

## 10주차 1차시 데이터 검색 4

### 【학습목표】

1. 특정 속성의 값이 같은 튜플을 모아 그룹을 만들어 검색하기 방법을 설명할 수 있다.
2. 그룹에 대한 조건을 추가하여 검색하는 방법을 설명할 수 있다.

### 학습내용1 : 그룹별 기본 검색

- 특정 속성 값이 같은 튜플을 모아 그룹을 만들어 검색
- GROUP BY 키워드를 사용
- 그룹에 대한 조건을 추가 ==> HAVING 키워드 사용
- GROUP BY 키워드가 없는 SELECT 문
  - . 테이블 전체를 하나의 그룹으로 하여 검색
- 기본 형식

```
SELECT [ ALL | DISTINCT ] 속성_리스트  
FROM 테이블_리스트  
[ WHERE 조건 ]  
[ GROUP BY 속성_리스트 [ HAVING 조건 ] ]  
[ ORDER BY 속성_리스트 [ ASC | DESC ] ] ;
```

- . GROUP BY 속성\_리스트 : 그룹을 나누는 기준이 되는 속성을 지정  
==> 그룹에 대한 조건 - HAVING 키워드와 함께 작성

#### 1. 질의 1 : GROUP BY 키워드

- Orders 테이블에서 주문 제품별 (item) 수량의 합계를 검색하시오.
- Orders 테이블의 데이터

```
mysql> SELECT * FROM orders;
+----+-----+-----+-----+-----+-----+
| num | customer | item | qty | date       | saddr |
+----+-----+-----+-----+-----+-----+
| 1   | apple   | E06  | 15  | 2015-06-01 | 경기 용인시 |
| 2   | apple   | E05  | 2   | 2015-06-01 | 경기 용인시 |
| 3   | eagle   | E01  | 3   | 2015-06-03 | 부산 남구   |
| 4   | bank    | E04  | 5   | 2015-06-03 | 서울 중로구 |
| 5   | bird    | E01  | 10  | 2015-06-04 | 충남 천안시 |
| 6   | bird    | E04  | 1   | 2015-06-04 | 충남 천안시 |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

- 질의문

```
SELECT item, SUM( qty ) as 총주문수량
FROM orders
GROUP BY item;
```

```
mysql> SELECT item, SUM( qty ) as 총주문수량
-> FROM orders
-> GROUP BY item;
+----+-----+
| item | 총주문수량 |
+----+-----+
| E01  | 13         |
| E04  | 6          |
| E05  | 2          |
| E06  | 15         |
+----+-----+
4 rows in set (0.02 sec)
```

[TIP] -----

- \* 그룹별로 검색할 때, 그룹을 나누는 기준 속성을 SELECT 절에 작성하는 것이 좋다
  - . SELECT 절에 작성하지 않아도 실행됨
  - ==> 어떤 그룹에 대한 검색 결과인지 확인하기 어렵다.

-----[ END TIP ]

- TIP에 대한 질의

- . 그룹을 위한 기준 속성(item)을 SELECT 절에 포함시키는 않은 경우

```
SELECT SUM( qty ) as 총주문수량
FROM orders
GROUP BY item;
```

```
mysql> SELECT SUM( qty ) as 총주문수량
-> FROM orders
-> GROUP BY item;
+-----+
| 총주문수량 |
+-----+
|          13 |
|           6 |
|           2 |
|          15 |
+-----+
4 rows in set (0.00 sec)

mysql>
```

## 2. 질의 2

- Orders 테이블에서 주문한 제품(item)별로 주문건수와 가장 많이 주문한 개수(qty)를 검색하시오. (주문건수는 주문건수, 가장 많이 주문한 개수는 최대주문수로 표기)

- . 주문한 제품(item)별 : GROUP BY item
- . 제품별 주문건수 : COUNT( \* )
- . 최대주문수 : MAX( qty )

- Orders 테이블의 데이터

```
mysql> SELECT * FROM orders;
+----+-----+-----+-----+-----+-----+
| num | customer | item | qty | date       | saddr |
+----+-----+-----+-----+-----+-----+
| 1 | apple | E06 | 15 | 2015-06-01 | 경기 용인시 |
| 2 | apple | E05 | 2 | 2015-06-01 | 경기 용인시 |
| 3 | eagle | E01 | 3 | 2015-06-03 | 부산 남구 |
| 4 | bank | E04 | 5 | 2015-06-03 | 서울 중로구 |
| 5 | bird | E01 | 10 | 2015-06-04 | 충남 천안시 |
| 6 | bird | E04 | 1 | 2015-06-04 | 충남 천안시 |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

