

12주차 1차시 내장 함수 1

【학습목표】

1. 내장 함수의 의미를 알아보고, 내장 함수의 종류를 설명할 수 있다.
2. SQL 질의문에서 사용되는 표현식에서 자동 데이터타입 변환 동작을 설명할 수 있다.
3. MySQL에서 제공하는 연산자의 종류에 대해 설명할 수 있다.
4. MySQL에서 제공하는 제어흐름함수에 대하여 알아보고 실습한다.

학습내용1 : 내장 함수 (Build-In Function)

- MySQL의 내장 함수 분류

분류	종류
연산자 (Operators)	+, -, AND, IS NULL 등
제어 흐름 함수 (Control Flow Functions)	CASE, IF(), IFNULL(), NULLIF()
날짜와 시간 함수 (Date & Time Functions)	CURTIME, DATE, DAY, TIMESTAMPS 등
숫자 함수 (Numeric Functions)	SIN, COS, ABS, ROUND, CEIL 등
문자열 함수 (String Functions)	LIKE, CONCAT 등
전체 텍스트 검색 함수 (Full-Text Search Functions)	MATCH() ... AGAINST 함수
타입 변환 함수 (Type Conversion Functions)	BINARY, CAST, CONVERT
비트 함수 (Bit Functions)	BIT_COUNT, &, ~, , ^, <<, >>
암호화 및 압축 함수 (Encryption & Compression Functions)	ENCODE, DECODE, COMPRESS, UNCOMPRESS 등
정보 함수 (Information Functions)	CHARSET, CURRENT_USER, SESSION_USER 등
GROUP BY 함수 (GROUPBY Functions)	AVG, COUNT, MAX, GROUP_CONCAT

- 내장함수는 상수나 속성이름을 입력 값으로 받아 단일 값을 결과로 반환
- 모든 내장 함수는 최초에 선언될 때 유효한 입력 값을 받아야 함
 - . 수학 함수의 입력 값 : 정수 또는 실수
- 선언에 위배된 값이 입력되면 질의는 실행을 중지하고 오류 메시지를 출력
- 내장함수는 SELECT절과 WHERE절, UPDATE SET절에 사용 가능
 - . 데이터 검색과 수정에만 내장함수를 사용

```

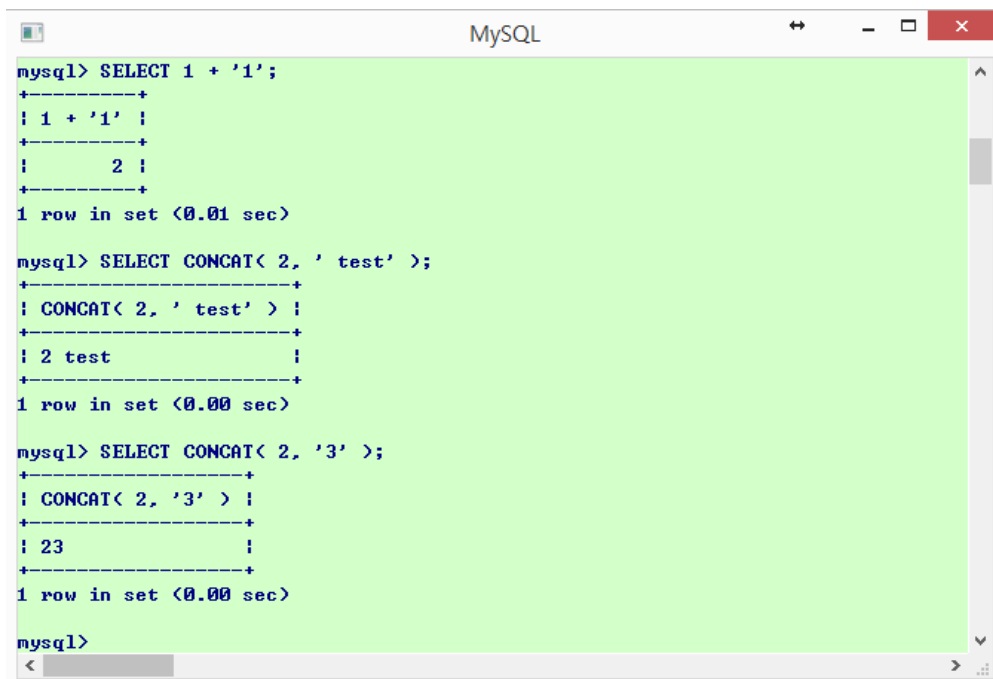
SELECT    ... 함수명(인자)
FROM      테이블이름
WHERE     ... 열이름=함수명(인자) ;

UPDATE   테이블이름
SET      ... 열이름=함수명(인자) ;

