

13주차 1차시 내장함수와 인덱스

【학습목표】

1. 그룹으로 데이터를 검색할 때 제공되는 GROUP BY 내장 함수를 사용하는 방법을 설명할 수 있다.
2. 데이터베이스의 인덱스를 생성, 삭제하는 방법을 설명할 수 있다.

학습내용1 : GROUP BY 함수

1. GROUP_CONCAT (수식)

- . GROUP으로부터 NULL이 아닌 값을 결합한 문자열 결과를 반환
- . 모두 NULL 인 값만 있는 경우, NULL을 반환
- . 기본 형식

```
GROUP_CONCAT( [ DISTINCT 표현식 [ , 표현식 .... ]  
              [ ORDER BY { 부호없는정수 | 속성이름 | 표현식 }  
                [ ASC | DESC ] [ , 속성이름 .... ] ]  
              [ SEPARATOR 문자값 ] )
```

질의 1 :

고객별로 주문한 물품의 리스트를 작성하시오. (주문한 물품의 중복 표시 허용)

A. Orders 테이블로부터 고객(customer)별 그룹 검색

==> GROUP BY customer

B. 고객별 물품 리스트 작성

==> GROUP_CONCAT(item)

- 질의문

```
SELECT customer,  
GROUP_CONCAT( item )  
FROM orders  
GROUP BY customer;
```

- 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	부산 남구
4	bank	E04	5	2015-06-03	서울 중로구
5	bird	E02	10	2015-06-04	충남 천안시
6	bird	E03	1	2015-06-04	충남 천안시
7	apple	E06	10	2015-06-10	NULL
8	apple	E05	10	2015-06-10	NULL

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

- 질의 실행 후 Orders 테이블 데이터

```
mysql> SELECT customer, GROUP_CONCAT( item )
-> FROM orders
-> GROUP BY customer;
```

customer	GROUP_CONCAT(item)
apple	E06,E05,E06,E05
bank	E04
bird	E02,E03
eagle	E01

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

```
mysql>
```

질의 2 :

고객별로 주문한 물품의 리스트를 작성하시오. (중복된 물품은 하나로 표시하되 내림차순으로 정렬하고, 각 물품을 구분하기 위하여 빈공간(' ')를 사용)

- Orders 테이블로부터 고객(customer)별 그룹 검색
=> GROUP BY customer
- 고객별 물품 리스트 작성
=> GROUP_CONCAT(item)
- 중복된 물품을 하나로 표시
=> GROUP_CONCAT(DISTINCT item)
- 물품을 내림차순으로 정렬
=> GROUP_CONCAT(DISTINCT item ORDER BY item DESC)
- 각 물품을 구분하기 위하여 빈공간(' ')를 사용
=> GROUP_CONCAT(DISTINCT item
ORDER BY item DESC SEPERATOR ' ')

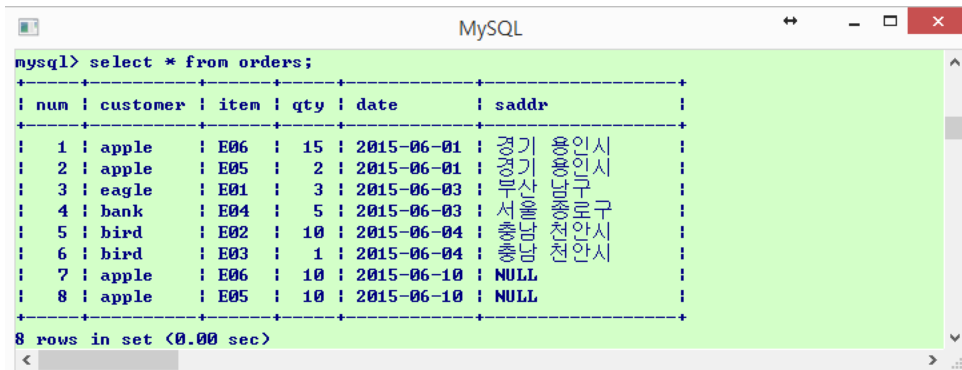
- 질의문

```

SELECT customer,
GROUP_CONCAT( DISTINCT item
              ORDER BY item DESC SEPARATOR ' ' )
FROM   orders
GROUP BY customer;

