

5장 SQL 주요 함수

5.1 DUAL 테이블과 SQL 함수 분류

- DUAL 테이블은 DUMMY라는 단 하나의 컬럼에 x라는 단 하나의 로우만을 저장하고 있으나 이 값은 아무런 의미가 없다.
- DUAL 테이블은 산술 연산의 결과를 한 줄로 얻기 위해서 오라클에서 제공하는 테이블이다.

```
SELECT *  
FROM DUAL;  
  
SELECT 24*60  
FROM EMP;  
  
SELECT 24*60  
FROM DUAL;  
  
SELECT SYSDATE  
FROM DUAL;  
-- 현재 날짜를 얻는 쿼리문
```

5.2 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.

5.2.1 절댓값 구하는 ABS 함수

- ABS 함수는 절대값을 구한다. 절대값은 방향은 없고 크기만 있는 것으로서 주어진 데이터가 음수일 경우 양수로 표현한다.

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

5.2.2 소수점 아래를 버리는 FLOOR 함수

- FLOOR 함수는 소수점 아래를 버린다. 34.5678를 FLOOR 함수에 적용하면 34가 구해진다.

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

5.2.3 특정 자릿수에서 반올림하는 ROUND 함수

- 34.5678를 반올림하면 35이다. 이와 같이 반올림한 결과를 구하기 위한 함수로 오라클에서는 ROUND가 제공된다.

```
SELECT 34.5678, ROUND(34.5678)
FROM DUAL;

SELECT 34.5678, ROUND(34.5678, 2)
FROM DUAL;

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

5.2.4 특정 자릿수에서 잘라내는 TRUNC 함수

- TRUNC 함수는 지정한 자리 수 이하를 버린 결과를 구해주는 함수이다.

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

5.2.5 나머지 구하는 MOD 함수

- MOD 함수는 나누기 연산을 한 후에 구한 몫이 아닌 나머지를 결과로 되돌려주는 함수이다.

```
SELECT MOD(27, 2), MOD(27, 5), MOD(27, 7)
FROM DUAL;
```

5.3 문자 처리 함수

5.3.1 UPPER 함수

- UPPER 함수는 입력한 문자값을 대문자로 변환하는 함수이다.

```
SELECT 'Welcome to Oracle', UPPER('Welcome to Oracle')
FROM DUAL;
```

5.3.2 LOWER 함수

- LOWER 함수는 문자열을 모두 소문자로 변경한다.

```
SELECT 'Welcome to Oracle', LOWER('Welcome to Oracle')
FROM DUAL;
```

5.3.3 INITCAP 함수

- INITCAP 함수는 문자열의 이니셜만 대문자로 변경한다.

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

5.3.4 LENGTH 함수

- LENGTH 함수는 컬럼에 저장된 데이터 값이 몇 개의 문자로 구성되었는지 길이를 알려주는 함수이다.

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

5.3.5 LENGTHB 함수

- LENGTHB 함수는 바이트 수를 알려주는 함수이다. 특히, 한글 1자는 2바이트를 차지한다. 그렇기 때문에 수행 결과를 보면 한글 3자로 구성된 '오라클'의 LENGTHB 함수의 결과는 9이 된다.

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

5.3.6 SUBSTR, SUBSTRB 함수

- SUBSTR 과 SUBSTRB 함수는 대상 문자열이나 컬럼의 자료에서 시작위치부터 선택 개수만큼의 문자를 추출한다. SUBSTRB 함수도 같은 형식이지만 명시된 개수만큼의 문자가 아닌 바이트 수를 잘라낸다는 점에서만 차이가 있다.

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

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

5.3.7 INSTR 함수

- INSTR 함수는 대상 문자열이나 칼럼에서 특정 문자가 나타나는 위치를 알려준다.

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

5.3.8 INSTRB 함수

- INSTRB 함수 역시 SUBSTRB 함수에서와 마찬가지로 문자의 위치를 알아내기 위한 바이트 기준으로 한다.

```
SELECT INSTR('데이터베이스', '이', 3, 1), INSTRB('데이터베이스', '이', 3, 1)
FROM DUAL;
```

5.3.9 LPAD/RPAD 함수

- LPAD(LEFT PADDING) 함수는 칼럼이나 대상 문자열을 명시된 자릿수에서 오른쪽에 나타내고, 남은 왼쪽 자리를 특정 기호로 채운다. RPAD(RIGHT PADDING) 함수는 반대로 칼럼이나 대상 문자열을 명시된 자릿수에서 왼쪽에 나타내고, 남은 오른쪽 자리를 특정 기호로 채운다.

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

5.3.10 LTRIM와 RTRIM 함수

- LTRIM 함수는 문자열의 왼쪽(앞)의 공백 문자들을 삭제한다. RTRIM 함수 역시 오른쪽(뒤)의 공백 문자를 잘라낸다.

