

# 9주차 2차시 데이터 검색 2

## 【학습목표】

- 1. 검색조건을 정확히 모르는 경우와 부분적으로 일치하는 데이터를 검색하기 위한 경우의 조건작성법을 설명할 수 있다.
- 2. 속성 값을 지정하여 저장하지 않는 경우, 자동으로 NULL값이 저장될 경우, 속성값이 NULL과 같은지를 확인하기 위한 조건을 작성하는 방법을 습득한다.

## 학습내용1 : LIKE를 이용한 검색

- 검색 조건을 정확히 모르는 경우
- 부분적으로 일치하는 데이터를 검색하려는 경우  
==> LIKE 키워드를 사용
- 검색 조건을 정확히 알면 = 연산자로 조건을 표현
- LIKE 키워드는 문자열을 이용하는 조건에만 사용할 수 있음
- LIKE 키워드와 같이 사용할 수 있는 기호

기호	의미
%	0개 이상의 문자를 의미 (문자의 내용과 개수는 상관 없음)
_	한 개의 문자만 의미 (문자의 내용은 상관 없음)

- LIKE 키워드의 사용 예

예	의미
LIKE '김%'	'김'으로 시작하는 문자열 ( '김'으로 시작하고, 길이는 상관 없음)
LIKE '%김'	'김'으로 끝나는 문자열 ( '김'으로 끝나기만 하면 길이는 상관 없음)
LIKE '%김%'	'김'을 포함하는 문자열
LIKE '김_ _'	'김'으로 시작하는 3개의 문자를 가진 문자열
LIKE '김_성'	'김'으로 시작하고 '성'으로 끝나며 3개의 문자를 가 진 문자열
LIKE '_창%'	두번째 문자가 '창'인 문자열

## 1. 질의 1

- Customer 테이블에서 성이 김씨인 name, account, grade를 검색하시오.
- Customer 테이블의 데이터

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

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 중구
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)

mysql>
```

-질의문

```
SELECT name, account, grade
FROM customer
WHERE name LIKE '김%';
```

```
mysql> SELECT name, account, grade
-> FROM customer
-> WHERE name LIKE '김%';
```

name	account	grade
김돌	pencil	Silver

```
1 row in set (0.02 sec)

mysql>
```

## 2. 질의 2

- Customer 테이블에서 account가 5자인 고객의 account, name, grade를 검색하시오.
- Customer 테이블의 데이터

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

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 용인구
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)

mysql>
```

- 질의문

```
SELECT account, name, grade
FROM customer
WHERE account LIKE '____';
```

```
mysql> SELECT account, name, grade
-> FROM customer
-> WHERE account LIKE '____';
```

account	name	grade
apple	이남이	VIP
eagle	박세재	Gold

```
2 rows in set (0.00 sec)

mysql>
```

### 3. 질의 3

- Customer 테이블에서 name(이름)의 마지막자가 '동'인 고객의 name, grade를 검색하시오. ( name을 3자로 가정한 경우 )

[TIP]

- \* 한글 코드가 2바이트라고 하여, 검색시 한글 한글자를 '\_' \_'으로 사용하면 안된다.
- \* 한글 문자도 밑줄 한 개로 표현한다.

- Customer 테이블의 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)
```

- 올바른 질의문

```
SELECT name, grade
FROM customer
WHERE name LIKE '_ _동';
```

- 잘못된 질의문

```
SELECT name, grade
FROM customer
WHERE name LIKE '_ _ _동';
```

```
mysql> SELECT name, grade
-> FROM customer
-> WHERE name LIKE '_ _동';
```

name	grade
홍길동	Gold

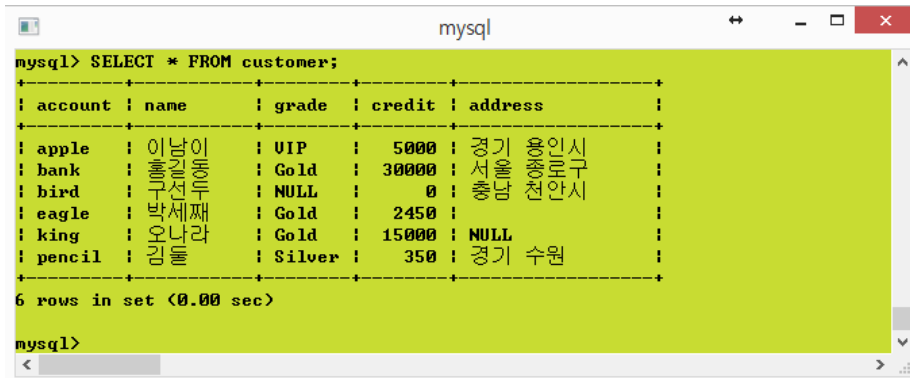
```
1 row in set (0.00 sec)
```

