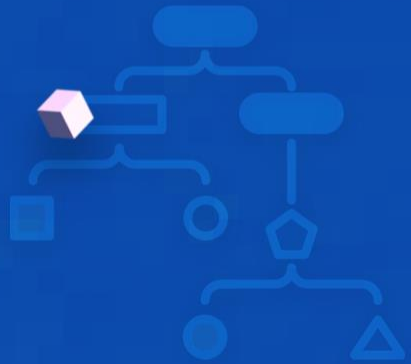
강원혁신플랫폼

리눅스프로그래밍

관계형 데이터베이스와 MariaDB



데이터를 테이블 형태로 구성하고, 테이블 간의 관계를 통해 데이터를 구성하는 데이터베이스로, 테이블, 열(Column), 행(Row)의 형태로 데이터가 작성되며, 각 테이블은 고유한 식별자(Primary Key)에 의해 구분되는 데이터베이스 유형을 무엇이라고 하나요?

관계형 데이터베이스 (relational database)

관계형 데이터베이스에서 데이터베이스의 구조와 구성 요소에 대한 전반적인 정의를 나타내며, 테이블, 열(Column), 제약 조건(Constraints), 관계 등을 정의하는데 사용하여 데이터베이스의 구조와 데이터의 유형을 결정하고 데이터의 일관성과 정확성을 보장하기 위해 사용하는 것을 무엇이라고 하나요?

스키마(Schema)

학습목표

학습 내용

1. 관계형 데이터베이스
2. MariaDB

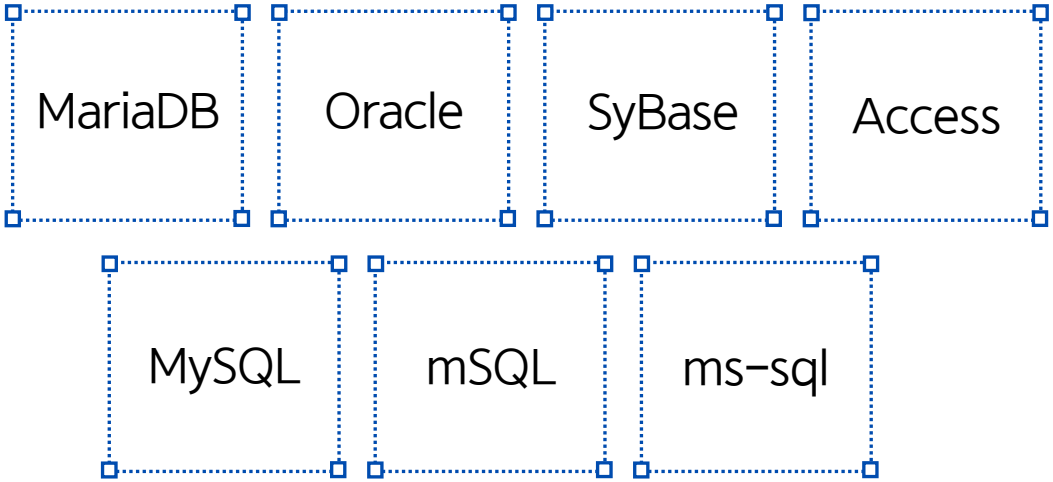
학습 목표

- 관계형 데이터베이스에 대해 설명할 수 있다.
- MariaDB scott/tiger 계정을 생성할 수 있다.

관계형 데이터베이스

관계형 데이터베이스 특징

- 정보의 관리 능력이 더 중요한 요소
- E. F. Codd가 개발한 관계형 데이터베이스 (relational database)



관계형 데이터베이스 특징

01 • 모든 정보는 테이블로 표현

02 • 물리적 저장장치에 독립적인 데이터를 위해 논리적 독립성

03 • 질의를 할 경우 SQL과 같은 고급 언어 제공

04 • 선택이나 조인(join) 같은 관계 대수 연산과 집합 연산을 제공

05 • 원 데이터 저장 구조와 다른 관점에서 처리 할 수 있도록 뷰(view) 제공

06 • 자료의 무결성, 권한, 트랜잭션 처리, 복구 등의 기능 제공

관계형 데이터베이스 테이블

특징

- RDB는 관계(Relation) 또는 테이블(Table)로 구성
- 관계는 열과 행으로 이루어진 이차원의 테이블
- 각 테이블은 고유한 이름을 가짐
- 관계는 속성(Attribute) 또는 필드(Field)로 구성
- 관계는 속성으로 이루어진 튜플(Tuple) 또는 레코드(Record)의 집합(Set)