```

- Orders 테이블 데이터



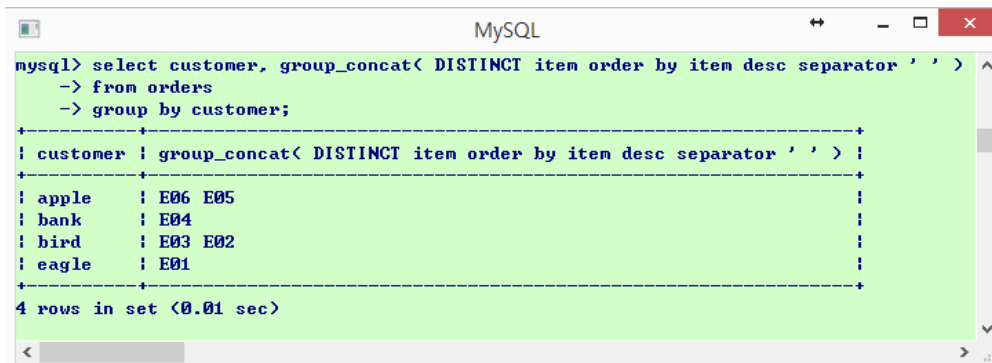
MySQL

```
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	E02	10	2015-06-04	충남 천안시
6	bird	E03	1	2015-06-04	충남 천안시
7	apple	E06	10	2015-06-10	NULL
8	apple	E05	10	2015-06-10	NULL

8 rows in set (0.00 sec)

- 질의 실행 후 Orders 테이블 데이터



MySQL

```
mysql> select customer, group_concat( DISTINCT item order by item desc separator ' ' )
-> from orders
-> group by customer;
```

customer	group_concat(DISTINCT item order by item desc separator ' ')
apple	E06 E05
bank	E04
bird	E03 E02
eagle	E01

4 rows in set (0.01 sec)

질의 3 :

고객이 주문한 판매금액이 10000원 이상이면 배송료를 면제하기 질의문을 작성하시오. 단, 같은 날 동일한 고객이 여러 건 주문한 경우, 묶음 배송으로 처리하고, 결과 테이블은 계정(orders.customer), 주문상품, 판매금액, 배송비(무료배송일 경우, '무료배송', 유료 배송비는 2500)를 반환한다.

- (1) 질의에 필요한 테이블 : orders, items
 - (2) 같은 판매일에 같은 고객이 주문한 경우 :
GROUP BY orders.date, orders.customer
 - (3) 주문 상품
 - 주문 상품 코드
GROUP_CONCAT(orders.item) AS '주문상품'
 - 주문 상품 이름
GROUP_CONCAT(items.name) AS '주문상품'
 - (4) 같은 판매일에 같은 고객이 주문한 판매금액 계산 :
SUM(orders.qty * items.price)
 - (5) 배송비를 계산 :
IF (SUM(orders.qty * items.price) > 10000, '무료배송', '2500')
AS '배송비'
- 질의문 (주문 상품 코드)

```
SELECT orders.customer,
       GROUP_CONCAT( orders.item ) AS '주문상품',
       SUM( orders.qty * items.price ) AS '판매금액',
       IF ( SUM( orders.qty * items.price ) > 10000,
           '무료배송', '2500' ) AS '배송비'
FROM   orders, items
WHERE  orders.item = items.code
GROUP BY orders.date, orders.customer;
```

- 질의문 (주문 상품 이름)

```
SELECT orders.customer,
       GROUP_CONCAT( items.name ) AS '주문상품',
       SUM( orders.qty * items.price ) AS '판매금액',
       IF ( SUM( orders.qty * items.price ) > 10000,
           '무료배송', '2500' ) AS '배송비'
FROM   orders, items
WHERE  orders.item = items.code
GROUP BY orders.date, orders.customer;
```

- Orders 테이블 데이터

MySQL

