



# SQL

✓ 원리를 알면 IT가 맛있다

## SQL for Beginners

chapter 04.

---

# 단일 함수

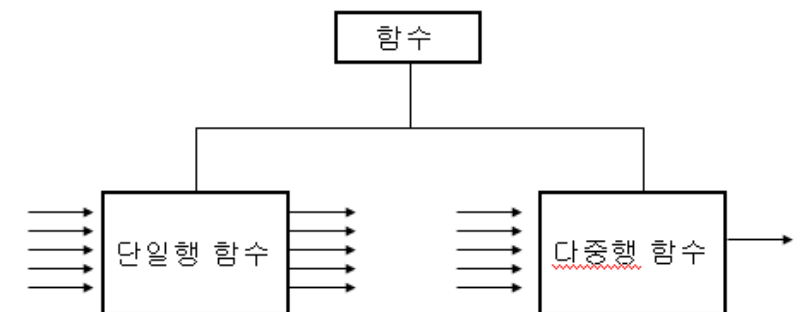
- 함수종류
- 단일행 함수
- 문자관련 함수
- 숫자관련 함수
- 날짜관련 함수
- 일반 함수 ( CASE, DECODE )

- 제공되는 함수들은 기본적인 query문을 더욱 강력하게 해주고 데이터값을 조작하는데 사용할 수 있다.

### ■ SQL 함수의 특징 및 이점

- 데이터에 계산을 수행할 수 있다.
- 개별적인 데이터 항목을 수정할 수 있다.
- 행의 그룹에 대해 결과를 조작할 수 있다.
- 출력을 위한 날짜와 숫자 형식을 조작할 수 있다.
- 열의 자료형을 변환할 수 있다.

### ■ 함수의 종류



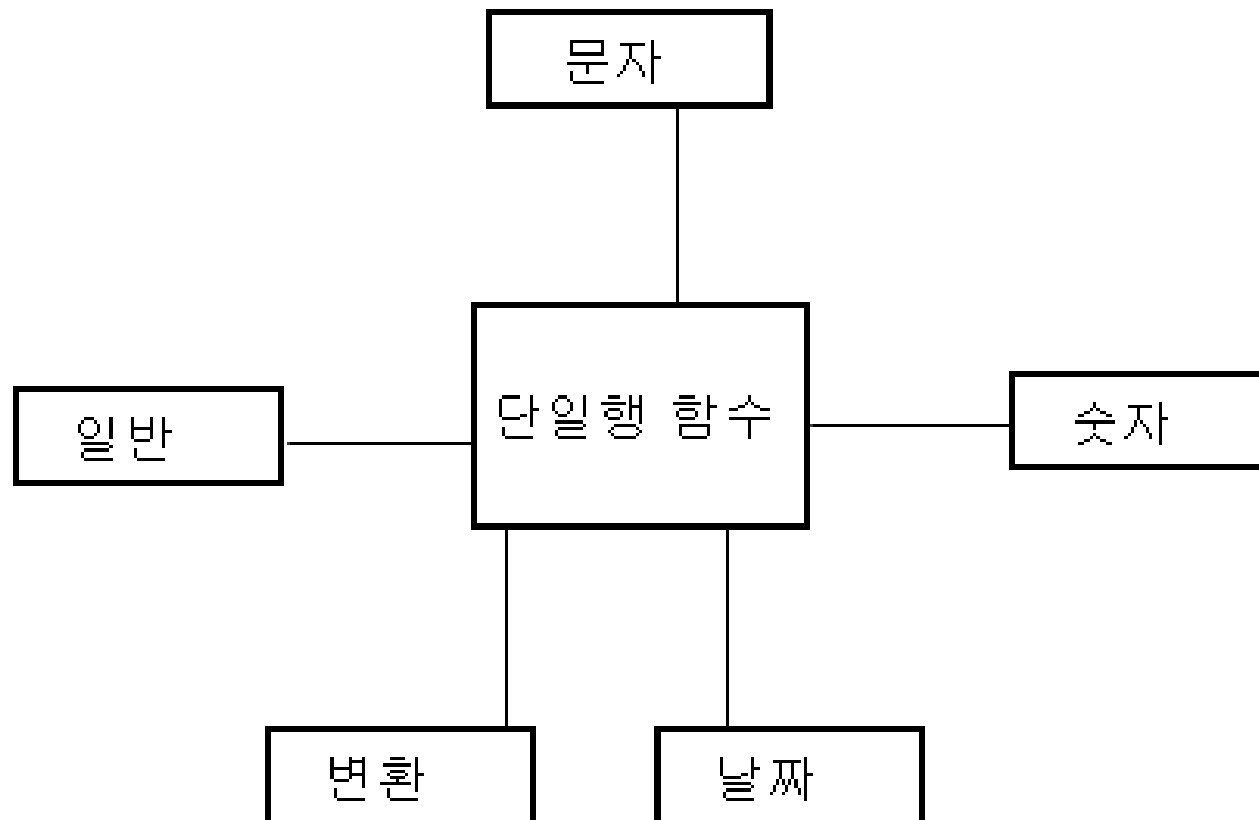
#### ■ 단일 행 함수

- : 모든 행에 대해서 각각 적용되어 행의 개수와 동일한 개수의 결과를 리턴.
- : SELECT, WHERE, ORDER BY절에 사용 가능하다.

