

ORACLE®

## 05.SQL 주요 함수



# 목차

1. DUAL 테이블
2. 숫자 함수
3. 문자 처리 함수
4. 날짜 함수
5. 형 변환 함수
6. DECODE 함수
7. CASE 함수
8. NVL 함수

# 1. DUAL 테이블

## ❖ DUAL

- DUMMY라는 단 하나의 컬럼으로 구성되어 있다.
- 이 컬럼의 최대 길이는 1이다.

형식	DESC DUAL;
이름      널? 유형 ----- DUMMY     VARCHAR2 (1)	

형식	SELECT * FROM DUAL;		
	<table border="1"><thead><tr><th>DUMMY</th></tr></thead><tbody><tr><td>1 X</td></tr></tbody></table>	DUMMY	1 X
DUMMY			
1 X			

- ❖ DUAL 테이블은 DUMMY라는 단 하나의 컬럼에 X라는 단 하나의 로우만을 저장하고 있으나 이 값은 아무런 의미가 없다.
- ❖ 쿼리문의 수행 결과가 하나의 로우로 출력되도록 하기 위해서 단 하나의 로우를 구성하고 있을 뿐이다.

## 2. 숫자 함수

구 분	설 명
ABS	절대값을 구한다.
COS	COSINE 값을 반환한다.
EXP	e(2.71828183...)의 n승을 반환한다.
FLOOR	소수점 아래를 잘라낸다.(버림)
LOG	LOG값을 반환한다.
POWER	POWER(m, n) m의 n승을 반환한다.
SIGN	SIGN (n) n<0이면 -1, n=0이면 0, n>0이면 1을 반환한다.
SIN	SINE값을 반환한다.
TAN	TANGENT값을 반환한다.
ROUND	특정 자릿수에서 반올림한다.
TRUNC	특정 자릿수에서 잘라낸다. (버림)
MOD	입력 받은 수를 나눈 나머지 값을 반환한다.

## 2.1 ABS, FLOOR

### ❖ ABS

- 절대값을 구하는 함수
- 방향은 없고 크기만 있는 함수로서 음수 입력 시 양수로 출력

### ❖ -10에 대한 절대값을 구하는 예제

예

```
SELECT -10, ABS(-10) FROM dual;
```

	-10	ABS(-10)
1	-10	10

### ❖ FLOOR

- 소수점 아래를 버리는 함수
- ❖ 34.5678에 대한 FLOOR함수를 적용한 예제

예

```
SELECT 34.5678, FLOOR(34.5678) FROM dual;
```

	34.5678	FLOOR(34.5678)
1	34.5678	34

## 2.2 ROUND(1)

- ❖ ROUND
  - 소수점을 반올림 하는 함수
- ❖ 34.5678에 대한 ROUND함수를 적용한 예제

예

**SELECT 34.5678, ROUND(34.5678) FROM dual;**

	34.5678	ROUND(34.5678)
1	34.5678	35

- ❖ 특정 자릿수에서 반올림 할 수 있도록 자릿수를 지정할 수 있다.

예

**SELECT 34.5678, ROUND(34.5678, 2) FROM dual;**

	34.5678	ROUND(34.5678,2)
1	34.5678	34.57

## 2.2 ROUND(2)

### ❖ ROUND

- 자릿수에 음수를 지정할 수 있는데, 이때 소수점이 아니라 일단위, 십단위, 백 단위 순으로 거슬로 올라가며 반올림이 이루어진다.

예

```
SELECT 34.5678, ROUND(34.5678, -1) FROM dual;
```

	34.5678	ROUND(34.5678,-1)
1	34.5678	30

## 2.3 TRUNC

### ◆ TRUNC

- 지정한 자리 수 이하를 버린 결과를 구해주는 함수
- 함수의 두번째 인자 값이 2이면 소수점 이하 세번째 자리에서 버림 연산을 하여 소수점 두 번째 자리까지 표시한다.
- 두 번째 인자 값이 0인 경우에는 소수점자리에서 버림 연산을 하고 -1인 경우는 일의 자리에서 버림 연산을 하며 두 번째 인자가 주어지지 않은 경우 0으로 간주되어 소수점자리에서 버림 연산을 수행된다.

예

```
SELECT TRUNC(34.5678,2), TRUNC(34.5678, -1),  
TRUNC(34.5678) FROM dual;
```

	TRUNC(34.5678,2)	TRUNC(34.5678,-1)	TRUNC(34.5678)
1	34.56	30	34

## 2.4 MOD

### ❖ MOD

- 나누기 연산을 한 뒤에 몫이 아닌 나머지를 결과로 되돌려 받는 함수.