```
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	E02	10	2015-06-04	충남천안시
6	bird	E03	1	2015-06-04	충남천안시
7	apple	E06	10	2015-06-10	NULL
8	apple	E05	10	2015-06-10	NULL

8 rows in set (0.00 sec)

- Items 테이블 데이터

MySQL

```
mysql> select * from items;
```

code	name	stocks	price
E01	김치	30	1575
E02	단무지	45	3413
E03	행유	15	7350
E04	주유	25	4788
E05	주스	20	2415
E06	라면	30	2688
ZZZ	삭제상품	0	0

7 rows in set (0.00 sec)

mysql>

- 질의문(주문 상품 코드) 실행 및 결과 데이터

MySQL

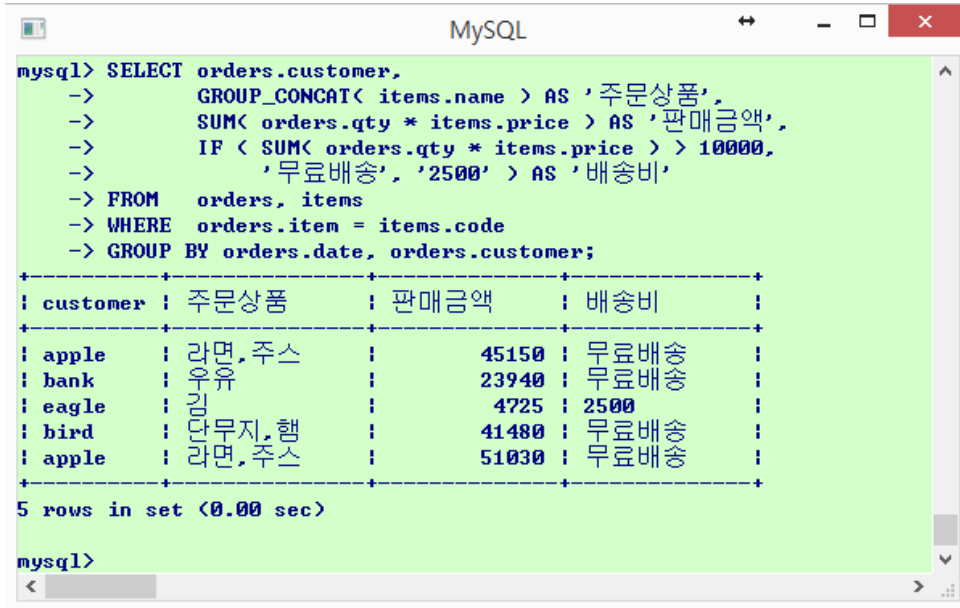
```
mysql> SELECT orders.customer,
-> GROUP_CONCAT( orders.item ) AS '주문상품',
-> SUM( orders.qty * items.price ) AS '판매금액',
-> IF ( SUM( orders.qty * items.price ) > 10000,
-> '무료배송', '2500' ) AS '배송비'
-> FROM orders, items
-> WHERE orders.item = items.code
-> GROUP BY orders.date, orders.customer;
```

customer	주문상품	판매금액	배송비
apple	E06,E05	45150	무료배송
bank	E04	23940	무료배송
eagle	E01	4725	2500
bird	E02,E03	41480	무료배송
apple	E06,E05	51030	무료배송

5 rows in set (0.00 sec)

mysql>

- 질의문(주문 상품 이름) 실행 및 결과 데이터



```
mysql> SELECT orders.customer,
->      GROUP_CONCAT( items.name ) AS '주문상품',
->      SUM( orders.qty * items.price ) AS '판매금액',
->      IF ( SUM( orders.qty * items.price ) > 10000,
->      '무료배송', '2500' ) AS '배송비'
-> FROM   orders, items
-> WHERE  orders.item = items.code
-> GROUP BY orders.date, orders.customer;
```

customer	주문상품	판매금액	배송비
apple	라면, 주스	45150	무료배송
bank	우유	23940	무료배송
eagle	김	4725	2500
bird	단무지, 햄	41480	무료배송
apple	라면, 주스	51030	무료배송

