

# **PRIMARY KEY, FOREIGN KEY**

Auto Mobile Robot

Exported on 06/26/2024

## Table of Contents

|                                   |    |
|-----------------------------------|----|
| 1 실습환경 만들기 .....                  | 4  |
| 1.1 실습할 데이터베이스로 이동 .....          | 4  |
| 2 PRIMARY KEY (기본키) .....         | 5  |
| 2.1 PRIMARY KEY 생성 문법 1 .....     | 5  |
| 2.2 PRIMARY KEY 생성 예제 1 - 1 ..... | 5  |
| 2.3 PRIMARY KEY 생성 예제 1 - 2 ..... | 6  |
| 2.4 PRIMARY KEY 삭제 문법 .....       | 7  |
| 2.5 PRIMARY KEY 삭제 예제 1 .....     | 7  |
| 2.6 PRIMARY KEY 삭제 예제 2 .....     | 7  |
| 2.7 PRIMARY KEY 생성 문법 2 .....     | 8  |
| 2.8 PRIMARY KEY 생성 예제 2 - 1 ..... | 8  |
| 2.9 PRIMARY KEY 생성 예제 2 - 2 ..... | 8  |
| 3 FOREIGN KEY (외래키) .....         | 10 |
| 3.1 FOREIGN KEY 생성 문법 .....       | 10 |
| 3.2 FOREIGN KEY 생성 예제 1 - 1 ..... | 10 |
| 3.3 FOREIGN KEY 생성 예제 1 - 2 ..... | 11 |
| 3.4 CONSTRAINT 확인 문법 .....        | 12 |
| 3.5 CONSTRAINT 확인 예제 .....        | 12 |
| 3.6 FOREIGN KEY 삭제 문법 .....       | 12 |
| 3.7 FOREIGN KEY 삭제 예제 .....       | 13 |
| 3.8 FOREIGN KEY 생성 문법 2 .....     | 13 |
| 3.9 FOREIGN KEY 생성 예제 2 .....     | 14 |
| 4 FOREIGN KEY 예제 .....            | 15 |
| 4.1 분석 - 1 .....                  | 15 |
| 4.2 분석 - 2 .....                  | 15 |
| 4.3 분석 - 3 .....                  | 16 |
| 4.4 Primary Key 설정 .....          | 16 |

|                             |    |
|-----------------------------|----|
| 4.5 Foreign Key 설정 - 1..... | 17 |
| 4.6 Foreign Key 설정 - 2..... | 17 |
| 4.7 Foreign Key 설정 - 3..... | 18 |
| 4.8 JOIN .....              | 19 |
| 5 연습 .....                  | 21 |

# 1 실습환경 만들기

## 1.1 실습할 데이터베이스로 이동

amrbase 사용 (이동)

```
USE amrbase;
```

```
|mysql> use amrbase;  
Database changed  
|mysql>
```

## 2 PRIMARY KEY (기본키)

- 테이블의 각 레코드를 식별
- 중복되지 않는 고유값을 포함
- NULL 값을 포함할 수 없음
- 테이블 당 하나의 기본키를 가짐

### 2.1 PRIMARY KEY 생성 문법 1

```
CREATE TABLE tablename
(
    column1 datatype NOT NULL,
    column2 datatype NOT NULL,
    ...
    CONSTRAINT constraint_name
    PRIMARY KEY (column1, column2, ...)
);
```

### 2.2 PRIMARY KEY 생성 예제 1 - 1

하나의 칼럼을 기본키로 설정하는 경우

```
CREATE TABLE person
(
    pid int NOT NULL,
    name varchar(16),
    age int,
    sex char,
    PRIMARY KEY (pid)
);
```

```
mysql> CREATE TABLE person
-> (
->   pid int NOT NULL,
->   name varchar(16),
->   age int,
->   sex char,
->   PRIMARY KEY (pid)
-> );
Query OK, 0 rows affected (0.01 sec)
```

```
[mysql> desc person;
```

| Field | Type        | Null | Key | Default | Extra |
|-------|-------------|------|-----|---------|-------|
| pid   | int         | NO   | PRI | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
| sex   | char(1)     | YES  |     | NULL    |       |

