

# 9강

## 숫자와 문자열 함수





## 1. 숫자 관련 함수들

함수	설명
ROUND	반올림
CEIL	올림
FLOOR	내림

```
SELECT ROUND(0.5), CEIL(0.4), FLOOR(0.6);
```



Number of Records: 1		
ROUND(0.5)	CEIL(0.4)	FLOOR(0.6)
1	1	0

```
SELECT Price, ROUND(price), CEIL(price), FLOOR(price)
FROM Products;
```



Number of Records: 77			
Price	ROUND(price)	CEIL(price)	FLOOR(price)
18.00	18	18	18
19.00	19	19	19
10.00	10	10	10



함수	설명
<b>GREATEST</b>	(괄호 안에서) 가장 큰 값
<b>LEAST</b>	(괄호 안에서) 가장 작은 값

← 영어도 됨 (마스크코드 게임)

**SELECT**

```
GREATEST(1, 2, 3),  
LEAST(1, 2, 3, 4, 5);
```



Number of Records: 1	
<b>GREATEST(1, 2, 3)</b>	<b>LEAST(1, 2, 3, 4, 5)</b>
3	1



※ 그룹 함수 - 조건에 따라 집계된 값을 가져옵니다.

함수	설명
MAX	가장 큰 값
MIN	가장 작은 값
COUNT	개수(NULL 값 제외)
SUM	총합
AVG	평균 값

```
SELECT MAX(Quantity), MIN(Quantity), COUNT(Quantity), SUM(Quantity), AVG(Quantity)
FROM OrderDetails
WHERE OrderDetailID BETWEEN 20 AND 30;
```



Number of Records: 1

MAX(Quantity)	MIN(Quantity)	COUNT(Quantity)	SUM(Quantity)	AVG(Quantity)
50	6	11	254	23.0909



함수	설명
<b>POW(A,B), POWER(A,B)</b>	A를 B만큼 제곱
<b>SQRT</b>	제곱근

```
SELECT POW(2, 3), POWER(5, 2), SQRT(16);
```



Number of Records: 1		
<b>POW(2, 3)</b>	<b>POWER(5, 2)</b>	<b>SQRT(16)</b>
8	25	4



함수	설명
<u>TRUNCATE(N,n)</u>	N을 소수점 n자리까지 선택

올림 / 내림 X

SELECT

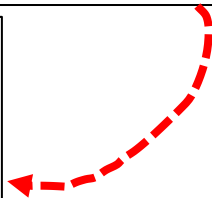
```
TRUNCATE(1234.5678, 1),
TRUNCATE(1234.5678, 2),
TRUNCATE(1234.5678, 3),
TRUNCATE(1234.5678, -1),
TRUNCATE(1234.5678, -2),
TRUNCATE(1234.5678, -3);
```



Number of Records: 1					
TRUNCATE(1234.5678, 1)	TRUNCATE(1234.5678, 2)	TRUNCATE(1234.5678, 3)	TRUNCATE(1234.5678, -1)	TRUNCATE(1234.5678, -2)	TRUNCATE(1234.5678, -3)
1234.5	1234.56	1234.567	1230	1200	1000

SELECT truncate(avg(price),1) from product group by category\_id

truncate(avg(price),1)
5000.0
2150.0
18250.0
2750.0
2050.0
3500.0





## 2. 문자열 관련 함수들

함수	설명
<b>UCASE, UPPER</b>	소문자를 대문자로 변경
<b>LCASE, LOWER</b>	대문자를 소문자로 변경

```
SELECT  
  UPPER('abcDEF'),  
  LOWER('abcDEF');
```



Number of Records: 1	
<b>UPPER('abcDEF')</b>	<b>LOWER('abcDEF')</b>
ABCDEF	abcdef



함수	설명
<b>CONCAT(문자A, 문자B, 문자C....)</b>	괄호 안의 모든 문자를 이어 붙임
<b>CONCAT_WS(구분자, 문자A, 문자B..)</b>	괄호 안의 문자A와 문자B 사이에 구분자를 이어 붙임

※ CONCAT( ), CONCAT\_WS( )란 데이터 조회 시 여러 컬럼에서 조회되는 데이터를 하나의 문자열로 출력해야 하는 경우가 있다. 이러한 상황에서 CONCAT(), CONCAT\_WS() 함수를 사용하여 간단하게 **문자열을 합칠** 수 있다.

```
select concat(city, ' ', district, ' ', detail) as '주소' from address;
```