5 rows in set (0.00 sec)

```
mysql>
```

2. GROUP BY 수정자

- 결과 테이블에 요약 결과 투플을 추가하기 위하여 WITH ROLLUP 수정자를 사용한다.
- 추가된 투플은 상위 레벨의 개요를 나타낸다.
- ORDER BY 키워드와 같이 사용할 수 없다.
- 그룹으로 지정한 속성에 대한 정렬은 속성이름 다음에 ASC 또는 DESC를 사용하여 지정
- 기본 형식

GROUP BY 속성리스트 [WITH ROLLUP]

질의 4 :

고객(customer)별 물품(item)별 구매 개수 현황을 만들되, 고객별 합계와 총 합계를 출력하시오.

- 고객별 물품별 구매 개수 현황을 만든다.
- 고객별 합계와 총합계를 구하여 결과 테이블에 삽입한다.

질의문 A : 고객별 물품별 구매 개수 현황

- 질의문

```
SELECT customer, item, sum( qty )
FROM   orders
GROUP BY customer, item;
```

- Orders 테이블 데이터

MySQL

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

num	customer	item	qty	date	saddr
1	apple	E06	15	2015-06-01	경기인사
2	apple	E05	2	2015-06-01	충청남도
3	eagle	E01	3	2015-06-03	부산광역시
4	bank	E04	5	2015-06-03	서울특별시
5	bird	E02	10	2015-06-04	충청남도
6	bird	E03	1	2015-06-04	충청남도
7	apple	E06	10	2015-06-10	NULL
8	apple	E05	10	2015-06-10	NULL

8 rows in set (0.00 sec)

- 질의 결과

MySQL

```
mysql> SELECT customer, item, sum( qty )
-> FROM orders
-> GROUP BY customer, item;
```

customer	item	sum(qty)
apple	E05	12
apple	E06	25
bank	E04	5
bird	E02	10
bird	E03	1
eagle	E01	3

6 rows in set (0.01 sec)

질의문 B : 고객별 합계와 총합계를 결과 테이블에 삽입

- 질의문

```
SELECT customer, item, sum( qty )
FROM orders
GROUP BY customer, item WITH ROLLUP;
```

- 질의 결과

MySQL

```
mysql> SELECT customer, item, sum( qty )
-> FROM orders
-> GROUP BY customer, item WITH ROLLUP;
```

customer	item	sum(qty)
apple	E05	12
apple	E06	25
apple	NULL	37
bank	E04	5
bank	NULL	5
bird	E02	10
bird	E03	1
bird	NULL	11
eagle	E01	3
eagle	NULL	3
NULL	NULL	56

11 rows in set (0.00 sec)

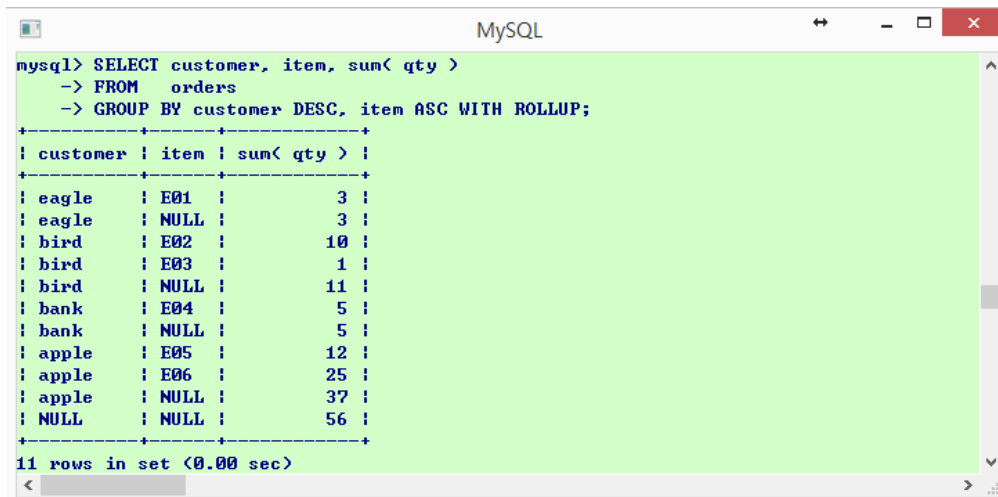
질의 5 :