예

**SELECT MOD(27,2), MOD(27,5), MOD(27,7) FROM dual;**

	MOD(27,2)	MOD(27,5)	MOD(27,7)
1	1	2	6

### 3. 문자 처리 함수

구 분	설 명
LOWER	소문자로 변환한다.
UPPER	대문자로 변환한다.
INITCAP	첫 글자만 대문자로 나머지 글자는 소문자로 변환한다.
CONCAT	문자의 값을 연결한다.
SUBSTR	문자를 잘라 추출한다. (한글 1Byte)
SUBSTRB	문자를 잘라 추출한다. (한글 2Byte)
LENGTH	문자의 길이를 반환한다.(한글 1Byte)
LENGTHB	문자의 길이를 반환한다.(한글 2Byte)
LPAD, RPAD	입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환한다.
TRIM	잘라내고 남은 문자를 표시한다.
CONVERT	char set을 변환한다.
CHR	ASCII 코드 값으로 변환한다.
ASCII	ASCII 코드 값을 문자로 변환한다.
REPLACE	문자열에서 특정 문자를 변경한다.

# 3.1 UPPER, LOWER

## ❖ UPPER

- 입력한 문자열을 모두 대문자로 변환하는 함수.

예

```
SELECT 'welcome to oracle', UPPER('welcome to oracle')
FROM DUAL;
```

'WELCOMETOORACLE'	UPPER('WELCOMETOORACLE')
welcom to oracle	WELCOM TO ORACLE

## ❖ LOWER

- 입력한 문자열을 모두 소문자로 변환하는 함수.

예

```
SELECT 'WELCOME TO ORACLE', LOWER('WELCOME TO
ORACLE') FROM DUAL;
```

'WELCOMETOORACLE'	LOWER('WELCOMETOORACLE')
WELCOME TO ORACLE	welcome to oracle

## 3.2 INITCAP, LENGTH

### ❖ INITCAP

- 입력한 문자열의 이니셜만 대문자로 변경한다.

예

```
SELECT 'welcome to oracle', INITCAP('welcome to oracle')
FROM DUAL;
```

'WELCOMTOORACLE'	INITCAP('WELCOMTOORACLE')
welcom to oracle	Welcom To Oracle

### ❖ LENGTH

- 데이터의 길이를 알려주는 함수

예

```
SELECT LENGTH('오라클'), LENGTH('Oracle') FROM DUAL;
```

LENGTH('오라클')	LENGTH('ORACLE')
1	3 6

## 3.3 SUBSTR(1)

### ❖ SUBSTR

- 대상 문자열이나 컬럼의 자료에서 시작위치부터 선택 개수만큼의 문자를 추출 한다.

**형식    SUBSTR(대상, 시작 위치, 추출할 개수)**

**예      SELECT SUBSTR('Welcome to Oracle', 4, 3) FROM DUAL;**

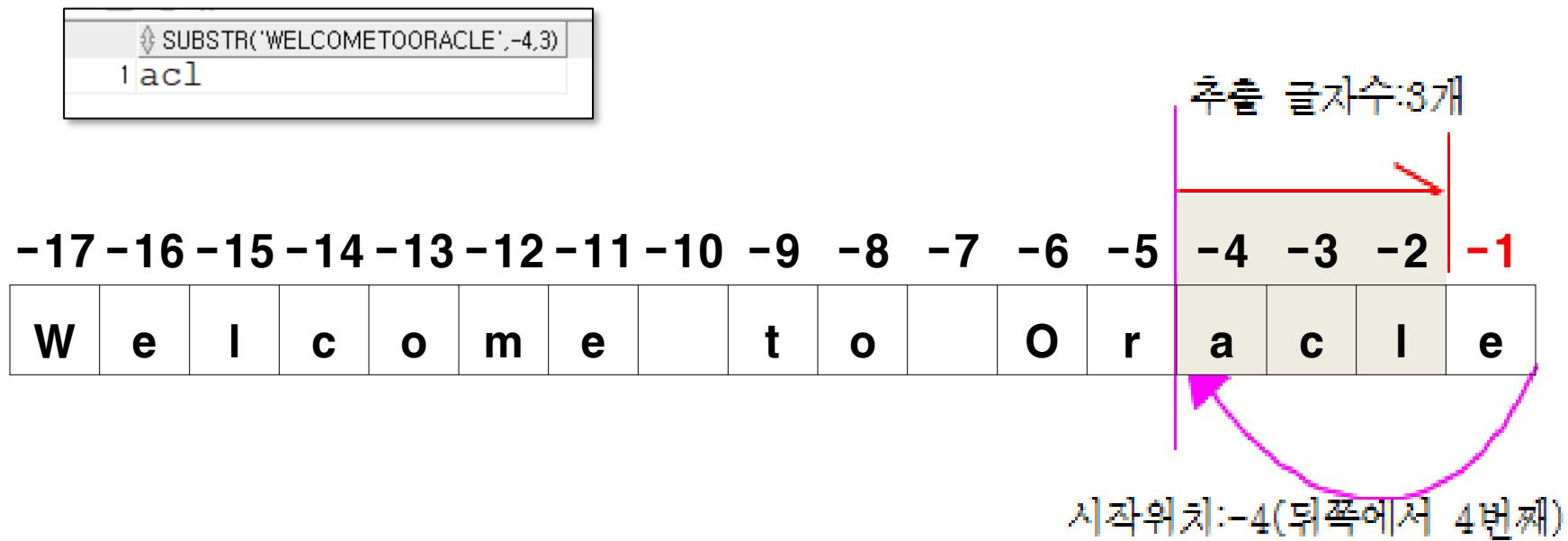
SUBSTR('WELCOME TO ORACLE', 4, 3)	
1	com