```

학습내용2 : 표현식에서 타입 변환

- 데이터 타입이 다른 피연산자를 처리하기 위하여 연산자에 호환되도록 피연산자의 타입을 변환한다.
- 일부 변환은 암시적으로 발생한다.
- 필요에 따라 숫자를 문자열로 또는 문자열을 숫자로 자동으로 변환한다.



```

mysql> SELECT 1 + '1';
+-----+
| 1 + '1' |
+-----+
|      2 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT CONCAT( 2, ' test' );
+-----+
| CONCAT( 2, ' test' ) |
+-----+
| 2 test               |
+-----+
1 row in set (0.00 sec)

mysql> SELECT CONCAT( 2, '3' );
+-----+
| CONCAT( 2, '3' ) |
+-----+
| 23               |
+-----+
1 row in set (0.00 sec)

mysql>

```

- CAST() 함수를 사용하여 명확하게 변환하는 것도 가능하다.

1. 연산자 (Operators)

술어	연산자	예
대입	=, :=	값을 대입
비교	=, <, !=, <=, >, >=	price < 2000
범위	BETWEEN	price BETWEEN 1000 AND 10000
집합	IN, NOT IN	price IN (10000, 20000, 30000)
패턴	LIKE	bookname LIKE '축구의 역사'
NULL	IS NULL, IS NOT NULL	price IS NULL
논리	AND, &&, OR,	(price < 20000) AND (bookname LIKE '축구의 역사')
비트	&, , ^, ~	29 & 15 → 13 (29:11101, 15:1111, 13:1101)
시프트	<<, >>	1 << 2 → 4

학습내용3 : 제어 흐름 함수 (Control Flow Functions)

Name	Description
CASE	CASE 연산자
IF()	IF/ELSE 구조
IFNULL()	NULL IF/ELSE 구조
NULLIF()	expr1 = expr2 이면 NULL을 반환

1. CASE 연산자

- 기본 형식 1

```

CASE 속성이름
  WHEN 값1 THEN 반환값1
  .....
  WHEN 값n THEN 반환값n
  [ ELSE 예외반환값 ]
END

```

- 기본 형식 2

```

CASE
  WHEN 조건1 THEN 반환값1
  .....
  WHEN 조건n THEN 반환값n
  [ ELSE 예외반환값 ]

```

END

- 속성이름의 값이 WHEN 절의 값과 같으면 반환값을 반환
- 예외반환값은 WHEN 절의 모든 값과 같지 않은 경우 반환되는 값

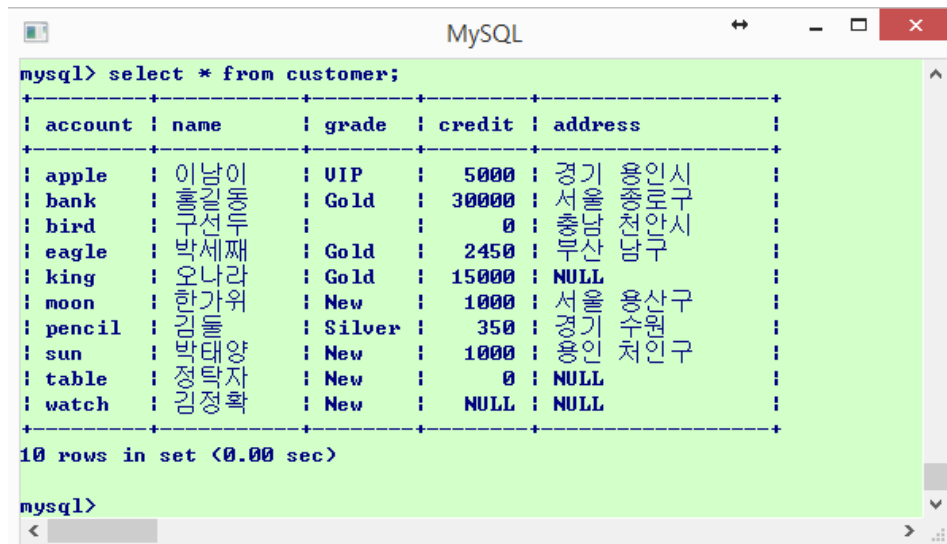
질의 1 :

Customer에서 계정에 대한 등급을 확인하되, 등급을 VIP는 최우수고객, Gold는 우수고객, Silver는 일반고객, New는 새내기, NULL인 경우 자료없음으로 표기하시오.