#### ■ 다중 행 함수 ( 그룹 함수 )

- : 검색되는 모든 행에 대해서 한번만 적용되고 한 건의 결과만을 리턴.

- 단일 행 함수의 종류



## □ 1) 단일행 함수 - 문자함수

SQL

### ■ INITCAP 함수

- 각 단어의 첫 문자를 대문자로 , 나머지 문자는 소문자로 변경.

Syntax	INITCAP(column   expression)
사 용 예	INITCAP('ORACLE SERVER') -> Oracle Server

문제> DEPT 테이블에서 첫글자만 대문자로 변경하여 모두 출력?

```
SQL> SELECT deptno, INITCAP(dname), INITCAP(loc)
```

```
2 FROM dept;
```

```
DEPTNO INITCAP(DNAME) INITCAP(LOC)
```

```
-----
```

```
10 Accounting      New York
```

```
20 Research        Dallas
```

```
30 Sales           Chicago
```

```
40 Operations      Boston
```

### ■ LOWER 함수

- 대소문자가 혼합되어 있거나 대문자인 문자열을 소문자로 변경.

Syntax	LOWER(column   expression)
사 용 예	LOWER('MANAGER') -> manager

문제> EMP 테이블에서 scott의 정보를 사원번호, 성명, 담당업무(소문자로), 부서번호를 출력 ?

```
SQL> SELECT empno, ename, LOWER(job), deptno
2 FROM emp
3 WHERE LOWER(ename) = 'scott';
```

EMPNO	ENAME	LOWER(JOB)	DEPTNO
7788	SCOTT	analyst	20

### ■ UPPER 함수

- 대문자가 혼합되어 있거나 소문자인 문자열을 대문자로 변경.

Syntax	UPPER(column   expression)
사 용 예	UPPER('manager') -> MANAGER

문제> EMP 테이블에서 scott의 정보를 사원번호, 성명, 담당업무, 부서번호를 출력 ?

```
SQL> SELECT empno, ename, job, deptno
```

```
2 FROM emp
```

```
3 WHERE ename = UPPER('scott');
```

EMPNO	ENAME	JOB	DEPTNO
7788	SCOTT	ANALYST	20



### ■ CONCAT 함수

- 두 개의 문자열을 합성합니다. CONCAT는 두 개의 매개변수만 사용 가능.
- CONCAT 함수는 Concatenation의 약자로 두 문자를 결합하는 역할.
- || 연산자와 같은 역할

Syntax	CONCAT(column1   expression1, column2   expression2)
사 용 예	INITCAP('ORACLE', 'SERVER') -> ORACLESERVER

문제> EMP 테이블에서 사원번호가 7369인 사원의 사원번호, 성명, 담당업무, Job를 출력 ?

```
SQL> SELECT EMPNO, ENAME, JOB, CONCAT(ENAME, JOB)
2  FROM EMP
3  WHERE EMPNO = 7369;
```

EMPNO	ENAME	JOB	CONCAT(ENAME, JOB)
7369	SMITH	CLERK	SMITHCLERK

### ■ LPAD 함수

- 왼쪽에 문자열을 끼워 넣는 역할을 한다. n은 반환되는 문자열의 전체 길이를 나타내며 첫 번째 문자열이 n보다 클 경우 첫 번째 문자열을 n개 문자열만큼 RETURN 한다.

Syntax	LPAD(column   expression, n, 'string')
사 용 예	LPAD('MILLER', 10, '*') -> ****MILLER

```
SQL> SELECT ename, LPAD(ename, 15, '*'), sal, LPAD(sal, 10, '*')
2 FROM emp
3 WHERE deptno = 10;
```

ENAME	LPAD(ENAME, 15, '*')	SAL	LPAD(SAL, 10, '*')
KING	*****KING	5000	*****5000
CLARK	*****CLARK	2450	*****2450
MILLER	*****MILLER	1300	*****1300

## □ 1) 단일행 함수 - 문자함수

### ■ RPAD 함수

- LPAD와 반대로 오른쪽에 문자열을 끼워 넣는 역할을 한다.