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	e	I	c	o	m	e		t	o		O	r	a	c	l	e

### 3.3 SUBSTR(2)

예

`SELECT SUBSTR('Welcome to Oracle', -4, 3) FROM DUAL;`



### 3.3 SUBSTR(3)

- ❖ SUBSTR 함수를 이용해서 입사일을 저장하고 있는 HIREDATE 컬럼에서 첫 글자부터 2개를 추출하면 사원들의 입사 년도를 구할 수 있다.
- ❖ 입사한 달만 출력하려면 HIREDATE 컬럼에서 네번째 글자부터 2개를 추출하면 된다.

예

```
SELECT SUBSTR(hiredate, 1, 2) as "년도", SUBSTR(hiredate, 4, 2) as "달" FROM emp;
```

	년도	달
1	07	03
2	07	04
3	05	02
4	03	09
5	05	04
6	03	10
7	04	01
8	04	03
9	96	10
10	05	04
11	07	03
12	07	08
13	02	10
14	07	11

## 3.4 INSTR

### ❖ INSTR

- 대상 문자열이나 컬럼에서 특정 문자가 나타나는 위치를 알려준다.

형식

**INSTR(대상, 찾을 글자, 시작 위치, 몇 번째 발견)**

예

**SELECT INSTR('WELCOME TO ORACLE', 'O', 6, 2) FROM dual;**

	INSTR('WELCOME TO ORACLE', 'O', 6, 2)
1	12

- ❖ '시작 위치'와 '몇 번째 발견'을 생략한 채 사용한 것으로 이들 값을 생략하면 모두 1로 간주되므로 시작 위치도 1이고 첫 번째 발견된 위치를 반환한다.

예

**SELECT INSTR('WELCOME TO ORACLE', 'O') FROM dual;**

	INSTR('WELCOME TO ORACLE', 'O')
1	5

## 3.5 LPAD, RPAD

### ❖ LPAD(Left Padding)

- 컬럼이나 대상 문자열을 명시된 자릿수에서 오른쪽에 나타내고, 남은 왼쪽 자리를 특정 기호로 채웁니다.

예

```
SELECT LPAD('Oracle', 20, '#') FROM dual;
```

SQL Statement
1 LPAD('ORACLE',20,'#')

1 #####Oracle

### ❖ RPAD(Right Padding)

- LPAD와 반대로 컬럼이나 대상 문자열을 명시된 자릿수에서 왼쪽에 나타내고, 남은 오른쪽 자리를 특정 기호로 채웁니다.

예

```
SELECT RPAD('Oracle', 20, '#') FROM dual;
```

SQL Statement
1 Oracle#####

## 3.6 LTRIM, RTRIM

### ❖ LTRIM

- 문자열의 왼쪽(앞)의 공백 문자들을 삭제한다.

예

```
SELECT LTRIM(' Oracle ') FROM dual;
```

LTRIM('ORACLE')
1 Oracle

### ❖ RTRIM

- LTRIM의 반대로 문자열의 오른쪽(뒤)의 공백 문자들을 삭제한다.

예

```
SELECT RTRIM(' Oracle ') FROM dual;
```

RTRIM('ORAC...')
1 Oracle

## 3.7 TRIM

### ◆ TRIM

- 칼럼이나 대상 문자열에서 특정 문자가 첫 번째 글자이거나 마지막 글자이면 잘라내고 남은 문자열만 반환합니다. (앞뒤의 'a'문자를 잘라낸다)

예

```
SELECT TRIM('a' FROM 'aaaaOracleaaaa') FROM dual;
```

ORA-00904:	TRIM('A'FROM'AAAAORACLEAAAA')
1	Oracle

### ◆ 공백을 포함한 문자열만을 TRIM함수로 표현하면 공백을 제거한다.

예

```
SELECT TRIM(' Oracle ') FROM dual;
```

ORA-00904:	TRIM('O...')
1	Oracle

## 4. 날짜 함수

구 분	설 명
SYSDATE	시스템에 저장된 현재 날짜를 반환한다.
MONTHS_BETWEEN	두 날짜 사이가 몇 개월인지를 반환한다.
ADD_MONTHS	특정 날짜에 개월 수를 더한다.
NEXT_DAY	특정 날짜에서 최초로 도래하는 인자로 받은 요일의 날짜를 반환한다.
LAST_DAY	해당 달의 마지막 날짜를 반환한다.
ROUND	인자로 받은 날짜를 특정 기준으로 반올림한다.
TRUNC	인자로 받은 날짜를 특정 기준으로 버린다.

