

## 3-1. SQL 함수1 - 문자/숫자/날짜형 함수

홍형경

[chariehong@gmail.com](mailto:chariehong@gmail.com)

2020.06

# 1. 함수 (Function) 란?

- 수학에서 사용하는  $y = f(x)$  와 개념이 동일
- 자주 사용되는 연산(계산)을 정의해 놓은 데이터베이스 객체 → 재사용 목적
- 절대값 처리
  - 양수는 양수 그대로, 0은 0, 음수는 양수로 반환
  - CASE WHEN value1 > 0 THEN value1  
WHEN value1 = 0 THEN 0  
ELSE value1 \* -1  
END
  - 매번 절대값을 구할 때 이 로직을 직접 구현하는 것은 비효율적
  - 따라서 함수로 구현해 놓고 이 함수를 호출하는 것이 바람직함

# 1. 함수 (Function) 란?

- 오라클에서는 기본적인 연산에 대 다양하고 많은 함수를 제공하고 있음  
→ 빌트인(built-in) 함수
- 함수는 특정 로직(연산)을 처리한 결과 값을 반환함
- 연산 대상(피연산자)과 반환 값의 데이터 유형에 따라 크게 문자형, 숫자형, 날짜형 함수로 구분

## 2. 숫자형 함수

- 매개변수, 즉 피연산자와 연산 결과가 모두 숫자인 함수
- 수학에서 사용하는 함수와 동일

## 2. 숫자형 함수

함수	기능	사용예	반환값
ABS (n)	n의 절대값 반환	SELECT ABS( -10 ) FROM DUAL;	10
CEIL ( n )	n과 같거나 큰 최소 정수 반환	SELECT CEIL( 5.5 ) FROM DUAL;	6
FLOOR ( n )	n과 같거나 작은 최대 정수 반환	SELECT FLOOR( 5.5 ) FROM DUAL;	5
EXP ( n )	e( $e = 2.71828183\dots$ )의 n승 반환	SELECT EXP( 5 ) FROM DUAL;	148.413159 ..
LN ( n )	n의 자연로그 값을 반환 ( $n > 0$ )	SELECT LN( 5 ) FROM DUAL;	1.60943791243 ...
LOG ( n2, n1 )	n2는 밑, n1은 진수. n1은 양수, n2 는 0과 1이 아닌 양수	SELECT LOG( 10, 1000 ) FROM DUAL;	3

## 2. 숫자형 함수

함수	기능	사용예	반환값
MOD ( n2, n1 )	n2를 n1로 나눈 나머지 반환	SELECT MOD( 17, 3 ) FROM DUAL;	2
POWER ( n2, n1 )	n2의 n1승을 반환	SELECT POWER( 5, 2 ) FROM DUAL;	25
ROUND ( n, i )	n의 소수점 기준 ( i+1 )번째에서 반올림한 값을 반환, 정수로 만들 시 i는0(소수점 첫째 자리 기준 반올림)	SELECT ROUND( 3.545, 2 ) FROM DUAL;	3.55
SIGN ( n )	n > 0이면 1, n < 0이면 -1, n=0이면 0 반환	SELECT SIGN( -19 ) FROM DUAL;	-1
SQRT ( n )	n의 제곱근 반환	SELECT SQRT( 3 ) FROM DUAL;	1.732 ...
TRUNC ( n1, n2 )	n1의 소수점 기준 n2 자리에서 절사, n2 생략 시 0이 적용	SELECT TRUNC( 3.545, 2 ) FROM DUAL;	3.54

### 3. 문자형 함수

- 매개변수, 즉 피연산자가 문자형인 함수, 반환 결과는 문자형 혹은 숫자형

함수	기능	사용예	반환값
CONCAT ( chr1, chr2 )	chr1과 chr2 문자를 결합한 결과 반환,    연산자와 같은 기능	SELECT CONCAT( 'A', 'B' ) FROM DUAL;	AB
INITCAP ( chr )	chr의 첫 번째 문자를 대문자로 변환	SELECT INITCAP( 'sql' ) FROM DUAL;	Sql
LOWER ( chr )	chr을 소문자로 변환	SELECT LOWER('SQL') FROM DUAL;	sql
UPPER ( chr )	chr을 대문자로 변환	SELECT UPPER( 'sql' ) FROM DUAL;	SQL
LPAD ( expr1, n, expr2 )	expr1을 반환하는데, expr2를 (n - expr1 길 이) 만큼 왼쪽을 채워 반환	SELECT LPAD( 'SQL', 5, '*' ) FROM DUAL;	**SQL