- 질의문

```
SELECT  item, COUNT( * ) AS 주문건수, MAX( qty ) AS 최대주문수
FROM    orders
GROUP BY item;
```



```
mysql> SELECT  item, COUNT( * ) AS 주문건수, MAX( qty ) AS 최대주문수
-> FROM    orders
-> GROUP BY item;
+-----+-----+-----+
| item | 주문건수 | 최대주문수 |
+-----+-----+-----+
| E01  | 2        | 10         |
| E04  | 2        | 5          |
| E05  | 1        | 2          |
| E06  | 1        | 15         |
+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>
```

## 학습내용2 : 그룹별 조건 검색

- 일반적인 검색조건 : WHERE 절에 작성
  - . 집계 함수는 WHERE 절에서는 사용할 수 없음
- 그룹에 대한 조건 : HAVING 절에 작성
  - . 집계 함수는 HAVING 절에서 사용 가능

### 1. 질의 3 : GROUP BY ... HAVING

- Orders 테이블에서 2건 이상 주문된 제품(item) 별로 제품의 주문건수와 가장 많이 주문된 개수를 검색하시오.  
(주문건수는 주문건수, 가장 많이 주문한 개수는 최대주문수로 표기)

- . 주문한 제품(item)별 : GROUP BY item HAVING COUNT(\*) >= 2
- . 제품별 주문건수 : COUNT( \* )
- . 최대주문수 : MAX( qty )

- Orders 테이블의 데이터

```
mysql> SELECT * FROM orders;
```

num	customer	item	qty	date	saddr
1	apple	E06	15	2015-06-01	경기인시
2	apple	E05	2	2015-06-01	경기인시
3	eagle	E01	3	2015-06-03	부산구
4	bank	E04	5	2015-06-03	서울로구
5	bird	E01	10	2015-06-04	충남천안시
6	bird	E04	1	2015-06-04	충남천안시

```
6 rows in set (0.00 sec)

mysql>
```

- 질의문

```
SELECT item, COUNT( * ) AS 주문건수, MAX( qty ) AS 최대주문수
FROM orders
GROUP BY item HAVING COUNT( * ) >= 2;
```

```
mysql> SELECT item, COUNT( * ) AS 주문건수, MAX( qty ) AS 최대주문수
-> FROM orders
-> GROUP BY item HAVING COUNT( * ) >= 2;
```

item	주문건수	최대주문수
E01	2	10
E04	2	5

```
2 rows in set (0.02 sec)

mysql>
```

- 질의 2와 질의 3의 결과 비교

item	주문건수	최대주문수
E01	2	10
E04	2	5
E05	1	2
E06	1	15

4 rows in set (0.01 sec)

item	주문건수	최대주문수
E01	2	10
E04	2	5

2 rows in set (0.02 sec)

```
mysql>
```

(1) 질의2 결과

(2) 질의3 결과

## 2. 질의 4 : GROUP BY ... HAVING

- Customer 테이블에서 적립금 평균이 5000원 이상인 등급 (grade)에 대해 등급별 고객 수와 적립금 평균을 검색하시오. ( 등급별 고객수를 고객수, 적립금 평균을 평균적립금으로 표기)

- . 등급별 그룹 : GROUP BY grade
- . 적립금 평균이 5000원 이상 : AVG( credit ) >= 5000
- . 등급별 고객수 : COUNT( \* )
- . 적립금 평균 : AVG( credit )

- Customer 테이블 데이터



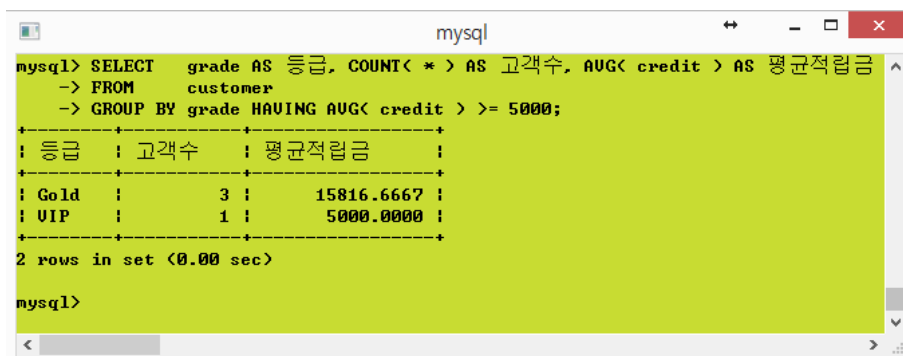
```
mysql> SELECT * FROM customer;
```

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 종로구
bird	구선두	NULL	0	충남 천안시
eagle	박세재	Gold	2450	
king	오나라	Gold	15000	NULL
pencil	김돌	Silver	350	경기 수원

6 rows in set (0.00 sec)

- 질의문

```
SELECT  grade AS 등급, COUNT( * ) AS 고객수, AVG( credit ) AS 평균적립금
FROM    customer
GROUP BY grade HAVING AVG( credit ) >= 5000;
```



```
mysql> SELECT  grade AS 등급, COUNT( * ) AS 고객수, AVG( credit ) AS 평균적립금
-> FROM
customer
-> GROUP BY grade HAVING AVG( credit ) >= 5000;
```

등급	고객수	평균적립금
Gold	3	15816.6667
VIP	1	5000.0000

2 rows in set (0.00 sec)

## 3. 질의 5 : GROUP BY

- Orders 테이블에서 각 주문고객(customer)이 주문한 제품의 총 주문수량을 주문 제품(item)별로 검색하고, 제품(item), 고객(customer), 총주문수량을 표시하시오.