# 4.1 SYSDATE

## ❖ SYSDATE

- 시스템에 저장된 현재 날짜를 반환하는 함수.

**예      SELECT SYSDATE FROM dual;**

SYSDATE
20/07/12

## ❖ 날짜 연산

- 날짜 형 데이터에 숫자를 더하면(날짜+숫자) 그 날짜로부터 그 기간만큼 지난 날짜를 계산하고, 날짜 형 데이터에 숫자를 빼면(날짜-숫자) 그 날짜로부터 그 기간만큼 이전 날짜를 구한다.

**예      SELECT SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일 FROM dual;**

어제	오늘	내일
20/07/11	20/07/12	20/07/13

## 4.2 ROUND(1)

### ❖ ROUND

- 숫자를 반올림하는 함수로 학습하였다. 하지만, 이 함수에 포맷 모델을 지정하면 숫자 이외에 날짜에 대해서도 반올림을 할 수 있다.

**형식    ROUND(date, format);**

포맷 모델	단위
CC, SCC	4자리 연도의 끝 두 글자를 기준으로 반올림
YYYY, YYYY, YEAR SYEAR, YY, YY, Y	년(7월 1일부터 반올림)
DDD, D, J	일을 기준
HH, HH12, HH24	시를 기준
Q	한 분기의 두 번째 달의 16일을 기준으로 반올림
MONTH, MON, MM, RM	월(16일을 기준으로 반올림)
DAY, DY, D	한주가 시작되는 날짜
MI	분을 기준

## 4.2 ROUND(2)

- ❖ ROUND 함수의 포맷 모델로 MONTH를 지정하였기에 특정 날짜(DATE)를 달(MONTH)을 기준으로 반올림한 날짜를 구한다.
- ❖ 일을 기준으로 16일보다 적으면 이번 달, 크면 다음달 1일을 구한다. 6월 9일 반올림하면 6월 1일이 되고 11월 17일은 반올림하여 12월 1이 된다. 또 한 1월 23일을 반올림하면 2월 1일 된다.
- ❖ ex) 입사일을 달을 기준으로 반올림 한 예제이다.

예

**SELECT hiredate, ROUND(hiredate, 'MONTH') FROM emp;**

	HIREDATE	ROUND(HIREDATE, 'MONTH')
1	07/03/01	07/03/01
2	07/04/02	07/04/01
3	05/02/10	05/02/01
4	03/09/02	03/09/01
5	05/04/07	05/04/01
6	03/10/09	03/10/01
7	04/01/08	04/01/01
8	04/03/08	04/03/01
9	96/10/04	96/10/01
10	05/04/07	05/04/01
11	07/03/01	07/03/01
12	07/08/09	07/08/01
13	02/10/09	02/10/01
14	07/11/09	07/11/01

# 4.3 TRUNC

## ❖ TRUNC

- TRUNC 함수 역시 숫자를 잘라내는 것뿐만 아니라 날짜를 잘라낼 수 있다. ROUND 함수와 마찬가지로 포맷 형식을 주어 다양한 기준으로 날짜를 잘라낼 수 있다.

**형식    TRUNC(date, format);**

- ❖ 특정 날짜(DATE)를 달(MONTH)을 기준으로 버림한 날짜를 구하기 위해서는 다음과 같이 표현한다.

**예        SELECT hiredate, TRUNC(hiredate, 'MONTH') FROM emp;**

	HIREDATE	TRUNC(HIREDATE, 'MONTH')
1	07/03/01	07/03/01
2	07/04/02	07/04/01
3	05/02/10	05/02/01
4	03/09/02	03/09/01
5	05/04/07	05/04/01
6	03/10/09	03/10/01
7	04/01/08	04/01/01
8	04/03/08	04/03/01
9	96/10/04	96/10/01
10	05/04/07	05/04/01
11	07/03/01	07/03/01
12	07/08/09	07/08/01
13	02/10/09	02/10/01
14	07/11/09	07/11/01

## 4.4 MONTHS\_BETWEEN(1)

### ❖ MONTHS\_BETWEEN

- 날짜와 날짜 사이의 개월 수를 구하는 함수.