관계형 데이터베이스 테이블

원소수

한 관계에 존재하는
튜플의 수

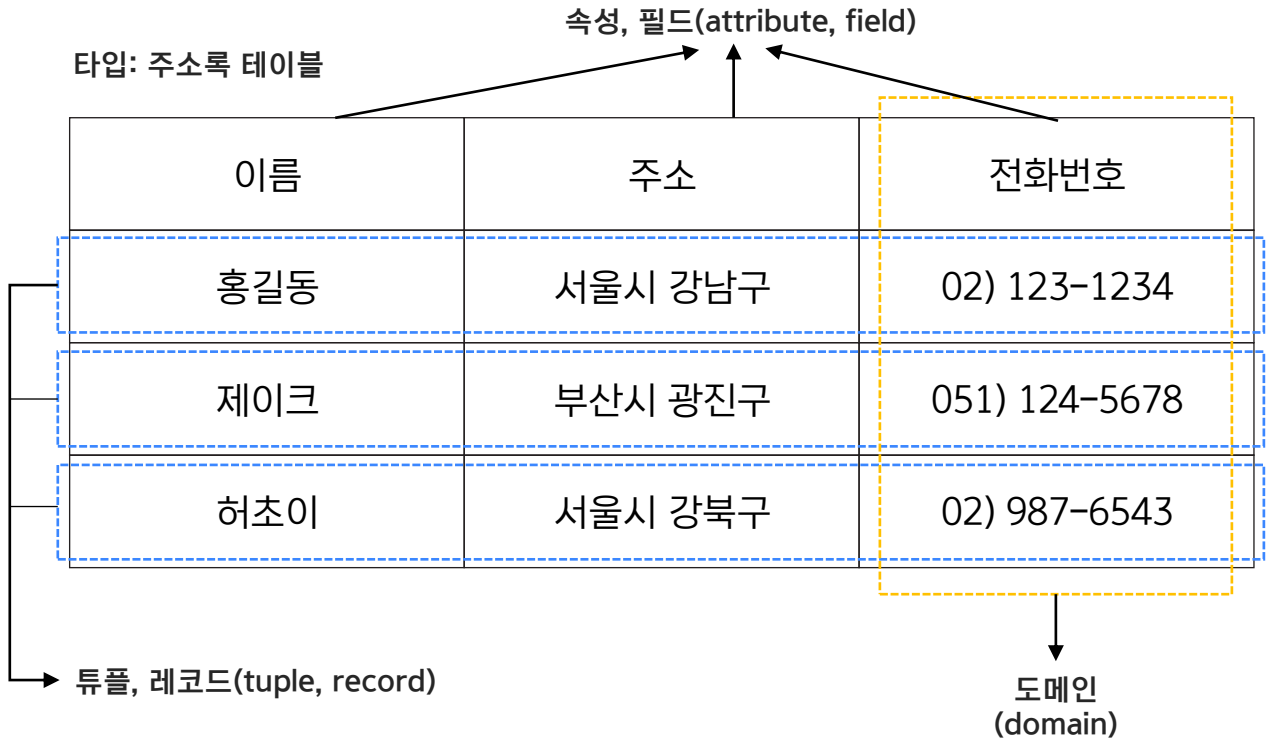
차수

한 관계에 있는
속성의 수

**영역
(Domain)**

한 관계에 있는
특정 속성이 가질 수 있는
같은 데이터 형식을 지닌
값(Value)의 집합

관계형 데이터베이스 테이블



SQL(Standard Query Language)

- DDL(Data Definition Language)

SQL(Standard Query Language)

- DML(Data Manipulation Language)

SQL(Standard Query Language)

구분	내용
SELECT	테이블에서 레코드를 가져올 때 사용
INSERT	테이블에 새로운 레코드를 추가
DELETE	테이블에서 레코드를 삭제
UPDATE	테이블에서 레코드의 임의 필드를 갱신
CREATE	새로운 테이블을 생성
DROP	테이블을 제거
ALTER	테이블을 수정(필드 추가 및 변경)

MariaDB

MariaDB 설치


```
$ sudo apt install mariadb-server mariadb-client

$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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)]>
```

MariaDB 내부 스키마 확인

스키마(Schema)