Syntax	RPAD(column   expression, n, 'string')
사 용 예	RPAD('MILLER', 10, '*') -> MILLER****

```
SQL> SELECT ename, RPAD(ename, 15, '*'), sal, RPAD(sal, 10, '*')  
2 FROM emp  
3 WHERE deptno = 10;
```

ENAME	RPAD(ENAME, 15, '*')	SAL	RPAD(SAL, 10, '*')
KING	KING*****	5000	5000*****
CLARK	CLARK*****	2450	2450*****
MILLER	MILLER*****	1300	1300*****

### ■ SUBSTR 함수

- m 번째 자리부터 길이가 n개인 문자열을 반환 한다. m이 음수일 경우에는 뒤에서 m번째 문자부터 반대 방향으로 n개의 문자를 RETURN 한다

Syntax	SUBSTR(column   expression, m [,n])
사 용 예	SUBSTR('000101-3234232', 8, 1) -> 3

### ■ LENGTH 함수

- 문자열의 길이를 숫자 값으로 RETURN 한다.

Syntax	LENGTH(column   expression )
사 용 예	LENGTH ('000101-3234232') -> 14

문제) EMP 테이블에서 20번 부서 중 이름의 길이 및 급여의 자릿수를 사원번호, 이름, 이름의 자릿수, 급여, 급여의 자릿수를 출력 ?

```
SQL> SELECT empno, ename, LENGTH(ename), sal, LENGTH(sal)
2 FROM emp
3 WHERE deptno = 20;
```

EMPNO	ENAME	LENGTH(ENAME)	SAL	LENGTH(SAL)
7566	JONES	5	2975	4
7902	FORD	4	3000	4
7369	SMITH	5	800	3

## □ 1) 단일행 함수 - 문자함수

SQL

### ■ REPLACE 함수

- 특정 문자열을 치환하는 함수 .

Syntax	REPLACE(column1   expression1, 'string1', 'string2')
사 용 예	REPLACE('JACK and JUE', 'J', 'BL') -> BLACK and BLUE

문제) EMP 테이블에서 JOB에 'A'를 '\$'로 바꾸어 출력 ?

```
SQL> SELECT ename, job, REPLACE(job, 'A', '$')
```

```
2 FROM emp;
```

ENAME	JOB	REPLACE(J
KING	PRESIDENT	PRESIDENT
BLAKE	MANAGER	M\$N\$GER
CLARK	MANAGER	M\$N\$GER

## □ 1) 단일행 함수 - 문자함수

SQL

### ■ INSTR 함수

- 문자열이 포함되어 있는지를 조사하여 문자열의 위치를 RETURN 한다.
- 지정한 문자열이 발견되지 않으면 0이 반환 됩니다.

Syntax	<code>INSTR(column   expression, m[,n])</code>
사 용 예	<code>INSTR('MILLER', 'L', 1, 2) -&gt; 4</code>

문제) EMP 테이블에서 이름 중 'L'자의 위치를 출력 ?

```
SQL> SELECT ename, INSTR(ename, 'L') e_null, INSTR(ename, 'L', 1, 1) e_11,  
2      INSTR(ename, 'L', 1, 2) e_12, INSTR(ename, 'L', 4, 1) e_41,  
3      INSTR(ename, 'L', 4, 2) e_42  
4 FROM emp  
5 ORDER BY ename;
```

ENAME	E_NULL	E_11	E_12	E_41	E_42
ADAMS	0	0	0	0	0
ALLEN	2	2	3	0	0
BLAKE	2	2	0	0	0
CLARK	2	2	0	0	0
FORD	0	0	0	0	0
JAMES	0	0	0	0	0
JONES	0	0	0	0	0
KING	0	0	0	0	0
MARTIN	0	0	0	0	0
MILLER	3	3	4	4	0

## □ 1) 단일행 함수 - 문자함수

SQL

### ■ LTRIM함수

- 문자열의 첫 문자부터 확인해서 지정 문자가 나타나는 동안 해당 문자를 제거.
- 지정문자가 생략되면 공백문자가 기본값이다.

Syntax	LTRIM(column1   expression1, column1   expression1)
사 용 예	LTRIM('MILLER', 'M') -> ILLER

문제) EMP 테이블에서 담당 업무 중 좌측에 'A'를 삭제하고, 급여 중 좌측의 1을 삭제하여 출력하여라.

```
SQL> SELECT ename, job, LTRIM(job, 'A'), sal, LTRIM(sal, 1)
```

```
2 FROM emp;
```

ENAME	JOB	LTRIM(JOB	SAL	LTRIM(SAL, 1)
FORD	ANALYST	NALYST	3000	3000
SMITH	CLERK	CLERK	800	800
SCOTT	ANALYST	NALYST	3000	3000
ADAMS	CLERK	CLERK	1100	00
MILLER	CLERK	CLERK	1300	300



## □ 1) 단일행 함수 - 문자함수

SQL

### ■ RTRIM함수

- 문자열의 끝 문자부터 확인해서 지정 문자가 나타나는 동안 해당 문자를 제거.
- 지정문자가 생략되면 공백문자가 기본값이다

Syntax	RTRIM(column1   expression1, column2   expression2)
사 용 예	RTRIM('MILLER', 'R') -> MILLE

문제) EMP 테이블에서 10번 부서에 대하여 담당 업무 중 우측에 'T'를 삭제하고 급여 중 우측의 0을 삭제하여 출력?.

```
SQL> SELECT ename, job, RTRIM(job, 'T'), sal, RTRIM(sal, 0)
2 FROM emp
3 WHERE deptno = 10;
```

ENAME	JOB	RTRIM(JOB	SAL	RTRIM(SAL,0)
KING	PRESIDENT	PRESIDEN	5000	5
CLARK	MANAGER	MANAGER	2450	245
MILLER	CLERK	CLERK	1300	13

### ■ TRIM함수

- 문자열에서 머리말, 꼬리말 또는 양쪽에 있는 지정 문자를 제거.  
( leading | trailing | both 가 생략되면 both가 기본 )
- 제거할 문자를 입력하지 않으면 기본적으로 공백이 제거 된다.