```
mysql> SELECT name, grade
-> FROM customer
-> WHERE name LIKE '_ _ _동';
```

Empty set (0.02 sec)

## 4. 질의 4

- Customer 테이블에서 name(이름)이 '동'으로 끝나는 고객의 name, credit을 검색하시오.  
( 이름의 길이가 몇 자인지 지정할 필요가 없는 경우 )
- Customer 테이블의 데이터



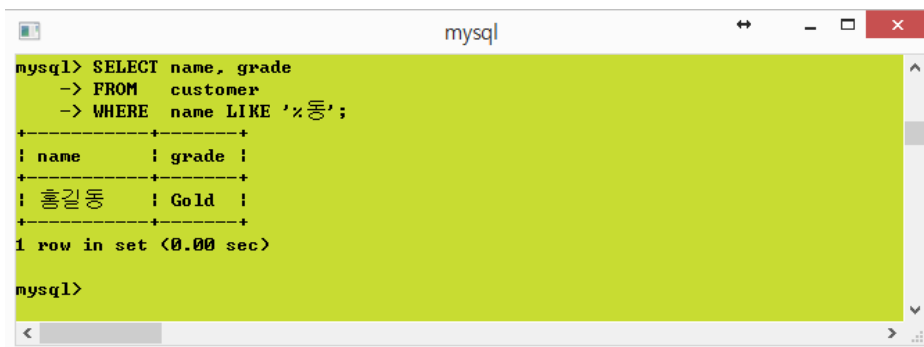
```
mysql> SELECT * FROM customer;
```

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 name, grade
FROM   customer
WHERE  name LIKE '%동';
```



```
mysql> SELECT name, grade
-> FROM   customer
-> WHERE  name LIKE '%동';
```

name	grade
홍길동	Gold

```
1 row in set (0.00 sec)
```

## 5. 질의 5

- Orders 테이블에서 '경기'로 발송된 주문의 item, date, saddr를 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	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

```
6 rows in set (0.00 sec)

mysql>
```

- 질의문

```
SELECT item, date, saddr
FROM   orders
WHERE  saddr LIKE '%경기%';
```

```
mysql> SELECT item, date, saddr
-> FROM   orders
-> WHERE  saddr LIKE '%경기%';
```

item	date	saddr
E06	2015-06-01	경기 용인시
E05	2015-06-01	경기 용인시

```
2 rows in set (0.00 sec)

mysql>
```

## 학습내용2 : NULL을 이용한 검색

- 검색 조건에서 속성의 값이 NULL인지 비교하기  
==> IS NULL 키워드 사용
- 검색 조건에서 속성의 값이 NULL이 아닌지 비교하기  
==> IS NOT NULL 키워드 사용

### 1. 질의 1

- Customer 테이블에서 주소(address)가 입력되지 않은 고객의 account, name을 검색하시오.

[TIP]-----

- \* NULL인지 비교하기 위하여 속성이름=NULL을 사용하면 안됨
- \* 튜플을 수정 (UPDATE) 할 경우

```
UPDATE customer
SET address=NULL
WHERE account='king';
```

-----[END TIP]

