

10주차 3차시 데이터 검색 6

【학습목표】

1. 질의문을 사용하여 검색한 결과로부터 다시 검색하는 부속 질의문을 이용한 검색방법에 대해 설명할 수 있다.
2. 속성의 숫자와 날짜 데이터 타입에 대해 설명할 수 있다.

학습내용1 : 부속 질의문을 이용한 검색이란?

- SELECT 문 안에 다른 SELECT 문을 포함
- 부속 질의문 / 서브 질의문 (Sub Query) : 내포된 SELECT 문
- 상위 질의문 / 주 질의문 (Main Query) : 다른 SELECT 문을 포함하는 SELECT 문

- 부속질의문
 - . 괄호로 묶어 작성
 - . ORDER BY 절을 사용할 수 없음
 - . 상위 질의문보다 먼저 수행됨
 - . 반환되는 행의 수에 따른 분류
 - * 단일행 부속 질의문
 - * 다중행 부속 질의문

- 상위질의문
 - . 부속질의문의 결과를 이용해 질의를 수행

- 부속질의문과 상위질의문을 연결하는 연산자
 - . 부속질의문의 종류에 따라 사용할 수 있는 연산자가 다름
 - . 단일행 부속 질의문 : 일반 비교 연산자를 사용할 수 있음
 - . 다중행 부속 질의문 : 일반 비교 연산자를 사용할 수 없음

- 다중 행 부속 질의문에 사용 가능한 연산자

연산자	설명
IN	부속 질의문의 결과 값 중 일치하는 것이 있으면 검색 조건이 참
NOT IN	부속 질의문의 결과 값 중 일치하는 것이 없으면 검색 조건이 참
EXISTS	부속 질의문의 결과 값이 하나라도 존재하면 검색 조건이 참
NOT EXISTS	부속 질의문의 결과 값이 하나도 존재하지 않으면 검색 조건이 참
ALL	부속 질의문의 결과 값 모두와 비교한 결과가 참이면 검색 조건을 만족 (비교 연산자와 함께 사용)
ANY 또는 SOME	부속 질의문의 결과 값 중 하나라도 비교한 결과가 참이면 검색 조건을 만족 (비교 연산자와 함께 사용)

학습내용2 : 부속 질의문을 이용한 검색

1. 질의 1

- 고객 (customer)테이블에서 'bank' 계정의 등급과 같은 모든 고객의 이름(name)과 적립금 (credit)을 검색하시오.

- 부속 질의문 : 계정이 bank 인 고객의 등급을 검색

```
SELECT grade
FROM customer
WHERE account='bank'
```

- 주질의문 : 부속 질의문의 결과인 등급과 동일한 고객의 이름과 적립금 검색

```
SELECT name, credit
FROM customer
WHERE grade = 부질의문_결과
```

- Customer 테이블의 데이터

Customer

<u>account</u>	name	grade	credit	address
bank	홍길동	Gold	30000	서울 종로구
apple	이남이	VIP	5000	경기 용인시
pencil	김돌	Silver	350	경기 수원
eagle	박세째	Gold	2450	부산 남구
bird	구선두	New	0	충남 천안시
king	오나라	Gold	15000	NULL

- 질의문

```

SELECT name, credit
FROM   customer
WHERE  grade = ( SELECT grade
                  FROM   customer
                  WHERE  account='bank' );

```

```

mysql> SELECT name, credit
-> FROM   customer
-> WHERE  grade = ( SELECT grade
->                  FROM   customer
->                  WHERE  account='bank' );
+-----+-----+
| name   | credit |
+-----+-----+
| 홍길동 | 30000  |
| 박세째 | 2450   |
| 오나라 | 15000  |
+-----+-----+
3 rows in set (0.03 sec)

mysql>

```

2. 질의 2

- Customer 테이블에서 적립금이 가장 많은 고객의 이름(name)과 적립금(credit)을 검색하시오.
- 부속질의문 : 적립금이 가장 많은 고객의 적립금 검색

```
SELECT MAX( credit ) FROM customer
```
- 주질의문 : 부속질의문 결과 고객의 이름(name)과 적립금(credit) 검색

```
SELECT name, credit
FROM customer
WHERE credit = 부속질의문_결과
```
- Customer 테이블의 데이터

Customer

<u>account</u>	name	grade	credit	address
bank	홍길동	Gold	30000	서울 종로구
apple	이남이	VIP	5000	경기 용인시
pencil	김돌	Silver	350	경기 수원
eagle	박세째	Gold	2450	부산 남구
bird	구선두	New	0	충남 천안시
king	오나라	Gold	15000	NULL

