

데이터베이스

↳ er다이어그램 삽입, 후보키, 기본키, 외래키 삽입

사용 개발 환경

MySQL

- 무료 오픈 소스 관계형 데이터베이스 관리 시스템(RDBMS)
- 쿼리를 사용하여 데이터를 검색, 수정, 삭제할 수 있는 소프트웨어

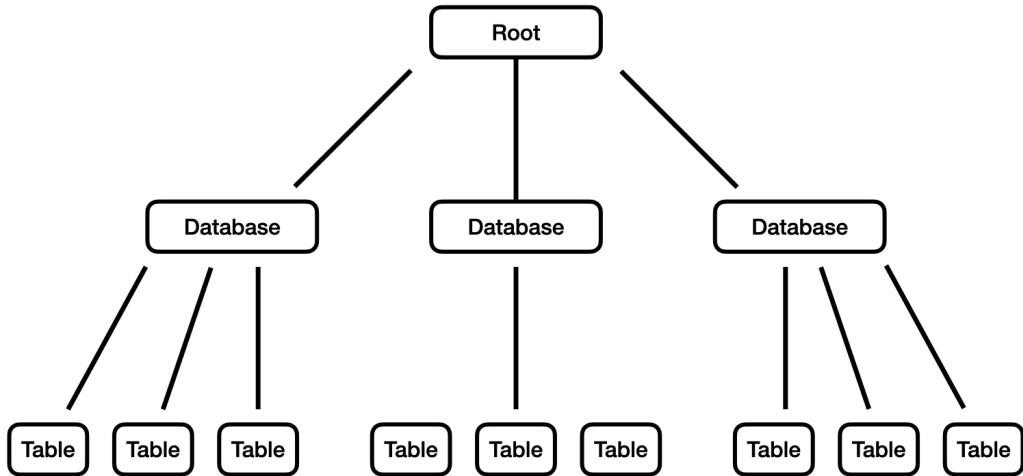


데이터 베이스 구성

- TrafficViolation 테이블은 위반 차량에 대한 기본정보를 포함한다
- CheckPoint는 단속위치에 대한 기본정보를 포함한다
- ResultOfJoin 테이블은 ios앱을 통해 유저에게 표시하는 테이블이 되며, 서버에서 조인을 통해 별도의 테이블을 생성하며 명칭은 임의로 지정한 것이다. CheckPoint의 부분참여, TrafficViolations의 전체 참여를 통해 구성된다

구조

- MySQL은 기본적으로 서버를 통해 Database들을 관리하게 된다
- 서버 구조는 계층적 구조(Tree) 구조를 이용하게 된다



- 처음 서버에 접속하게 되면 현재 위치는 root가 되며 각 Database를 탐색, Database 별로 table을 가지는 구조다

Database

- test

테이블 이름

- TrafficViolations

테이블 구조

```

car_number VARCHAR(255) NOT NULL PRIMARY KEY,
overspeed INT NOT NULL,
location VARCHAR(255) NOT NULL FOREIGN KEY CASCADE DELETE,
violation_time TIME NOT NULL, PRIMARY KEY
violation_date DATE NOT NULL, PRIMARY KEY
image_path VARCHAR(255)
  
```

테이블 생성시

```
CREATE TABLE TrafficViolations (
    car_number VARCHAR(255) NOT NULL,
    overspeed INT NOT NULL,
    location VARCHAR(255) NOT NULL,
    violation_time TIME NOT NULL,
    violation_date DATE NOT NULL,
    image_path VARCHAR(255),
    PRIMARY KEY (car_number, violation_time, violation_date),
    FOREIGN KEY (location) REFERENCES CheckPoint(location) ON DELETE CASCADE
);
```