- CASE 문을 사용
- 비교하기 위한 속성은 grade
- 질의문

```
SELECT account,
CASE grade WHEN 'Gold' THEN '우수고객'
           WHEN 'Silver' THEN '일반고객'
           WHEN 'New' THEN '새내기'
           WHEN 'VIP' THEN '최우수고객'
           ELSE '자료없음'
END AS '등급'
FROM customer;
```

- Customer 테이블 데이터



The image shows a MySQL command window with the following content:

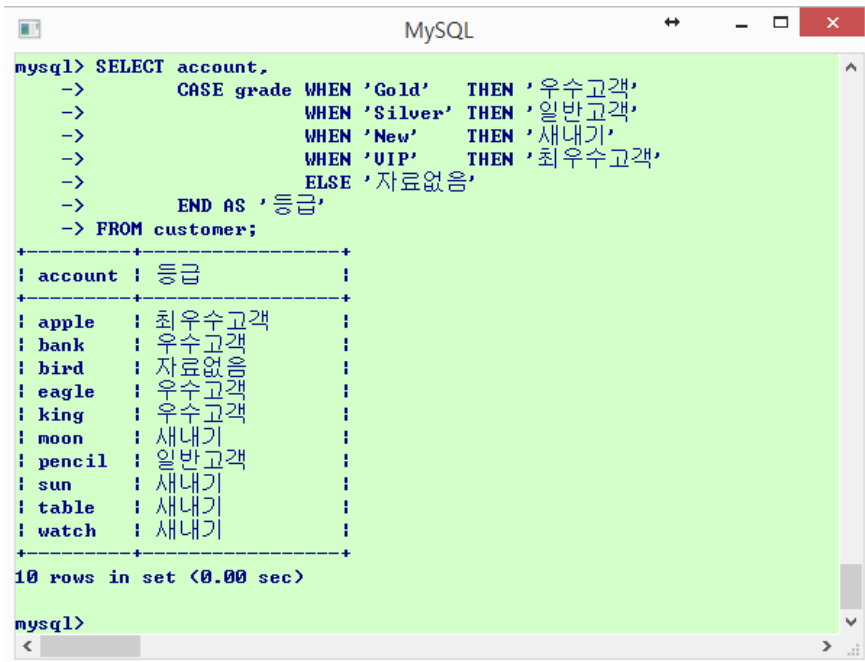
```
mysql> select * from customer;
```

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 종로구
bird	구선두		0	충남 천안시
eagle	박세재	Gold	2450	부산 남구
king	오나라	Gold	15000	NULL
moon	한가위	New	1000	서울 용산구
pencil	김들	Silver	350	경기 수원시
sun	박태양	New	1000	용인 처인구
table	정탁자	New	0	NULL
watch	김정확	New	NULL	NULL

10 rows in set (0.00 sec)

```
mysql>
```

- 질의문 실행 및 결과 데이터



```
mysql> SELECT account,
-> CASE grade WHEN 'Gold' THEN '우수고객',
-> WHEN 'Silver' THEN '일반고객',
-> WHEN 'New' THEN '새내기',
-> WHEN 'VIP' THEN '최우수고객',
-> ELSE '자료없음',
-> END AS '등급',
-> FROM customer;
```

account	등급
apple	최우수고객
bank	우수고객
bird	자료없음
eagle	우수고객
king	우수고객
moon	새내기
pencil	일반고객
sun	새내기
table	새내기
watch	새내기

```
10 rows in set (0.00 sec)

mysql>
```

2. IF () 함수

- 기본 형식

IF (표현식1, 표현식2, 표현식 3)

- 표현식1 이 참이면, 표현식2를 반환
- 표현식1 이 거짓이면, 표현식3을 반환

질의 2 :

고객이 주문한 판매금액이 10000원을 초과하면 배송료를 면제하기 위한 질의문을 작성하시오. 단, 결과 테이블은 주문번호(orders.num), 계정(orders.customer), 주문물품(orders.item), 판매금액, 배송비(무료배송일 경우, '무료배송', 유료 배송비는 2500)를 반환한다.