주소
▶ 서울 강남구 역삼동 123-1
서울 서초구 서초동 88-3
서울 마포구 합정동 33-10
부산 해운대구 좌동 12-2
대구 중구 동성로 55
인천 남동구 구월동 77-8
광주 북구 신안동 19-3
대전 유성구 궁동 45-9
울산 남구 삼산동 90-4
수원 영통구 원천동 24
성남 분당구 정자동 300
고양 일산동구 마두동 22-1

```
select concat(city,null,district,null,detail) as '주소' from address;
```



주소
▶ NULL
NULL
NULL
NULL
NULL
NULL

※ CONCAT( ) 순서대로 합쳐주는 것은 하지만 문자열 사이에 NULL이 포함되어 있다면 NULL을 포함한다는 점에 주의하자!!





```
select concat_ws(' ',city,district,detail) as '주소' from address;
```



주소
▶ 서울 강남구 역삼동 123-1
서울 서초구 서초동 88-3
서울 마포구 합정동 33-10
부산 해운대구 좌동 12-2
대구 중구 동성로 55
인천 남동구 구월동 77-8
광주 북구 신안동 19-3
대전 유성구 궁동 45-9
울산 남구 삼산동 90-4
수원 영통구 원천동 24

```
select concat_ws('-',city,null,district,null,detail) as '주소' from address;
```



주소
▶ 서울_강남구_역삼동 123-1
서울_서초구_서초동 88-3
서울_마포구_합정동 33-10
부산_해운대구_좌동 12-2
대구_중구_동성로 55
인천_남동구_구월동 77-8
광주_북구_신안동 19-3
대전_유성구_궁동 45-9
울산_남구_삼산동 90-4
수원_영통구_원천동 24
성남_분당구_정자동 300
고양_일산동구_마두동 22-1
용인_수지구_죽전동 101
창원_성산구_상남동 77
전주_완산구_서신동 18-5

※ CONCAT\_WS( ) 문자열 사이에 NULL이 포함되어 있어도 NULL을 반환하지 않는다는 점에 주의하자!!



함수	설명
<b>SUBSTR(문자열, 시작위치, 길이), SUBSTRING(문자열, 시작위치, 길이)</b>	시작부터 길이만큼 문자를 반환한다, 단, 길이가 생략되면 문자열의 끝까지 반환한다.
<b>LEFT(문자열, 길이)</b>	왼쪽부터 길이만큼 문자를 반환한다.
<b>RIGHT(문자열, 길이)</b>	오른쪽부터 길이만큼 문자를 반환한다.

SELECT

```
SUBSTR('ABCDEFGF', 3),
SUBSTR('ABCDEFGF', 3, 2),
SUBSTR('ABCDEFGF', -4),
SUBSTR('ABCDEFGF', -4, 2);
```

시작위치  
길이

문자열, 기호 포함



Number of Records: 1			
SUBSTR('ABCDEFGF', 3)	SUBSTR('ABCDEFGF', 3, 2)	SUBSTR('ABCDEFGF', -4)	SUBSTR('ABCDEFGF', -4, 2)
CDEFG	CD	DEFG	DE

SELECT LEFT('ABCDEFGF', 3), RIGHT('ABCDEFGF', 3);



Number of Records: 1	
LEFT('ABCDEFGF', 3)	RIGHT('ABCDEFGF', 3)
ABC	EFG



```
select substr(detail, 1, 3) as '주소 동' from address;
```



	주소 동
▶	역삼동
	서초동
	합정동
	좌동
	동성로
	구월동
	신안동
	궁동
	삼산동



함수	설명
<b>LENGTH</b>	문자열의 바이트 수를 반환
<b>CHAR_LENGTH, CHARACTER_LENGTH</b>	문자열의 개수를 반환

※ 할당된 Bit 크기 또는 문자 크기를 반환한다, CHAR\_LENGTH( )는 문자의 개수를 반환하면, LENGTH( )는 할당된 Byte 수를 반환한다. (단, 공백도 문자로 인식함을 주의하자!) ←

```
select detail, char_length(detail) as '주소길이', length(detail) as '바이트수' from address;
```



detail	주소길이	바이트수
▶ 역삼동 123-1	9	15
서초동 88-3	8	14
합정동 33-10	9	15
좌동 12-2	7	11
동성로 55	6	12
구월동 77-8	8	14
신안동 19-3	8	14
궁동 45-9	7	11
삼산동 90-4	8	14