-- 0 을 제거 합니다.

```
SQL> SELECT TRIM(0 FROM 0009872348900) "TRIM Example" FROM DUAL;
```

TRIM Example

98723489

```
SQL> SELECT EMPNO, JOB,  
2  TRIM(LEADING 'S' FROM JOB) AS LEADING,  
3  TRIM(TRAILING 'N' FROM JOB) AS TRAILING  
4  FROM EMP  
5  WHERE EMPNO = 7844
```

EMPNO	JOB	LEADING	TRAILING
7844	SALESMAN	ALESMAN	SALESMA

#### • DUAL 테이블 특징

- 사용자 SYS가 소유하며  
모든 사용자가 사용 가능.

- SELECT~FROM 형식을  
맞추기 위한 일종의 dummy테이블.

예> select sysdate from dual;  
select 123\*345 from dual;

## □ 1) 단일행 함수 - 숫자함수

SQL

### ■ 숫자 함수

함 수	사 용 목 적
ABS	절대값을 계산
CEIL	올림값
EXP	주어진 값의 e 의 승수를
FLOOR	내림값
LN	자연로그(e) 값
MOD	나머지를 구함
POWER	거듭제곱
ROUND	숫자를 반올림
SIGN	양수, 음수, 0 인지를 구분
SQRT	제곱근
TRUNC	숫자를 절삭
CHR	ASCII 값에 해당하는 문자를 구함

## □ 1) 단일행 함수 - 숫자함수

### ■ CEIL 함수 (올림값)

- 주어진 숫자보다 크거나 같은 최소 정수 리턴.

Syntax	CEIL(column1   expression1)
사 용 예	CEIL(10.1) -> 11

### ■ FLOOR 함수 (버림값)

- 주어진 숫자보다 작거나 같은 최대 정수 리턴.

Syntax	FLOOR(column1   expression1)
사 용 예	FLOOR(10.7) -> 10

### ■ MOD 함수

- 숫자의 나머지를 구하는 함수.
- n으로 나누어 남은 값을 반환한다. n이 0일 경우 그 자체를 반환.

Syntax	MOD(column1   expression1, n)
사 용 예	MOD(10, 3) -> 1

문제) EMP 테이블에서 급여를 30으로 나눈 나머지를 구하여 출력 ?.

```
SQL> SELECT sal, MOD(sal, 30)
```

```
2 FROM emp
```

```
3 WHERE deptno = 10;
```

```
      SAL MOD(SAL, 30)
```

```
-----
```

```
5000      20
```

```
2450      20
```

```
1300      10
```

## □ 1) 단일행 함수 - 숫자함수

SQL

### ■ ROUND 함수

- 숫자를 소수점 n자리에서 반올림한다. 생략하면 기본값은 0.
- n이 양수이면 소수 자리를, 음수이면 정수 자리를 반올림한다.

Syntax	ROUND(column1   expression1, n)
사 용 예	ROUND(456.789, 2) -> 456.79

문제) 다음의 결과를 분석 ?

```
SQL> SELECT ROUND(4567.678), ROUND(4567.678,0), ROUND(4567.678,2), ROUND(4567.678,-2)
2 FROM dual;
```

```
ROUND(4567.678) ROUND(4567.678,0) ROUND(4567.678,2) ROUND(4567.678,-2)
-----
4568              4568              4567.68              4600
```

### ■ TRUNC 함수

- 숫자를 소수 n자리에서 절삭한다. 생략하면 기본값은 0.
- n이 양수이면 소수 자리를, 음수이면 정수 자리를 절삭.

Syntax	TRUNC(column1   expression1, n)
사 용 예	TRUNC(456.789, 2) -> 456.78

문제) 다음의 결과를 분석 ?

```
SQL> SELECT TRUNC(4567.678),TRUNC(4567.678, 0), TRUNC(4567.678, 2),TRUNC(4567.678, -2)  
2 FROM dual;
```

```
TRUNC(4567.678) TRUNC(4567.678,0) TRUNC(4567.678,2) TRUNC(4567.678,-2)  
-----  
4567          4567          4567.67          4500
```



## □ 1) 단일행 함수 – 날짜함수

- 오라클은 세기, 년, 월, 일, 시, 분, 초를 내부 숫자(7 Byte) 형식으로 날짜를 저장 한다.