```
SELECT LTRIM(' Oracle ')
FROM DUAL;
SELECT RTRIM(' Oracle ')
FROM DUAL;
```

5.3.11 TRIM 함수

- TRIM 함수는 칼럼이나 대상 문자열에서 특정 문자가 첫 번째 글자이거나 마지막 글자이면 잘라내고 남은 문자열만 반환한다.

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

5.3.12 REPLACE 함수

- REPLACE()는 문자열을 치환하는 함수이다. 예를 들어 담당자의 실수로 도서의 제목을 잘못 입력한 경우 REPLACE 함수를 사용하면 일일이 변경하지 않고 한꺼번에 변경할 수 있다.

```
CONN MADANG/MADANG;
SELECT BOOKID, REPLACE(BOOKNAME, '야구', '농구') BOOKNAME, PUBLISHER, PRICE FROM BOOK;
```

5.4 날짜 함수

5.4.1 현재 날짜를 반환하는 SYSDATE 함수

- SYSDATE 함수는 시스템에 저장된 현재 날짜를 반환하는 함수이다.

```
SELECT SYSDATE
FROM DUAL;
```

5.4.2 날짜 연산

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

```
SELECT SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일
FROM DUAL;
```

5.4.3 특정 기준으로 반올림하는 ROUND 함수

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

```
-- 형식
ROUND (date, format)

-- 예
```

```
SELECT HIREDATE, ROUND(HIREDATE, 'MONTH')
FROM EMP;
```

5.4.4 특정 기준으로 버리는 TRUNC 함수

- 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)

-- 예
SELECT HIREDATE, TRUNC(HIREDATE, 'MONTH')
FROM EMP;
```

5.4.5 두 날짜 사이의 날수를 구하는 MONTHS_BETWEEN 함수

- MONTHS_BETWEEN 함수는 날짜와 날짜 사이의 개월 수를 구하는 함수이다.

```
-- 형식
MONTHS_BETWEEN (date1, date2)

-- 예
SELECT ENAME, SYSDATE, HIREDATE, MONTHS_BETWEEN (SYSDATE, HIREDATE)
FROM EMP;
```

5.4.6 개월 수를 더하는 ADD_MONTHS 함수

- ADD_MONTHS 함수는 특정 개월 수를 더한 날짜를 구하는 함수이다.

```
-- 형식
ADD_MONTHS (date, number)

-- 예
SELECT ENAME, HIREDATE, ADD_MONTHS(HIREDATE, 6)
FROM EMP;
```

5.4.7 해당 요일의 가장 가까운 날짜를 반환하는 NEXT_DAY 함수

- NEXT_DAY 함수는 해당 날짜를 기준으로 최초로 도래하는 요일에 해당되는 날짜를 반환하는 함수이다.

```
-- 형식
NEXT_DAY (date, 요일)

-- 예
```

```
SELECT SYSDATE, NEXT_DAY(SYSDATE, '수요일')
FROM DUAL;
```

5.4.8 해당 달의 마지막 날짜를 반환하는 LAST_DAY 함수

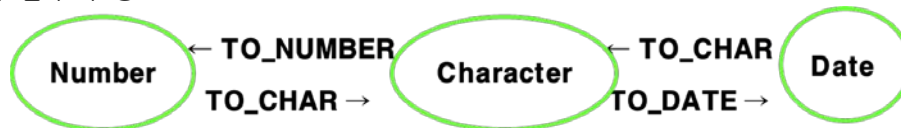
- LAST_DAY 함수는 해당 날짜가 속한 달의 마지막 날짜를 반환하는 함수이다.

```
SELECT HIREDATE, LAST_DAY(HIREDATE)
FROM EMP;
```



5.5 형 변환 함수

- 오라클을 사용하다 보면 숫자, 문자, 날짜의 데이터 형을 다른 데이터형으로 변환해야 하는 경우가 생긴다.
- 이럴 때 사용하는 함수가 형 변환 함수이다. 형 변환 함수로는 TO_NUMBER, TO_CHAR, TO_DATE가 있다.
- 형 변환 함수의 종류



5.5.1 문자형으로 변환하는 TO_CHAR 함수

(1) 날짜형을 문자형으로 변환하기

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

```
-- 형식
TO_CHAR ( 날짜 데이터, '출력형식' )

-- 예
SELECT SYSDATE, TO_CHAR(SYSDATE, 'YYYY-MM-DD')
FROM DUAL;
```

(2) 숫자형을 문자형으로 변환하기

```
SELECT ENAME, SAL, TO_CHAR (SAL, 'L999,999')
FROM EMP;

SELECT TO_CHAR (123456, '000000000'), TO_CHAR (123456, '999,999,999')
FROM DUAL;
```

5.5.2 날짜형으로 변환하는 TO_DATE 함수

- 날짜 형은 세기, 년도, 월, 일, 시간, 분, 초와 같이 날짜와 시간에 대한 정보를 저장한다.
- 오라클에서 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 지정한다.
- TO_DATE 함수는 문자열을 날짜 형으로 변환한다.

```
SELECT ENAME, HIREDATE FROM EMP
WHERE HIREDATE=TO_DATE(19810220,'YYYYMMDD');

SELECT TRUNC(SYSDATE-TO_DATE('2008/01/01','YYYY/MM/DD'))
FROM DUAL;
```

5.5.3 숫자형으로 변환하는 TO_NUMBER 함수

- TO_NUMBER 함수는 특정 데이터를 숫자형으로 변환해 주는 함수이다.