질의4의 질의인 고객(customer)별 물품(item)별 구매 개수 현황을 만들되, 고객별 합계와 총 합계를 출력하시오. 단, 고객은 내림차순으로 정렬하고 물품은 오름차순으로 정렬하시오.

- ORDER BY를 사용하지 않는다.
- 질의3의 질의문에 고객을 내림차순, 물품을 오름차순 정렬
- 질의문

```
SELECT customer, item, sum( qty )
FROM orders
GROUP BY customer DESC, item ASC WITH ROLLUP;
```

- 질의 결과



customer	item	sum(qty)
eagle	E01	3
eagle	NULL	3
bird	E02	10
bird	E03	1
bird	NULL	11
bank	E04	5
bank	NULL	5
apple	E05	12
apple	E06	25
apple	NULL	37
NULL	NULL	56

11 rows in set (0.00 sec)

질의 6 :

질의5 의 결과에서 6개의 튜플만 제한하여 보이시오.

- LIMIT 키워드는 결과 테이블의 반환 튜플 수를 제한한다.
- LIMIT 키워드는 ROLLUP 키워드 후에 적용한다.
- 질의문

```
SELECT customer, item, sum( qty )
FROM orders
GROUP BY customer DESC, item ASC WITH ROLLUP
LIMIT 6 ;
```



```

MySQL
mysql> SELECT customer, item, sum( qty )
-> FROM orders
-> GROUP BY customer DESC, item ASC WITH ROLLUP
-> LIMIT 6;
+-----+-----+-----+
| customer | item | sum( qty ) |
+-----+-----+-----+
| eagle   | E01  |          3 |
| eagle   | NULL |          3 |
| bird    | E02  |         10 |
| bird    | E03  |          1 |
| bird    | NULL |         11 |
| bank    | E04  |          5 |
+-----+-----+-----+
6 rows in set (0.00 sec)

```

질의 7 :

질의5의 결과에서 각 고객별 합계의 NULL을 “Sub-total” 로 표현하고, 고객 전체 합계의 NULL을 “Total”로 표현하시오.

