20200507 DB과제 기초4

B반 신석환

#실습 1

[실습 #1]

1. 학생 명부 테이블 구성하기
   1) No, 이름, 학교, 학년 등 Column 7개 이상
   2) 아래 조건대로 출력하기
      (1) 현재 상태 그대로 테이블 구조 출력
      (2) 학년 기준으로 오름차순으로 출력하기
      (3) 학교를 기준으로 그룹화 하기
      (4) 학교를 기준으로 그룹화하고 내림차순으로 출력하기

1

```
mysql> CREATE TABLE STUDENT(
    -> NO INT NOT NULL AUTO_INCREMENT PRIMARY KEY, NAME CHAR(10), SCHOOL CHAR(10),
GRADE CHAR(10), ADDRESS CHAR(20), PHONE_NUMBER CHAR(20), ETC CHAR(20)
    -> );
Query OK, 0 rows affected (0.52 sec)

mysql> DESC STUDENT;
+--------------+---------+------+-----+---------+----------------+
| Field        | Type    | Null | Key | Default | Extra          |
+--------------+---------+------+-----+---------+----------------+
| NO           | int     | NO   | PRI | NULL    | auto_increment |
| NAME         | char(10)| YES  |     | NULL    |                |
| SCHOOL       | char(10)| YES  |     | NULL    |                |
| GRADE        | char(10)| YES  |     | NULL    |                |
| ADDRESS      | char(20)| YES  |     | NULL    |                |
| PHONE_NUMBER | char(20)| YES  |     | NULL    |                |
| ETC          | char(20)| YES  |     | NULL    |                |
+--------------+---------+------+-----+---------+----------------+
7 rows in set (0.09 sec)

mysql> INSERT INTO STUDENT(NAME, SCHOOL, GRADE, ADDRESS, PHONE_NUMBER, ETC) VALUES
    -> ('William', 'jamsil', '3', 'Seoul', '01011112222', ''),
    -> ('Shin', 'seoungnam', '2', 'SeoungNam', '01022223333', ''),
    -> ('Seok', 'jeju', '2', 'jeju', '01033334444', ''),
    -> ('Hwan', 'busan', '1', 'deagu', '01044445555', ''),
    -> ('Juan', 'jamsil', '2', 'Seoul', '01055556666', ''),
    -> ('Lee', 'jeju', '3', 'canada', '01066667777', '');
Query OK, 6 rows affected (0.16 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> select * from student;
+----+---------+----------+-------+-----------+--------------+------+
| NO | NAME    | SCHOOL   | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+----+---------+----------+-------+-----------+--------------+------+
|  1 | William | jamsil   | 3     | Seoul     | 01011112222  |      |
|  2 | Shin    | seoungnam| 2     | SeoungNam | 01022223333  |      |
|  3 | Seok    | jeju     | 2     | jeju      | 01033334444  |      |
|  4 | Hwan    | busan    | 1     | deagu     | 01044445555  |      |
|  5 | Juan    | jamsil   | 2     | Seoul     | 01055556666  |      |
|  6 | Lee     | jeju     | 3     | canada    | 01066667777  |      |
+----+---------+----------+-------+-----------+--------------+------+
6 rows in set (0.00 sec)
```

2

```
mysql> select * from student order by grade;
+----+---------+-----------+-------+-----------+--------------+------+
| NO | NAME    | SCHOOL    | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+----+---------+-----------+-------+-----------+--------------+------+
|  4 | Hwan    | busan     | 1     | deagu     | 01044445555  |      |
|  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
|  5 | Juan    | jamsil    | 2     | Seoul     | 01055556666  |      |
|  1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|  6 | Lee     | jeju      | 3     | canada    | 01066667777  |      |
+----+---------+-----------+-------+-----------+--------------+------+
6 rows in set (0.00 sec)
```

3

```
mysql> select * from student group by school;
+----+---------+-----------+-------+-----------+--------------+------+
| NO | NAME    | SCHOOL    | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+----+---------+-----------+-------+-----------+--------------+------+
|  1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
|  4 | Hwan    | busan     | 1     | deagu     | 01044445555  |      |
+----+---------+-----------+-------+-----------+--------------+------+
4 rows in set (0.00 sec)
```

