

SQL & ORM



| Background

- ✓ SQL
- ✓ Database

| Goal

- ✓ 테이블의 생성
- ✓ 데이터의 생성 및 조회
- ✓ SQL과 ORM 비교

| Problem

Table Name : countries

name	data type
room_num	text
check_in	text
check_out	text
grade	text
price	integer

id	room_num	check_in	check_out	grade	price
1	B203	2019-12-31	2020-01-03	suite	900
2	1102	2020-01-04	2020-01-08	suite	850
3	303	2020-01-01	2020-01-03	deluxe	500
4	807	2020-01-04	2020-01-07	superior	300



1. SQL Query

위 countries 테이블을 바탕으로 아래 문제에 해당하는 SQL query문을 작성하고 실행하시오.

- 1) countries 테이블을 생성하시오.
- 2) 데이터를 입력하시오.
- 3) 테이블의 이름을 hotels로 변경하시오.
- 4) 객실 가격을 내림차순으로 정렬하여 상위 2개의 room_num과 price를 조회하시오.
- 5) grade 별로 분류하고 분류된 grade 개수를 내림차순으로 조회하시오.
- 6) 객실의 위치가 지하 혹은 등급이 deluxe인 객실의 모든 정보를 조회하시오.
- 7) 지상층 객실이면서 2020년 1월 4일에 체크인 한 객실의 목록을 price 오름차순으로 조회하시오.



2. SQL ORM 비교하기

주어진 정보를 활용하여 작성된 SQL문과 대응하는 ORM문을 작성하고 실행하시오.

Table Name : users

name	data type
id	integer(pk)
first_name	text
last_name	text
age	integer
country	text
phone	text
balance	integer

SQL & ORM



1) user 테이블 전체 데이터를 조회하시오.

```
SELECT * FROM users_user;
```

2) id가 19인 사람의 age를 조회하시오.

```
SELECT age FROM users_user WHERE id = 19;
```

3) 모든 사람의 age를 조회하시오.

```
SELECT age FROM users_user;
```

4) age가 40 이하인 사람들의 id와 balance를 조회하시오.

```
SELECT id, balance FROM users_user WHERE age <= 40;
```

5) last_name이 '김'이고 balance가 500 이상인 사람들의 first_name을 조회하시오.

```
SELECT first_name FROM users_user  
WHERE last_name = '김' AND balance >= 500;
```

6) first_name이 '수'로 끝나면서 행정구역이 경기도인 사람들의 balance를 조회하시오.

```
SELECT balance FROM users_user  
WHERE first_name LIKE '%수' AND country = '경기도';
```

7) balance가 2000 이상이거나 age가 40 이하인 사람의 총 인원수를 구하시오.

```
SELECT COUNT(*) FROM users_user  
WHERE balance >= 2000 OR age <= 40;
```

8) phone 앞자리가 '010'으로 시작하는 사람의 총원을 구하시오.

```
SELECT COUNT(*) FROM users_user  
WHERE phone LIKE '010%';
```

SQL & ORM



9) 이름이 '김옥자'인 사람의 행정구역을 경기도로 수정하시오.

```
UPDATE users_user SET country = '경기도'
WHERE first_name = '옥자' AND last_name = '김';

SELECT country FROM users_user
WHERE first_name = '옥자' AND last_name = '김'
```

10) 이름이 '백진호'인 사람을 삭제하시오.

```
DELETE FROM users_user
WHERE first_name = '진호' AND last_name = '백';

SELECT * FROM users_user
WHERE first_name = '진호' AND last_name = '백';
```

11) balance를 기준으로 상위 4명의 first_name, last_name, balance를 조회하시오.

```
SELECT first_name, last_name, balance FROM users_user
ORDER BY balance DESC LIMIT 4;
```

12) phone에 '123'을 포함하고 age가 30미만인 정보를 조회하시오.

```
SELECT * FROM users_user
WHERE phone LIKE '%123%' AND age < 30;
```

13) phone이 '010'으로 시작하는 사람들의 행정 구역을 중복 없이 조회하시오.

```
SELECT DISTINCT country FROM users_user
WHERE phone LIKE '010%';
```

14) 모든 인원의 평균 age를 구하시오.

```
SELECT AVG(age) FROM users_user;
```

SQL & ORM



15) 박씨의 평균 balance를 구하시오.

```
SELECT AVG(balance) FROM users_user  
WHERE last_name = '박';
```

16) 경상북도에 사는 사람 중 가장 많은 balance의 액수를 구하시오.

```
SELECT MAX(balance) FROM users_user  
WHERE country = '경상북도';
```

17) 제주특별자치도에 사는 사람 중 balance가 가장 많은 사람의 first_name을 구하시오.

```
SELECT first_name FROM users_user  
WHERE country = '제주특별자치도' ORDER BY balance DESC LIMIT 1;
```