



# 23.05.10-AI7기

## 1. 지난시간 복습!

SQL 명령어 문법순서

### 2. DML

### 3. ORDER BY / SORT , SORTED

퀴즈1— [미션3]발언권이 강한 주주는?

퀴즈2 - [미션2]평균구하기

### 4. JOIN

### 5. GROUP BY/ HAVING

집계함수?

퀴즈3— [미션1]가게db분석하기

퀴즈4

퀴즈5 —[미션2] 인기있는물건

프로그래머스 SQL KIT

## 1. 지난시간 복습!

### 1. 이론

\* MySQL은 ( RDBMS / NoSQL ) 이다.

: RDBMS

\* ( ) 란 행과 열의 2차원으로 이뤄진 데이터형식이다.

: DB (RDB) , <테이블>

### 2. DDL

\* 테이블의 속성(열)/빠대는 그대로 남기고 데이터만 초기화하고 싶을때 사용하는 명령어는?

TRUNCATE ( 비교: DROP, DELETE )

\* PRIMARY KEY가 가진 특성 2가지는 무엇인가?

: 중복X(고유하다) , NOT NULL

1 1 2 3 3 4 3 5

: 1 2 3 4 5

### 3. DML

\* 처음보는 자료를 확인할때 사용하는 쿼리. 모든 정보를 조회한다.

(테이블명 사용할것)

: SELECT \* FROM 테이블명 ;

#### 4. WHERE

----- copy + paste

```
create table market (  
  id int,  
  name varchar(15),  
  score float ,  
  stock int);
```

```
insert into market values (1000, 'apple', 77.5, 33) ,  
(1001, 'banana' , 80.0, 50) , (1002, 'cat', 95.5, 11),  
(1003, 'doll', 65.5, 5), ( 1004, 'egg', 77.5, 20) ,  
(1005, 'fish', 79.5, 25) ,( 1006, 'goose', 83.5, 8);  
-----
```

\* score가 70 ~ 80인 id, name을 구하라.

```
: select id , name from market where score between 70 and 80;  
      where score >=70 and score<=80;
```

\* name의 중간에 'o'가 들어가는 name, score를 구하라. % 와일드카드

```
: select name, score from market where name like '%o%' ;
```

\* 이름이 ('apple', 'cat' ) 인 모든정보를 구하라.

```
: select * from market where name = 'apple' or name='cat';  
      where name in ('apple','cat');
```

\* score/stock을 반올림하여 소수 2번째자리까지 나타내는 새로운 칼럼 ratio 을 만들어라.

```
: select *, round( (score/stock), 2 ) as ratio from market ;
```

## SQL 명령어 문법순서

# 문법순서

```
SELECT 컬럼명 - FROM 테이블명 - WHERE 테이블 조건  
- GROUP BY 컬럼명 HAVING 그룹 조건 - ORDER BY 컬럼명 ;
```

## 2. DML

- SELECT ~ / INSERT INTO ~ VALUES ()
- UPDATE : 데이터 수정

```
update 테이블명 set ( ) ;
```

- DELETE : 데이터 삭제

```
delete from 테이블명 조건;
```

### 3. ORDER BY / SORT , SORTED

고유/중복없음 : UNIQUE , DISTINCT

정렬: SORT, SORTED, ORDER BY

ascending : 오름차순 출석, 순위 1, 2, 3, 4, 5/ descending : 내림차순 : 성적 100 , 98, 93, 88

- ORDER BY (ASC 기본) / (DESC : descending)

```
SELECT * FROM 테이블명 ORDER BY 칼럼 ;
```

```
SELECT * FROM 테이블명 ORDER BY 칼럼 DESC;
```

\*\* DDL의 DESC 와 비교!

DESC : describe

```
DESC 테이블명 ;
```

### 퀴즈1— [미션3]발언권이 강한 주주는?

1. agree, name 기준으로 오름차순
2. agree 내림차순, stock은 오름차순

```
select * from shareholder order by agree, name ;
```

```
select * from shareholder order by agree desc, stock ;
```