-Default Date Type은 RR-MM-DD(변경 가능) 이다.

```
select *
from NLS_SESSION_PARAMETERS;
```

PARAMETER	VALUE
1 NLS_LANGUAGE	KOREAN
2 NLS_TERRITORY	KOREA
3 NLS_CURRENCY	₩
4 NLS_ISO_CURRENCY	KOREA
5 NLS_NUMERIC_CHARACTERS	.,
6 NLS_CALENDAR	GREGORIAN
7 NLS_DATE_FORMAT	RR/MM/DD
8 NLS_DATE_LANGUAGE	KOREAN

### ■ RR 날짜 형식

: 년도를 2자리로 입력 시 적절하게 1900년대 또는 2000년대의 4자리 연도로 변환.

현재 년도	명시된 날짜	RR 형식	YY 형식
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

		명시된 두 자리 연도	
		0-49	50-99
현재 년도의 두 자리값	0-49	리턴되는 날짜는 현재의 세기	리턴되는 날짜는 현재 세기 이 전의 세기
	50-99	리턴 날짜는 현재 세기 이후의 세기	리턴 날짜는 현재 세기



### ■ SYSDATE 함수

: 데이터베이스 서버에 설정되어 있는 날짜를 리턴.

```
SQL> SELECT SYSDATE FROM DUAL;
```

```
SYSDATE
```

```
-----  
09/09/03
```

```
SQL> SELECT SYSTIMESTAMP FROM DUAL;
```

```
SYSTIMESTAMP
```

```
-----  
09/09/03 15:10:00.765000 +09:00
```

### ■ 날짜 연산

연산	결과	설명
날짜 + 숫자	날짜	날짜에 일수를 더한다
날짜 - 숫자	날짜	날짜에서 일수를 뺀다
날짜 - 날짜	숫자(일수)	두 날짜의 차이(일수)를 계산한다
날짜 + 숫자/24	날짜	날짜에 시간을 더한다

문제) EMP 테이블에서 현재까지 근무일 수가 몇주 몇일 인가를 출력. 단 근무일수가 많은 사람 순으로 출력 ?

```
SQL> SELECT ename, hiredate, sysdate, sysdate - hiredate "Total Days",  
2 TRUNC((sysdate - hiredate) / 7, 0) Weeks,  
3 ROUND(MOD((sysdate - hiredate), 7), 0) DAYS  
4 FROM emp  
5 ORDER BY sysdate - hiredate DESC;
```

## □ 1) 단일행 함수 - 날짜함수

SQL

### ■ 날짜 함수

- 날짜 함수는 오라클 날짜에 대해 연산을 한다. MONTHS\_BETWEEN을 제외하고는 DATE형을 RETURN 한다.

날짜 함수	설 명
MONTHS_BETWEEN	두 날짜 사이의 월수를 계산
ADD_MONTHS	월을 날짜에 더합니다.
NEXT_DAY	명시된 날짜로부터 다음 요일에 대한 날짜를 나타냅니다.
LAST_DAY	월의 마지막 날을 계산 합니다.
ROUND	날짜를 반올림 합니다.
TRUNC	날짜를 절삭 합니다.

## □ 1) 단일행 함수 - 날짜함수

SQL

### ■ MONTHS\_BETWEEN 함수

- 날짜와 날짜 사이의 월수를 계산한다.
- 결과는 음수 또는 양수가 될 수 있다.
- 결과의 비정수 부분은 월의 부분을 나타낸다.

Syntax	MONTHS_BETWEEN(date1, date2)
사 용 예	MONTHS_BETWEEN(sysdate, hiredate) -> 212.04794

위 예에서 212 는 월을 나타내고 .04794 는 월의 일부분을 나타냅니다.

문제) EMP 테이블에서 10번 부서 중 현재까지의 근무 월수를 계산하여 출력?

```
SQL> SELECT ename, hiredate, SYSDATE, MONTHS_BETWEEN(SYSDATE, hiredate) m_between,  
2 TRUNC(MONTHS_BETWEEN(SYSDATE, hiredate), 0) t_between  
3 FROM emp  
4 WHERE deptno = 10  
5 ORDER BY MONTHS_BETWEEN(SYSDATE, hiredate) DESC;
```

### ■ ADD\_MONTHS 함수

- 날짜에 월을 더한다.(ADD\_MONTHS(hiredate,10))
- 날짜에 월을 뺀다.(ADD\_MONTHS(hiredate,-10))
- 결과는 날짜형이다. .

Syntax	ADD_MONTHS(date1, n)
사 용 예	ADD_MONTHS(hiredate, 5)

문제) EMP 테이블에서 10번 부서 중 입사 일자로부터 5개월이 지난 후 날짜를 계산하여 출력 ?

```
SQL> SELECT ename, hiredate, ADD_MONTHS(hiredate, 5) a_month  
2 FROM emp  
3 WHERE deptno = 10  
4 ORDER BY hiredate DESC;
```

### ■ NEXT\_DAY 함수

- 명시된 요일의 돌아오는 날짜를 계산 한다.
- 요일이 아니라 숫자도 기술 가능(SUNDAY:1, MONDAY:2, . . . . .)
- NLS\_LANG이 KOREAN 로 되어 있으면 한글도 사용 가능(일요일, 월요일, 화요일, . . . . .) .