형식

**MONTHS\_BETWEEN(date1, date2);**

### ❖ ex) 각 직원들의 근무한 개월 수를 구하는 예제이다.

예

**SELECT ename, sysdate, hiredate, MONTHS\_BETWEEN(sysdate,hiredate) FROM emp;**

	ENAME	SYSDATE	HIREDATE	MONTHS_BETWEEN(SYSDATE,HIREDATE)
1	김사랑	20/07/12 07/03/01	160.378824671445639...	
2	한예슬	20/07/12 07/04/02	159.346566606929510...	
3	오지호	20/07/12 05/02/10	185.088502090800477...	
4	이병현	20/07/12 03/09/02	202.346566606929510...	
5	신농협	20/07/12 05/04/07	183.185276284348864...	
6	장농건	20/07/12 03/10/09	201.120760155316606...	
7	이문세	20/07/12 04/01/08	198.153018219832735...	
8	감우성	20/07/12 04/03/08	196.153018219832735...	
9	안성기	20/07/12 96/10/04	285.282050477897252...	
10	이병현	20/07/12 05/04/07	183.185276284348864...	
11	조향기	20/07/12 07/03/01	160.378824671445639...	
12	강혜정	20/07/12 07/08/09	155.120760155316606...	
13	박충훈	20/07/12 02/10/09	213.120760155316606...	
14	조인성	20/07/12 07/11/09	152.120760155316606...	

## 4.4 MONTHS\_BETWEEN(2)

- ❖ 결과가 소수점 이하 까지 구해진다. 소수점 이하 자리는 한 달이 되지 못한 일수를 나타내기 때문에. 이를 TRUNC 함수를 사용하면 소수점 이하를 절삭할 수 있다.

예

```
SELECT ename, sysdate, hiredate,  
TRUNC(MONTHS_BETWEEN(sysdate, hiredate)) FROM emp;
```

	ENAME	SYSDATE	HIREDATE	TRUNC(MONTHS_BETWEEN(SYSDATE,HIREDATE))
1	김사랑	20/07/12	07/03/01	160
2	한예슬	20/07/12	07/04/02	159
3	오지호	20/07/12	05/02/10	185
4	이병현	20/07/12	03/09/02	202
5	신농협	20/07/12	05/04/07	183
6	장동건	20/07/12	03/10/09	201
7	이문세	20/07/12	04/01/08	198
8	감우성	20/07/12	04/03/08	196
9	안성기	20/07/12	96/10/04	285
10	이병현	20/07/12	05/04/07	183
11	조향기	20/07/12	07/03/01	160
12	강혜정	20/07/12	07/08/09	155
13	박충훈	20/07/12	02/10/09	213
14	조인성	20/07/12	07/11/09	152

## 4.5 ADD\_MONTHS

### ❖ ADD\_MONTHS

- 특정 개월 수를 더한 날짜를 구하는 함수.

**형식    ADD\_MONTHS(date, number)**

### ❖ ex) 입사 날짜에서 6개월을 추가하는 예제이다.

**예**

```
SELECT ename, hiredate, ADD_MONTHS(hiredate, 6) FROM emp;
```

	ENAME	HIREDATE	ADD_MONTHS(HIREDATE,6)
1	김사랑	07/03/01	07/09/01
2	한예슬	07/04/02	07/10/02
3	오지호	05/02/10	05/08/10
4	이병현	03/09/02	04/03/02
5	신농협	05/04/07	05/10/07
6	장동건	03/10/09	04/04/09
7	이문세	04/01/08	04/07/08
8	감우성	04/03/08	04/09/08
9	안성기	96/10/04	97/04/04
10	이병현	05/04/07	05/10/07
11	조향기	07/03/01	07/09/01
12	강혜정	07/08/09	08/02/09
13	박충훈	02/10/09	03/04/09
14	조인성	07/11/09	08/05/09

## 4.6 LAST\_DAY

- ❖ LAST\_DAY
  - 해당 날짜가 속한 달의 마지막 날짜를 반환하는 함수.
- ❖ ex) 입사한 달의 마지막 날을 구하는 예제이다.

예

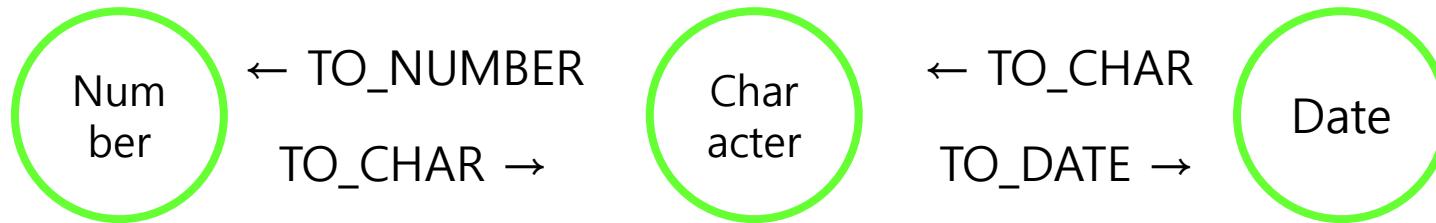
```
SELECT hiredate, LAST_DAY(hiredate) FROM emp;
```