### 3. 문자형 함수

함수	기능	사용예	반환값
RPAD ( expr1, n, expr2 )	expr1을 반환하는데, expr2를 (n - expr1 길이) 만큼 오른쪽을 채워 반환	SELECT RPAD('SQL', 5, '*') FROM DUAL;	SQL**
LTRIM ( expr1, expr2 )	expr1의 왼쪽에서 expr2를 제거한 결과를 반환	SELECT LTRIM('**SQL**', '*') FROM DUAL;	SQL**
RTRIM ( expr1, expr2 )	expr1의 오른쪽에서 expr2를 제거한 결과를 반환	SELECT RTRIM('**SQL**', '*') FROM DUAL;	**SQL
SUBSTR ( chr, n1, n2 )	<ul style="list-style-type: none"> <li>- chr에서 n1에서 시작해 n2 만큼 잘라낸 결과를 반환</li> <li>- n1을 0으로 명시하면 1이 적용</li> <li>- n1이 음수이면 chr 오른쪽 끝에서부터 거꾸로 세어 가져옴</li> <li>- n2를 생략하면 n1부터 끝까지 반환</li> </ul>	SELECT SUBSTR('ABCDEFGF', 3, 2) FROM DUAL;	CD



### 3. 문자형 함수

함수	기능	사용예	반환값
TRIM ( chr )	chr의 양쪽 끝 공백을 제거한 결과를 반환	SELECT TRIM(' ABCD EFG ') FROM DUAL;	ABCD EFG
ASCII ( chr )	chr문자의 ASCII 코드 값을 반환	SELECT ASCII('A') FROM DUAL;	65
LENGTH ( chr )	chr 문자의 글자 수를 반환	SELECT LENGTH('SQL') FROM DUAL;	3
LENGTHB(chr)	chr 문자의 바이트수 반환	SELECT LENGTHB('김') FROM DUAL;	3

### 3. 문자형 함수

함수	기능	사용예	반환값
<code>REPLACE ( chr, serch_str, rep_str )</code>	Chr에서 serch_str을 찾아 rep_str로 대체	<code>SELECT REPLACE('ABC', 'B', 'D') FROM DUAL;</code>	ADC
<code>INSTR ( chr1,chr2, n1, n2 )</code>	<ul style="list-style-type: none"><li>- chr1에서 chr2 문자를 찾아 그 시작 위치 반환</li><li>- n1은 chr1에서 몇 번째 문자부터 찾을 것인지를 나타냄. 생략 시 1이 적용</li><li>- n2는 chr1에서 chr2 문자를 찾을 때 일치하는 문자의 몇 번째 위치를 반환할지를 나타냄. 생략 시 1이 적용됨</li></ul>	<code>SELECT INSTR('ABABAB', 'A', 2) FROM DUAL;</code>	3

## 4. 날짜형 함수

- 날짜, 즉 데이터 형이 DATE나 TIMESTAMP인 데이터를 대상으로 연산을 수행하는 함수

함수	기능	사용예	반환값
<b>SYSDATE</b>	현재 일자와 시간을 반환 (오라클 설치된 서버시간)	<b>SELECT SYSDATE FROM DUAL;</b>	<b>2019-10-10 10:10:10</b>
<b>ADD_MONTHS</b> ( date, n )	date 날짜에 n개월을 더한 날짜를 반환	<b>SELECT ADD_MONTHS(SYSDATE, 1 ) FROM DUAL</b>	<b>2019-11-10 10:10:10</b>
<b>MONTHS_BETWEEN</b> ( date1, date2 )	date1과 date2 두 날짜 사이의 개월 수를 반환. date1 > date2 이면 양수, 반대면 음수	<b>SELECT MONTHS_BETWEEN(SYSDATE + 30, SYSDATE ) FROM DUAL;</b>	<b>1</b>
<b>LAST_DAY</b> ( date )	date가 속한 월의 마지막 일자를 반환	<b>SELECT LAST_DAY ( SYSDATE ) FROM DUAL;</b>	<b>2019-10-31 10:10:10</b>

## 4. 날짜형 함수

함수	기능	사용예	반환값
NEXT_DAY ( date, expr )	date 날짜를 기준으로 expr에 명시한 날짜 반환. expr : '월요일' ... or 1~ 7형태로 쓸 수도 있고 1~7까지 숫자를 쓸 수도 있음(1은 일요일, 7은 토요일)	<b>SELECT NEXT_DAY ( SYSDATE, ' 화요일' ) FROM DUAL;</b>	<b>2019-10-15 10:10:10</b>
ROUND ( date, format )	date를 format 기준으로 반올림한 날짜 반환. format은 YYYY, MM, DD, HH, HH24, MI 등 사용 가능, 생략 시 DD	<b>SELECT ROUND ( SYSDATE, 'YEAR' ) FROM DUAL;</b>	<b>2020-01-01 00:00:00</b>
TRUNC ( date, format )	date를 format 기준으로 잘라낸 날짜 반환. format은 ROUND 함수와 동일하게 사용 가능	<b>SELECT TRUNC( SYSDATE, 'YEAR' ) FROM DUAL;</b>	<b>2019-01-01 10:10:10</b>

