SQL #1. QUERY (1)

I. 기본 SELECT

SQL*Plus에서 SQL 문장 작성 시 유의 사항

- 모든 SQL 문장은 세미콜론(;)으로 끝낸다.
- SQL 문장은 한 줄로 입력하거나 여러 줄로 보기 좋게 나누어 입력한다.
- SQL 문장은 대소문자를 가리지 않는다. (Case insensitive)
- data 값은 대소문자를 가린다. (Case sensitive)

SELECT 문장은 Table에서 data를 검색하기 위한 문장으로 응용 프로그램 작성 시 가장 많이 사용하는 문장이다.

최소한 SELECT절과 FROM절이 있어야 SELECT 문장이 가능하다.

Default Column Heading : column 명이 대문자로 display된다.
Default Data justification : Number 값은 right-justified, Character와 Date 값은 left-justified 된다.

실습계정 : scott

```
SQL > conn scott;
```

SELECT 기본

```
SQL> SELECT ename FROM emp;
```

한화면에 보기 힘들다면 pagesize와 linesize를 조정하여 한 화면에서 볼 수 있도록 만들자.

```
SQL>  SET pagesize 1000
```
(*1000줄을 한 페이지로 설정하는 SQL*Plus 명령 실행)

```
SQL>  /
```
(* SQL*Plus의 buffer에 들어있는 마지막 SQL 문장 다시 실행)

SQL> select ename from emp;
SQL> Select Ename From Emp;
SQL> SELECT ename
FROM emp;
SQL> SELECT empno, ename FROM emp;
SQL> SELECT ename, empno FROM emp;
SQL> SELECT ename, ename FROM emp;
SQL> SELECT * FROM emp;

산술연산자
+, -, *, / : 사칙연산
사칙연산의 순서: * / + -,
괄호 사용가능

SQL> SELECT sal, -sal FROM emp;
SQL> SELECT sal, sal*1.1 FROM emp;
SQL> SELECT sal, comm, sal + comm FROM emp;
SQL> SELECT sal, -sal + 100 * -2 FROM emp;
SQL> SELECT sal, (-sal + 100) * -2 FROM emp;

모든 사원의 연봉을 출력하기 위한 문장들이다. 다음을 실습해 보시오.
SQL> SELECT empno, ename, sal, sal*12 FROM emp;
SQL> SELECT empno, ename, sal, sal*12 + comm FROM emp;
SQL> SELECT empno, ename, sal, sal + comm * 12 FROM emp;
SQL> SELECT empno, ename, sal, (sal + comm) * 12 FROM emp;

NULL: 아무런 값이 입력되어 있지 않음을 표시
검색을 할때는 IS NULL 이용
NULL을 포함한 모든 계산의 결과는 NULL

다음 문장을 실행해 보시오. Column이 Null인 경우 어떻게 표현되는가?
Comm 값이 있는 사원은 어떤 사원들인가?
사번 7844인 사원의 커미션은 얼마인가?

SQL> SELECT empno, job, comm FROM emp;

(* Null인 값은 비어있는 것으로 표현된다. JOB이 SALESMAN인 사원들에게만 커미션이 적용되

며, 사번 7844인 사원의 커미션은 0이다.)

다음은 사원들의 연봉을 계산하는 문장이다.
Comm 값이 Null인 경우 연봉은 얼마인가?
연봉 계산한 수식의 column heading은 어떻게 나타나는가?

SQL> SELECT sal, comm, (sal+comm)*12 FROM emp;
(* comm 값이 Null인 row의 경우 (sal+comm)*12를 한 결과도 모두 Null이 된다.
또한, expression 전체가 column heading으로 나타난다.)

다음은 사번과 커미션을 출력하면서 커미션이 없는 사원의 경우 Null이 아니라 0으로 출력하도록 하는 문장이다. 실행해 보시오.

SQL> SELECT empno, ename, NVL(comm, 0) comm FROM emp;

다음은 매니저가 없는 , 즉 최고 직급의 사원인 경우 'No Manager'라고 출력되도록 하는 문장이다. 실행하여 Error 메시지를 적어보고 Error가 나는 이유를 설명하시오.