- 질의문

```
SELECT name, credit
FROM customer
WHERE credit = ( SELECT MAX( credit ) FROM customer );
```

```
mysql> SELECT name, credit
-> FROM customer
-> WHERE credit = ( SELECT MAX( credit ) FROM customer );
+-----+-----+
| name      | credit |
+-----+-----+
| 홍길동    | 30000  |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

- 잘못된 질의문

```
SELECT name, credit
FROM customer
WHERE credit = MAX( credit );
```

```
mysql>
mysql>
mysql> SELECT name, credit
      -> FROM customer
      -> WHERE credit = MAX( credit );
ERROR 1111 (HY000): Invalid use of group function
mysql>
```

==> 집계 함수

- . WHERE 절에서는 사용할 수 없음
- . SELECT 절과 HAVING 절에서만 사용할 수 있음

- 잘못된 질의문

```
SELECT name, account, MAX( credit )
FROM customer;
```

```
mysql> SELECT name, account, MAX( credit )
      -> FROM customer;
+-----+-----+-----+
| name   | account | MAX( credit ) |
+-----+-----+-----+
| 이남이 | apple   | 30000         |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

```
mysql> SELECT * FROM customer;
+-----+-----+-----+-----+-----+
| account | name   | grade | credit | address |
+-----+-----+-----+-----+-----+
| apple   | 이남이 | VIP   | 5000   | 경기 용인시 |
| bank    | 홍길동 | Gold  | 30000  | 서울 중도구 |
| bird    | 구선두 | NULL  | 0       | 충남 천안시 |
| eagle   | 박세재 | Gold  | 2450   | NULL       |
| king    | 오나라 | Gold  | 15000  | NULL       |
| pencil  | 김돌   | Silver | 350    | 경기 수원   |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

[TIP] =====

- 단일행 부속질의문에서 사용할 수 있는 연산자
 - * 비교연산자(=, <, >, >=, <=)
- 다중행 부속질의문
 - * 일반 비교 연산자를 사용할 수 없음
 - * IN, NOT, EXISTS, NOT EXISTS, ALL, ANY, SOME 사용

===== [END TIP]

3. 질의 3

- apple 고객이 주문한 제품의 이름을 검색하시오. (조인 검색의 질의1)

[TIP] -----

- * IN 연산자 : 부분집합 연산자
 - A IN B : A 가 B의 부분집합이면 참
 - A NOT IN B : A가 B의 부분집합이 아니면 참

----- [END TIP]

- 부속질의문 : apple 고객이 주문한 제품 코드 검색

```
SELECT code FROM orders WHERE customer='apple';
```

- 주질의문 : 제품코드가 부속질의문에서 검색한 결과에 속해 있는지 확인

```
SELECT name FROM items WHERE code IN 부속질의문_결과;
```

- Orders 와 Items 테이블

Items

<u>code</u>	name	stocks	price
E01	김	30	1500
E02	단무지	45	3250
E03	햄	15	7000
E04	우유	25	4560
E05	주스	20	2300
E06	라면	30	2560
ZZZ	삭제상품	0	0

Orders

num	customer	item	qty	date	saddr
00001	apple	E06	15	2015-06-01	경기 용인시
00002	apple	E05	2	2015-06-01	경기 용인시
00003	eagle	E01	3	2015-06-03	부산 남구
00004	bank	E04	5	2015-06-03	서울 종로구
00005	bird	E02	10	2015-06-04	충남 천안시
00006	bird	E03	1	2015-06-04	충남 천안시
00007	apple	E06	10	2015-06-10	NULL
00008	apple	E05	10	2015-06-10	NULL

- 질의문

```

SELECT name
FROM   items
WHERE  code IN ( SELECT item
                  FROM   orders
                  WHERE  customer='apple' );

```

- 결과

```

mysql> SELECT name
-> FROM   items
-> WHERE  code IN ( SELECT item
->                  FROM   orders
->                  WHERE  customer='apple' );
+-----+
| name |
+-----+
| 주스 |
| 라면 |
+-----+
2 rows in set (0.19 sec)

```

- 부속 질의문과 결과

```
SELECT item
FROM   orders
WHERE  customer='apple';
```



```
mysql> SELECT item
-> FROM   orders
-> WHERE  customer='apple';
+-----+
| item |
+-----+
| E06  |
| E05  |
| E06  |
| E05  |
+-----+
4 rows in set (0.00 sec)

mysql>
```

- 주 질의문과 결과

```
SELECT name
FROM   items
WHERE  code IN ('E06', 'E05', 'E06', 'E05' );
```



```
mysql> SELECT name
-> FROM   items
-> WHERE  code IN ('E06', 'E05', 'E06', 'E05' );
+-----+
| name |
+-----+
| 주스 |
| 라면 |
+-----+
2 rows in set (0.00 sec)

mysql>
```


4. 질의 4

- apple 고객이 주문하지 않은 제품의 이름(name)과 재고(qty)를 검색하시오.
- 부속질의문 : apple 고객이 주문한 제품 코드 검색