## 5. SQL 함수 실습

### (1) 숫자형 함수 – ABS (절대값 반환)

```
SELECT ABS(-7), ABS(0), ABS(7.8)  
FROM DUAL;
```

ABS(-7)	ABS(0)	ABS(7.8)
7	0	7.8

## 5. SQL 함수 실습

### (1) 숫자형 함수 – CEIL, FLOOR

```
SELECT CEIL(7.6), FLOOR(7.6)  
FROM DUAL;
```

CEIL(7.6)	FLOOR(7.6)
8	7

## 5. SQL 함수 실습

## (1) 숫자형 함수 – EXP, LN, LOG

```
SELECT EXP(5), LN(148.413159102576603421115580040552279624), LOG(10, 10000)
FROM DUAL;
```

[illegible]

## 5. SQL 함수 실습

## (1) 숫자형 함수 – EXP, LN, LOG

```
SELECT EXP(LN(5))
FROM DUAL;
```

[illegible]



## 5. SQL 함수 실습

### (1) 숫자형 함수 – MOD, SIGN

```
SELECT MOD(17, 3), SIGN(-19), SIGN(0)  
FROM DUAL;
```

MOD(17,3)	SIGN(-19)	SIGN(0)
2	-1	0

## 5. SQL 함수 실습

### (1) 숫자형 함수 – POWER, SQRT

```
SELECT POWER(2,3), SQRT(3)
FROM DUAL;
```

POWER(2,3)	SQRT(3)
8	1.73205080756887729352744634150587236694

## 5. SQL 함수 실습

### (1) 숫자형 함수 – ROUND, TRUNC

```
SELECT ROUND(3.545, 2), ROUND(3.545, 1), TRUNC(3.545, 2), TRUNC(3.545, 1)
FROM DUAL;
```

ROUND(3,545,2)	ROUND(3,545,1)	TRUNC(3,545,2)	TRUNC(3,545,1)
3.55	3.5	3.54	3.5

## 5. SQL 함수 실습

### (2) 문자형 함수 – CONCAT

```
SELECT CONCAT('A', 'B'), 'A' || 'B' || 'C'  
FROM DUAL;
```

CONCAT('A','B')	'A'    'B'    'C'
AB	ABC

## 5. SQL 함수 실습

### (2) 문자형 함수 – INITCAP, UPPER, LOWER

```
SELECT INITCAP('abc'), UPPER('abc'), LOWER('AL|bC'), INITCAP('홍gildong')  
FROM DUAL;
```

INITCAP('ABC')	UPPER('ABC')	LOWER('AL BC')	INITCAP('홍GILDONG')
Abc	ABC	aL bc	홍Gildong

## 5. SQL 함수 실습

### (2) 문자형 함수 – INITCAP, UPPER, LOWER

```
SELECT *  
FROM employees  
WHERE first_name = 'steven';
```

EMPLO...	FIRST_N...	LAST_N...	EMAIL	PHONE...	HIRE_DA...	JOB_ID	SALARY	COMMI...	MANAG...	DEPART...
----------	------------	-----------	-------	----------	------------	--------	--------	----------	----------	-----------

```
SELECT *  
FROM employees  
WHERE UPPER(first_name) = 'STEVEN';
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	2003-06-17 00:00:00	AD_PRES	24000	(null)	(null)	90
128	Steven	Markle	SMARKLE	650.124.1434	2008-03-08 00:00:00	ST_CLERK	2200	(null)	120	50

## 5. SQL 함수 실습

### (2) 문자형 함수 – LPAD, RPAD

```
SELECT LPAD( 'SQL', 5, '*' ), RPAD('SQL', 5, '*')  
FROM DUAL;
```

⚙ LPAD('SQL',5,'*')	⚙ RPAD('SQL',5,'*')
***SQL	SQL***

## 5. SQL 함수 실습

### (2) 문자형 함수 – LPAD, RPAD