	HIREDATE	LAST_DAY(HIREDATE)
1	07/03/01	07/03/31
2	07/04/02	07/04/30
3	05/02/10	05/02/28
4	03/09/02	03/09/30
5	05/04/07	05/04/30
6	03/10/09	03/10/31
7	04/01/08	04/01/31
8	04/03/08	04/03/31
9	96/10/04	96/10/31
10	05/04/07	05/04/30
11	07/03/01	07/03/31
12	07/08/09	07/08/31
13	02/10/09	02/10/31
14	07/11/09	07/11/30

# 5. 형 변환 함수

## ❖ 형 변환 함수

- 오라클을 사용하다 보면 숫자, 문자, 날짜의 데이터 형을 다른 데이터형으로 변환해야 하는 경우가 생기는데 이럴 때에 사용하는 함수이다.
- TO\_NUMBER, TO\_CHAR, TO\_DATE 가 있다.



구 분	설 명
TO_CHAR	날짜형 혹은 숫자형을 문자형으로 <b>변환</b> 한다.
TO_DATE	문자형을 <b>날짜형으로</b> 변환한다.
TO_NUMBER	문자형을 <b>숫자형으로</b> 변환한다.

# 5.1 TO\_CHAR(1)

## ❖ TO\_CHAR(날짜->문자)

- DATE 형태의 데이터를 지정한 양식에 의해 VARCHAR2 형의 문자로 변환한다.

형식

**TO\_CHAR(date, 'format')**

## ❖ 날짜 출력 형식의 종류

종류	의미
YYYY	년도 표현(4자리)
YY	년도 표현(2자리)
MM	월을 숫자로 표현
MON	월을 알파벳으로 표현
DAY	요일 표현
DY	요일을 약어로 표현

# 5.1 TO\_CHAR(2)

- ❖ ex) 현재 날짜를 기본 형식과 다른 형태로 출력하는 예제.

예

```
SELECT sysdate, TO_CHAR(sysdate, 'YYYY-MM-DD') FROM dual;
```

SYSDATE	TO_CHAR(SYSDATE, 'YYYY-MM...')
20/07/12	2020-07-12

- ❖ ex) 년도를 2자리(YY)로 출력하고 월은 문자(MON)로 표시하고 요일을 약어(DY)로 표시한 예제.

예

```
SELECT hiredate, TO_CHAR(hiredate, 'YY/MON/DD DY') FROM emp;
```

HIREDATE	TO_CHAR(HIREDATE, 'YY/MON/DDDY')
07/03/01	07/03/01 목
07/04/02	07/04/02 금
05/02/10	05/02/10 화
03/09/02	03/09/02 일
05/04/07	05/04/07 목
03/10/09	03/10/09 목
04/01/08	04/01/08 일
04/03/08	04/03/08 일
96/10/04	96/10/04 목
05/04/07	05/04/07 목
07/03/01	07/03/01 목
07/08/09	07/08/09 목
02/10/09	02/10/09 수
07/11/09	07/11/09 목

# 5.1 TO\_CHAR(3)

## ❖ 시간 출력 형식의 종류

종류	의미
AM 또는 PM	오전(AM), 오후(PM) 시각 표시
A.M 또는 P.M	오전(A.M), 오후(P.M) 시각 표시
HH또는 HH12	시간(1~12)
HH24	24시간으로 표현(0~23)
MI	분 표현
SS	초 표현

## ❖ ex) 현재 날짜와 시간을 출력하는 예제.

예

```
SELECT TO_CHAR(sysdate, 'YYYY/MM/DD HH24:MI:SS') FROM  
dual;
```

TO_CHAR(SYSDATE,'YYYY/MM/DDHH24:MI:SS')
1 2020/07/12 18:15:34

## 5.1 TO\_CHAR(4)

- ❖ TO\_CHAR(숫자->문자)

예

```
SELECT TO_CHAR(1230000) FROM dual;
```

TO_CHAR(1230000)
1230000

- ❖ 숫자 출력을 나열한 표

구 분	설 명
0	자릿수를 나타내며 자릿수가 맞지 않을 경우 0으로 채운다.
9	자릿수를 나타내며 자릿수가 맞지 않아도 채우지 않는다.
L	각 지역별 통화 기호를 앞에 표시한다.
.	소수점
,	천 단위 자리 구분

## 5.1 TO\_CHAR(5)

- ❖ ex) 지역별 통화 기호를 앞에 붙이고 천 단위마다 콤마를 붙여서 출력하는 예제(₩1,230,000 )

예

```
SELECT ename, sal, TO_CHAR(sal, 'L999,999') FROM emp;
```

	ENAME	SAL	TO_CHAR(SAL, 'L999,999')
1	김사랑	300	₩300
2	한예슬	250	₩250
3	오지호	500	₩500
4	이병현	600	₩600
5	신농협	450	₩450
6	장농건	480	₩480
7	이문세	520	₩520
8	감우성	500	₩500
9	안성기	1000	₩1,000
10	이병현	500	₩500
11	조향기	280	₩280
12	강혜정	300	₩300
13	박충훈	560	₩560
14	조인성	250	₩250

## 5.1 TO\_CHAR(6)

- ❖ 9는 자릿수를 나타내며 자릿수가 맞지 않으면 채우지 않는다. 하지만 0은 자릿수를 나타내며 자릿수가 맞지 않을 경우 0으로 채우게 된다.