SQL> SELECT NVL(mgr, 'No Manager') FROM emp;

*** Error 발생. 이유는 :**

column alias를 사용한 문장들이다. column heading이 어떻게 나타나는지를 기록하고, Error가 나는 문장에 대해서는 이유를 설명하시오.

SQL> SELECT sal*12 annual_salary FROM emp;
SQL> SELECT sal*12 Annual_Salary FROM emp;
SQL> SELECT sal*12 Annual Salary FROM emp;

*** Error 발생. 이유?**

SQL> SELECT sal*12 "Annual Salary" FROM emp;
SQL> SELECT sal*12 AS "Annual Salary" FROM emp;

연결연산자: 문자열을 연결(Concatunation)

SQL> SELECT empno||ename FROM emp;
SQL> SELECT empno||ename||hiredate FROM emp;
(* number나 date 값은 default 형태의 character 값으로 자동 변환된 후 연결된다.)

Literal: 다음을 실습해 보시오.
' ' → Literal , SQL문에 사용되는 데이터 등에 사용
" " → SQL의 Alias 이름, 테이블의 이름 등 SQL 파서에서 인식할 문자열에 사용

SQL> SELECT ename|| ' ' || sal FROM emp;
SQL> SELECT ename || ' is working as a ' || job FROM emp;

FROM 절의 table을 바꿔가며 literal을 출력해 보시오.

DUAL Table: 화면에 특정 연산 결과 등을 출력할 때 사용. 한 개의 Row를 갖는 가상의 테이블
- SELECT의 결과는 항상 테이블의 row수만큼 나오므로 별도의 table이 필요없는 질의에서 사용

결과 row의 개수는 table의 row 수와 같다.

SQL> SELECT 'Korea Fighting' FROM emp;
(* Korea Fighting 이라는 literal이 14 번 출력된다. 이유는 :

)

SQL> SELECT 'Korea Fighting' FROM dept;
(* Korea Fighting 이라는 literal이 4 번 출력된다. 이유 :

)

literal이나 literal들의 연산결과를 출력해 볼 때는 sys사용자 소유의 dual이라는 dummy table을 활용한다.

SQL> SELECT 'Korea Fighting' FROM dual;

SQL> SELECT 10 + 20 FROM dual;
SQL> SELECT 'Red' || ' ' || 'Devil' FROM dual;

dual table을 이용하여 server의 현재 시각이나 현재 접속중인 DB 사용자를 조회해 볼 수 있다.

SQL> SELECT sysdate, user FROM dual;

DISTINCT : 중복제거

==========

DISTINCT: 다음 문장들을 실행하고 결과를 비교하시오.
몇 개의 튜플이 출력되는가??

SQL> SELECT job FROM emp;
(? rows selected)
SQL> SELECT DISTINCT job FROM emp;
(? rows selected)
SQL> SELECT deptno FROM emp;
(? rows selected)
SQL> SELECT DISTINCT deptno FROM emp;
(? rows selected)

다음은 여러 column에 대한 중복 값을 제거하는 문장이다. 결과를 비교하시오.

SQL> SELECT deptno, job FROM emp;
(? rows selected)
SQL> SELECT DISTINCT deptno, job FROM emp;
(? rows selected)
SQL> SELECT DISTINCT job,deptno FROM emp;
(? rows selected)

DATE 처리

Date 값에 대해 조건을 줄 때는 현재 session의 NLS_DATE_FORMAT에 맞춰 주도록 한다.

현재 Date포멧은?

```
SQL> SELECT value FROM nls_session_parameters
          WHERE parameter = 'NLS_DATE_FORMAT';
```

그외에도 각종 SESSION 파라메터를 확인해보자.

```
SQL> select * from nls_session_parameters;
```

날짜 출력 포멧 변경, RR은 Y2K고려한 2자리 년도

```
SQL> ALTER SESSION SET nls_date_format = 'DD-MON-RR';

SQL> SELECT empno, ename FROM emp WHERE hiredate >= '01-JAN-82';
```

ORA-01843: not a valid month 에러가 발생한다면?