- 후보기는 아래와 같았다
(차량번호, 위반시간, 위반 날짜)
(차량번호, 이미지 경로)
(차량번호, 위반시간, 위반 날짜, 이미지 경로)
(차량번호, 위반날짜, 이미지 경로)
(차량번호, 이미지 경로, 위반시간)
(차량번호, 이미지 경로, 위반날짜, 위반 시간, 위반 위치)
(이미지 경로, 위반날짜, 위반시간)
(이미지 경로)
(이미지 경로, 위반시간)
(이미지 경로, 위반날짜)
(이미지 경로, 위반 위치)
(차량번호, 이미지 경로, 위반 날짜)
(차량번호, 이미지 경로, 위반 시간)
(차량번호, 이미지 경로, 위반 시간, 위반 날짜, 위반 위치)
- 위 후보기 중 ios 앱 내에서 검색을 위해서는 차량번호가 필요했다는 점, 동일 시간에 두개 이상의 차량이 존재할 수 없다는 점, 최소한의 속성을 이용해서 기본 키를 선정해야 한다는 점에서 기본 키를 (차량번호, 위반 시간, 위반 날짜)로 지정하였다
- join을 실행할때 각 테이블의 location을 중심으로 이뤄지기 때문에 단속 위치에 대한 정보가 중요하게 작용
- 그러므로 CheckPoint 테이블에서 location 속성 중 하나가 삭제된 경우 CheckPoint 테이블의 location을 참조하는 TrafficViolations 테이블에서도 동일한 값을 가진 모든 레코드를 제거하도록 작동 해야한다
- 참조 결과에 따른 레코드를 삭제했을때 결과는 아래와 같다

```

mysql> insert into TrafficViolations(car_number, overspeed, speed_limit, location, violation_time, violation_date,image_path)
-> VALUES ('A1239', 90, 60, '충청남도 태안군 소월면 농협', '13:45:00', '2023-10-01', '/Users/yoonwoosang/Desktop/programming/swift/project/Database\ Query/server/testImage/car1.png');
Query OK, 1 row affected (0.01 sec)

mysql> select * from TrafficViolations;
+-----+-----+-----+-----+-----+-----+-----+
| car_number | overspeed | speed_limit | location | violation_time | violation_date | image_path |
+-----+-----+-----+-----+-----+-----+-----+
| A1239 | 90 | 60 | 충청남도 태안군 소월면 농협 | 13:45:00 | 2023-10-01 | /Users/yoonwoosang/Desktop/programming/swift/project/Database\ Query/server/testImage/car1.png |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from
-> CheckPoint;
+-----+-----+-----+
| location | lat | lon |
+-----+-----+-----+
| 충청남도 서산시 서산휴게소 | 36.7387 | 126.5656 |
| 충청남도 서산시 운산면 신창고 | 36.7172 | 126.5691 |
| 충청남도 서산시 운산면 운산터널 | 36.7846 | 126.5654 |
| 충청남도 서산시 해미면 한서대 입구 | 36.6854 | 126.5793 |
| 충청남도 서산시 해미면 해미 IC | 36.7132 | 126.5625 |
| 충청남도 태안군 소월면 농협 | 36.3695 | 126.5789 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> delete from CheckPoint
-> where CheckPoint.location = '충청남도 태안군 소월면 농협';
Query OK, 1 row affected (0.00 sec)

mysql> select * from CheckPoint;
+-----+-----+-----+
| location | lat | lon |
+-----+-----+-----+
| 충청남도 서산시 서산휴게소 | 36.7387 | 126.5656 |
| 충청남도 서산시 운산면 신창고 | 36.7172 | 126.5691 |
| 충청남도 서산시 운산면 운산터널 | 36.7846 | 126.5654 |
| 충청남도 서산시 해미면 한서대 입구 | 36.6854 | 126.5793 |
| 충청남도 서산시 해미면 해미 IC | 36.7132 | 126.5625 |
+-----+-----+-----+
5 rows in set (0.01 sec)

```

- CheckPoint에 존재하는 레코드를 삭제하면 TrafficViolation내에 존재하는 동일한 값을 가진 레코드가 삭제되는 것을 확인할 수 있다

Database

- test

테이블 이름

- CheckPoint

테이블 구조

location VARCHAR(255) NOT NULL PRIMARY KEY,
 lat Double(7,4) NOT NULL,
 lon Double(7,4) NOT NULL,
 speed_limit INT NOT NULL