4 rows in set (0.00 sec)

## 2.3 PRIMARY KEY 생성 예제 1 - 2

여러개의 칼럼을 기본키로 설정하는 경우

```
CREATE TABLE animal
(
  name varchar(16) NOT NULL,
  type varchar(16) NOT NULL,
  age int,
  PRIMARY KEY (name, type)
);
```

```
mysql> CREATE TABLE animal
-> (
->   name varchar(16) NOT NULL,
->   type varchar(16) NOT NULL,
->   age int,
->   CONSTRAINT PK_Person PRIMARY KEY (name, type)
-> );
Query OK, 0 rows affected (0.01 sec)
```

```
[mysql> desc animal;
```

| Field | Type        | Null | Key | Default | Extra |
|-------|-------------|------|-----|---------|-------|
| name  | varchar(16) | NO   | PRI | NULL    |       |
| type  | varchar(16) | NO   | PRI | NULL    |       |
| age   | int         | YES  |     | NULL    |       |

3 rows in set (0.00 sec)

## 2.4 PRIMARY KEY 삭제 문법

```
ALTER TABLE tablename
DROP PRIMARY KEY;
```

## 2.5 PRIMARY KEY 삭제 예제 1

하나의 칼럼이 기본키로 설정된 경우

```
ALTER TABLE person
DROP PRIMARY KEY;
```

```
[mysql> alter table person drop primary key;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
[mysql> desc person;
```

| Field | Type        | Null | Key | Default | Extra |
|-------|-------------|------|-----|---------|-------|
| pid   | int         | NO   |     | NULL    |       |
| name  | varchar(16) | YES  |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
| sex   | char(1)     | YES  |     | NULL    |       |

```
4 rows in set (0.01 sec)
```

## 2.6 PRIMARY KEY 삭제 예제 2

여러개의 칼럼이 기본키로 설정된 경우 (삭제하는 방법은 동일)

```
ALTER TABLE animal
DROP PRIMARY KEY;
```

```
[mysql> alter table animal drop primary key;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

[mysql> desc animal;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | varchar(16)   | NO   |     | NULL    |       |
| type  | varchar(16)   | NO   |     | NULL    |       |
| age   | int           | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

## 2.7 PRIMARY KEY 생성 문법 2

```
ALTER TABLE tablename
ADD PRIMARY KEY (column1, column2, ...);
```

## 2.8 PRIMARY KEY 생성 예제 2 - 1

하나의 칼럼을 기본키로 지정하는 경우

```
ALTER TABLE person
ADD PRIMARY KEY (id);
```

```
[mysql> alter table person add primary key (pid);
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

[mysql> desc person;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| pid   | int           | NO   | PRI | NULL    |       |
| name  | varchar(16)   | YES  |     | NULL    |       |
| age   | int           | YES  |     | NULL    |       |
| sex   | char(1)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

## 2.9 PRIMARY KEY 생성 예제 2 - 2

여러개의 칼럼을 기본키로 지정하는 경우



```
ALTER TABLE animal
ADD CONSTRAINT PK_animal PRIMARY KEY (name, type);
```

```
[mysql> alter table animal add constraint pk_animal primary key (name, type);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
[mysql> desc animal;
```

| Field | Type        | Null | Key | Default | Extra |
|-------|-------------|------|-----|---------|-------|
| name  | varchar(16) | NO   | PRI | NULL    |       |
| type  | varchar(16) | NO   | PRI | NULL    |       |
| age   | int         | YES  |     | NULL    |       |

```
3 rows in set (0.00 sec)
```

### 3 FOREIGN KEY (외래키)

한 테이블을 다른 테이블과 연결해주는 역할이며, 참조되는 테이블의 항목은 그 테이블의 기본키 (혹은 단일값)

#### 3.1 FOREIGN KEY 생성 문법

```
CREATE TABLE tablename
(
    column1 datatype NOT NULL,
    column2 datatype NOT NULL,
    column3 datatype,
    column4 datatype,
    ...
    CONSTRAINT constraint_name
    PRIMARY KEY (column1, column2, ...),
    CONSTRAINT constraint_name
    FOREIGN KEY (column3, column4, ...) REFERENCES REF_tablename(REF_column)
);
```

