SQL Basic Cheat Sheet

SQL 쿼리 순서

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. ORDER BY
6. LIMIT

1. 데이터 가져오기

```
SELECT * FROM table1;
SELECT col1,col2,col3
FROM table1;
```

2. 조건에 맞는 데이터 검색하기

1) 조건문 (WHERE)

```
SELECT *
FROM table1
WHERE col1 > 30;
```

2) 비교연산자 (=, >, >=, <, <=, <>)

```
SELECT * FROM table1
WHERE col1 = 30;
```

(col1 값이 30을 가짐)

```
SELECT * FROM table1
WHERE col1 <> 30;
```

(col1값이 30을 제외한 모두)
(<> 대신 != 도 가능)

문자도 가능

```
SELECT * FROM table1
WHERE  col2 < 'B';
```

(알파벳 순서에 따라 문자 'B' 이전 데이터들만)
(즉, 'A'로 시작하는 모든 데이터가 검색)

3) 논리연산자 (AND, OR)

```
SELECT * FROM table1
WHERE col3 = 'London'
AND col1 > 30;

SELECT * FROM Customers
WHERE col1 <= 3 OR col1 >= 30;
```

4) LIKE

```
SELECT * FROM table1
WHERE col4 LIKE 'Qu%';
```

- %, _ 기호 포함하고 싶을 때는
  이스케이프 문자(\ ) 이용

- NOT LIKE 'a%' : 'a'로 시작하지 않는 ~

- LIKE 심화
  % : 와일드 카드
  _ : 1글자 와일드카드

- SELECT * FROM table 1
  WHERE col4 = 'Q___'
  'Q'로 시작하되, 뒤에 3개의 문자가 뒤따라옴

5) IN, BETWEEN

```
SELECT * FROM table1
WHERE col3 IN ('London','Seoul');

SELECT * FROM table1
WHERE col1 BETWEEN 3 AND 10;
```

6) IS NULL

```
SELECT * FROM table1
WHERE col4 IS NULL;
```

3. 보고싶은 데이터 요약하기

집계 함수

sample

Id	Name	Visits
1	A	1
2	A	2
3	B	3
4	C	5
5	NULL	NULL

1) COUNT

```
SELECT COUNT(*) FROM sample;
> 5
SELECT COUNT(Name)
FROM sample;
> 4
SELECT COUNT(DISTINCT Name)
FROM sample;
> 3
```

2) SUM

```
SELECT SUM(Visits)
FROM sample;
```

3) AVG

```
SELECT AVG(Visits)
FROM sample;
> (1+2+3+5) / 4 = 2.75
SELECT SUM(Visits)/COUNT(*)
FROM sample;
> (1+2+3+5) / 5 = 2.2
```

4) MAX/MIN

```
SELECT MAX(Visits)
FROM sample;
SELECT MIN(Visits)
FROM sample;
```

2) GROUP BY

```
SELECT CategoryID,SUM(Price)
FROM Products
GROUP BY CategoryID;
```

(그룹화의 기준이 되는 컬럼은
SELECT 구문에 반드시 적어주기)

2) HAVING

```
SELECT CategoryID,COUNT(*)
FROM Products
GROUP BY CategoryID
HAVING COUNT(*) <= 10
```

4. 데이터 순서 정렬하기

ORDER BY

오름차순(Default)

```
ASC ascending
```

내림차순

```
DESC descending
```

```
SELECT *
FROM Products
WHERE Price >= 20
ORDER BY price DESC;
```

(Price가 20이상인 값들 중 비싼 순으로 정렬)

LIMIT

```
SELECT *
FROM Products
ORDER BY price DESC
LIMIT 1;
```

(가장 비싼 물건 1개 출력)

SQL Basic Cheat Sheet

SQL 문자열 자르기

```
SELECT LEFT("20140323", 4)
FROM sample;
> 2014

SELECT RIGHT("20140323", 4)
FROM sample;
> 0323

SELECT SUBSTR("20140323", 1, 4)
FROM sample;
> 2014

SELECT SUBSTR("20140323", 3)
FROM sample;
> 140323
```

SQL 소수점 처리

```
SELECT CEIL(5.5)
> 6

SELECT FLOOR(5.5)
> 5

SELECT ROUND(5.5569, 2)
> 5.56
```

MYSQL 시간 더하기, 빼기

```
SELECT DATE_ADD(NOW(), INTERVAL 1 DAY)

SELECT DATE_SUB(NOW(), INTERVAL 1 SECOND)
```

CASE문 만들기

table1

Id	Color
1	Red
2	Blue
Null	Unspecified

```
SELECT Id
     , CASE Id
            WHEN 1 THEN 'Red'
            WHEN 2 THEN 'Blue'
            ELSE 'Unspecified'
       END AS Color
FROM table1;
```

- ELSE를 생략할 경우에는 ELSE NULL이 자동으로 지정
  WHEN절의 조건에 아무것도 부합하지 않은 데이터가 있는 경우
  ELSE 절에 값을 지정해주지 않으면 해당 값은 자동으로 NULL값 반환