- (1) 질의에 필요한 테이블 : orders, items
- (2) 판매금액 계산 : orders.qty * items.price
- (3) 배송비를 계산 :

IF ((orders.qty * items.price) > 10000, '무료배송', '2500') AS '배송비'

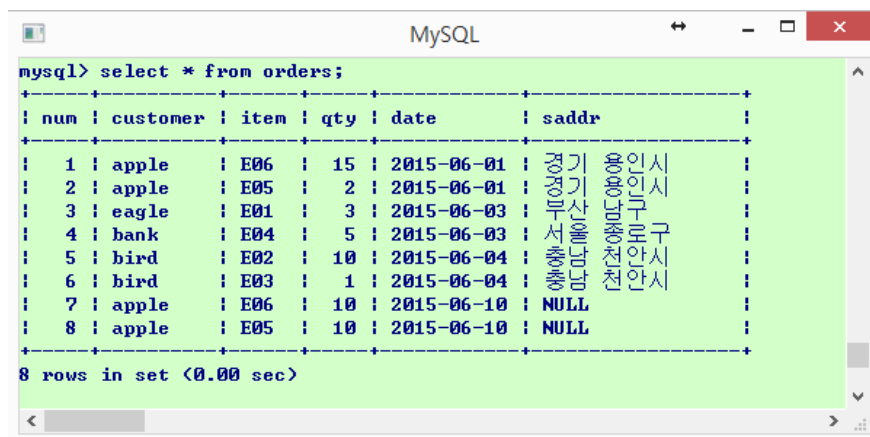
- 질의문

```

SELECT orders.num, orders.customer, orders.item,
       ( orders.qty * items.price ) AS '판매금액',
       IF ( ( orders.qty * items.price ) > 10000,
           '무료배송', '2500' ) AS '배송비'
FROM   orders, items
WHERE  orders.item = items.code;

```

- Orders 테이블 데이터

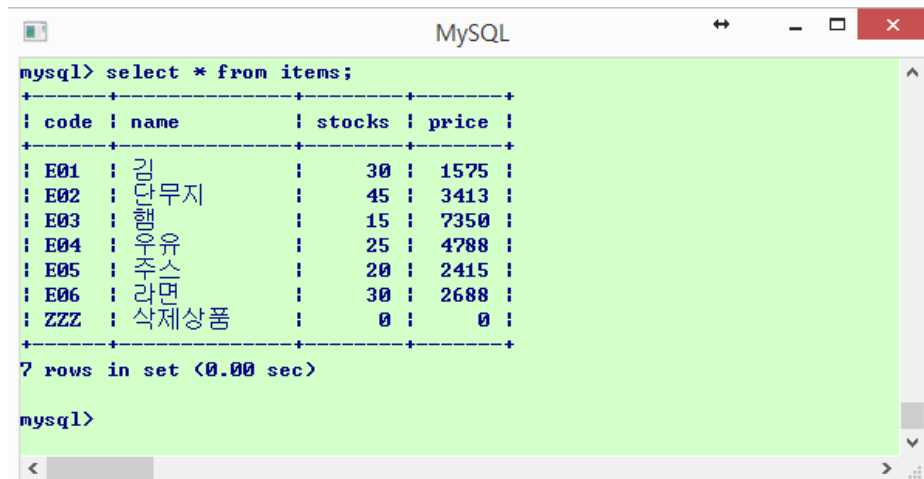


```
mysql> select * from orders;
```

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	E02	10	2015-06-04	충청남도 천안시
6	bird	E03	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)