4

```
mysql> select * from student group by school order by school desc;
+----+---------+-----------+-------+-----------+--------------+------+
| NO | NAME    | SCHOOL    | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+----+---------+-----------+-------+-----------+--------------+------+
|  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
|  1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|  4 | Hwan    | busan     | 1     | deagu     | 01044445555  |      |
+----+---------+-----------+-------+-----------+--------------+------+
4 rows in set (0.00 sec)
```

#실습 2

[실습 #2]

1. 전화번호부의 즐겨찾기 테이블 구성하기

 1) No(AUTO_INCREMENT, NOT NULL, PRIMARY KEY), 이름, 전화번호 필드 생성

 2) 아래 조건대로 출력하기

  (1) 현재 상태의 테이블 구조 출력

  (2) 'No'를 제외한 이름과 전화번호만 입력하여 데이터 5개 입력 후 내용 전체 출력

1

```
mysql> desc numbers;
+--------+----------+------+-----+---------+----------------+
| Field  | Type     | Null | Key | Default | Extra          |
+--------+----------+------+-----+---------+----------------+
| No     | int      | NO   | PRI | NULL    | auto_increment |
| name   | char(20) | YES  |     | NULL    |                |
| number | char(20) | YES  |     | NULL    |                |
+--------+----------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> insert into numbers(name, number) values
    -> ('Shin', '01011112222'), ('Seok', '01022223333'), ('Hwan', '01033334444'), ('William', '01044445555'), ('Juan', '01055556666');
Query OK, 5 rows affected (0.10 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from numbers;
+----+---------+-------------+
| No | name    | number      |
+----+---------+-------------+
|  1 | Shin    | 01011112222 |
|  2 | Seok    | 01022223333 |
|  3 | Hwan    | 01033334444 |
|  4 | William | 01044445555 |
|  5 | Juan    | 01055556666 |
+----+---------+-------------+
5 rows in set (0.00 sec)
```

2

#실습 3

[실습 #3]

1. 물품 가격 정리 테이블 구축하기

 1) No(AUTO_INCREMENT, NOT NULL, PRIMARY KEY), 물품명, 가격 필드 생성

 2) 데이터를 아래와 같이 입력하기

  (1) 대소문자를 섞은 물품명을 활용하여 데이터 5개 입력

  (2) 1번에서 입력한 똑같은 물품명을 소문자로만 하여 데이터 5개 입력

 3) 아래 조건대로 출력하기

  (1) 대소문자 구분없이 전체 목록 출력하기

  (2) 대소문자 구분하여 데이터 출력해보기(5개)

1

```
mysql> select * from price;
+----+---------+-------+
| No | name    | price |
+----+---------+-------+
|  1 | Pencil  |  1000 |
|  2 | pEn     |  2000 |
|  3 | noTe    |   500 |
|  4 | maSk    |  1500 |
|  5 | Shose   |  5000 |
|  6 | pencil  |  1000 |
|  7 | pen     |  2000 |
|  8 | note    |   500 |
|  9 | mask    |  1500 |
| 10 | shoes   |  5000 |
+----+---------+-------+
10 rows in set (0.00 sec)
```

2

```
mysql> select * from price where binary(name) like '%P%';
+----+---------+-------+
| No | name    | price |
+----+---------+-------+
|  1 | Pencil  |  1000 |
+----+---------+-------+
1 row in set (0.00 sec)

mysql> select * from price where binary(name) like '%S%';
+----+-------+-------+
| No | name  | price |
+----+-------+-------+
|  4 | maSk  |  1500 |
|  5 | Shose |  5000 |
+----+-------+-------+
2 rows in set (0.00 sec)
```

2

```
mysql> select * from price where binary(name) like '%T%';
+----+------+-------+
| No | name | price |
+----+------+-------+
|  3 | noTe |   500 |
+----+------+-------+
1 row in set (0.00 sec)

mysql> select * from price where binary(name) like '%E%';
+----+------+-------+
| No | name | price |
+----+------+-------+
|  2 | pEn  |  2000 |
+----+------+-------+
1 row in set (0.00 sec)

mysql> select * from price where binary(name) like '%e%';
+----+--------+-------+
| No | name   | price |
+----+--------+-------+
|  1 | Pencil |  1000 |
|  3 | noTe   |   500 |
|  5 | Shose  |  5000 |
|  6 | pencil |  1000 |
|  7 | pen    |  2000 |
|  8 | note   |   500 |
| 10 | shoes  |  5000 |
+----+--------+-------+
7 rows in set (0.00 sec)
```