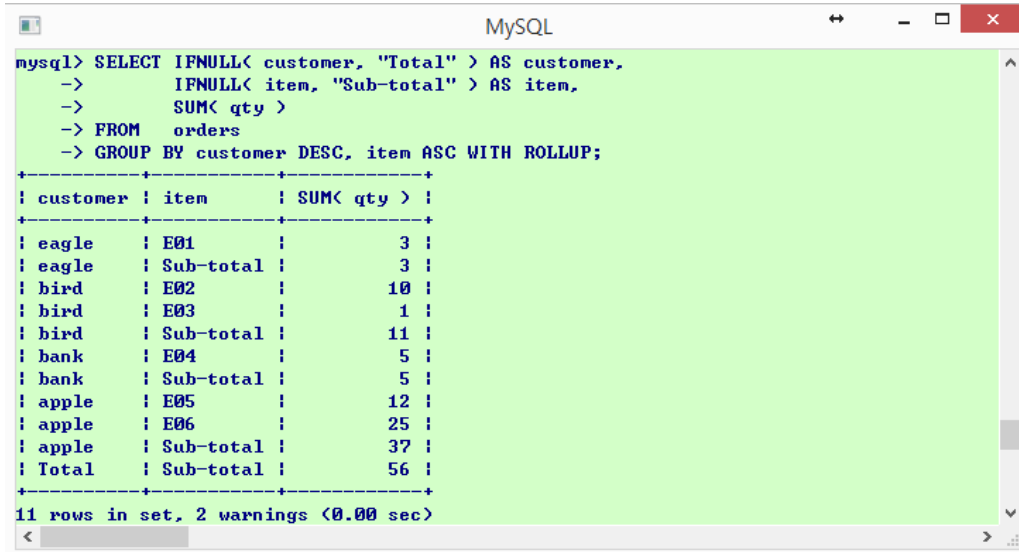
- A. 고객별 합계의 NULL을 “Sub-total”
 - IFNULL (item, “Sub-total”) AS item
- B. 전체 합계의 NULL을 “Total”
 - IFNULL (customer, “Total”) AS customer
- 질의문

```

SELECT IFNULL( customer, "Total" ) AS customer,
       IFNULL( item, "Sub-total" ) AS item,
       sum( qty )
FROM orders
GROUP BY customer DESC, item ASC WITH ROLLUP;

```

- 질의문 실행 결과



```
mysql> SELECT IFNULL< customer, "Total" > AS customer,
-> IFNULL< item, "Sub-total" > AS item,
-> SUM< qty >
-> FROM orders
-> GROUP BY customer DESC, item ASC WITH ROLLUP;
```

customer	item	SUM< qty >
eagle	E01	3
eagle	Sub-total	3
bird	E02	10
bird	E03	1
bird	Sub-total	11
bank	E04	5
bank	Sub-total	5
apple	E05	12
apple	E06	25
apple	Sub-total	37
Total	Sub-total	56

11 rows in set, 2 warnings (0.00 sec)

학습내용2 : 인덱스 (Index)

- 테이블에서 자주 사용하는 속성을 별도로 생성
- 테이블의 검색 동작 속도를 향상시키기 위하여 사용
- 테이블에 대한 킷값을 가지고 있는 사본 테이블
- 사용자들은 인덱스를 볼 수 없다.
- MySQL에서는 기본키에 대하여 인덱스가 자동 생성된다.
- 대부분의 MySQL 인덱스 (기본키, UNIQUE, INDEX와 FULLTEXT)는 B-tree에 저장된다.
 - . B-tree는 Binary Tree와 다르다.
 - . B-tree : 여러 개의 자식 노드를 가질 수 있다.
 - . Binary Tree : 2개의 자식 노드를 가진다.
- 장점
 - . WHERE 절이나 ORDER BY 처리에서 빠른 처리 속도
- 단점
 - . 인덱스가 설정된 테이블을 수정할 때 처리 속도가 떨어짐
- 인덱스 생성시 고려 사항
 - . WHERE절에서 자주 사용되는 속성을 선택한다.
 - . 조인에 자주 사용되는 속성을 선택한다.
 - . 인덱스가 많으면 속도가 느려진다.

1. 인덱스 생성

- 기본 형식

```
CREATE INDEX 인덱스명
ON 테이블명( 속성명, ... );
```

또는

```
ALTER TABLE 테이블명 ADD INDEX 인덱스명 [ 인덱스타입 ]
( 인덱스속성명, ... );
```

질의 1 :

CREATE INDEX를 사용하여 Customer 테이블의 name 속성에 대하여 인덱스(nameidx)를 생성하시오.