```
SELECT employee_id,  
       phone_number,  
       LPAD(phone_number, 20, ' ') phone_number2  
FROM employees  
ORDER BY 1;
```

EMPLOY...	PHONE_NUMBER	PHONE_NUMBER2
100	515.123.4567	515.123.4567
101	515.123.4568	515.123.4568
102	515.123.4569	515.123.4569
103	590.423.4567	590.423.4567
104	590.423.4568	590.423.4568
105	590.423.4569	590.423.4569
106	590.423.4560	590.423.4560
107	590.423.5567	590.423.5567
108	515.124.4569	515.124.4569
109	515.124.4169	515.124.4169
110	515.124.4269	515.124.4269
111	515.124.4369	515.124.4369
112	515.124.4469	515.124.4469
113	515.124.4567	515.124.4567
114	515.127.4561	515.127.4561
115	515.127.4562	515.127.4562
116	515.127.4563	515.127.4563
117	515.127.4564	515.127.4564
118	515.127.4565	515.127.4565
119	515.127.4566	515.127.4566
120	650.123.1234	650.123.1234
121	650.123.2234	650.123.2234
122	650.123.3234	650.123.3234
123	650.123.4234	650.123.4234



## 5. SQL 함수 실습

### (2) 문자형 함수 – LTRIM, RTRIM

```
SELECT LTRIM('**SQL**', '*'), RTRIM('**SQL**', '*')  
FROM DUAL;
```

⚙ LTRIM('**SQL**', '*')	⚙ RTRIM('**SQL**', '*')
SQL**	**SQL

## 5. SQL 함수 실습

### (2) 문자형 함수 – SUBSTR

```
SELECT SUBSTR('ABCDEFGH', 1, 2) FIRSTS  
      ,SUBSTR('ABCDEFGH', 0, 2) SECONDS  
      ,SUBSTR('ABCDEFGH', 3, 2) THIRDS  
      ,SUBSTR('ABCDEFGH', 3 )  FOURTHS  
      ,SUBSTR('ABCDEFGH', -3)  FIFTHS  
      ,SUBSTR('ABCDEFGH', -3, 2) SIXTHS  
FROM DUAL;
```

⚡ FIRSTS	⚡ SECONDS	⚡ THIRDS	⚡ FOURTHS	⚡ FIFTHS	⚡ SIXTHS
AB	AB	CD	CDEFG	EFG	EF

## 5. SQL 함수 실습

### (2) 문자형 함수 – TRIM, ASCII, LENGTH, LENGTHB

```
SELECT TRIM(' AB C D '), ASCII('a'), LENGTH('A B C'), LENGTHB('A B 강')  
FROM DUAL;
```

TRIM('ABCD')	ASCII('A')	LENGTH('ABC')	LENGTHB('AB강')
AB C D	97	5	7

## 5. SQL 함수 실습

### (2) 문자형 함수 – REPLACE

```
SELECT REPLACE('산은 산이요 물은 물이다', '산', '언덕')  
FROM DUAL;
```

```
SQL> REPLACE('산은산이요물은물이다','산','언덕')  
언덕은 언덕이요 물은 물이다
```

## 5. SQL 함수 실습

### (2) 문자형 함수 – REPLACE

```
SELECT TRIM(' AB C D '), REPLACE(' AB C D ', ' ', '')  
FROM DUAL;
```

TRIM('ABCD')	REPLACE('ABCD',' ','')
AB C D	ABCD

## 5. SQL 함수 실습

### (2) 문자형 함수 – INSTR

```
SELECT INSTR('ABCABCABC', 'C')  
       ,INSTR('ABCABCABC', 'c')  
       ,INSTR('ABCABCABC', 'C', 2)  
       ,INSTR('ABCABCABC', 'C', 2, 2)  
FROM DUAL;
```

INSTR('ABCABCABC', 'C')	INSTR('ABCABCABC', 'c')	INSTR('ABCABCABC', 'C', 2)	INSTR('ABCABCABC', 'C', 2, 2)
3	0	3	6

## 5. SQL 함수 실습

### (3) 날짜형 함수 – SYSDATE

```
SELECT SYSDATE  
FROM DUAL;
```

SYSDATE
2020-01-12 12:47:36

## 5. SQL 함수 실습

### (3) 날짜형 함수 – ADD\_MONTHS

```
SELECT ADD_MONTHS(SYSDATE, 1),  
       ADD_MONTHS(SYSDATE, -1),  
       ADD_MONTHS(SYSDATE, 0)  
FROM DUAL;
```

ADD_MONTHS(SYSDATE,1)	ADD_MONTHS(SYSDATE,-1)	ADD_MONTHS(SYSDATE,0)
2020-02-12 12:49:01	2019-12-12 12:49:01	2020-01-12 12:49:01



## 5. SQL 함수 실습

### (3) 날짜형 함수 – MONTHS\_BETWEEN