#### 3.2 FOREIGN KEY 생성 예제 1 - 1

CREATE TABLE 에서 FOREIGN KEY 를 지정하는 경우

```
CREATE TABLE orders
(
    oid int not null,
    order_no varchar(16),
    pid int,
    PRIMARY KEY (oid),
    CONSTRAINT FK_person FOREIGN KEY (pid) REFERENCES person(pid)
);
```

```
mysql> CREATE TABLE orders
-> (
->   oid int not null,
->   order_no varchar(16),
->   pid int,
->   PRIMARY KEY (oid),
->   CONSTRAINT FK_person FOREIGN KEY (pid) REFERENCES person(pid)
-> );
Query OK, 0 rows affected (0.01 sec)

[mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| oid   | int       | NO   | PRI | NULL    |       |
| order_no | varchar(16) | YES | MUL | NULL    |       |
| pid   | int       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

### 3.3 FOREIGN KEY 생성 예제 1 - 2

CREATE TABLE 에서 FOREIGN KEY 를 지정하는 경우, CONSTRAINT 를 생략할 수 있다.

```
CREATE TABLE job
(
  jid int not null,
  name varchar(16),
  pid int,
  PRIMARY KEY (jid),
  FOREIGN KEY (pid) REFERENCES person(pid)
);
```

```
mysql> CREATE TABLE job
-> (
->   jid int not null,
->   name varchar(16),
->   pid int,
->   PRIMARY KEY (jid),
->   FOREIGN KEY (pid) REFERENCES person(pid)
-> );
Query OK, 0 rows affected (0.01 sec)

[mysql> desc job;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| jid   | int       | NO   | PRI | NULL    |       |
| name  | varchar(16) | YES | MUL | NULL    |       |
| pid   | int       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

### 3.4 CONSTRAINT 확인 문법

자동 생성된 CONSTRAINT 를 확인하는 방법

```
SHOW CREATE TABLE tablename;
```

### 3.5 CONSTRAINT 확인 예제

```
SHOW CREATE TABLE job;
```

```
[mysql> show create table job;
+-----+
| Table | Create Table
+-----+
| job   | CREATE TABLE `job` (
  `jid` int NOT NULL,
  `name` varchar(16) DEFAULT NULL,
  `pid` int DEFAULT NULL,
  PRIMARY KEY (`jid`),
  KEY `pid` (`pid`),
  CONSTRAINT `job_ibfk_1` FOREIGN KEY (`pid`) REFERENCES `person` (`pid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)
```

### 3.6 FOREIGN KEY 삭제 문법

```
ALTER TABLE tablename
DROP FOREIGN KEY FK_constraint;
```

### 3.7 FOREIGN KEY 삭제 예제

```
ALTER TABLE orders
DROP FOREIGN KEY FK_person;
```

```
[mysql> alter table orders drop foreign key FK_person;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0]
```

```
[mysql> desc orders;
```

| Field    | Type        | Null | Key | Default | Extra |
|----------|-------------|------|-----|---------|-------|
| oid      | int         | NO   | PRI | NULL    |       |
| order_no | varchar(16) | YES  |     | NULL    |       |
| pid      | int         | YES  | MUL | NULL    |       |

3 rows in set (0.00 sec)

```
[mysql> show create table orders;
```

| Table  | Create Table  |
|--------|---|
| orders | <pre>CREATE TABLE `orders` (   `oid` int NOT NULL,   `order_no` varchar(16) DEFAULT NULL,   `pid` int DEFAULT NULL,   PRIMARY KEY (`oid`),   KEY `FK_person` (`pid`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci</pre> |

1 row in set (0.00 sec)

### 3.8 FOREIGN KEY 생성 문법 2

Table 이 생성된 이후에도 ALTER TABLE 을 통해 FOREIGN KEY 를 지정할 수 있다.

```
ALTER TABLE tablename
ADD FOREIGN KEY (column) REFERENCES REF_tablename(REF_column);
```