- 질의문

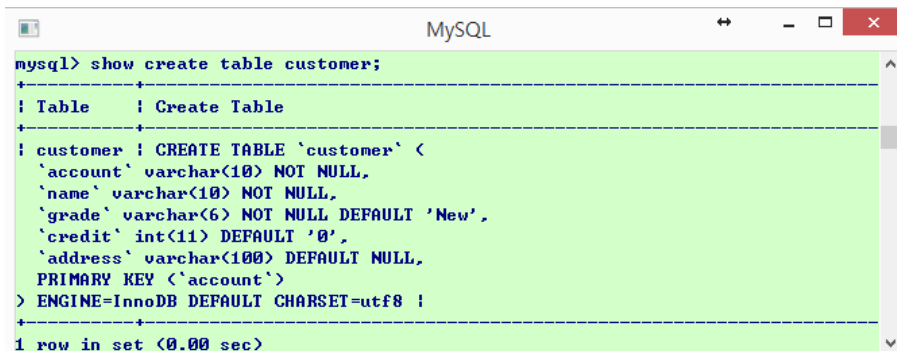
```
CREATE INDEX nameidx ON customer( name );
```



```
mysql> CREATE INDEX nameidx ON customer( name );
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

- 테이블 생성 코드 (인덱스 생성 전)



```
mysql> show create table customer;
+-----+-----+
| Table | Create Table |
+-----+-----+
| customer | CREATE TABLE `customer` (
  `account` varchar(10) NOT NULL,
  `name` varchar(10) NOT NULL,
  `grade` varchar(6) NOT NULL DEFAULT 'New',
  `credit` int(11) DEFAULT '0',
  `address` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`account`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

- 테이블 생성 코드 (인덱스 생성 후)

```
mysql> show create table customer;
+-----+-----+
| Table | Create Table |
+-----+-----+
| customer | CREATE TABLE `customer` (
  `account` varchar(10) NOT NULL,
  `name` varchar(10) NOT NULL,
  `grade` varchar(6) NOT NULL DEFAULT 'New',
  `credit` int(11) DEFAULT '0',
  `address` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`account`),
  KEY `nameidx` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

2. 인덱스 조회

- 기본 형식

SHOW INDEX FROM 테이블명;

질의 2 :

Customer 테이블에 대하여 질의1에서 생성한 인덱스를 검색하시오.

- 질의문

SHOW INDEX FROM customer;

- 질의1의 인덱스 생성전 customer 테이블의 인덱스

```
mysql> show index from customer;
+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation |
+-----+-----+-----+-----+-----+-----+
| customer | 0 | PRIMARY | 1 | account | A |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> show index from customer;
+-----+-----+-----+-----+-----+-----+-----+
| Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comme |
+-----+-----+-----+-----+-----+-----+-----+
| 8 | NULL | NULL | | BTREE | | |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- 질의1의 인덱스 생성후 customer 테이블의 인덱스

MySQL

```
mysql> SHOW INDEX FROM customer;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation
customer	0	PRIMARY	1	account	A
customer	1	nameidx	1	name	A

2 rows in set (0.00 sec)

MySQL

Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
8	NULL	NULL		BTREE		
8	NULL	NULL		BTREE		

3. 인덱스 삭제

- 기본 형식

DROP INDEX 인덱스명 ON 테이블명 ;

또는

ALTER TABLE 테이블명 DROP INDEX 인덱스명 ;

질의 3 :

질의1에서 생성한 nameidx 인덱스를 삭제하시오.

- 질의문

ALTER TABLE customer DROP INDEX nameidx;

또는

DROP INDEX nameidx ON customer;

- 질의문 실행 및 확인

MySQL

```
mysql>
mysql> ALTER TABLE customer DROP INDEX nameidx;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

```

MySQL
mysql> DROP INDEX nameidx ON customer;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show index from customer;
+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation |
+-----+-----+-----+-----+-----+-----+
| customer | 0 | PRIMARY | 1 | account | A |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

【학습정리】

1. GROUP_CONCAT은 그룹으로 묶은 속성 값 중에서 NULL이 아닌 값을 결합한 문자열 결과를 반환한다.
2. 결과 테이블에 그룹별 요약 결과 튜플을 추가하기 위하여 WITH ROLLUP을 사용한다.
3. 데이터 검색 동작 속도를 향상시키기 위하여 테이블에서 자주 사용하는 속성을 별도로 인덱스로 지정한다.