※ 인코딩 종류에 따른 문자 Byte 크기

한글(한글자 크기)

영어(한글자 크기)

EUR-KR

2BYTE

1BYTE

UTF-8

3BYTE

1BYTE



함수	설명
<b>TRIM(문자열)</b>	문자열의 앞, 뒤 공백을 모두 제거
<b>TRIM(방향 자를 문자열 FROM 문자열)</b>	LEADING(앞), BOTH(양쪽), TRAILING(뒤) 문자열 제거
<b>LTRIM(문자열)</b>	문자열의 왼쪽 공백을 제거
<b>RTRIM(문자열)</b>	문자열의 오른쪽 공백을 제거

※ LTRIM(문자열)/RTRIM(문자열)은中间的 공백을 제거되지 않는다.

```
select ltrim(' 이것이 '), rtrim('이것이 '), ltrim('A B'), rtrim('콩 팥')
```



<b>ltrim(' 이것이 ')</b>	<b>rtrim('이것이 ')</b>	<b>ltrim('A B')</b>	<b>rtrim('콩 팥')</b>
이것이	이것이	A B	콩 팥

```
select trim(' 이것이 '), trim(both('ㅋ') from 'ㅋㅋㅋ 재미있어요 ㅋㅋㅋ')
```



<b>trim(' 이것이 ')</b>	<b>trim(both('ㅋ') from 'ㅋㅋㅋ 재미있어요 ㅋㅋㅋ')</b>
이것이	재미있어요

```
select trim(LEADING('ㅋ') from 'ㅋㅋㅋ 재미있어요 ㅋㅋㅋ'), trim(TRAILING('ㅋ') from 'ㅋㅋㅋ 재미있어요 ㅋㅋㅋ')
```



<b>trim(LEADING('ㅋ') from 'ㅋㅋㅋ 재미있어요 ㅋㅋㅋ')</b>	<b>trim(TRAILING('ㅋ') from 'ㅋㅋㅋ 재미있어요 ㅋㅋㅋ')</b>
재미있어요 ㅋㅋㅋ	ㅋㅋㅋ 재미있어요



함수	설명
<b>LPAD(문자열, 길이, 채울 문자열)</b>	문자열을 길이만큼 늘린 후 채울 문자열을 왼쪽에 이어 붙임
<b>RPAD(문자열, 길이, 채울 문자열)</b>	문자열을 길이만큼 늘린 후에 채울 문자열을 오른쪽에 이어 붙임

※ 문자열을 길이만큼 늘린 후에 빈 곳을 채울 문자열로 채운다. (특정 문자로 자릿수 채울 때 사용)

```
SELECT LPAD('ABC',5,'#'), RPAD('ABC',5,'#')
```



<b>LPAD('ABC',5,'##')</b>	<b>RPAD('ABC',5,'##')</b>
##ABC	ABC##

```
select category_id, lpad(category_name,10,'*') from category;
```



category_id	lpad(category_name,10,'*')
6	*****간식
8	*****공산품
1	*****과일
15	*****기타
12	*****문구류
14	****반려동물용품
7	*****생활용품
4	*****수산물
13	*****스포츠용품
3	*****육류
5	*****음료
10	*****의류
9	*****전자제품
11	*****주방용품
2	*****채소

```
select rpad(substr(user_name, 1, 1), 3, '**') from customer;
```

↳ 김\*\*

```
select replace(user_name, substr(user_name, 2, 1), '*') from customer;
```

↳ 김\*수

```
select concat(replace(user_name, substr(user_name, 2, 1), '*'), ' ', '출석') from customer;
```

↳ 김\*수 출석



찾기

함수	설명
<b>REPLACE(문자열, 원래 문자열, 바꿀 문자열)</b>	문자열에서 원래 문자열을 찾아 바꿀 문자열로 바꿔준다.