#복습

[복습]

1. 회원관리 테이블 구축하기

 1) No(AUTO_INCREMENT, NOT NULL, PRIMARY KEY), 이름, 전화번호, 생년월일 등

 2) 칼럼 최소 7개 생성

 3) 데이터는 아래와 같이 입력하기

  (1) 대소문자를 섞은 이름으로 5개 입력

  (2) 1번에서 입력한 이름을 소문자로만 5개 입력

 4) 아래 조건대로 출력하기

  (1) 대소문자 구분없이 전체 목록 출력하기

  (2) 대소문자 구분하여 데이터 출력해보기(5개)

  (3) 맨마지막에 'etc' 칼럼을 생성하여 테이블 구조 출력

1

```
mysql> select * from customers;
+----+---------+-------------+--------+------+-----------------+---------+
| No | name    | number      | birday | sex  | customer_number | level   |
+----+---------+-------------+--------+------+-----------------+---------+
|  1 | Shin    | 01011112222 | 871211 | M    | 111111          | new     |
|  2 | sEok    | 01033332222 | 891201 | M    | 111112          | new     |
|  3 | hwAn    | 01033334444 | 990201 | F    | 121112          | element |
|  4 | William | 01033335555 | 690201 | M    | 122112          | mid     |
|  5 | juaN    | 01011115555 | 590201 | F    | 122132          | high    |
|  6 | shin    | 01011112222 | 871211 | M    | 111111          | new     |
|  7 | seok    | 01033332222 | 891201 | M    | 111112          | new     |
|  8 | hwan    | 01033334444 | 990201 | F    | 121112          | element |
|  9 | william | 01033335555 | 690201 | M    | 122112          | mid     |
| 10 | juan    | 01011115555 | 590201 | F    | 122132          | high    |
+----+---------+-------------+--------+------+-----------------+---------+
10 rows in set (0.00 sec)
```

2

```
mysql> select * from customers where binary(name) like '%S%';
+----+------+-------------+--------+------+-----------------+-------+
| No | name | number      | birday | sex  | customer_number | level |
+----+------+-------------+--------+------+-----------------+-------+
|  1 | Shin | 01011112222 | 871211 | M    | 111111          | new   |
+----+------+-------------+--------+------+-----------------+-------+
1 row in set (0.00 sec)

mysql> select * from customers where binary(name) like '%s%';
+----+------+-------------+--------+------+-----------------+-------+
| No | name | number      | birday | sex  | customer_number | level |
+----+------+-------------+--------+------+-----------------+-------+
|  2 | sEok | 01033332222 | 891201 | M    | 111112          | new   |
|  6 | shin | 01011112222 | 871211 | M    | 111111          | new   |
|  7 | seok | 01033332222 | 891201 | M    | 111112          | new   |
+----+------+-------------+--------+------+-----------------+-------+
3 rows in set (0.00 sec)

mysql> select * from customers where binary(name) like '%W%';
+----+---------+-------------+--------+------+-----------------+-------+
| No | name    | number      | birday | sex  | customer_number | level |
+----+---------+-------------+--------+------+-----------------+-------+
|  4 | William | 01033335555 | 690201 | M    | 122112          | mid   |
+----+---------+-------------+--------+------+-----------------+-------+
1 row in set (0.00 sec)

mysql> select * from customers where binary(name) like '%a%';
+----+---------+-------------+--------+------+-----------------+---------+
| No | name    | number      | birday | sex  | customer_number | level   |
+----+---------+-------------+--------+------+-----------------+---------+
|  4 | William | 01033335555 | 690201 | M    | 122112          | mid     |
|  5 | juaN    | 01011115555 | 590201 | F    | 122132          | high    |
|  8 | hwan    | 01033334444 | 990201 | F    | 121112          | element |
|  9 | william | 01033335555 | 690201 | M    | 122112          | mid     |
| 10 | juan    | 01011115555 | 590201 | F    | 122132          | high    |
+----+---------+-------------+--------+------+-----------------+---------+
5 rows in set (0.00 sec)

mysql> select * from customers where binary(name) like '%m%';
+----+---------+-------------+--------+------+-----------------+-------+
| No | name    | number      | birday | sex  | customer_number | level |
+----+---------+-------------+--------+------+-----------------+-------+
|  4 | William | 01033335555 | 690201 | M    | 122112          | mid   |
|  9 | william | 01033335555 | 690201 | M    | 122112          | mid   |
+----+---------+-------------+--------+------+-----------------+-------+
2 rows in set (0.00 sec)
```


```
mysql> select * from customers;
+----+---------+-------------+--------+------+-----------------+---------+------+
| No | name    | number      | birday | sex  | customer_number | level   | etc  |
+----+---------+-------------+--------+------+-----------------+---------+------+
|  1 | Shin    | 01011112222 | 871211 | M    | 111111          | new     | NULL |
|  2 | sEok    | 01033332222 | 891201 | M    | 111112          | new     | NULL |
|  3 | hwAn    | 01033334444 | 990201 | F    | 121112          | element | NULL |
|  4 | William | 01033335555 | 690201 | M    | 122112          | mid     | NULL |
|  5 | juaN    | 01011115555 | 590201 | F    | 122132          | high    | NULL |
|  6 | shin    | 01011112222 | 871211 | M    | 111111          | new     | NULL |
|  7 | seok    | 01033332222 | 891201 | M    | 111112          | new     | NULL |
|  8 | hwan    | 01033334444 | 990201 | F    | 121112          | element | NULL |
|  9 | william | 01033335555 | 690201 | M    | 122112          | mid     | NULL |
| 10 | juan    | 01011115555 | 590201 | F    | 122132          | high    | NULL |
+----+---------+-------------+--------+------+-----------------+---------+------+
10 rows in set (0.00 sec)
```