### 3.9 FOREIGN KEY 생성 예제 2

```
ALTER TABLE orders
ADD FOREIGN KEY (pid) REFERENCES person(pid);
```

```
mysql> ALTER TABLE orders
[ -> ADD FOREIGN KEY (pid) REFERENCES person(pid);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
[mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| oid   | int  | NO   | PRI | NULL    |       |
| order_no | varchar(16) | YES |     | NULL    |       |
| pid   | int  | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
[mysql> show create table orders;
```

```
+-----+
| Table | Create Table
+-----+
| orders | CREATE TABLE `orders` (
  `oid` int NOT NULL,
  `order_no` varchar(16) DEFAULT NULL,
  `pid` int DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `pid` (`pid`),
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`pid`) REFERENCES `person` (`pid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)
```



## 4 FOREIGN KEY 예제

police\_station 과 crime\_status 테이블 사이에 관계 (Foreign Key) 를 설정해 봅시다.

AWS RDS (database-1) 의 amrbase 에서 작업합니다.

### 4.1 분석 - 1

police\_station.name 과 crime\_status.police\_station 을 매칭하여 관계를 맺도록 하겠습니다.

```
mysql> select count(distinct name) from police_station;
+-----+
| count(distinct name) |
+-----+
|                    31 |
+-----+
1 row in set (0.01 sec)

mysql> select count(distinct police_station) from crime_status;
+-----+
| count(distinct police_station) |
+-----+
|                    31 |
+-----+
1 row in set (0.00 sec)
```

### 4.2 분석 - 2

경찰서 이름이 각 테이블에서 표시되는 형식이 다릅니다.

```
mysql> select distinct name from police_station limit 3;
+-----+
| name |
+-----+
| 서울 중부경찰서 |
| 서울 종로경찰서 |
| 서울 남대문경찰서 |
+-----+
3 rows in set (0.01 sec)

mysql> select distinct police_station from crime_status limit 3;
+-----+
| police_station |
+-----+
| 중부 |
| 종로 |
| 남대문 |
+-----+
3 rows in set (0.01 sec)
```

## 4.3 분석 - 3

crime\_status.police\_station 을 police\_station.name 과 같게 만들어서 비교하도록 합니다.

```
mysql> select c.police_station, p.name
-> from crime_status c, police_station p
-> where p.name like concat('서울 ', c.police_station, '경찰서')
-> group by c.police_station, p.name;
```

| police_station | name      |
|----------------|-----------|
| 중부             | 서울 중부경찰서  |
| 종로             | 서울 종로경찰서  |
| 남대문            | 서울 남대문경찰서 |
| 서대문            | 서울 서대문경찰서 |
| 혜화             | 서울 혜화경찰서  |
| 용산             | 서울 용산경찰서  |
| 성북             | 서울 성북경찰서  |
| 동대문            | 서울 동대문경찰서 |
| 마포             | 서울 마포경찰서  |
| 영등포            | 서울 영등포경찰서 |
| 성동             | 서울 성동경찰서  |
| 동작             | 서울 동작경찰서  |
| 광진             | 서울 광진경찰서  |
| 서부             | 서울 서부경찰서  |
| 강북             | 서울 강북경찰서  |
| 금천             | 서울 금천경찰서  |
| 중랑             | 서울 중랑경찰서  |
| 강남             | 서울 강남경찰서  |
| 관악             | 서울 관악경찰서  |
| 강서             | 서울 강서경찰서  |
| 강동             | 서울 강동경찰서  |
| 종암             | 서울 종암경찰서  |
| 구로             | 서울 구로경찰서  |
| 서초             | 서울 서초경찰서  |
| 양천             | 서울 양천경찰서  |
| 송파             | 서울 송파경찰서  |
| 노원             | 서울 노원경찰서  |
| 방배             | 서울 방배경찰서  |
| 은평             | 서울 은평경찰서  |
| 도봉             | 서울 도봉경찰서  |
| 수서             | 서울 수서경찰서  |

31 rows in set (0.01 sec)