※ Replace( )는 찾는 문자열의 대소문자를 구분함을 주의하자

```
SELECT REPLACE('이것이 MySQL이다.','이것이','This is')
```



```
REPLACE('이것이 MySQL이다.','이것이','This is')
```

```
This is MySQL이다.
```

```
select replace(district,'강','서') from address where city='서울' and detail like '%역삼%';
```



```
replace(district,'강','서')
```

```
▶ 서남구
```



```
select category_name, replace(category_name, '과일', '채소1') from category;
```



category_name	replace(category_name, '과일', '채소1')
▶ 간식	간식
공산품	공산품
과일	채소1
기타	기타
문구류	문구류
반려동물용품	반려동물용품
생활용품	생활용품
수산물	수산물
스포츠용품	스포츠용품
육류	육류
음료	음료
의류	의류
전자제품	전자제품
주방용품	주방용품
채소	채소





함수	설명
<b>INSTR(문자열, 찾는 문자열)</b>	문자열중에서 찾는 문자열의 첫 위치 반환, 없을 시 0

※ instr( )는 특정 문자열에서 찾고자 하는 문자가 있는지 확인하고 있으면 시작자리수를, 없는 경우는 0을 반환한다. (단, 찾는 문자열은 대소문자를 구분하지 않는다)

```
SELECT instr('Hello','lo'), instr('Hello','new')
```



<code>instr('Hello','lo')</code>	<code>instr('Hello','new')</code>
4	0



```
select city, district, detail, instr(detail, '동') from address;
```



	city	district	detail	instr(detail, '동')
▶	서울	강남구	역삼동 123-1	3
	서울	서초구	서초동 88-3	3
	서울	마포구	합정동 33-10	3
	부산	해운대구	좌동 12-2	2
	대구	중구	동성로 55	1
	인천	남동구	구월동 77-8	3
	광주	북구	신안동 19-3	3
	대전	유성구	궁동 45-9	2
	울산	남구	삼산동 90-4	3
	수원	영통구	원천동 24	3
	성남	분당구	정자동 300	3
	고양	일산동구	마두동 22-1	3
	용인	수지구	죽전동 101	3
	창원	성산구	상남동 77	3
	전주	완산구	서신동 18-5	3

# 9강

## 연습문제





## ※ 6강에서 작성한 테이블들을 이용하여 아래 문제를 풀어 주세요

1. address 테이블에서 'city + district + detail'을 하나의 문자열로 합쳐 '전체주소'로 출력하시오.

(CONCAT 사용)

2. customer 테이블에서 user\_name, email, phone을 '/'로 구분하여 하나의 문자열로 출력하시오.

(CONCAT\_WS 사용)

3. address 테이블에서 detail 컬럼에서 '동'까지의 첫 2글자를 잘라서 '행정동' 이름만 출력하시오.

(SUBSTR 사용)

4. product 테이블에서 product\_name의 글자수를 CHAR\_LENGTH로 출력하시오. '상품명길이' 컬럼명 사용

5. address 테이블의 detail 값이 몇 바이트인지 LENGTH로 확인하여 '바이트수'라는 컬럼명으로 출력하시오.



## ※ 6강에서 작성한 테이블들을 이용하여 아래 문제를 풀어 주세요

6. category 테이블에서 category\_name을 길이 10으로 하고, 빈 공간은 '#'로 채워서 왼쪽 정렬되도록 출력하시오.  
(LPAD 사용)
7. address 테이블에서 district 컬럼의 '구'를 '군'으로 바꾸어 출력하시오. (REPLACE 사용)
8. product 테이블에서 product\_name에서 '1kg' 문자열을 '1KG'로 변경하여 출력하시오. (REPLACE 사용)
9. address 테이블에서 detail 안에 '동'이라는 단어가 몇 번째에 위치하는지 검색하시오. (INSTR 사용)
10. customer 테이블에서 user\_name과 phone을 연결해 '고객연락처' 형태로 출력하시오. (CONCAT 사용)  
예: 김철수(010-1111-1111)



## ※ 6강에서 작성한 테이블들을 이용하여 아래 문제를 풀어 주세요

11. orders 테이블에서 order\_date의 앞 4자리(연도)만 잘라서 출력하시오. (SUBSTR 사용)
12. product 테이블에서 product\_name과 price를 '상품명: 가격원' 형태로 결합한 새로운 컬럼을 출력하시오.  
(CONCAT\_WS 또는 CONCAT 사용)
13. category 테이블에서 category\_name의 첫 글자만 SUBSTR로 추출하시오.
14. address 테이블에서 city, district를 '-'로 연결한 주소 ID 문자열을 출력하시오. (CONCAT\_WS 사용)  
예: 서울-강남구
15. customer 테이블에서 email 컬럼에서 '@'의 위치를 INSTR로 찾고 '이메일\_골뱅이\_위치'라는 컬럼명으로 출력하시오.