중 상품명(ProductName)이 'C'로 시작하고 가격(Price)이 20보다 큰 상품의 상품명과 가격을 가격이 비싼순으로 조회해주세요.

Table: Products

Q. 상품(Products)의 카테고리아이디(CategoryID) 별로 상품가격의 합, 가장 비싼 상품 가격, 가장 저렴한 상품 가격을 구하세요.

Table: Products

실습

Q. 상품(Products)의 카테고리아이디(CategoryID) 별로 상품개수와 상품개수가 10개가 넘을경우 많음 아니면 적음이 표시되어있는 칼럼을 하나 추가하고 상품수가 많은 순서대로 조회해주세요.

Table: Products

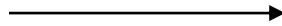
Q. 고객(Customers)의 국가(Country)별 고객수와 백분위 ($\text{국가별고객수} / \text{전체고객수} * 100$)을 구하세요

Table: Customers

- 데이터 조인

| 이름 | 전화번호 |
|-----|---------------|
| 홍길동 | 010-3242-5931 |
| 이동진 | 010-3943-1992 |
| 박철우 | 010-6123-4453 |

| 이름 | 구입일 | 상품 | 수량 |
|-----|------------|----|----|
| 홍길동 | 2022-06-11 | 우유 | 1 |
| 홍길동 | 2022-06-13 | 식빵 | 2 |
| 이동진 | 2022-06-20 | 치즈 | 1 |
| 박철우 | 2022-06-10 | 소금 | 1 |
| 박철우 | 2022-06-12 | 우유 | 3 |



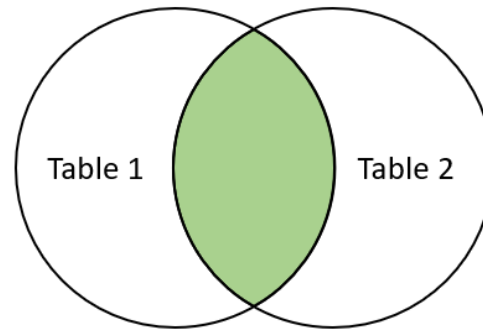
| 이름 | 전화번호 | 구입일 | 상품 | 수량 |
|-----|---------------|------------|----|----|
| 홍길동 | 010-3242-5931 | 2022-06-11 | 우유 | 1 |
| 홍길동 | 010-3242-5931 | 2022-06-13 | 식빵 | 2 |
| 이동진 | 010-3943-1992 | 2022-06-20 | 치즈 | 1 |
| 박철우 | 010-6123-4453 | 2022-06-10 | 소금 | 1 |
| 박철우 | 010-6123-4453 | 2022-06-12 | 우유 | 3 |

- INNER JOIN

```
- SELECT column_name(s),  
FROM table_1  
INNER JOIN table_2  
ON table_1.column_name = table_2.column_name;
```

Q. 주문이력이 있는 고객명(CustomerName)과 주문일(OrderDate)를 조회해주세요

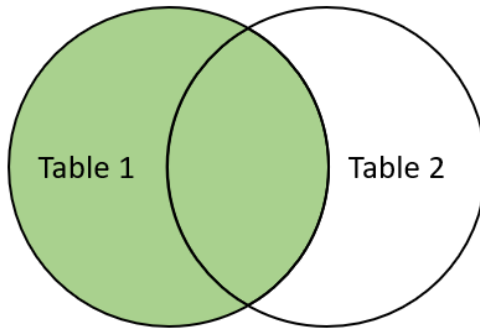
Table: Customers, Orders



INNER JOIN

– LEFT JOIN

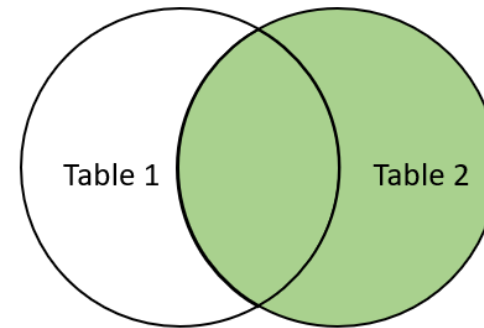
```
– SELECT column_name(s),  
FROM table_1  
LEFT JOIN table_2  
ON table_1.column_name = table_2.column_name;
```