. 각 주문 고객 : ORDER BY item, customer

==> 1차 그룹 : item

2차 그룹 : customer

. SELECT item, customer, SUM( qty )

==> 2차 그룹 후에 수량의 합계를 계산

- Orders 테이블 데이터

```
mysql> select * from orders;
```

num	customer	item	qty	date	saddr
1	apple	E06	15	2015-06-01	경기인시
2	apple	E05	2	2015-06-01	경기인시
3	eagle	E01	3	2015-06-03	부산구
4	bank	E04	5	2015-06-03	서울구
5	bird	E01	10	2015-06-04	충남구
6	bird	E04	1	2015-06-04	충남구
7	apple	E06	10	2015-06-10	NULL
8	apple	E05	10	2015-06-10	NULL

8 rows in set (0.00 sec)

## - 질의문

```
SELECT  item, customer, SUM( qty ) AS 총주문수량
FROM    orders
GROUP BY item, customer;
```

```
mysql> SELECT  item, customer, SUM( qty ) AS 총주문수량
-> FROM    orders
-> GROUP BY item, customer;
+-----+-----+-----+
| item | customer | 총주문수량 |
+-----+-----+-----+
| E01  | bird     | 10        |
| E01  | eagle    | 3         |
| E04  | bank     | 5         |
| E04  | bird     | 1         |
| E05  | apple    | 12        |
| E06  | apple    | 25        |
+-----+-----+-----+
6 rows in set (0.02 sec)

mysql>
```

## 4. 질의 6 : GROUP BY

- Orders 테이블에서 각 주문고객(customer)이 주문한 제품의 총 주문수량을 주문 고객(customer)별로 검색하고, 고객(customer), 제품(item), 총주문수량을 표시하시오.

. 각 주문 고객 : ORDER BY customer, item

==> 1차 그룹 : customer

2차 그룹 : item

. SELECT customer, item, SUM( qty )

==> 2차 그룹 후에 수량의 합계를 계산

## - Orders 테이블 데이터

```
mysql> select * from orders;
+-----+-----+-----+-----+-----+-----+
| num | customer | item | qty | date       | saddr |
+-----+-----+-----+-----+-----+-----+
| 1   | apple    | E06  | 15  | 2015-06-01 | 경기 용인시 |
| 2   | apple    | E05  | 2   | 2015-06-01 | 경기 용인시 |
| 3   | eagle    | E01  | 3   | 2015-06-03 | 부산광역시 |
| 4   | bank     | E04  | 5   | 2015-06-03 | 서울특별시 |
| 5   | bird     | E01  | 10  | 2015-06-04 | 충청남도 천안시 |
| 6   | bird     | E04  | 1   | 2015-06-04 | 충청남도 천안시 |
| 7   | apple    | E06  | 10  | 2015-06-10 | NULL |
| 8   | apple    | E05  | 10  | 2015-06-10 | NULL |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```



## - 질의문

```
SELECT  item, customer, SUM( qty ) AS 총주문수량
FROM    orders
GROUP BY customer, item;
```



```
mysql> SELECT  customer, item, SUM( qty ) AS 총주문수량
-> FROM    orders
-> GROUP BY customer, item;
+-----+-----+-----+
| customer | item | 총주문수량 |
+-----+-----+-----+
| apple   | E05  | 12         |
| apple   | E06  | 25         |
| bank    | E04  | 5          |
| bird    | E01  | 10         |
| bird    | E04  | 1          |
| eagle   | E01  | 3          |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

## 5. 정렬 검색 Review

## - 그룹별 검색 기본 형식

```
SELECT [ ALL | DISTINCT ] 속성_리스트
FROM   테이블_리스트
[ WHERE 조건 ]
[ GROUP BY 속성_리스트 [ HAVING 조건 ] ]
[ ORDER BY 속성_리스트 [ ASC | DESC ] ] ;
```

## 【학습정리】

1. GROUP BY 키워드를 사용하여 특정 속성의 값이 같은 튜플을 모아 그룹을 만들어 검색할 수 있다.
2. GROUP BY를 사용하여 그룹을 만들 때, HAVING 키워드를 사용하여 조건을 명시할 수 있고, 조건에 집계함수를 사용할 수 있다.