## 4.4 Primary Key 설정

police\_station.name 을 Primary Key 로 설정합니다.

```
ALTER TABLE police_station
ADD PRIMARY KEY (name);
```



```
mysql> ALTER TABLE police_station
-> ADD PRIMARY KEY (name);
Query OK, 31 rows affected (0.18 sec)
Records: 31 Duplicates: 0 Warnings: 0
```

```
mysql> DESC police_station;
```

| Field   | Type         | Null | Key | Default | Extra |
|---------|--------------|------|-----|---------|-------|
| name    | varchar(16)  | NO   | PRI | NULL    |       |
| address | varchar(128) | YES  |     | NULL    |       |

```
2 rows in set (0.01 sec)
```

## 4.5 Foreign Key 설정 - 1

crime\_status 테이블에 Foreign Key 로 사용할 Column 추가

```
ALTER TABLE crime_status
ADD COLUMN reference VARCHAR(16);
```

```
mysql> ALTER TABLE crime_status
-> ADD COLUMN reference VARCHAR(16);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC crime_status;
```

| Field          | Type        | Null | Key | Default | Extra |
|----------------|-------------|------|-----|---------|-------|
| year           | year        | YES  |     | NULL    |       |
| police_station | varchar(16) | YES  |     | NULL    |       |
| crime_type     | varchar(16) | YES  |     | NULL    |       |
| status_type    | char(2)     | YES  |     | NULL    |       |
| case_number    | int         | YES  |     | NULL    |       |
| reference      | varchar(16) | YES  |     | NULL    |       |

```
6 rows in set (0.01 sec)
```

## 4.6 Foreign Key 설정 - 2

Foreign Key 생성

```
ALTER TABLE crime_status
ADD FOREIGN KEY (reference) REFERENCES police_station(name);
```

```
mysql> ALTER TABLE crime_status
-> ADD FOREIGN KEY (reference) REFERENCES police_station(name);
Query OK, 310 rows affected (0.12 sec)
Records: 310 Duplicates: 0 Warnings: 0

mysql> DESC crime_status;
```

| Field          | Type        | Null | Key | Default | Extra |
|----------------|-------------|------|-----|---------|-------|
| year           | year        | YES  |     | NULL    |       |
| police_station | varchar(16) | YES  |     | NULL    |       |
| crime_type     | varchar(16) | YES  |     | NULL    |       |
| status_type    | char(2)     | YES  |     | NULL    |       |
| case_number    | int         | YES  |     | NULL    |       |
| reference      | varchar(16) | YES  | MUL | NULL    |       |

```
6 rows in set (0.02 sec)
```

## 4.7 Foreign Key 설정 - 3

Foreign Key 값 Update

```
UPDATE crime_status c, police_station p
SET c.reference = p.name
WHERE p.name LIKE concat('서울', c.police_station, '경찰서');
```

```
mysql> UPDATE crime_status c, police_station p
-> SET c.reference = p.name
-> WHERE p.name LIKE concat('서울 ', c.police_station, '경찰서 ');
Query OK, 310 rows affected (0.04 sec)
Rows matched: 310  Changed: 310  Warnings: 0
```

```
mysql> SELECT distinct police_station, reference FROM crime_status;
```

| police_station | reference |
|----------------|-----------|
| 중부             | 서울 중부경찰서  |
| 종로             | 서울 종로경찰서  |
| 남대문            | 서울 남대문경찰서 |
| 서대문            | 서울 서대문경찰서 |
| 혜화             | 서울 혜화경찰서  |
| 용산             | 서울 용산경찰서  |
| 성북             | 서울 성북경찰서  |
| 동대문            | 서울 동대문경찰서 |
| 마포             | 서울 마포경찰서  |
| 영등포            | 서울 영등포경찰서 |
| 성동             | 서울 성동경찰서  |
| 동작             | 서울 동작경찰서  |
| 광진             | 서울 광진경찰서  |
| 서부             | 서울 서부경찰서  |
| 강북             | 서울 강북경찰서  |
| 금천             | 서울 금천경찰서  |
| 중랑             | 서울 중랑경찰서  |
| 강남             | 서울 강남경찰서  |
| 관악             | 서울 관악경찰서  |
| 강서             | 서울 강서경찰서  |
| 강동             | 서울 강동경찰서  |
| 종암             | 서울 종암경찰서  |
| 구로             | 서울 구로경찰서  |
| 서초             | 서울 서초경찰서  |
| 양천             | 서울 양천경찰서  |
| 송파             | 서울 송파경찰서  |
| 노원             | 서울 노원경찰서  |
| 방배             | 서울 방배경찰서  |
| 은평             | 서울 은평경찰서  |
| 도봉             | 서울 도봉경찰서  |
| 수서             | 서울 수서경찰서  |

```
31 rows in set (0.01 sec)
```