LEFT JOIN

– RIGHT JOIN

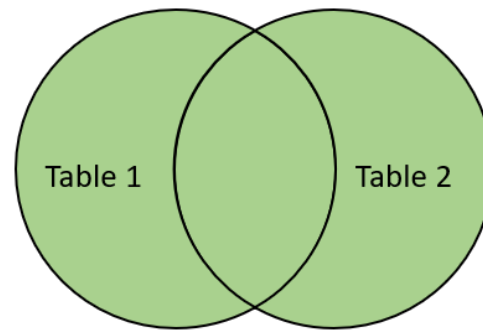
```
– SELECT column_name(s),  
FROM table_1  
RIGHT JOIN table_2  
ON table_1.column_name = table_2.column_name;
```



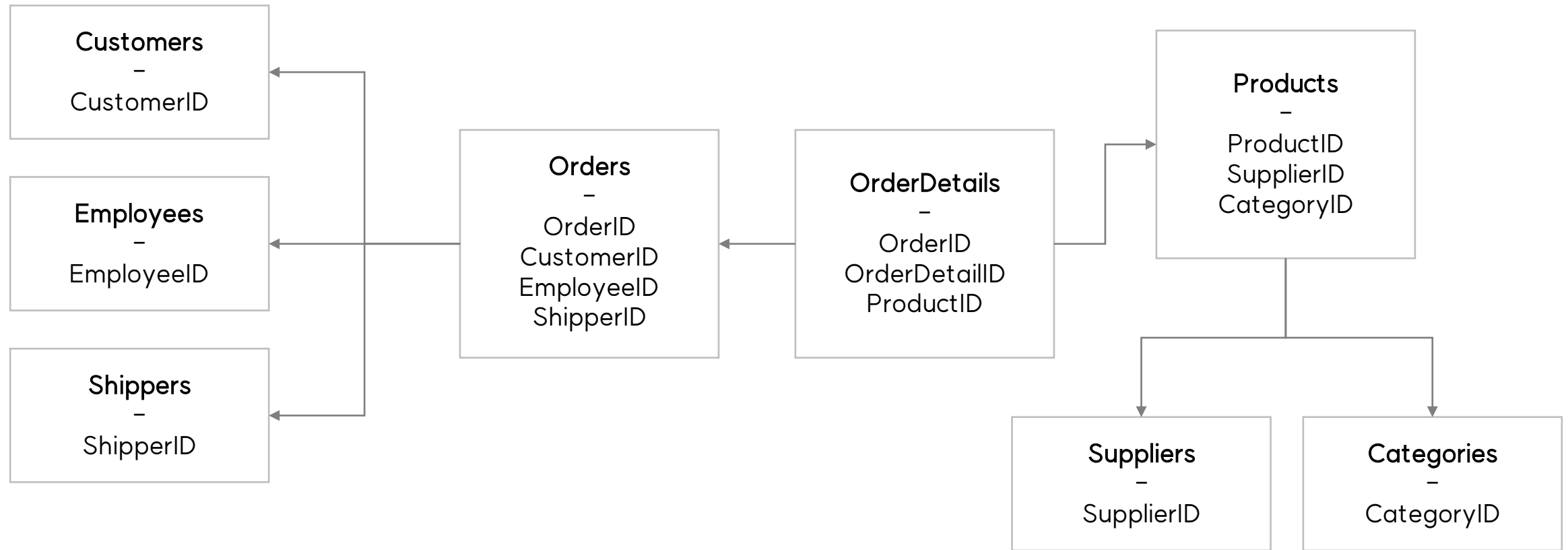
RIGHT JOIN

– OUTER JOIN

```
– SELECT column_name(s),  
FROM table_1  
FULL OUTER JOIN table_2  
ON table_1.column_name = table_2.column_name;
```