#예습과제1 - Join

Join이란?

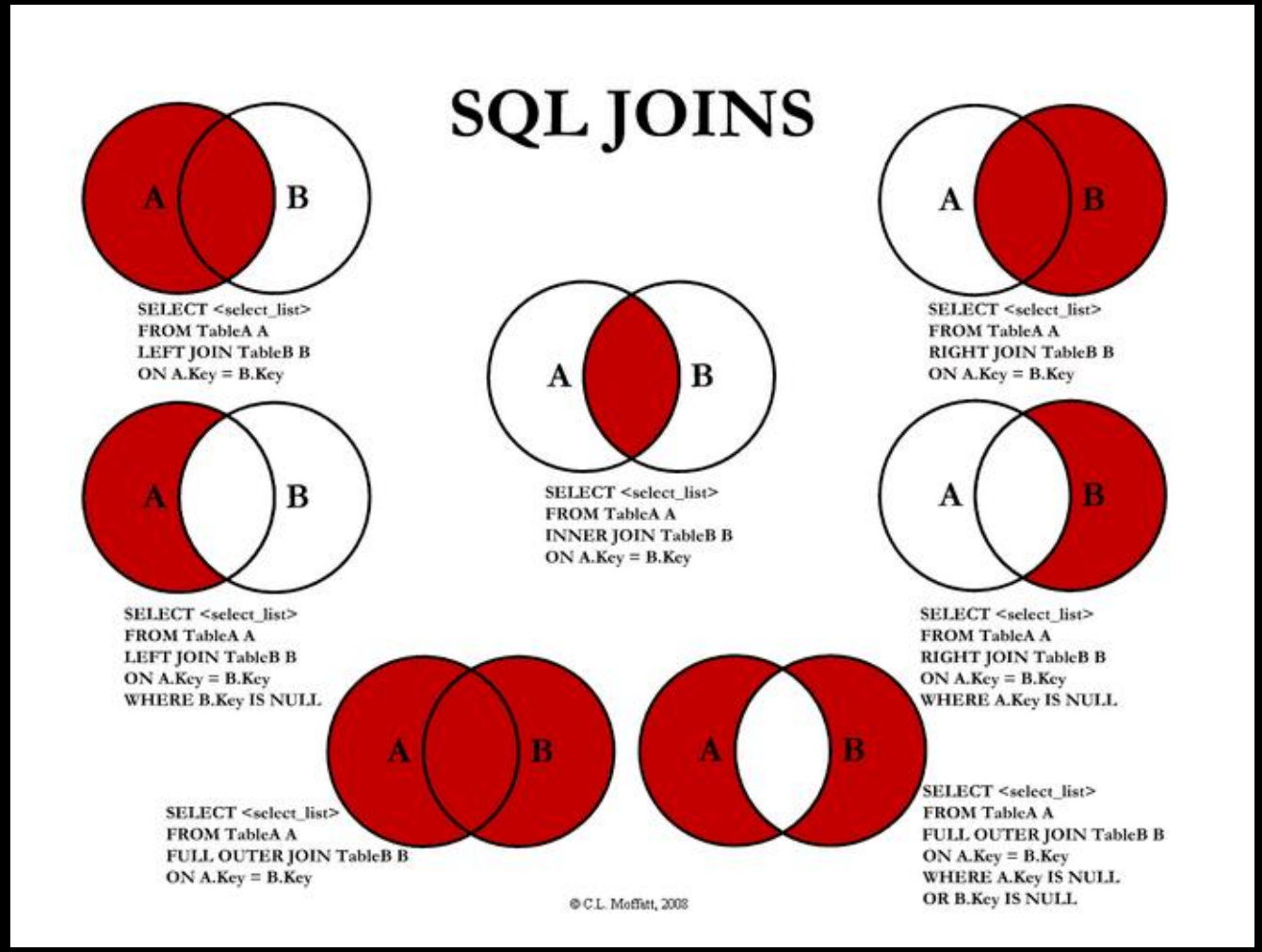
두 개 이상의 테이블이나 데이터베이스를 연결하여 데이터를 검색하는 방법

-INNER JOIN
쉽게 말해 교집합.
기준테이블과 Join한 테이블의 중복값을 보여준다

-LEFT OUTER JOIN
기준테이블의 값 + INNER JOIN값

-RIGHT OUTER JOIN
LEFT OUTER JOIN의 반대

-FULL OUTER JOIN
합집합 두 테이블의 데이터 모두를 검색함



#예습과제1 - Union

UNION이란?

Union은 여러 개의 SQL 문을 합쳐 하나의 SQL 문으로 만들어주는 방법입니다.
두개의 쿼리의 합집합을 만들어준다고 생각하면 될 듯합니다.

Union 과 UnionAll 의 차이 점

하지만 Union은 두 쿼리의 결과의 중복값을 제거해서 보여주고,
UnionAll은 중복된값도 전부 다 보여준다는 차이점이 있습니다.

Union 사용시 주의점
- 칼럼명이 같아야 한다. (같지 않을 경우 AS를 사용하여 같게 만들어주면 됨)
- 칼럼 별 데이터 타입이 같아야 합니다

#예습과제1 - SubQuery

SubQuery이란?

서브쿼리(Subquery)란 하나의 SQL 문 안에 포함되어 있는 또 다른 SQL 문을 말한다.
서브쿼리는 메인쿼리가 서브쿼리를 포함하는 종속적인 관계이다.

```
#메인쿼리
SELECT *
FROM db_table
WHERE table_fk IN (
          #서브쿼리
          SELECT table_fk FROM db_table_other WHERE ..
          )
```

Subquery를 사용 할 수 있는 곳

- SELECT, FROM, WHERE, HAVING, ORDER BY, INSERT문의 VALUES, UPDATE문의 SET

#예습과제1 - SubQuery

서브쿼리의 종류

① 단일행 서브쿼리(Single Row Subquery)

```
SELECT *
FROM Player
WHERE Team_ID = (
            SELECT Team_ID
            FROM Player
            WHERE Player_name = "yonglae"
            )
ORDER BY Team_name;
```

② 다중행 서브쿼리(Multiple Row Subquery)

```
SELECT *
FROM Team
WHERE Team_ID IN (
            SELECT Team_ID
            FROM Player
            WHERE Player_name = "yonglae"
            )
ORDER BY Team_name;
```

③ 다중컬럼 서브쿼리(Multi Column Subquery)

```
SELECT *
FROM Player
WHERE (Team_ID, Height) IN (
            SELECT Team_ID, MIN(Height)
            FROM Player
            GROUP BY Team_ID
            )
ORDER BY Team_ID, Player_name;
```