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

5.6 NULL을 다른 값으로 변환하는 NVL 함수

- NVL 함수는 NULL을 0 또는 다른 값으로 변환하기 위해서 사용하는 함수이다.

```
SELECT ENAME, SAL, COMM, SAL*12+COMM,
NVL(COMM, 0), SAL*12+NVL(COMM, 0)
FROM EMP
ORDER BY JOB;
```

5.7 선택을 위한 DECODE 함수

- DECODE 함수는 프로그램 언어에서 가장 많이 사용되는 switch case 문과 같은 기능을 갖는다. 즉, 여러 가지 경우에 대해서 선택할 수 있도록 한다.

```
-- 형식
DECODE (표현식, 조건1, 결과1,
        조건2, 결과2,
        조건3, 결과3,
        기본결과n
)

-- 예
SELECT ENAME, DEPTNO,
       DECODE(DEPTNO, 10, 'ACCOUNTING',
               20, 'RESEARCH',
               30, 'SALES',
               40, 'OPERATIONS' )
```



```
AS DNAME  
FROM EMP;
```

5.8 조건에 따라 서로 다른 처리가 가능한 CASE 함수

- CASE 함수 역시 여러 가지 경우에 대해서 하나를 선택하는 함수이다.
- 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 수도 있다.
- CASE 함수는 프로그램 언어의 if else if else 와 유사한 구조를 갖는다.

```
-- 형식  
CASE 표현식 WHEN 조건1 THEN 결과1  
            WHEN 조건2 THEN 결과2  
            WHEN 조건3 THEN 결과3  
            ELSE 결과n  
END  
  
-- 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  
FROM EMP;
```

[과제] 과제-05-01.TXT

SQL> CONN SCOTT/TIGER 로 접속하여 SQL문을 작성하세요.

1. 사원테이블(EMP)에서 입사일(HIREDATE)을 4자리 연도로 출력되도록 SQL문을 작성하세요? (ex. 1980/01/01)

<정답>

2. 사원테이블(EMP)에서 MGR컬럼의 값이 null 인 데이터의 MGR의 값을 CEO로 출력하는 SQL문을 작성 하세요?

<정답>

3. 사원 테이블(EMP)에서 가장 최근에 입사한 직원명을 출력하는 SQL문을 작성 하세요? (힌트: 서브쿼리와 MAX()함수 이용)

<정답>

4. 사원 테이블(EMP)에서 최대 급여를 받는 직원명과 최대급여금액을 출력하는 SQL문을 작성하세

요? (힌트: 서브쿼리와 MAX()함수 이용)

<정답>

[과제] 과제-05-02.TXT

SQL> CONN ORA_USER/HONG 로 접속하여 SQL 문을 작성하세요.

1. 직원테이블(employees)에는 phone_number 라는 컬럼에 직원의 전화번호가 ###.###.#### 형태로 저장되어 있다. 여기서 처음 3 자리 숫자 대신 서울 지역번호인 (02)를 붙여 전화번호를 출력하도록 쿼리를 작성해 보자.

<정답>

2. 현재일자 기준으로 직원테이블의 입사일자(hire_date)를 참조해서 근속년수가 10 년 이상인 직원을 다음과 같은 형태의 결과를 출력하도록 쿼리를 작성해보자. (근속년수가 많은 직원순서대로 결과를 나오도록 하자)

직원번호 직원명 입사일자 근속년수

<정답>

3. 고객 테이블(CUSTOMERS)에는 고객 전화번호(cust_main_phone_number) 컬럼이 있다. 이 컬럼 값은 ###-###-#### 형태인데, '-' 대신 '/'로 바꿔 출력하는 쿼리를 작성해 보자. (힌트: REPLACE() 함수 사용)

<정답>

4. 고객 테이블(CUSTOMERS)의 고객 전화번호(cust_main_phone_number) 컬럼을 다른 문자로 대체(일종의 암호화)하도록 쿼리를 작성해 보자. (힌트: TRANSLATE() 함수 사용)

<정답>

5. 고객 테이블(CUSTOMERS)에는 고객의 출생년도(cust_year_of_birth) 컬럼이 있다. 현재일 기준으로 이 컬럼을 활용해 30 대, 40 대, 50 대를 구분해 출력하고, 나머지 연령대는 '기타'로 출력하는 쿼리를 작성해보자.

<정답>

6. 5 번 문제는 30~50 대 까지만 표시했는데, 전 연령대를 표시하도록 쿼리를 작성하는데, 이번에는 DECODE 대신 CASE 표현식을 사용해보자.

<정답>