

UNION

Auto Mobile Robot

Exported on 06/21/2024

Table of Contents

1 실습환경 만들기	3
1.1 실습할 데이터베이스로 이동	3
1.2 기존 테스트용 데이터 확인	3
1.3 테스트용 테이블 생성	3
1.4 테스트 데이터 추가 1	4
1.5 테스트 데이터 추가 2	4
1.6 데이터 확인 1	5
1.7 데이터 확인 2	5
2 UNION	6
2.1 UNION 문법	6
2.2 UNION 예제 1 - UNION ALL	6
2.3 UNION 예제 1 - UNION	7
2.4 UNION 예제 2 - UNION ALL	7
2.5 UNION 예제 2 - UNION	8
2.6 UNION 예제 3	8
2.7 UNION 예제 3 - 1	8
2.8 UNION 예제 3 - 2	9
2.9 UNION 예제 3 -3	9
3 연습	10

1 실습환경 만들기

1.1 실습할 데이터베이스로 이동

amrbase 사용 (이동)

```
USE amrbase;
```

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

1.2 기존 테스트용 데이터 확인

```
SELECT * FROM celeb;
```

```
mysql> select * from celeb;
```

ID	NAME	birthday	AGE	SEX	JOB_TITLE	AGENCY
1	아이유	1993-05-16	29	F	가수, 텔런트	EDAM엔터테인먼트
2	이미주	1994-09-23	28	F	가수	올림엔터테인먼트
3	송강	1994-04-23	28	M	텔런트	나무엑터스
4	강동원	1981-01-18	41	M	영화배우, 텔런트	YG엔터테인먼트
5	유재석	1972-08-14	50	M	MC, 개그맨	안테나
6	차승원	1970-06-07	48	M	영화배우, 모델	YG엔터테인먼트
7	이수현	1999-05-04	23	F	가수	YG엔터테인먼트

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

1.3 테스트용 테이블 생성

```
CREATE TABLE test1
(
  no int
);

CREATE TABLE test2
(
  no int
```

```
);
```

```
mysql> create table test1
[ -> (
[ -> no int
[ -> );
Query OK, 0 rows affected (0.02 sec)

mysql> create table test2
[ -> (
[ -> no int
[ -> );
Query OK, 0 rows affected (0.01 sec)
```

1.4 테스트 데이터 추가 1

```
INSERT INTO test1 VALUES (1);
INSERT INTO test1 VALUES (2);
INSERT INTO test1 VALUES (3);
```

```
mysql> INSERT INTO test1 VALUES (1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO test1 VALUES (2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO test1 VALUES (3);
Query OK, 1 row affected (0.00 sec)
```

1.5 테스트 데이터 추가 2

```
INSERT INTO test2 VALUES (5);
INSERT INTO test2 VALUES (6);
INSERT INTO test2 VALUES (3);
```

```
mysql> INSERT INTO test2 VALUES (5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO test2 VALUES (6);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO test2 VALUES (3);
Query OK, 1 row affected (0.00 sec)
```

1.6 데이터 확인 1

```
SELECT * FROM test1;
```

```
mysql> select * from test1;
+-----+
| no    |
+-----+
|      1 |
|      2 |
|      3 |
+-----+
3 rows in set (0.00 sec)
```

1.7 데이터 확인 2

```
SELECT * FROM test2;
```

```
mysql> select * from test2;
+-----+
| no    |
+-----+
|      5 |
|      6 |
|      3 |
+-----+
3 rows in set (0.00 sec)
```

2 UNION

여러 개의 SQL문을 합쳐서 하나의 SQL 문으로 만들어주는 방법 (주의. 칼럼의 개수가 같아야함)

2.1 UNION 문법

- UNION : 중복된 값을 제거하여 알려준다.
- UNION ALL : 중복된 값도 모두 보여준다.

```
SELECT column1, column2, ... FROM tableA
UNION | UNION ALL
SELECT column1, column2, ... FROM tableB;
```

2.2 UNION 예제 1 - UNION ALL

test1 의 모든 데이터와 test2 의 모든 데이터를 중복된 값을 포함하여 검색

```
SELECT * FROM test1
UNION ALL
SELECT * FROM test2;
```

```
mysql> select * from test1
[
  -> UNION ALL
[
  -> select * from test2;
+-----+
| no    |
+-----+
| 1     |
| 2     |
| 3     |
| 5     |
| 6     |
| 3     |
+-----+
6 rows in set (0.00 sec)
```

2.3 UNION 예제 1 - UNION

test1 의 모든 데이터와 test2 의 모든 데이터를 중복된 값을 제거하여 검색

```
SELECT * FROM test1
UNION
SELECT * FROM test2;
```

```
[mysql> select * from test1
[      -> UNION
[      -> select * from test2;
```

no
1
2
3
5
6

5 rows in set (0.00 sec)