관계형 데이터베이스에서 스키마(Schema)는
데이터베이스의 구조와 구성 요소에 대한
전반적인 정의를 나타냄

MariaDB 내부 스키마 확인

- 용도
  - 스키마는 테이블, 열(Column), 제약 조건(Constraints), 관계 등을 정의하는데 사용
  - 데이터베이스의 구조와 데이터의 유형을 결정하고 데이터의 일관성과 정확성을 보장하기 위해 사용

MariaDB 내부 스키마 확인


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

MariaDB 프로세스 관리


```
$ mariadb --version
mariadb  Ver 15.1 Distrib 10.6.12-MariaDB, for debian-linux-gnu (x86_64) using
EditLine wrapper
```

MariaDB 프로세스 종료

```
$ sudo /etc/init.d/mariadb stop
Stopping mariadb (via systemctl): mariadb.service.
```

MariaDB 프로세스 시작

```
$ sudo /etc/init.d/mariadb start
Starting mariadb (via systemctl): mariadb.service.
```

MariaDB 프로세스 관리

MariaDB 프로세스 상태 확인

```
$ sudo /etc/init.d/mariadb status
● mariadb.service - MariaDB 10.6.12 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-07-14 09:42:35 KST; 11min ago
```

MariaDB mysql.user 확인


```
$ sudo mariadb
ERROR 2002 (HY000): Can't connect to local server through socket
'/run/mysqld/mysqld.sock' (2)
$ sudo /etc/init.d/mariadb start
Starting mariadb (via systemctl): mariadb.service.
$ sudo mariadb
MariaDB [(none)]> select user from mysql.user;
```

MariaDB mysql.user 확인


```
+-------------+
| User        |
+-------------+
| mariadb.sys |
| mysql       |
| root        |
+-------------+
3 rows in set (0.001 sec)
```

MariaDB scott/tiger 계정 생성


```
MariaDB [(none)]> create user scott@localhost identified by 'tiger';
Query OK, 0 rows affected (0.012 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
```

MariaDB scott/tiger 계정 생성


```
MariaDB [(none)]> select user from mysql.user;
+-------------+
| User        |
+-------------+
| mariadb.sys |
| mysql       |
| root        |
| scott       |
+-------------+
4 rows in set (0.000 sec)
MariaDB [(none)]> quit
Bye
```

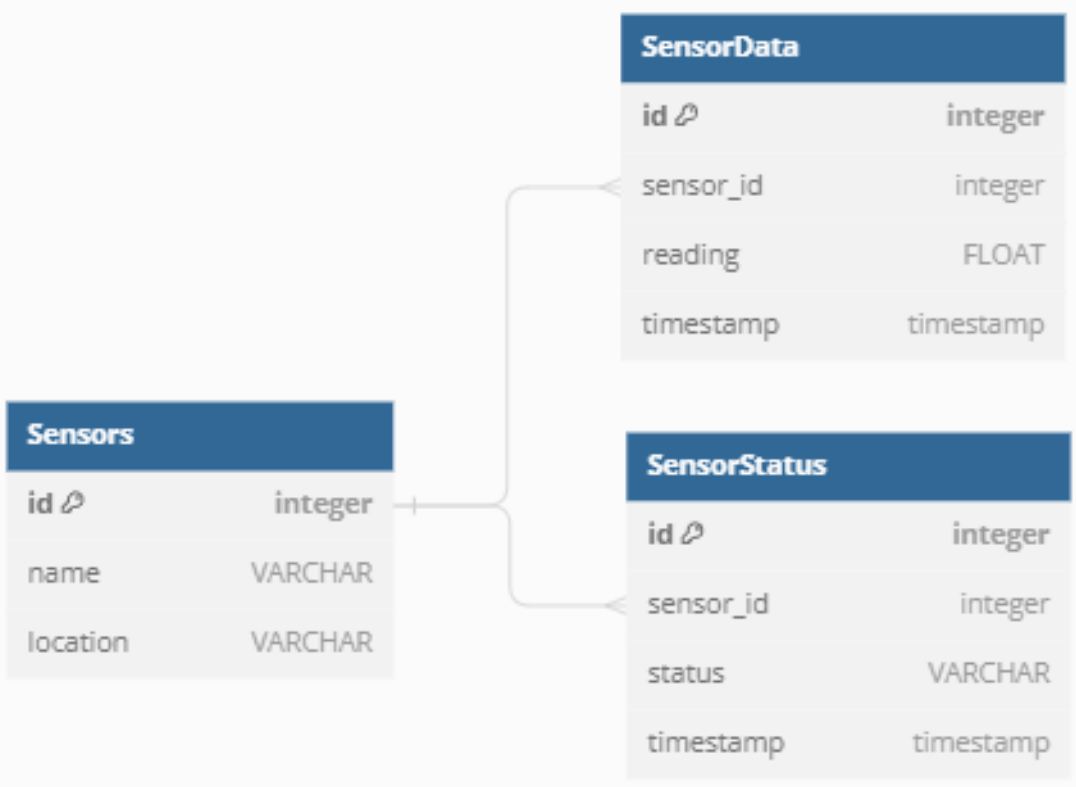
MariaDB scott/tiger DB 생성, grant


```
$ sudo mariadb -u root

MariaDB [(none)]> CREATE DATABASE mydb;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> GRANT ALL ON mydb.* TO 'scott'@'localhost';
Query OK, 0 rows affected (0.012 sec)

MariaDB [(none)]> quit
Bye

$
```