현재 어떤 언어로 설정되어있는가?

```
SQL> SELECT value FROM nls_session_parameters WHERE parameter = NLS_LANGUAGE;
```

KOREAN 으로 설정되어있다면, 아래와 같이 입력해보자.

```
SQL> SELECT empno, ename FROM emp WHERE hiredate >= '01-1월-82';
```

현재 session에서 NLS_LANGUAGE 변경

```
SQL> ALTER SSESSION SET nls_language = 'AMERICAN';
```

다시 확인 해보자

```
SQL> SELECT value FROM nls_session_parameters WHERE parameter = NLS_LANGUAGE;
```

다시 한국어로 변경하고 싶다면,

```
SQL> ALTER SSESSION SET nls_language = 'KOREAN;

SQL> ALTER SESSION SET nls_date_format = 'RR/MM/DD';

SQL> SELECT empno, ename FROM emp WHERE hiredate >= '82/01/01';
```

집합에 대한 연산자:
Any: 주어진 그룹중에 하나라도
In : 그룹에 포함되면

==========

ANY / IN 연산자

BOSTON이나 DALLAS 에 위치한 부서를 출력하시오.

SQL> SELECT dname, loc FROM dept WHERE loc IN ('BOSTON', 'DALLAS');

SQL> SELECT dname, loc FROM dept WHERE loc =ANY ('BOSTON', 'DALLAS');

30, 40번 부서에 속하지 않는 사원들을 출력하시오.

SQL> SELECT ename,deptno FROM emp WHERE deptno NOT IN (30,40);

SQL> SELECT ename,deptno FROM emp WHERE deptno <>ALL (30,40);

DALLAS의 20번 부서, 또는 CHICAGO의 30번 부서를 출력하시오.

SQL> SELECT * FROM dept WHERE (deptno,loc) IN ((20,'DALLAS'),(30,'CHICAGO'));

#==========

기타 연산자:
BETWEEN: 사이에
LIKE: 문자열 부분 매칭

 # 급여가 2000에서 3000 사이인 사원을 출력하시오.

SQL> SELECT ename, job, sal FROM emp WHERE sal BETWEEN 2000 AND 3000;

이름이 A 로 시작되는 사원을 출력하시오.

SQL> SELECT ename FROM emp WHERE ename LIKE 'A%';

사번이 8번으로 끝나는 사원을 출력하시오.

SQL> SELECT empno, ename FROM emp WHERE empno LIKE '%8';

82년도에 입사한 사원을 출력하시오.

SQL> SELECT ename, hiredate FROM emp WHERE hiredate LIKE '82%';

부서명에 X_Y 가 포함되어 있는 부서를 출력하시오.

SQL> SELECT dname, loc FROM dept WHERE dname LIKE '%X/_Y%' ESCAPE '/';

IS NULL, IS NOT NULL : Null인가? 아닌가? (= 을 사용하면 결과가 안나옴)

커미션 지급 대상인 사원을 출력하시오.

SQL> SELECT ename, comm FROM emp WHERE comm IS NOT NULL;

==========

아래 두 문장의 결과를 비교해 보고 차이점을 설명하시오.

SQL> SELECT ename, comm FROM emp WHERE comm IS NULL;

SQL> SELECT ename, comm FROM emp WHERE comm = NULL;

각종 질의 실습

사번이 7788인 사원의 이름과 급여를 출력하시오.

SQL> SELECT ename, sal FROM emp WHERE empno = 7788;

급여가 3000 이 넘는 직종을 출력하시오.

SQL> SELECT job FROM emp WHERE sal > 3000;

PRESIDENT 를 제외한 사원들의 이름과 직종을 출력하시오.

SQL> SELECT ename, job FROM emp WHERE job <> 'PRESIDENT';

BOSTON 지역에 있는 부서의 번호와 이름을 출력하시오.

SQL> SELECT deptno, dname FROM dept WHERE loc = 'BOSTON';

직종이 CLERK 인 사원 중에서 급여가 1000 이상인 사원을 출력하시오.

SQL> SELECT ename, job, sal FROM emp WHERE job = 'CLERK' AND sal >= 1000;

commission을 받는 사원을 출력하시오.

SQL> SELECT ename FROM emp WHERE comm IS NOT NULL;

SQL> SELECT ename FROM emp WHERE NOT comm IS NULL;

10번 부서와 20번 부서에 속한 사원을 출력하시오.

SQL> SELECT ename, deptno FROM emp WHERE deptno = 10 OR deptno = 20;

SQL> SELECT ename, deptno FROM emp WHERE deptno IN (10, 20);

10번과 20번 부서에 속하지 않는 사원의 이름과 부서번호를 출력하시오.