```
SELECT code FROM orders WHERE customer='apple';
```

- 주질의문 : 제품코드가 부속질의문에서 검색한 결과에 속해 있지 않으면 참

```
SELECT name FROM items WHERE code NOT IN 부속질의문_결과;
```

- Orders 와 Items 테이블

Items

code	name	stocks	price
E01	김	30	1500
E02	단무지	45	3250
E03	햄	15	7000
E04	우유	25	4560
E05	주스	20	2300
E06	라면	30	2560
ZZZ	삭제상품	0	0

Orders

num	customer	item	qty	date	saddr
00001	apple	E06	15	2015-06-01	경기 용인시
00002	apple	E05	2	2015-06-01	경기 용인시
00003	eagle	E01	3	2015-06-03	부산 남구
00004	bank	E04	5	2015-06-03	서울 종로구
00005	bird	E02	10	2015-06-04	충남 천안시
00006	bird	E03	1	2015-06-04	충남 천안시
00007	apple	E06	10	2015-06-10	NULL
00008	apple	E05	10	2015-06-10	NULL

- 질의문

```
SELECT name
FROM items
WHERE code NOT IN ( SELECT item
                     FROM orders
                     WHERE customer='apple' );
```

- 결과

```
mysql> SELECT name
-> FROM items
-> WHERE code NOT IN ( SELECT item
-> FROM orders
-> WHERE customer='apple' );
+-----+
| name |
+-----+
| 김단무지 |
| 햄우유 |
| 삭제상품 |
+-----+
5 rows in set (0.06 sec)

mysql>
```

5. 속성의 데이터 타입 - 숫자와 날짜

- 부속 질의문
 - . SELECT 문 안에 다른 SELECT 문을 포함
 - . 부속 질의문 / 서브 질의문 (Sub Query) : 내포된 SELECT 문
 - . 상위 질의문 / 주 질의문 (Main Query) : 다른 SELECT 문을 포함하는 SELECT 문

표. 데이터 타입 (숫자)

데이터 타입	설명	크기
TINYINT(size)	-128 ~ 127 (부호없는 정수* : 0 ~ 255) 최대 숫자를 괄호안에 지정할 수 있다.	2 ⁸
SMALLINT(size)	-32768 ~ 32767 (부호없는 정수* : 0 ~ 65535)	2 ¹⁶
MEDIUMINT(size)	-8388608 ~ 8388607 (부호없는 정수* : 0 ~ 16777215)	2 ²⁴
INT(size)	-2147483648 ~ 2147483647 (부호없는 정수* : 0 ~ 4294967295)	2 ³²
BIGINT(size)	-9223372036854775808 to 9223372036854775807 (부호없는 정수* : 0 to 18446744073709551615)	2 ⁶⁴
FLOAT(size,d)	실수 (size : 최대 자릿수, d : 소수점의 오른쪽 자릿수)	
DOUBLE(size, d)	실수 (size : 최대 자릿수, d : 소수점의 오른쪽 자릿수)	
DECIMAL(size,d)	고정 소수점으로 표현된 실수의 문자열	

* 부호없는 정수 : UNSIGNED 옵션으로 정수타입을 지정

표. 데이터 타입 (날짜)

데이터 타입	설명
DATE()	YYYY-MM-DD 형식의 날짜 (1000-01-01 ~ 9999-12-31)
DATETIME(*)	YYYY-MM-DD HH:MI:SS 형식의 날짜와 시간 (1000-01-01 00:00:00 ~ 9999-12-31 23:59:59)
TIMESTAMP(*)	타임스탬프, 유닉스 기점 (1970-01-01 00:00:00 UTC) 이후의 초 범위 : 1970-01-01 00:00:01 UTC ~ 2038-01-09 03:14:07 UTC INSERT 와 UPDATE query에서 자동으로 현재 날짜와 시간으로 설정된다. 여러가지 형식 제공 : YYYYMMDDHHMISS, YYMMDDHHMISS, YYYYMMDD, YYMMDD
TIME()	HH:MI:SS 형식의 시간
YEAR()	2자리 또는 4자리의 년도 (4자리 범위 : 1901 ~ 2155, 2자리범위 : 70 ~ 69 = 1970 ~ 2069)

* DATETIME과 TIMESTAMP가 같은 형식을 반환하지만, 다르게 동작됨.

【학습정리】

1. 여러 개의 테이블을 연결해주는 속성인 조인 속성을 이용하여 원하는 데이터를 여러 테이블로부터 검색할 수 있다.
2. 조인 속성은 외래키를 사용하며, 반드시 사용되는 속성 값의 도메인은 같아야 한다.