- Items 테이블 데이터



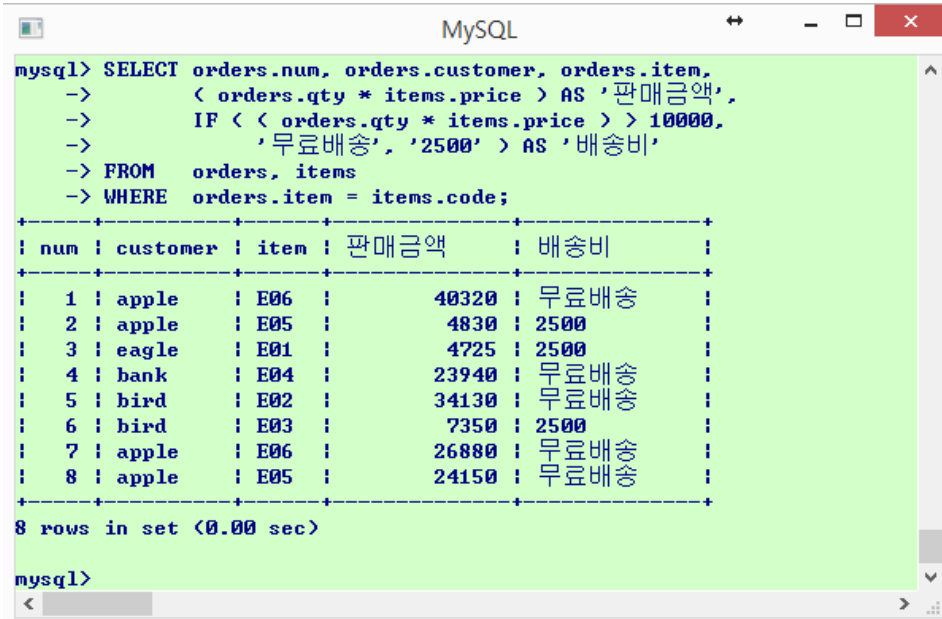
```
mysql> select * from items;
```

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

7 rows in set (0.00 sec)

```
mysql>
```

- 질의문 실행 및 결과 데이터



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

num	customer	item	판매금액	배송비
1	apple	E06	40320	무료배송
2	apple	E05	4830	2500
3	eagle	E01	4725	2500
4	bank	E04	23940	무료배송
5	bird	E02	34130	무료배송
6	bird	E03	7350	2500
7	apple	E06	26880	무료배송
8	apple	E05	24150	무료배송

8 rows in set (0.00 sec)

질의 3 :

고객이 주문한 판매금액이 10000원을 초과하면 배송료를 면제하기 위한 질의문을 작성하시오. 단, 같은 날 동일한 고객이 여러 건 주문한 경우, 묶음 배송으로 처리하고, 결과 테이블은 계정(orders.customer), 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) 같은 판매일에 같은 고객이 주문한 판매금액 계산 :
SUM(orders.qty * items.price)
- (3) 배송비를 계산 :
IF (SUM(orders.qty * items.price) > 10000, '무료배송', '2500')
AS '배송비'

- 질의문

```
SELECT orders.customer,
      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	김	30	1575
E02	다무지	45	3413
E03	행우유	15	7350
E04	주스	25	4788
E05	라면	20	2415
E06	삭제상품	30	2688
ZZZ	삭제상품	0	0

7 rows in set (0.00 sec)

mysql>

- 질의문 실행 및 결과 데이터

MySQL