예

```
SELECT TO_CHAR(123456, '0000000000'),  
       TO_CHAR(123456, '999,999,999')  
FROM dual;
```

	TO_CHAR(123456, '0000000000')	TO_CHAR(123456, '999,999,999')
1	000123456	123,456

## 5.2 TO\_DATE(1)

### ❖ TO\_DATE(문자->날짜)

- VARCHAR2 형태의 데이터를 지정한 양식에 의해 DATE 형의 날짜로 변환한다.
- 날짜 형은 세기, 년도, 월, 일, 시간, 분, 초와 같이 날짜와 시간에 대한 정보를 저장한다.
- 기본 날짜 형식은 'YY/MM/DD'형식으로 '년/월/일' 예를 들면 '06/03/08' 식으로 나타낸다.
- 만일 년도를 4자리로 출력하려면 'YYYY/MM/DD' 형식으로 지정해야 한다.

**형식    TO\_DATE('문자', 'format')**

## 5.2 TO\_DATE[2]

- ❖ ex) 숫자형태인 20040108 을 TO\_DATE 함수를 사용해서 날짜형으로 변환하는 예제.

예

```
SELECT ename, hiredate FROM emp WHERE hiredate =  
TO_DATE('20040108','YYYYMMDD');
```

	ENAME	HIREDATE
1	이문세	04/01/08

- ❖ ex) 올해 며칠이 지났는지 현재 날짜에서 2020/01/01을 뺀 결과를 출력하는 예제.

예

```
SELECT TRUNC(sysdate - TO_DATE('2020/01/01',  
'YYYY/MM/DD')) FROM dual;
```

TRUNC(SYSDATE-TO_DATE('2020/01/01','YYYY/MM/DD'))
193

## 5.3 TO\_NUMBER

- ❖ TO\_NUMBER(문자->숫자)
  - 특정 데이터를 숫자형으로 변환해 주는 함수.
- ❖ ex) '20,000'을 '10,000'의 차이를 알아보기 위해서 빼기를 하는 예제.

예

```
SELECT TO_NUMBER('20,000', '99,999') - TO_NUMBER('10,000', '99,999') FROM dual;
```

TO_NUMBER('20,000', '99,999')-TO_NUMBER('10,000', '99,999')
10000

## 6. DECODE(1)

### ❖ DECODE

- 프로그램 언어에서 가장 많이 사용되는 switch case 문과 같은 기능을 갖는다.
- 여러 가지 경우에 대해서 선택할 수 있다.

형식

**DECODE(표현식, 조건1, 결과1,  
조건2, 결과2,  
조건3, 결과3,  
기본결과n  
)**

## 6. DECODE(2)

- ❖ ex) 부서 번호에 해당되는 부서명을 구하는 예제.(DECODE 함수 사용)

예

```
SELECT ename, deptno, DECODE(deptno, 10, 'ACCOUNTING',
                                20, 'RESEARCH',
                                30, 'SALES',
                                40, 'OPERATIONS')
          as dname – 생략가능
FROM emp;
```

	ENAME	DEPTNO	DNAME
1	김사랑	20	RESEARCH
2	한예슬	30	SALES
3	오지호	30	SALES
4	이병현	20	RESEARCH
5	신동협	30	SALES
6	장농건	30	SALES
7	이분세	10	ACCOUNTING
8	감우성	30	SALES
9	안성기	20	RESEARCH
10	이병현	10	ACCOUNTING
11	조향기	30	SALES
12	강혜정	20	RESEARCH
13	박충훈	20	RESEARCH
14	조인성	10	ACCOUNTING

# 7. CASE(1)

## ❖ CASE

- 프로그램 언어의 if else if else 와 유사한 구조를 갖는다.
- 여러 가지 경우에 대해서 선택할 수 있다.
- DECODE 함수와 차이점이 있다면 DECODE 함수는 조건이 일치(= 비교 연산자) 하는 경우에 대해서만 적용되는 반면, CASE 함수는 다양한 비교 연산자를 이용하여 조건을 제시할 수 있으므로 범위를 지정할 수 있다.