테이블 생성시

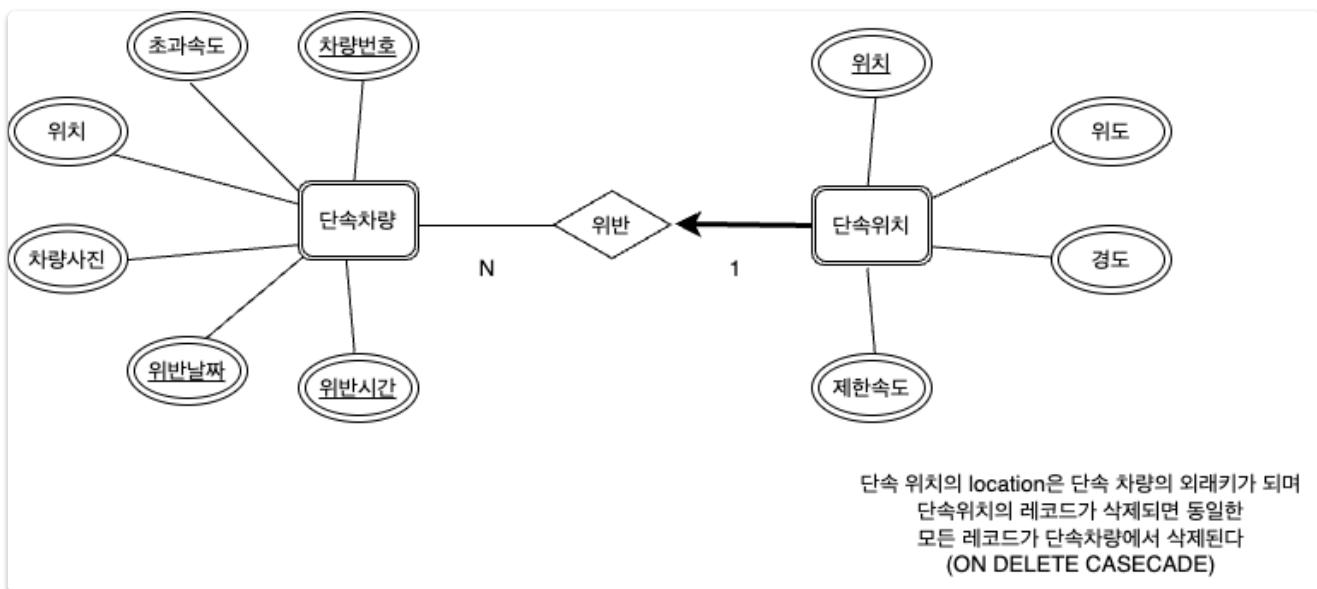
```

CREATE TABLE CheckPoint (
    location VARCHAR(255) NOT NULL,
    lat DOUBLE(8, 4) NOT NULL,
    lon DOUBLE(8, 4) NOT NULL,
    speed_limit INT NOT NULL,
);

```

ERD

- 결과 조회 테이블에 대해서 CheckPoint는 부분참여, TrafficViolations는 전체참여이다
- CheckPoint는 일, TrafficViolations는 다의 관계를 가진다. 즉 join의 결과로 생성되는 새로운 결과조회의 테이블은 하나의 CheckPoint 레코드에 대해 여러개의 TrafficViolations의 레코드가 묶이게 된 형태를 가진다
- TrafficViolation은 location을 외래키로 가진다. 또한 CheckPoint의 location에 대한 참조를 통해 CheckPoint의 location 속성중 해당 값이 삭제되면 TrafficViolations의 모든 동일 location 을 가진 레코드가 삭제된다
- TrafficViolations 의 기본 키는 차량번호, 위반시간, 위반 날짜이며, CheckPoint의 기본 키는 위치 이다