```
mysql> SELECT orders.num, orders.customer,
-> 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;
```

num	customer	판매금액	배송비
1	apple	45150	무료배송
4	bank	23940	무료배송
3	eagle	4725	2500
5	bird	41480	무료배송
7	apple	51030	무료배송

5 rows in set (0.01 sec)

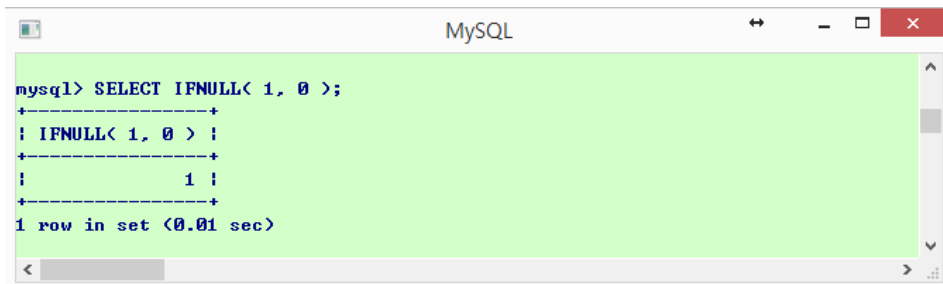
3. IFNULL() 함수

- 기본 형식

IFNULL (표현식1, 표현식2)

- 표현식1이 NULL이 아니면, 표현식1을 반환
- 표현식1이 NULL이면, 표현식2를 반환

- 예제 1 :



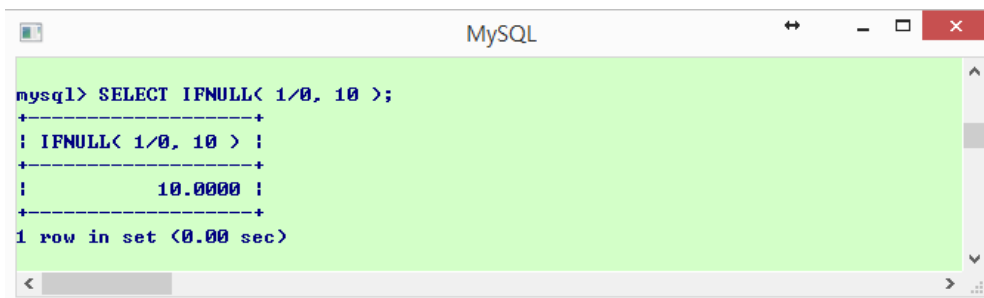
```
mysql> SELECT IFNULL< 1, 0 >;
+-----+
| IFNULL< 1, 0 > |
+-----+
|             1 |
+-----+
1 row in set (0.01 sec)
```

- 예제 2 :



```
mysql> SELECT IFNULL< NULL, 10 >;
+-----+
| IFNULL< NULL, 10 > |
+-----+
|                10 |
+-----+
1 row in set (0.00 sec)
```

- 예제 3 :



```
mysql> SELECT IFNULL< 1/0, 10 >;
+-----+
| IFNULL< 1/0, 10 > |
+-----+
|          10.0000 |
+-----+
1 row in set (0.00 sec)
```

- 예제 4 :

```

MySQL
mysql> SELECT IFNULL( 1/0, 'yes' );
+-----+
| IFNULL( 1/0, 'yes' ) |
+-----+
| yes                  |
+-----+
1 row in set (0.00 sec)

mysql>

```

질의 4 :

고객(customer) 테이블의 계정(account), 고객명(name), 주소(address)를 검색하되, 주소가 NULL로 입력된 경우, '주소없음'으로 표기하고, 각 속성명을 계정, 고객명, 주소로 출력하시오.

1. 각 속성명 '계정', '고객명'으로 표기
account AS '계정', name AS '고객명'
2. 주소가 NULL인 경우 '주소없음'으로 표기
IFNULL(address, '주소없음') AS '주소'

- 질의문

```

SELECT account AS '계정', name AS '고객명',
       IFNULL( address, '주소없음' ) AS '주소'
FROM   customer;

```

- Customer 테이블 데이터

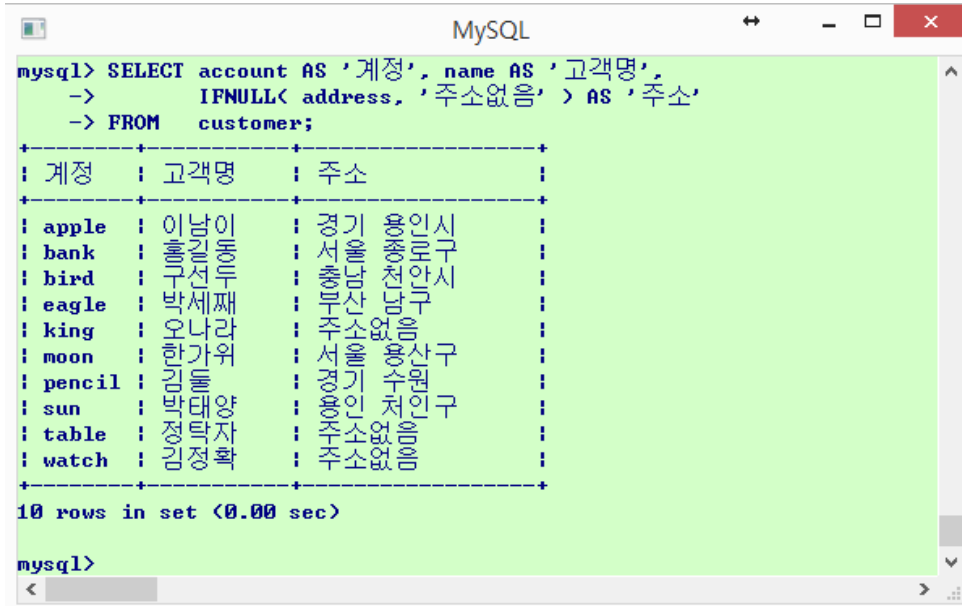
```