TABLE PIVOT

sample

categoryid	price
1	3
1	4
2	70
2	60

```
SELECT AVG(CASE
               WHEN categoryid = 1 THEN price
               ESLE NULL
           END) AS category1_avg_price
     , AVG(CASE
               WHEN categoryid = 2 THEN price
               ELSE NULL
           END) AS category2_avg_price
FROM sample;
```

→

category1_avg_price	category2_avg_price
3.5	65

- 세로로 표시되는 테이블 결과물을 가로로 보고싶을 때 사용하는 쿼리
  CASE문을 이용하여 각각의 컬럼에 맞는 데이터만 출력하고
  나머지는 null 값을 가지도록 하여, 각 컬럼에서 보고싶은
  연산(COUNT, SUM, AVG 등등)의 결과를 보여줌

SQL UNION

Users_recent

Id	Contact	Address
A	010-****-1234	광진구 자양동
B	010-****-7700	송파구 석촌동

Users_past

Id	Contact	Address
C	010-****-4676	분당구 판교동

1) UNION

- 컬럼명이 같아야한다. (같지 않을 경우 AS를 사용하여 같게 만들어주면 됨)
- 컬럼별 데이터타입이 같아야한다.
- 두 테이블에 중복으로 들어있는 데이터는 모두 표시하지 않고
  하나의 데이터만 표시 (DISTINCT)

```
SELECT *
FROM Users_recent

UNION

SELECT *
FROM Users_past;
```

Id	Contact	Address
A	010-****-1234	광진구 자양동
B	010-****-7700	송파구 석촌동
C	010-****-4676	분당구 판교동

2) UNION ALL

- 중복되는 데이터들을 생략하지 않고 원본의 모든 데이터를 단순히 이어붙인 결과

```
SELECT *
FROM Users_recent

UNION ALL

SELECT *
FROM Users_past;
```

Id	Contact	Address
A	010-****-1234	광진구 자양동
B	010-****-7700	송파구 석촌동
C	010-****-4676	분당구 판교동
B	010-****-7700	송파구 석촌동

- MySQL에는 지원되지 않지만, SQL을 이용해 교집합, 차집합도 구할 수 있다.
- 교집합은 INTERSECT, 차집합은 EXCEPT를 사용

SQL Basic Cheat Sheet

leetcode 183. Customers Who Never Order 참고

Customers

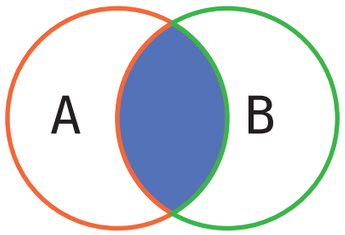
Id	Name
1	Joe
2	Henry
3	Sam
4	Max

Orders

Id	CustomerID
1	3
2	1

SQL JOIN

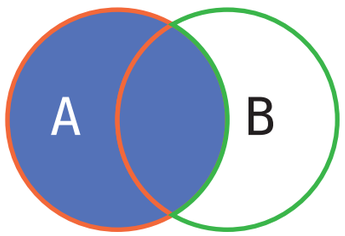
1) INNER JOIN



Id	Name	orderid
1	Joe	2
3	Sam	1

```
SELECT C.Id, C.Name, O.Id AS orderid,
FROM Customers AS C
     INNER JOIN Orders AS O ON C.Id = O.CustomerId;
```

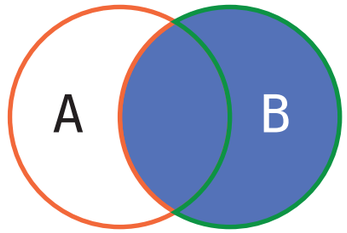
2) LEFT OUTER JOIN



Id	Name	orderid
3	Sam	1
1	Joe	2
2	Henry	null
4	Max	null

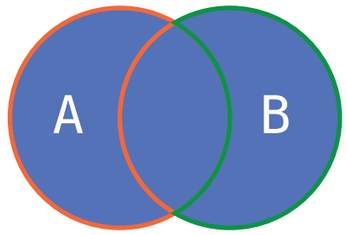
```
SELECT C.Id, C.Name, O.Id AS orderid
FROM Customers AS C
     LEFT JOIN Orders AS O ON C.Id = O.CustomerId;
```

3) RIGHT OUTER JOIN




```
SELECT C.Id, C.Name AS Customers
FROM Customers AS C
     RIGHT JOIN Orders AS O ON C.Id = O.CustomerId;
```

4) FULL OUTER JOIN




```
SELECT C.Id, C.Name AS Customers
FROM Customers AS C
     LEFT JOIN Orders AS O ON C.Id = O.CustomerId

UNION

SELECT C.ID, C.Name AS Customers
FROM Customers AS C
     RIGHT JOIN Orders AS O ON C.Id = O.CustomerId;
```