MariaDB scott/tiger 접속


```
$ sudo mariadb -u scott -ptiger

MariaDB [(none)]> use mydb;
MariaDB [mydb]> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
+----------------+
0 rows in set (0.001 sec)
```

MariaDB scott/tiger ERD



MariaDB scott/tiger Drop Table


```
$ sudo mariadb -u scott -ptiger

MariaDB [mydb]> DROP TABLE IF EXISTS SensorStatus;
Query OK, 0 rows affected (0.018 sec)

MariaDB [mydb]> DROP TABLE IF EXISTS SensorData;
Query OK, 0 rows affected (0.018 sec)

MariaDB [mydb]> DROP TABLE IF EXISTS Sensors;
Query OK, 0 rows affected (0.019 sec)
```

MariaDB scott/tiger Create Table


```
$ sudo mariadb -u scott -ptiger mydb

MariaDB [mydb]> CREATE TABLE Sensors (
    ->   id INT PRIMARY KEY AUTO_INCREMENT,
    ->   name VARCHAR(50),
    ->   location VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.020 sec)
```

MariaDB scott/tiger Create Table


```
MariaDB [mydb]> CREATE TABLE SensorData (
    ->   id INT PRIMARY KEY AUTO_INCREMENT,
    ->   sensor_id INT,
    ->   reading FLOAT,
    ->   timestamp DATETIME,
    ->   FOREIGN KEY (sensor_id) REFERENCES Sensors(id)
    -> );
Query OK, 0 rows affected (0.028 sec)
```

MariaDB scott/tiger Create Table


```
MariaDB [mydb]> CREATE TABLE SensorStatus (
    ->   id INT PRIMARY KEY AUTO_INCREMENT,
    ->   sensor_id INT,
    ->   status VARCHAR(20),
    ->   timestamp DATETIME,
    ->   FOREIGN KEY (sensor_id) REFERENCES Sensors(id)
    -> );
Query OK, 0 rows affected (0.023 sec)
```

MariaDB scott/tiger Create Table


```
MariaDB [mydb]> show tables;
+----------------+
| Tables_in_mydb |
+----------------+
| SensorData     |
| SensorStatus   |
| Sensors        |
+----------------+
3 rows in set (0.001 sec)
```

MariaDB scott/tiger Insert


```
MariaDB [mydb]> INSERT INTO Sensors (name, location)
    -> VALUES ('Sensor1', 'Location1'),
    ->        ('Sensor2', 'Location2'),
    ->        ('Sensor3', 'Location3');
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [mydb]> INSERT INTO SensorData (sensor_id, reading, timestamp)
    -> VALUES (1, 25.5, CURRENT_TIMESTAMP),
    ->        (2, 30.2, CURRENT_TIMESTAMP),
    ->        (1, 26.8, CURRENT_TIMESTAMP),
    ->        (3, 18.9, CURRENT_TIMESTAMP);
```

MariaDB scott/tiger Insert


```
Query OK, 4 rows affected (0.002 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [mydb]> INSERT INTO SensorStatus (sensor_id, status, timestamp)
    -> VALUES (1, 'Active', CURRENT_TIMESTAMP),
    ->        (2, 'Inactive', CURRENT_TIMESTAMP),
    ->        (3, 'Active', CURRENT_TIMESTAMP);
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [mydb]>
```