Syntax	NEXT_DAY(date1, 'string'   n
사 용 예	NEXT_DAY(hiredate, 'FRIDAY') NEXT_DAY(hiredate, '금요일')

문제) EMP 테이블에서 사원번호가 7839 중 입사 일자로부터 돌아오는 수요일을 계산하여 출력?

```
SQL> SELECT EMPNO, ENAME, HIREDATE, NEXT_DAY(HIREDATE, '수')  
2 FROM EMP  
3 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	NEXT_DAY
7839	KING	81/11/17	81/11/18

## □ 1) 단일행 함수 - 날짜함수

SQL

### ■ LAST\_DAY 함수

- 월의 마지막 날짜를 계산
- 윤년, 평년은 자동 계산

Syntax	LAST_DAY(date1)
사 용 예	LAST_DAY(hiredate)

```
SQL> SELECT EMPNO, ENAME, HIREDATE, LAST_DAY(HIREDATE)
2 FROM EMP
3 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	LAST_DAY
7839	KING	81/11/17	81/11/30

## □ 1) 단일행 함수 - 날짜함수

SQL

### ■ ROUND 함수

- 날짜를 가장 가까운 년도 또는 월로 반올림할 수 있다.

```
SQL> SELECT EMPNO, ENAME, HIREDATE,  
2 ROUND(HIREDATE, 'MONTH'),  
3 ROUND(HIREDATE, 'YEAR')  
4 FROM EMP  
5 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	ROUND(HI	ROUND(HI
7839	KING	81/11/17	81/12/01	82/01/01

### ■ TRUNC 함수

- 날짜를 가장 가까운 년도 또는 월로 절삭할 수 있다.

```
SQL> SELECT EMPNO, ENAME, HIREDATE,  
2 TRUNC(HIREDATE, 'MONTH'),  
3 TRUNC(HIREDATE, 'YEAR')  
4 FROM EMP  
5 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	TRUNC(HI	TRUNC(HI
7839	KING	81/11/17	81/11/01	81/01/01



## □ 1) 단일행 함수 - 변환함수

### ■ 변환 함수

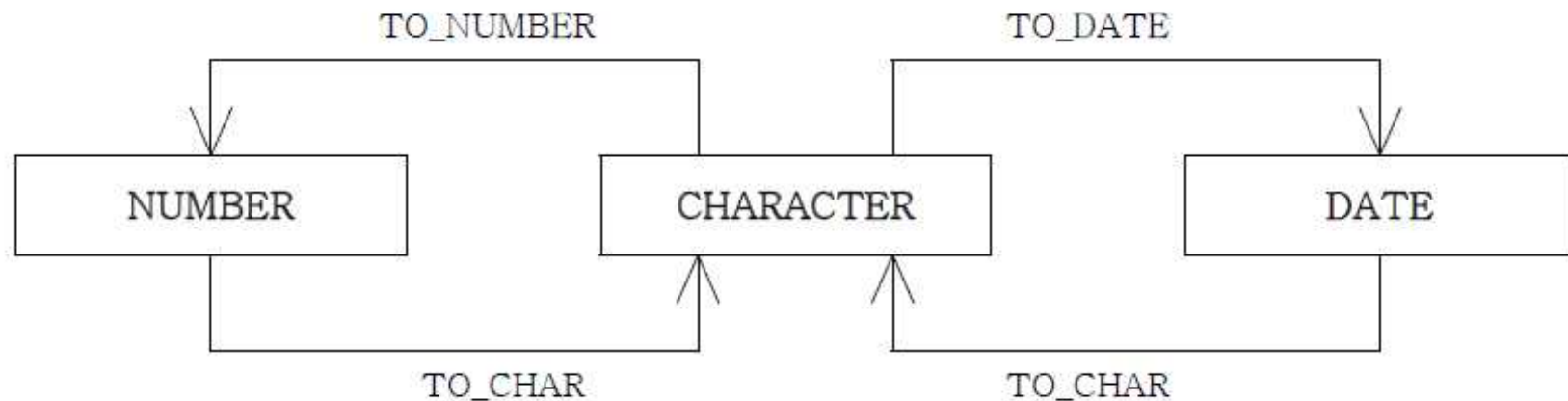
: 데이터 타입을 다른 데이터 타입으로 변환하는 함수.