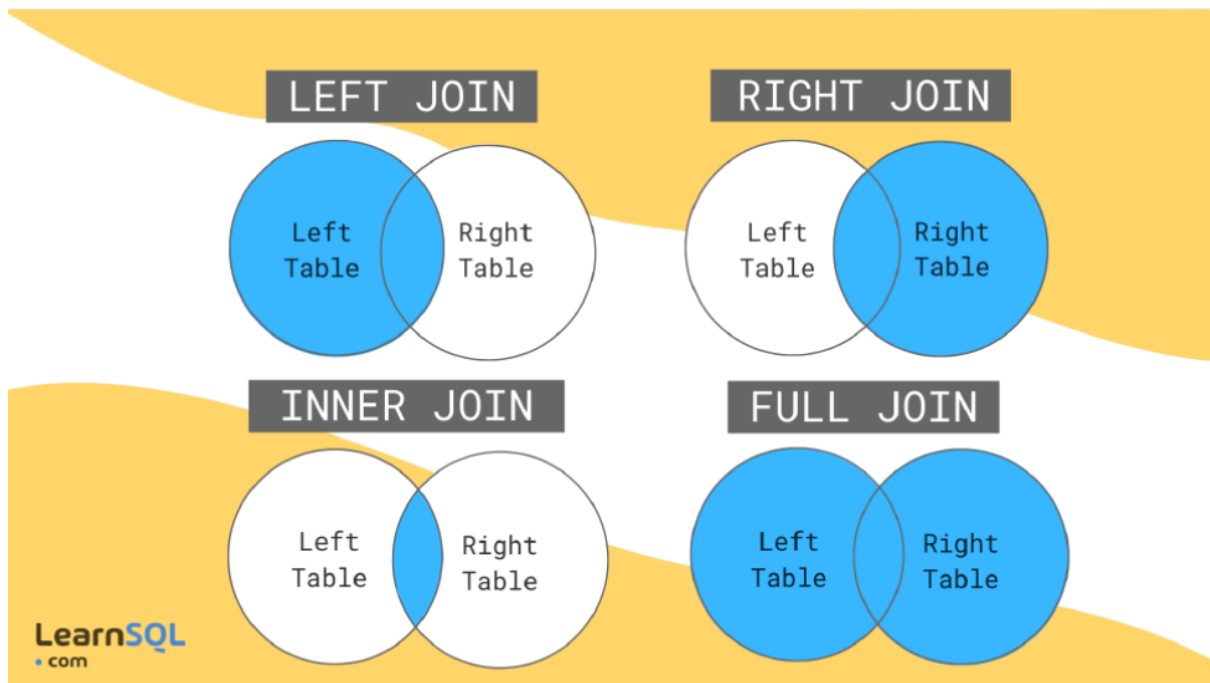
## 퀴즈2 - [미션2]평균구하기

-- 1. 영어 내림차순, 수학 내림차순으로 조회하기

```
select * from test order by eng desc, math desc;
```

## 4. JOIN

SQL 포트폴리오: big query(구글 빅쿼리) / SQLD 자격증



```
SELECT * FROM 테이블A LEFT JOIN 테이블B ON A.칼럼명=B.칼럼명;
```

### • [미션1] 판매기록 조회하기

```
select * from sale
right join product
on sale.product_id=product.id;
```

```
select * from sale A
right join product B
on A.product_id=B.id;
```

## 5. GROUP BY/ HAVING

- select /from /where/ group by /having /order by 순서

### 집계함수?

: 수학적함수: AVG() , MAX(), COUNT(몇개인가?) , SUM(합계, 총합)...

## 퀴즈3— [미션1]가게db분석하기

```
-- 퀴즈 집계함수의 컬럼명을 다음과 같이 바꾸시오.  
-- 1. cnt  
-- 2. sum_stock  
-- 3. max_selling
```

```
select count(*) as cnt from product;  
select sum(stock) sum_stock from product;  
select max(selling_price) as max_selling from product;
```

- GROUP BY는 반드시 집계함수와 함께 사용! ≠ 집계함수는 반드시 GROUP BY와 함께 사용된다.

```
SELECT *, 집계함수() FROM 테이블명  
GROUP BY 컬럼명  
HAVING 조건;
```

- GROUP BY : 부서별 , 학년별, 00별 평균, 갯수 등을 구할때

## 퀴즈4

```

create table market (
  id int,
  code varchar(10),
  name varchar(15),
  score float ,
  stock int);

insert into market values (1000, 'fruit', 'apple', 77.5, 33) ,
(1001, 'fruit', 'banana' , 80.0, 50) , (1002,'animal', 'cat', 95.5, 11),
(1003,'stuff', 'doll', 65.5, 5), ( 1004, 'food','egg', 77.5, 20) ,
(1005, 'food','fish', 79.5, 25) ,( 1006,'animal', 'goose', 83.5, 8),
(1007,'food','ham', 82.5, 50), (1008, 'animal','iguana',94.5, 30),
(1009,'food','jam',78.5, 3);

```

- code별 갯수가 3개 이상인 code,갯수를 구하시오.

```

-- code별 갯수가 3개 이상인 code,갯수를 구하시오.
select code,count(*) from market
group by code
having count(*) >=3;

```

```

create table orders (
  sort varchar(15),
  amount int,
  owner varchar(15)
);

insert into orders values
('fruit', 150, 'james'),
('fruit', 3000, 'john'),
('food',100, 'mina'),
('food',205, 'lauren'),
('animal',30, 'jenny'),
('stuff',100, 'amy');

```

- order 테이블과 market 테이블을 inner join 하고, order 테이블의 owner 기준 오름차순하시오.

```

select * from market inner join orders on market.code=orders.sort
order by orders.owner;

select * from market A inner join orders B on A.code=B.sort
order by B.owner;

```

## 퀴즈5 —[미션2] 인기있는물건

```
-- sum(amount)를 기준으로 내림차순 정렬
select name, sum(amount) as sum from sale group by name
having sum(revenue)>=50000
order by sum(amount) desc;
```

## 프로그래머스 SQL KIT

[https://school.programmers.co.kr/learn/challenges?tab=sql\\_practice\\_kit](https://school.programmers.co.kr/learn/challenges?tab=sql_practice_kit)

- 요즘 코테에도 SQL 문제가 1,2개씩 포함됨. level.3까지 풀수있다면 아무문제나 가능 .