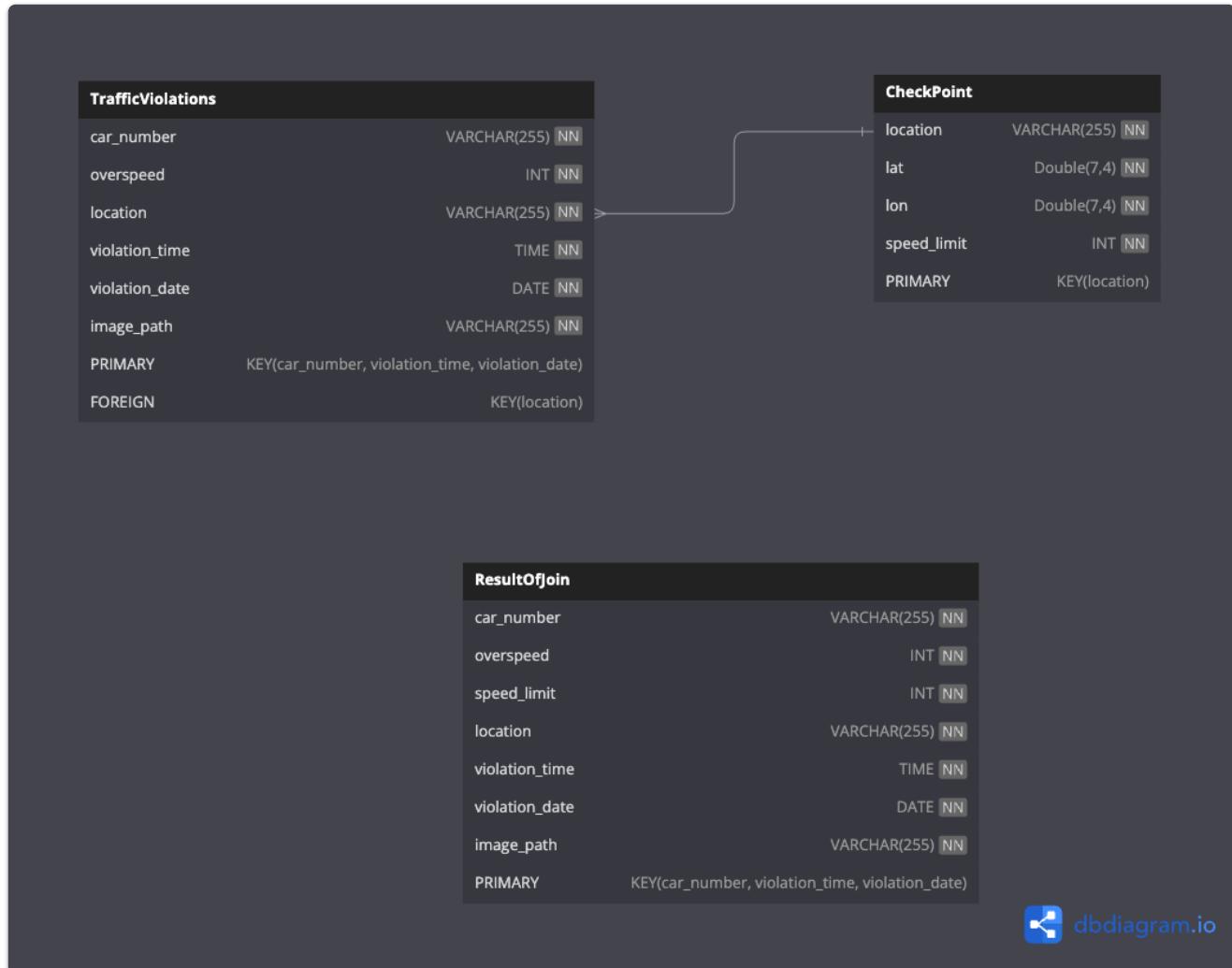


논리적 데이터베이스 모델

- TrafficViolation 테이블은 위반 차량에 대한 기본정보를 포함한다
- CheckPoint는 단속위치에 대한 기본정보를 포함한다
- ResultOfJoin 테이블은 ios앱을 통해 유저에게 표시하는 테이블이 되며, 서버에서 조인을 통해 별도의 테이블을 생성하며 명칭은 임의로 지정한 것이다. CheckPoint의 부분참여, TrafficViolations의 전체 참여를 통해 구성된다

- 아래 링크는 밑의 사진에 대한 코드와 테이블 모델을 작성한 사이트에 대한 링크다

<https://dbdiagram.io/d/TrafficViolation-Project-656828043be149578706c542>



- 해당 이미지는 dbdiagram.io라는 사이트에서 제작한 테이블 모델이다
- 해당 이미지를 제작할때 사용한 코드의 구성은 아래와 같다

```

TABLE TrafficViolations {
    car_number VARCHAR(255) [not null]
    overspeed INT [not null]
    location VARCHAR(255) [not null]
    violation_time TIME [not null]
    violation_date DATE [not null]
    image_path VARCHAR(255) [not null]
    PRIMARY KEY (car_number, violation_time, violation_date)
    FOREIGN KEY (location)
}
```

```
// 메모  
Note: "위반 차량에 대한 정보를 나타내는 테이블"  
}
```

```
TABLE CheckPoint{  
  
location VARCHAR(255) [not null]  
  
lat Double(7,4) [not null]  
  
lon Double(7,4) [not null]  
  
speed_limit INT [not null]  
  
PRIMARY KEY (location)
```

```
//메모  
Note: "위반 단속 위치를 나타내는 테이블"  
}
```

```
TABLE ResultOfJoin{  
  
car_number VARCHAR(255) [not null]  
  
overspeed INT [not null]  
  
speed_limit INT [not null]  
  
location VARCHAR(255) [not null]  
  
violation_time TIME [not null]  
  
violation_date DATE [not null]  
  
image_path VARCHAR(255) [not null]  
  
PRIMARY KEY (car_number, violation_time, violation_date)
```

```
// 메모  
Note: "ios 앱 내부에서 유저에게 표시하기 위한 테이블"
```

}

```
// 참조에 관한 부분을 정의한다  
// TrafficViolations.location 는 CheckPoint.location를 참조하며  
// ON DELETE CASCADE 관계를 가진다
```

Ref: [TrafficViolations.location > CheckPoint.location](#)