### ■ 데이터 타입의 변환 방법

#### 1. 묵시적 형 변환 ( 권장 안함 )

변환 전	변환 후
VARCHAR2 또는 CHAR	NUMBER
VARCHAR2 또는 CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

#### 2. 명시적 형 변환 ( 함수 이용 )



### ■ 자동 형 변환된 경우

```
SQL> SELECT EMPNO , ENAME , HIREDATE
2 FROM EMP
3 WHERE HIREDATE = '82/01/23';
```

EMPNO	ENAME	HIREDATE
7934	MILLER	82/01/23

```
SQL> SELECT EMPNO , ENAME , HIREDATE
2 FROM EMP
3 WHERE HIREDATE = TO_DATE( '82/01/23' );
```

EMPNO	ENAME	HIREDATE
7934	MILLER	82/01/23

```
SQL> SELECT EMPNO , ENAME
2 FROM EMP
3 WHERE EMPNO = '7900';
```

EMPNO	ENAME
7900	JAMES

```
SQL> SELECT EMPNO , ENAME
2 FROM EMP
3 WHERE EMPNO = TO_NUMBER('7900');
```

EMPNO	ENAME
7900	JAMES

### ■ 명시적 형 변환 함수

함 수	사 용 목 적
TO_CHAR	숫자나 문자값을 지정한 형식의 VARCHAR2 문자열로 변환 합니다.
TO_NUMBER	숫자를 포함하는 문자열을 숫자로 변환 합니다.
TO_DATE	날짜를 나타내는 문자열을 명시된 날짜로 변환 합니다.

### ■ TO\_CHAR 함수

- 숫자, 날짜, 문자열을 지정한 형식의 VARCHAR2 문자열로 변환하는 함수 .

Syntax	TO_CHAR(date, 'fmt' )
사 용 예	TO_CHAR(hiredate, 'YY/MM/DD') -> 81/11/17

## □ 1) 단일행 함수 - 변환함수

SQL

### ■ 날짜 형식 모델

형식	설명
YYYY	년도를 숫자로 표시
YEAR	년도를 영문으로 표시
MM	월을 숫자로 표시
MONTH	월을 영문으로 표시
MON	월을 영문 3자리로 축약해서 표시
DY	요일을 영문 3자리로 축약해서 표시, 한글 환경에서는 한글 1자로 표시
DAY	요일을 영문으로 표시, 한글 환경에서는 한글 3자로 표시
DD	일을 숫자로 표시

### ■ 시간 형식 모델

형식	설명
AM 또는 PM	오전 또는 오후를 표시
A.M. 또는 P.M.	오전 또는 오후를 표시
HH 또는 HH12 또는 HH24	시간 또는 1~12시 또는 1~24시
MI	분(0~59)
SS	초(0~59)
SSSSS	자정 이후 초(0~86399)

## □ 1) 단일행 함수 - 변환함수

SQL

### ■ 숫자 형식 모델

형식	설명
9	숫자로 표시
0	숫자의 앞부분을 0으로 표시
\$	달러 표시
L	지역 화폐 단위 표시
.	소수점 표시
,	1000 단위 구분자 표시

문제) EMP 테이블에서 10번 부서 중 입사 일자를 '1998년 1월 1일'의 형태로 출력?

```
SQL> SELECT ename,hiredate,T0_CHAR(hiredate,'YYYY"년" MM"월" DD"일"') 입사일  
2 FROM emp  
3 WHERE deptno = 10;
```

ENAME	HIREDATE	입사일
CLARK	1981-06-09 12:00:00	1981년 06월 09일
KING	1981-11-17 12:00:00	1981년 11월 17일
MILLER	1982-01-23 12:00:00	1982년 01월 23일

문제) EMP 테이블에서 부서 20중 급여 앞에 \$를 삽입하고 3자리마다 ,를 출력?

```
SQL> SELECT empno, ename, job, sal, T0_CHAR(sal, '$999,999')  
2 FROM emp  
3 WHERE deptno = 20  
4 ORDER BY sal DESC;
```

### ■ TO\_NUMBER 함수

- 숫자를 포함하는 문자열을 숫자로 변환하는 함수 .

Syntax	TO_NUMBER(char )
사 용 예	TO_NUMBER('1234') -> 1234

### ■ TO\_DATE 함수

- 날짜를 나타내는 문자열을 명시된 날짜로 변환 .

Syntax	TO_DATE(char [, 'fmt' ] )
사 용 예	TO_DATE('19990220181030', 'YYYYMMDDHH24MISS') → 1999/02/20 18:10:30

```
SQL> select TO_DATE('19990220181030', 'YYYYMMDDHH24MISS') from dual;
```

```
TO_DATE('1999022018
```

```
-----  
1999-02-20 06:10:30
```