MySQL
mysql> select * 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   | 박세재 | Gold  | 2450   | 부산 남구   |
| king    | 오나라 | Gold  | 15000  | NULL        |
| moon    | 한가위 | New   | 1000   | 서울 용산구 |
| pencil  | 김돌돌 | Silver | 350    | 경기 수원시 |
| sun     | 박태양 | New   | 1000   | 용인 처인구 |
| table   | 정택자 | New   | 0      | NULL        |
| watch   | 김정확 | New   | NULL   | NULL        |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>

```

- 질의문 실행 및 결과 데이터



```
mysql> SELECT account AS '계정', name AS '고객명',
-> IFNULL< address, '주소없음' > AS '주소'
-> FROM customer;
```

계정	고객명	주소
apple	이남이	경기 용인시
bank	홍길동	서울 종로구
bird	구선두	충남 천안시
eagle	박세재	부산 남구
king	오나라	주소없음
moon	한가위	서울 용산구
pencil	김돌	경기 수원시
sun	박태양	용인 처인구
table	정탁자	주소없음
watch	김정확	주소없음

```
10 rows in set (0.00 sec)

mysql>
```

4. NULLIF () 함수

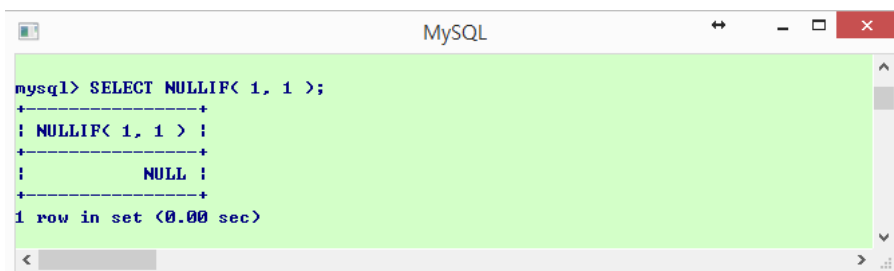
- 기본 형식

NULLIF(표현식1, 표현식2)

- 표현식1과 표현식2가 같으면 NULL을 반환
- 표현식1과 표현식2가 다르면 표현식1을 반환
- 동일한 함수

CASE WHEN 표현식1 = 표현식2 THEN NULL ELSE 표현식1 END

- 예제 1 :

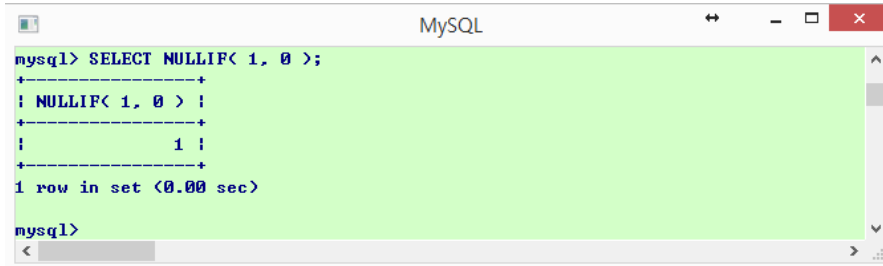


```
mysql> SELECT NULLIF< 1, 1 >;
```

NULLIF< 1, 1 >
NULL

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

- 예제 2 :



```
mysql> SELECT NULLIF< 1, 0 >;
+-----+
| NULLIF< 1, 0 > |
+-----+
|                |
+-----+
1 row in set (0.00 sec)
```

【학습정리】

1. SQL에는 데이터베이스를 편리하게 사용하기 위하여 내장함수를 제공한다.
2. 데이터 타입은 연산자에 따라 자동으로 변환되기도 하지만, 명시적으로 변환하여 사용하는 것이 좋다.
3. 내장함수는 데이터 검색과 수정을 위하여 사용한다.