## 4.8 JOIN

Foreign Key 를 기준으로 두 테이블을 연관시켜 검색할 수 있다.

```
SELECT c.police_station, p.address
FROM crime_status c, police_station p
WHERE c.reference = p.name
GROUP BY c.police_station;
```

```
mysql> SELECT c.police_station, p.address
-> FROM crime_status c, police_station p
-> WHERE c.reference = p.name
-> GROUP BY c.police_station;
```

| police_station | address                            |
|----------------|------------------------------------|
| 강남             | 서울특별시 강남구 테헤란로 114길 11             |
| 강동             | 서울특별시 강동구 성내로 33                   |
| 강북             | 서울특별시 강북구 오패산로 406                 |
| 강서             | 서울특별시 양천구 화곡로 73                   |
| 관악             | 서울특별시 관악구 관악로 5길 33                |
| 광진             | 서울특별시 광진구 광나루로 447 광진소방서 임시청사 (능동) |
| 구로             | 서울특별시 구로구 가마산로 235                 |
| 금천             | 서울특별시 관악구 남부순환로 1435               |
| 남대문            | 서울특별시 중구 한강대로 410                  |
| 노원             | 서울특별시 노원구 노원로 283                  |
| 도봉             | 서울특별시 도봉구 노해로 403                  |
| 동대문            | 서울특별시 동대문구 약령시로 21길 29             |
| 동작             | 서울특별시 동작구 노량진로 148                 |
| 마포             | 서울특별시 마포구 마포대로 183                 |
| 방배             | 서울특별시 서초구 방배천로 54                  |
| 서대문            | 서울특별시 서대문구 통일로 113                 |
| 서부             | 서울특별시 은평구 은평로 9길 15                |
| 서초             | 서울특별시 서초구 반포대로 179                 |
| 성동             | 서울특별시 성동구 왕십리광장로 9                 |
| 성북             | 서울특별시 성북구 보문로 170                  |
| 송파             | 서울특별시 송파구 중대로 221                  |
| 수서             | 서울특별시 강남구 개포로 617                  |
| 양천             | 서울특별시 양천구 목동동로 99                  |
| 영등포            | 서울특별시 영등포구 국회대로 608                |
| 용산             | 서울특별시 용산구 원효로 89길 24               |
| 은평             | 서울특별시 은평구 연서로 365                  |
| 종로             | 서울특별시 종로구 율곡로 46                   |
| 중암             | 서울특별시 성북구 중암로 135                  |
| 중랑             | 서울특별시 중랑구 중랑역로 137                 |
| 중부             | 서울특별시 중구 수표로 27                    |
| 혜화             | 서울특별시 종로구 창경궁로 112-16              |

31 rows in set (0.01 sec)

## 5 연습

1. 다음과 같이 study\_id 가 PRIMARY KEY, patient\_id 가 person 테이블의 pid 와 연결된 FOREIGN KEY 로 지정된 study 테이블을 생성하세요.

| Field      | Type | Null | Key | Default | Extra |
|------------|------|------|-----|---------|-------|
| study_id   | int  | NO   | PRI | NULL    |       |
| study_date | date | YES  |     | NULL    |       |
| study_time | time | YES  |     | NULL    |       |
| patient_id | int  | YES  | MUL | NULL    |       |

2. 생성한 테이블의 PRIMARY KEY 를 삭제하세요.
3. 생성한 테이블의 FOREIGN KEY 를 삭제하세요.
4. study 테이블의 patient\_id 를 person 테이블의 pid 와 연결된 FOREIGN KEY 로 등록하세요.
5. study 테이블의 study\_id 를 PRIMARY KEY로 등록하세요.