MariaDB scott/tiger Select


```
MariaDB [mydb]> select * from Sensors;
+----+---------+-----------+
| id | name    | location  |
+----+---------+-----------+
|  1 | Sensor1 | Location1 |
|  2 | Sensor2 | Location2 |
|  3 | Sensor3 | Location3 |
+----+---------+-----------+
3 rows in set (0.000 sec)
```

MariaDB scott/tiger Select


```
MariaDB [mydb]> select * from SensorData;
+----+-----------+---------+---------------------+
| id | sensor_id | reading | timestamp           |
+----+-----------+---------+---------------------+
|  1 |         1 |    25.5 | 2023-07-14 10:00:00 |
|  2 |         2 |    30.2 | 2023-07-14 11:30:00 |
|  3 |         1 |    26.8 | 2023-07-14 13:45:00 |
|  4 |         3 |    18.9 | 2023-07-14 15:15:00 |
+----+-----------+---------+---------------------+
4 rows in set (0.000 sec)
```

MariaDB scott/tiger Select, Join


```
MariaDB [mydb]> select b.id, a.name, b.sensor_id, b.reading, b.timestamp
from Sensors a, SensorData b WHERE a.id=b.sensor_id;
+----+---------+-----------+---------+---------------------+
| id | name    | sensor_id | reading | timestamp           |
+----+---------+-----------+---------+---------------------+
|  1 | Sensor1 |         1 |    25.5 | 2023-07-14 10:00:00 |
|  2 | Sensor2 |         2 |    30.2 | 2023-07-14 11:30:00 |
|  3 | Sensor1 |         1 |    26.8 | 2023-07-14 13:45:00 |
|  4 | Sensor3 |         3 |    18.9 | 2023-07-14 15:15:00 |
+----+---------+-----------+---------+---------------------+
4 rows in set (0.000 sec)
```

MariaDB scott/tiger Select, Join


```
MariaDB [mydb]> select b.id, b.sensor_id, b.reading, b.timestamp from
Sensors a, SensorData b WHERE a.id=b.sensor_id;
+----+-----------+---------+---------------------+
| id | sensor_id | reading | timestamp           |
+----+-----------+---------+---------------------+
|  1 |         1 |    25.5 | 2023-07-14 10:00:00 |
|  2 |         2 |    30.2 | 2023-07-14 11:30:00 |
|  3 |         1 |    26.8 | 2023-07-14 13:45:00 |
|  4 |         3 |    18.9 | 2023-07-14 15:15:00 |
+----+-----------+---------+---------------------+
4 rows in set (0.000 sec)
```

MariaDB scott/tiger Select, Join


```
MariaDB [mydb]> select b.id, a.name, b.sensor_id, b.reading, b.timestamp,
c.status from Sensors a, SensorData b, SensorStatus c WHERE
a.id=b.sensor_id and a.id=c.sensor_id;
+----+---------+-----------+---------+---------------------+----------+
| id | name    | sensor_id | reading | timestamp           | status   |
+----+---------+-----------+---------+---------------------+----------+
|  1 | Sensor1 |         1 |    25.5 | 2023-07-14 10:00:00 | Active   |
|  2 | Sensor2 |         2 |    30.2 | 2023-07-14 11:30:00 | Inactive |
|  3 | Sensor1 |         1 |    26.8 | 2023-07-14 13:45:00 | Active   |
|  4 | Sensor3 |         3 |    18.9 | 2023-07-14 15:15:00 | Active   |
+----+---------+-----------+---------+---------------------+----------+
4 rows in set (0.000 sec)
```

MariaDB scott/tiger Select, Join, Group By


```
MariaDB [mydb]> select count(sensor_id) from SensorData GROUP BY(sensor_id);
+------------------+
| count(sensor_id) |
+------------------+
|                2 |
|                1 |
|                1 |
+------------------+
3 rows in set (0.000 sec)
```

MariaDB scott/tiger Select, Join, Group By


```
MariaDB [mydb]> select count(b.sensor_id), a.name, b.sensor_id from Sensors
a, SensorData b, SensorStatus c  WHERE a.id=b.sensor_id and
a.id=c.sensor_id GROUP BY(b.sensor_id);
+--------------------+---------+-----------+
| count(b.sensor_id) | name    | sensor_id |
+--------------------+---------+-----------+
|                  2 | Sensor1 |         1 |
|                  1 | Sensor2 |         2 |
|                  1 | Sensor3 |         3 |
+--------------------+---------+-----------+
3 rows in set (0.001 sec)
MariaDB [mydb]>
```

요약정리

01 • 관계형 데이터베이스

02 • MariaDB