- Customer 테이블의 데이터



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

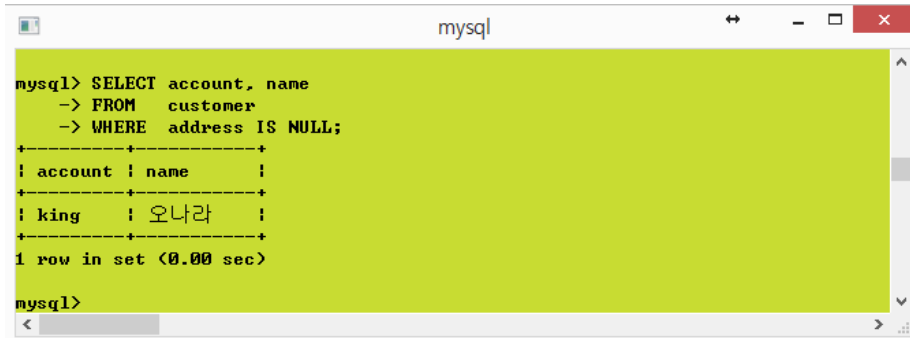
account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 종로구
bird	구선두	NULL	0	충남 천안시
eagle	박세재	Gold	2450	
king	오나라	Gold	15000	NULL
pencil	김돌	Silver	350	경기 수원

```
6 rows in set (0.00 sec)

mysql>
```

- 옳은 질의문

```
SELECT account, name
FROM customer
WHERE address IS NULL;
```

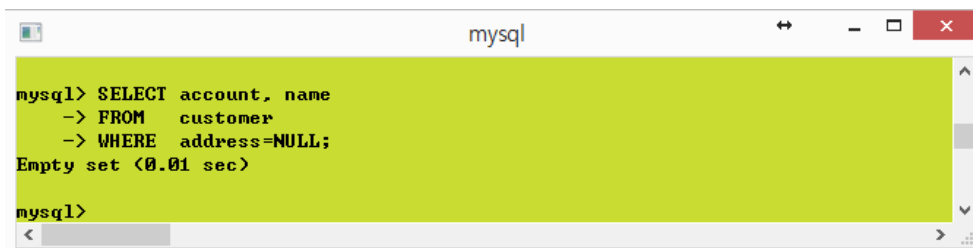


```
mysql> SELECT account, name
-> FROM customer
-> WHERE address IS NULL;
+-----+-----+
| account | name   |
+-----+-----+
| king    | 오나라 |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

- 잘못된 질의문

```
SELECT account, name
FROM customer
WHERE address=NULL;
```



```
mysql> SELECT account, name
-> FROM customer
-> WHERE address=NULL;
Empty set (0.01 sec)

mysql>
```

## 2. 질의 2 : 문자열 데이터 타입과 NULL 값

- Customer 테이블에서 주소(address)가 입력된 고객의 이름(name)과 주소(address)를 검색하시오.

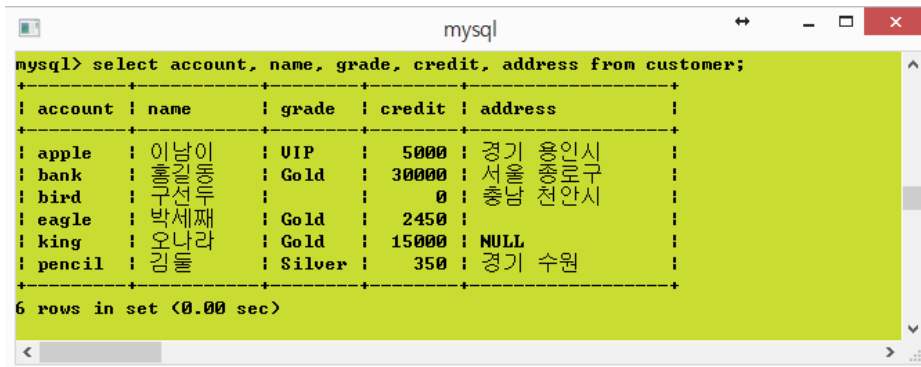
[TIP]-----

- \* CHAR 데이터타입의 값을 ' '와 같이 데이터를 전혀 입력하지 않은 것과 NULL은 다름
- \* INT의 0 값은 NULL과 다름
- \* 문자열 값이 입력되지 않은 속성은 NULL 값을 주는 것이 좋음
- \* INT 값을 입력하지 않을 경우, DEFAULT를 사용하여 0 으로 초기화하는 것이 좋음

-----[END TIP]



## - Customer 테이블의 데이터