2.4 UNION 예제 2 - UNION ALL

성별이 여자인 데이터를 검색하는 쿼리와 소속사가 YG엔터테인먼트인 데이터를 검색하는 쿼리를 UNION ALL 로 실행

```
SELECT name, sex, agency FROM celeb WHERE sex = 'F'
UNION ALL
SELECT name, sex, agency FROM celeb WHERE agency = 'YG엔터테인먼트';
```

```
mysql> SELECT name, sex, agency FROM celeb WHERE sex = 'F'
-> UNION ALL
-> SELECT name, sex, agency FROM celeb WHERE agency = 'YG엔터테이먼트';
```

name	sex	agency
아이유	F	EDAM엔터테이먼트
이미주	F	올림엔터테이먼트
이수현	F	YG엔터테이먼트
강동원	M	YG엔터테이먼트
차승원	M	YG엔터테이먼트
이수현	F	YG엔터테이먼트

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

2.5 UNION 예제 2 - UNION

성별이 여자인 데이터를 검색하는 쿼리와 소속사가 YG엔터테이먼트인 데이터를 검색하는 쿼리를 UNION 으로 실행

```
SELECT name, sex, agency FROM celeb WHERE sex = 'F'
UNION
SELECT name, sex, agency FROM celeb WHERE agency = 'YG엔터테이먼트';
```

```
mysql> SELECT name, sex, agency FROM celeb WHERE sex = 'F'
-> UNION
-> SELECT name, sex, agency FROM celeb WHERE agency = 'YG엔터테이먼트';
```

name	sex	agency
아이유	F	EDAM엔터테이먼트
이미주	F	올림엔터테이먼트
이수현	F	YG엔터테이먼트
강동원	M	YG엔터테이먼트
차승원	M	YG엔터테이먼트

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

2.6 UNION 예제 3

가수가 직업인 연예인의 이름, 직업을 검색하는 쿼리와, 1980년대에 태어난 연예인의 이름, 생년월일, 나이를 검색하는 쿼리를 UNION으로 실행

2.7 UNION 예제 3 - 1

가수가 직업인 연예인의 이름, 직업을 검색하는 쿼리와, 1980년대에 태어난 연예인의 이름, 생년월일, 나이를 검색하는 쿼리를 UNION으로 실행

```
SELECT name, job_title FROM celeb WHERE job_title LIKE '%가수%';
```



```
mysql> SELECT name, job_title FROM celeb WHERE job_title LIKE '%가수%';
```

name	job_title
아이유	가수, 텔런트
이미주	가수
이수현	가수

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

2.8 UNION 예제 3-2

가수가 직업인 연예인의 이름, 직업을 검색하는 쿼리와, **1980년대에 태어난 연예인의 이름, 생년월일, 나이를 검색하는 쿼리**를 UNION으로 실행

```
SELECT name, birthday, age FROM celeb
WHERE birthday BETWEEN '1980-01-01' AND '1989-12-31';
```

```
mysql> SELECT name, birthday, age FROM celeb
-> WHERE birthday BETWEEN '1980-01-01' AND '1989-12-31';
```

name	birthday	age
강동원	1981-01-18	41

```
1 row in set (0.00 sec)
```

2.9 UNION 예제 3-3

가수가 직업인 연예인의 이름, 직업을 검색하는 쿼리와, **1980년대에 태어난 연예인의 이름, 생년월일, 나이를 검색하는 쿼리**를 UNION으로 실행

```
SELECT name, job_title FROM celeb WHERE job_title LIKE '%가수%'
UNION
SELECT name, birthday, age FROM celeb
WHERE age BETWEEN '1980-01-01' AND '1989-12-31';
```

```
mysql> SELECT name, job_title FROM celeb WHERE job_title LIKE '%가수%'
-> UNION
-> SELECT name, birthday, age FROM celeb
-> WHERE age BETWEEN '1980-01-01' AND '1989-12-31';
ERROR 1222 (21000): The used SELECT statements have a different number of columns
```

3 연습

1. 직업이 가수인 (가수를 포함하는) 데이터를 검색하는 쿼리와 직업이 텔런트인 (텔런트를 포함하는) 데이터를 검색하는 쿼리를 중복을 제거하여 합쳐서 실행하세요.

2. 성이 이씨인 데이터를 검색하는 쿼리와 1970년대생을 검색하는 쿼리를 중복을 포함하여 합쳐서 실행하세요.

<주의> 혹시 실행중에 birthday 칼럼이름이 brithday 로 되어있는 경우, 칼럼명 변경 방법을 상기시켜서 변경 실습을 먼저함