형식

CASE 표현식 WHEN 조건1 THEN 결과1  
WHEN 조건1 THEN 결과1  
WHEN 조건1 THEN 결과1  
ELSE 결과n  
END

## 7. CASE(2)

- ❖ ex) 부서 번호에 해당되는 부서명을 구하는 예제.(CASE 함수 사용)

예

```
SELECT ename, deptno, CASE
WHEN deptno=10 THEN 'ACCOUNTING'
WHEN deptno=20 THEN 'RESEARCH'
WHEN deptno=30 THEN 'SALES'
WHEN deptno=40 THEN 'OPERATIONS'
END
as dname – 생략가능
FROM emp;
```

	ENAME	DEPTNO	DNAME
1	김사랑	20	RESEARCH
2	한예슬	30	SALES
3	오지호	30	SALES
4	이병헌	20	RESEARCH
5	신농협	30	SALES
6	장농건	30	SALES
7	이문세	10	ACCOUNTING
8	감우성	30	SALES
9	안성기	20	RESEARCH
10	이병헌	10	ACCOUNTING
11	조향기	30	SALES
12	강혜정	20	RESEARCH
13	박승훈	20	RESEARCH
14	조인성	10	ACCOUNTING

# 8. NVL 함수

## ❖ NVL

- NULL을 0 또는 다른 값으로 변환하기 위해서 사용하는 함수.

예

```
SELECT ename, sal, comm, sal*12+comm, NVL(comm,0),  
       sal*12+NVL(comm,0) FROM emp ORDER BY job;
```

	ENAME	SAL	COMM	SAL*12+COMM	NVL(COMM,0)	SAL*12+NVL(COMM,0)
1	신동협	450	200	5600	200	5600
2	오지호	500	100	6100	100	6100
3	이병현	500	(null)	(null)	0	6000
4	한예슬	250	80	3080	80	3080
5	장농건	480	(null)	(null)	0	5760
6	이병현	600	(null)	(null)	0	7200
7	이문세	520	(null)	(null)	0	6240
8	박승훈	560	(null)	(null)	0	6720
9	김사랑	300	(null)	(null)	0	3600
10	조향기	280	(null)	(null)	0	3360
11	강혜정	300	(null)	(null)	0	3600
12	조인성	250	(null)	(null)	0	3000
13	안성기	1000	(null)	(null)	0	12000
14	감우성	500	0	6000	0	6000

# 문제

1. 10월에 입사한 사원을 출력해 보시오.(SUBSTR 함수를 사용한다)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1006	장동건	부장	1008	03/10/09	480	(null)	30
1009	안성기	사장	(null)	96/10/04	1000	(null)	20
1013	박승훈	부장	1003	02/10/09	560	(null)	20

# 문제

2. 2003년도에 입사한 사원을 알아내기 위한 쿼리문을 작성하시오.(SUBSTR 함수를 사용하시오)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1004	이병현	부장	1008	03/09/02	600	(null)	20
1006	장농건	부장	1008	03/10/09	480	(null)	30

# 문제

3. 이름이 “기”로 끝나는 사원을 검색하시오.(**LIKE** 연산자를 사용하지 말고 **SUBSTR**함수를 사용하시오.)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1009	안성기	사장	(null)	96/10/04	1000	(null)	20
1011	조향기	사원	1007	07/03/01	280	(null)	30

# 문제

4. 직급에 따라 급여를 인상하도록 한다. 직급이 '부장'인 사원은 5%, '과장'인 사원은 10%, '대리'인 사원은 15%, '사원'인 사원은 20%, 이 외의 직급은 급여 그대로 출력한다.(**DECODE** 함수를 사용하시오.)

EMPNO	ENAME	JOB	SAL	UPSAL
1001	김사랑	사원	300	360
1002	한예슬	대리	250	287.5
1003	오지호	과장	500	550
1004	이병현	부장	600	630
1005	신농협	과장	450	495
1006	장농건	부장	480	504
1007	이문세	부장	520	546
1008	감우성	차장	500	500
1009	안성기	사장	1000	1000
1010	이병현	과장	500	550
1011	조향기	사원	280	336
1012	강혜정	사원	300	360
1013	박승훈	부장	560	588
1014	조인성	사원	250	300

# 문제

5. 입사일을 연도는 4자리(YYYY), 월은 숫자(MON)로 표시하시오.

HIREDATE	CDATE
07/03/01	2007/3월 /01
07/04/02	2007/4월 /02
05/02/10	2005/2월 /10
03/09/02	2003/9월 /02
05/04/07	2005/4월 /07
03/10/09	2003/10월 /09
04/01/08	2004/1월 /08
04/03/08	2004/3월 /08
96/10/04	1996/10월 /04
05/04/07	2005/4월 /07
07/03/01	2007/3월 /01
07/08/09	2007/8월 /09
02/10/09	2002/10월 /09
07/11/09	2007/11월 /09