- ERD



실습

Q. Tokyo에 위치한 공급자(Supplier)가 생산한 상품(Products) 목록 조회

Table: Suppliers, Products

Q. 분류(CategoryName)가 Seafood인 상품명(ProductName) 조회

Table: Categories, Products

Q. 공급자(Supplier)가 공급한 상품의 공급자 국가(Country) 별로 상품 카테고리(ProductName)의 상품건수와 평균가격 조회

Table: Suppliers, Products, Categories

Q. 주문별 주문자명(CustomerName), 직원명(LastName), 배송자명(ShipperName), 주문상세갯수

Table: Orders, OrderDetails, Customers, Employees, Shippers

실습

Q. 판매량(Quantity) 상위 TOP 3 공급자(supplier) 목록 조회

Table: Suppliers, Products, OrderDetails

Q. 상품분류(Category)별, 고객지역별(City) 판매량 많은 순 정렬

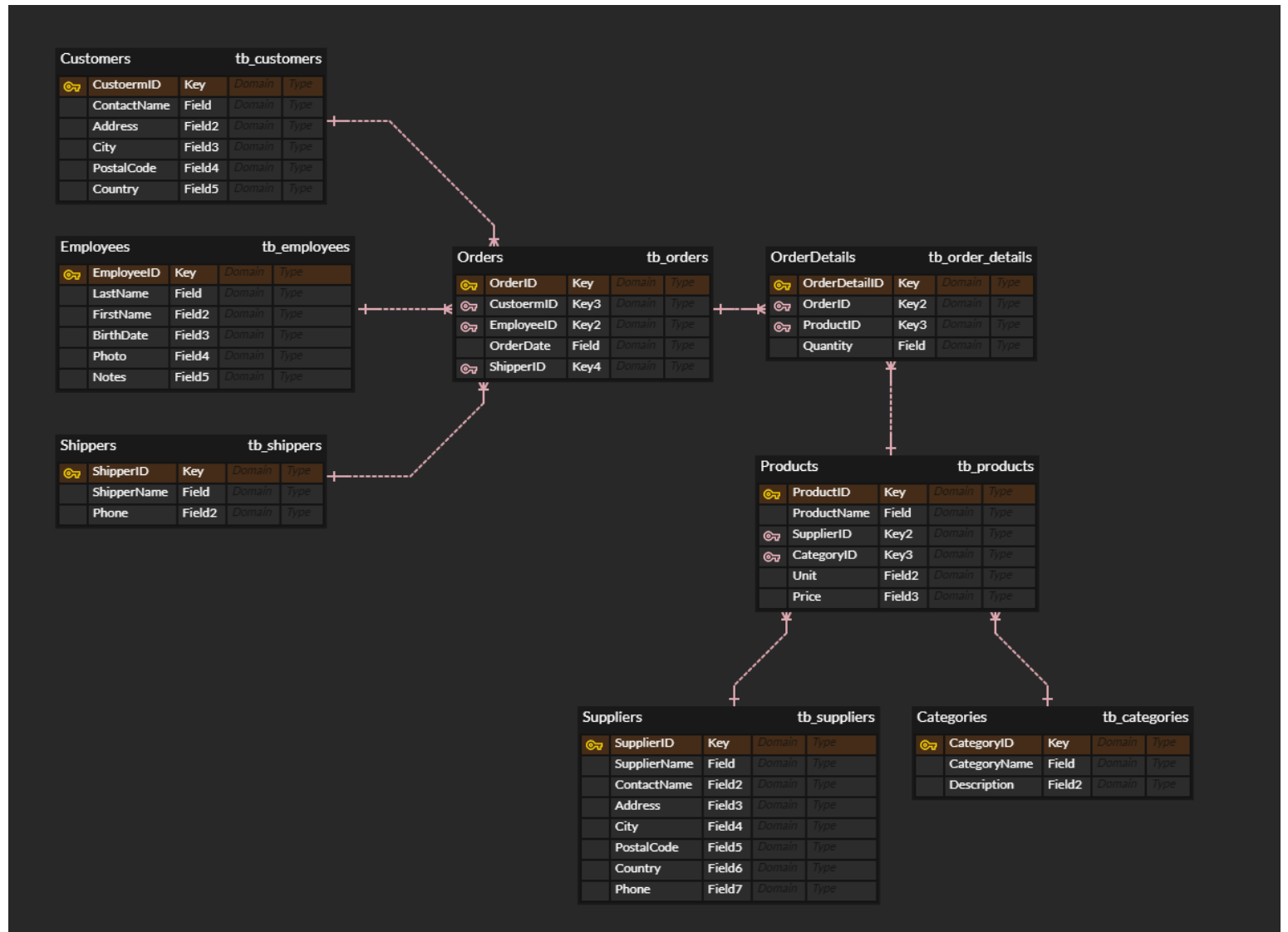
Table: Order, OrderDetails, Customers, Categories, Products