```
mysql> select account, name, grade, credit, address from customer;
```

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

6 rows in set (0.00 sec)

## - 질의문

```
SELECT name, address
FROM   customer
WHERE  address IS NOT NULL;
```



```
mysql> SELECT name, address
-> FROM   customer
-> WHERE  address IS NOT NULL;
```

name	address
이남이	경기 용인시
홍길동	서울 종로구
구선두	충남 천안시
박세채	
김돌	경기 수원

5 rows in set (0.00 sec)

```
mysql>
```

## - 잘못된 질의문

. NULL 이 아닌 조건은 ' < > NULL '을 사용하면 안됨.

```
SELECT name, address
FROM   customer
WHERE  address <> NULL;
```

```
mysql> SELECT name, address
-> FROM customer
-> WHERE address <> NULL;
Empty set (0.00 sec)

mysql>
```

- 속성값은 ' '와 같이 빈 문자열을 사용한 질의문

```
SELECT name, address
FROM customer
WHERE address='';
```

```
mysql> SELECT name, address
-> FROM customer
-> WHERE address='';
+-----+-----+
| name | address |
+-----+-----+
| 박세재 |      |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

### 3. 질의 3 : INT 데이터 타입의 NULL 크기 비교

- Items 테이블에서 단가(price) 값을 입력하지 않은 제품의 코드(code), 이름(name)을 검색하시오.
- Items 테이블의 데이터

```
mysql> select * from items;
+----+----+-----+-----+
| code | name | stocks | price |
+----+----+-----+-----+
| E01 | 김 | 30 | 1500 |
| E02 | 단무지 | 45 | 3250 |
| E03 | 행 | 15 | 7000 |
| E04 | 우유 | 25 | 4560 |
| E05 | 주스 | 20 | 2300 |
| E06 | 라면 | 30 | NULL |
| ZZZ | 삭제상품 | 0 | 0 |
+----+----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

- 질의문

```
SELECT code, name
FROM items
```

WHERE price IS NULL;

```
mysql> SELECT code, name
-> FROM items
-> WHERE price IS NULL;

+----+-----+
| code | name |
+----+-----+
| E06 | 라면 |
+----+-----+
1 row in set (0.00 sec)

mysql>
```

#### 4. 질의 4

- Items 테이블에서 단가(price)가 0원인 제품의 코드(code), 이름(name)을 검색하시오.
- Items 테이블의 데이터

```
mysql> select * from 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 | NULL |
| ZZZ | 삭제상품 | 0 | 0 |
+----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

- 질의문

```
SELECT code, name
FROM items
WHERE price=0;
```

```
mysql> SELECT code, name
-> FROM items
-> WHERE price=0;

+----+-----+
| code | name |
+----+-----+
| ZZZ | 삭제상품 |
+----+-----+
1 row in set (0.00 sec)

mysql>
```

## 5. 질의 5

- Items 테이블에서 단가가 2000원 미만인 제품의 코드(code), 제품명(name), 단가(price)를 검색하시오.

[TIP] -----

- \* NULL 값은 숫자로 비교할 수 없다.
- \* NULL 값은 다른 값과 크기를 비교하면 항상 FALSE 이다.

price > 2000

price < 2000

price <> 2000

price >= 2000

price <= 2000

price = 2000

-----[ END TIP ]

- Items 테이블의 데이터

```
mysql> select * from 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	NULL
ZZZ	삭제상품	0	0

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

```
mysql>
```

- 질의문

```
SELECT code, name
FROM items
WHERE price < 2000;
```

```
mysql> SELECT code, name
-> FROM items
-> WHERE price < 2000;
```

code	name
E01	김
ZZZ	삭제상품

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

```
mysql>
```

### 【학습정리】

1. 검색 조건을 정확히 모르거나 부분적으로 일치하는 데이터를 검색하는 경우에 LIKE를 사용한다.
2. 속성의 값이 입력되지 않아 NULL로 저장된 경우, NULL과 같은지를 비교하기 위하여 IS NULL을 사용하고, NULL과 다른지를 비교하기 위하여 IS NOT NULL을 사용한다.