## ■ CASE 함수 ( ANSI )

```
CASE expr WHEN comparison_expr1 THEN return_expr1
      [WHEN comparison_expr2 THEN return_expr2
      WHEN comparison_exprn THEN return_exprn
      ELSE else_expr]
END
```

```
SQL> SELECT EMPNO, ENAME, SAL, JOB,
2      CASE JOB WHEN 'ANALYST' THEN SAL*1.1
3              WHEN 'CLERK' THEN SAL*1.2
4              WHEN 'MANAGER' THEN SAL*1.3
5              WHEN 'PRESIDENT' THEN SAL*1.4
6              WHEN 'SALESMAN' THEN SAL*1.5
7              ELSE SAL
8      END "급여"
9      FROM EMP;
```

```
SQL> SELECT EMPNO, ENAME,
2      CASE WHEN SAL >= 0 AND SAL <= 1000 THEN 'E'
3            WHEN SAL > 1000 AND SAL <= 2000 THEN 'D'
4            WHEN SAL > 2000 AND SAL <= 3000 THEN 'C'
5            WHEN SAL > 3000 AND SAL <= 4000 THEN 'B'
6            WHEN SAL > 4000 AND SAL <= 5000 THEN 'A'
7            END "등급"
8      FROM EMP;
```

EMPNO	ENAME	SAL	JOB	급여
7369	SMITH	800	CLERK	960
7499	ALLEN	1600	SALESMAN	2400
7521	WARD	1250	SALESMAN	1875
7566	JONES	2975	MANAGER	3867.5
7654	MARTIN	1250	SALESMAN	1875



### ■ DECODE 함수 ( 오라클 함수 )

```
DECODE(col/expression, search1, result1  
      [, search2, result2, ... .]  
      [, default])
```

```
SQL> SELECT EMPNO, ENAME, SAL, JOB,  
2      DECODE(JOB, 'ANALYST' , SAL*1.1,  
3                'CLERK'   , SAL*1.2,  
4                'MANAGER' , SAL*1.3,  
5                'PRESIDENT', SAL*1.4,  
6                'SALESMAN' , SAL*1.5, SAL) "급여"  
7 FROM EMP;
```

EMPNO	ENAME	SAL	JOB	급여
7369	SMITH	800	CLERK	960
7499	ALLEN	1600	SALESMAN	2400
7521	WARD	1250	SALESMAN	1875
7566	JONES	2975	MANAGER	3867.5
7654	MARTIN	1250	SALESMAN	1875

### ■ 실습 예제

1. 사원테이블에서 입사일이 12월인 사원의 사번, 사원명, 입사일 검색하시오
2. 다음과 같은 결과를 검색할 수 있는 SQL 문장을 작성하시오.

EMPNO	ENAME	급여
7369	SMITH	*****800
7499	ALLEN	*****1600
...		
7934	MILLER	*****1300

- 3 다음과 같은 결과를 검색할 수 있는 SQL 문장을 작성하시오.

EMPNO	ENAME	입사일
7369	SMITH	1980-12-17
7499	ALLEN	1981-02-20
...		
7934	MILLER	1982-01-23

4. 사원 테이블에서 영업사원과 비영업사원을 구별하여 다음과 같이 출력하시오.

EMPNO	ENAME	COMM	사원종류
7369	SMITH	(null)	일반사원
7499	ALLEN	300	영업사원
7521	WARD	500	영업사원
7566	JONES	(null)	일반사원
7654	MARTIN	1400	영업사원
7698	BLAKE	(null)	일반사원
7782	CLARK	(null)	일반사원
7788	SCOTT	(null)	일반사원
7839	KING	(null)	일반사원
7844	TURNER	0	영업사원
7876	ADAMS	(null)	일반사원
7900	JAMES	(null)	일반사원
7902	FORD	(null)	일반사원
7934	MILLER	(null)	일반사원

5. 부서 테이블에서 부서 위치를 미국의 동부, 중부, 서부로 구분하여 다음과 같이 출력하시오.

LOC	AREA
NEW YORK	EAST
DALLAS	CENTER
CHICAGO	CENTER
BOSTON	EAST

6. 사원 테이블에서 급여가 2000 이상이면 보너스를 1000으로, 1000 이상이면 500으로 , 1000 미만이면 0으로 계산하여 출력하시오.

ENAME	SAL	BONUS
SMITH	800	0
ALLEN	1600	500
WARD	1250	500
JONES	2975	1000
MARTIN	1250	500
BLAKE	2850	1000
CLARK	2450	1000
SCOTT	3000	1000
KING	5000	1000
TURNER	1500	500
ADAMS	1100	500
JAMES	950	0
FORD	3000	1000
MILLER	1300	500

7. 사원 테이블에서 급여에 따라 사번, 이름, 급여, 등급을 검색하는 SQL 문장을 작성하시오  
( CASE 함수와 DECODE 각각 구현. DECODE 사용시 SIGN함수 이용한다.)

급여	등급
0~1000	E
1001~2000	D
2001~3000	C
3001~4000	B
4001~5000	A

## ■ SIGN 함수

-주어진 숫자가 양수인지 음수인지 또는 0인지를 구하는 함수

```
SQL> SELECT SIGN(100) , SIGN(0) , SIGN(-3) FROM dual;
```

SIGN(100)	SIGN(0)	SIGN(-3)
-----	-----	-----
1	0	-1



**Thank you**

---