Q. 고객국가(Country)가 USA이고, 상품별 판매량(Quantity)의 합이 많은순으로 정렬

Table: Customers, Products, Orders, OrderDetails

ERD Cloud실습

- <https://www.erdcloud.com/>



Foreign Key

- 두 개의 테이블이 특정 칼럼을 기준으로 참조할 때 사용
- 부모가 되는 테이블의 칼럼은 **unique** 해야한다.

Foreign Key

| 이름 | 전화번호 |
|-----|---------------|
| 홍길동 | 010-3242-5931 |
| 이동진 | 010-3943-1992 |
| 박철우 | 010-6123-4453 |

| 이름 | 구입일 | 상품 | 수량 |
|-----|------------|----|----|
| 홍길동 | 2022-06-11 | 우유 | 1 |
| 홍길동 | 2022-06-13 | 식빵 | 2 |
| 이동진 | 2022-06-20 | 치즈 | 1 |
| 박철우 | 2022-06-10 | 소금 | 1 |
| 박철우 | 2022-06-12 | 우유 | 3 |

생성

- CREATE TABLE **table_name**
(
...
FOREIGN KEY (**column**) REFERENCES **table_name** (**column**)
);
- ALTER TABLE **table_name** ADD CONSTRAINT **key_name** FOREIGN KEY(**column**)
REFERENCES **table_name**(**column**);

삭제

- ALTER TABLE **table_name** DROP FOREIGN KEY **key_name**;

INDEX

- 인덱스는 큰 테이블에서 소량 데이터를 검색할 때 사용
- 테이블에서 자주 사용하는 속성에 사용 (조건문, 조인)
- SELECT에선 성능향상, INSERT, UPDATE, DELETE에선 속도 감소
- explain `table_name`

생성

- CREATE INDEX `index_name` ON `table_name`(`column`);

확인

- SHOW INDEX FROM `table_name`;

삭제

- DROP INDEX `index_name` ON `table_name`;

VIEW

- 실제 테이블을 기반으로 만드는 가상의 테이블
- VIEW를 통한 데이터 수정은 가능하나, 제한적이다
- 복잡한 SQL을 간단하게 표시할 수 있음
- 필요한 데이터만 사용자에게 보여줄 수 있음

생성

```
- CREATE VIEW view_name (column1, column2, ...)
AS
SELECT column1, column2, ...
FROM table_name;
```

삭제

- DROP VIEW view_name;

실습

Q. Tokyo에 위치한 공급자(Supplier)가 생산한 상품 목록에 대한 VIEW 생성

Table: Suppliers, Products

Q. 분류(CategoryName)가 Seafood인 상품명(ProductName)에 대한 VIEW 생성

Table: Categories, Products