```
SELECT SYSDATE + 31  
      ,SYSDATE - 31  
      ,MONTHS_BETWEEN(SYSDATE + 31, SYSDATE )  
      ,MONTHS_BETWEEN(SYSDATE - 31, SYSDATE )  
FROM DUAL;
```

SYSDATE+31	SYSDATE-31	MONTHS_BETWEEN(SYSDATE+31,SYSDATE)	MONTHS_BETWEEN(SYSDATE-31,SYSDATE)
2020-02-12 12:56:26	2019-12-12 12:56:26	1	-1

## 5. SQL 함수 실습

### (3) 날짜형 함수 – LAST\_DAY, NEXT\_DAY

```
SELECT LAST_DAY(SYSDATE)
       , NEXT_DAY(SYSDATE, '금')
FROM DUAL;
```

⚡ LAST_DAY(SYSDATE)	⚡ NEXT_DAY(SYSDATE, '금')
2020-01-31 13:00:11	2020-01-17 13:00:11

## 5. SQL 함수 실습

### (3) 날짜형 함수 – ROUND

SELECT SYSDATE

    ,ROUND(SYSDATE, 'YYYY') YEARS

    ,ROUND(SYSDATE, 'MM') MONTHS

    ,ROUND(SYSDATE, 'DD') DAYS

    ,ROUND(SYSDATE, 'HH24') HOURS24

    ,ROUND(SYSDATE, 'MI') MINUTES

    ,ROUND(SYSDATE) DEFAULTS

FROM DUAL ;

	SYSDATE
1	2020-01-12 13:17:35

YEARS	MONTHS
2020-01-01 00:00:00	2020-01-01 00:00:00

DAYS	HOURS24
2020-01-13 00:00:00	2020-01-12 13:00:00

MINUTES	DEFAULTS
2020-01-12 13:18:00	2020-01-13 00:00:00

## 5. SQL 함수 실습

### (3) 날짜형 함수 – TRUNC

SELECT SYSDATE

,TRUNC(SYSDATE, 'YYYY') YEARS

,TRUNC(SYSDATE, 'MM') MONTHS

,TRUNC(SYSDATE, 'DD') DAYS

,TRUNC(SYSDATE, 'HH24') HOURS24

,TRUNC(SYSDATE, 'MI') MINUTES

,TRUNC(SYSDATE) DEFAULTS

FROM DUAL ;

SYSDATE
2020-01-12 13:24:52

YEARS	MONTHS
2020-01-01 00:00:00	2020-01-01 00:00:00

DAYS	HOURS24
2020-01-12 00:00:00	2020-01-12 13:00:00

MINUTES	DEFAULTS
2020-01-12 13:24:00	2020-01-12 00:00:00

## 5. SQL 함수 실습

### (4) 날짜 연산자

- 날짜에 개월 수를 더하거나 빼기

→ **ADD\_MONTHS**

- 날짜에 일(day)을 더하거나 빼기

→ **+, -**

```
SELECT SYSDATE + 1 nextday  
       ,SYSDATE - 1 previousday  
FROM DUAL;
```

SYSDATE
2020-01-12 13:24:52

NEXTDAY	PREVIOUSDAY
2020-01-13 13:24:52	2020-01-11 13:24:52

# 학습정리

- 자주 사용되는 연산 로직을 정의해 재사용하는 데이터베이스 객체를 함수라 한다.
- 오라클에서는 이런 함수들을 제공하고 있는데 이들을 SQL 함수 혹은 빌트인 함수라고 한다.
- 오라클의 빌트인 함수 중 가장 기본적인 함수는 크게 숫자형, 문자형, 날짜형 함수로 구분할 수 있다.
- 기본 함수들은 매개변수를 입력 받아 연산을 수행한 후 단일 값을 반환한다.

## Quiz

1. INITCAP, UPPER, LOWER는 영문자를 대소문자로 변환하는 함수입니다. 다음 문장처럼 매개변수로 한글이 입력되면 그 결과는 어떻게 될까요?

```
SELECT UPPER('홍길동')  
FROM DUAL;
```

## Quiz

2. 다음 문자열은 보헤미안 랩소디 가사 첫 부분입니다. 이 중에서 'fantasy?' 만 반환하도록 SUBSTR 함수를 작성해 보세요.

**'Is this the real life? Is this just fantasy?'**



# Quiz

3. 현재 일자 기준 익월 1일을 반환하는 `select` 문을 작성해 보세요.