SQL> SELECT ename, deptno FROM emp WHERE deptno <> 10 AND deptno <> 20;

SQL> SELECT ename, deptno FROM emp WHERE deptno NOT IN (10,20);

급여가 2000에서 3000 사이인 사원을 출력하시오.

SQL> SELECT ename, job, sal FROM emp WHERE sal >= 2000 AND sal <= 3000;

SQL> SELECT ename, job, sal FROM emp WHERE sal BETWEEN 2000 AND 3000;

정렬: ORDER BY

==========

ORDER BY

Null 값은 오름차순의 경우 맨 마지막에, 내림자순의 경우 맨 처음에 display된다.

SQL> SELECT ename, comm FROM emp ORDER BY comm;

SQL> SELECT ename, comm FROM emp ORDER BY comm DESC;

급여가 적은 사원부터 출력하시오.

SQL> SELECT ename, sal FROM emp ORDER BY sal;

급여가 많은 사원부터 출력하시오.

SQL> SELECT ename, sal FROM emp ORDER BY sal DESC;

급여가 많은 사원부터 출력하되 급여가 같은 경우 이름 순서대로 출력하시오.

SQL> SELECT ename, sal FROM emp ORDER BY sal DESC, ename;

급여가 많은 사원부터 출력하되 급여가 같은 경우 이름 순서대로 출력하시오.
(ORDER BY 절에 숫자 사용: 몇번째 컬럼?)

SQL> SELECT ename, sal FROM emp ORDER BY 2 DESC, 1;

SELECT 절에 나타나지 않은 column에 대해서도 정렬이 가능하다.

SQL> SELECT ename FROM emp ORDER BY sal DESC;

SQL> SELECT sal, ename FROM emp ORDER BY sal DESC;

II. Single-Row Function

NUMBER 함수

#ROUND, TRUNC 는 첫 번째 argument를 소수점 아래 두 번째 argument자리까지 표현한다.

```
SQL> SELECT sal, ROUND(sal, -3), TRUNC(sal, -3) FROM emp;
```

#한페이지에 안보이면 페이지 가로 세로 조정

```
SQL> set pagesize 100
SQL> set linesize 100
SQL> /
```

#ROUND, TRUNC 함수의 두 번째 argument를 생략하면 default로 0 이다.

```
SQL> SELECT ROUND(45.925), ROUND(45.925, 0), TRUNC(45.925), TRUNC(45.925, 0)
            FROM dual;
```

#FLOOR, CEIL 함수는 argument가 1개이며, TRUNC나 ROUND로 바꾸어 표현이 가능하다.
#FLOOR나 CEIL은 정수부분만 표시한다.
#TRUNC나 ROUND는 자리수 표현, 자리수 지정 없으면 정수부분

```
SQL> SELECT FLOOR(45.925), CEIL(45.925) FROM dual;
SQL> SELECT TRUNC(45.925), ROUND(45.925) FROM dual;
```

 # 각종 수학 함수 다음을 실습해 보시오.

```
SQL> SELECT MOD(10, 3), MOD(10, -3), MOD(-10, 3), MOD(45.925, 10) FROM dual;
SQL> SELECT ABS(-15), ABS(15) FROM dual;
SQL> SELECT SIGN(-15),SIGN(15) FROM dual;
SQL> SELECT SIN(3.141592/2) FROM dual;
SQL> SELECT EXP(4) FROM dual;
(* e = 2.71828183 ... )
```

문자열 함수

```
SQL> SELECT CONCAT('Oracle','DBMS'), INITCAP('Oracle DBMS'),
            LOWER('Oracle DBMS'), UPPER('Oracle DBMS')
           FROM dual;
SQL> SELECT LPAD('Oracle DBMS', 13, 'x'), RPAD('Oracle DBMS', 13, 'x') FROM dual;
```

```
SQL> SELECT LPAD(ename, 15)||'    '||RPAD(job,20) FROM emp;
```

 # TRIM: 주어진 문자가 아닌 문자가 나올때까지 지운다.

```
SQL> SELECT ename , LTRIM(ename, 'AB'), RTRIM(ename, 'SR') FROM emp;
SQL> SELECT 'The job of ' || INITCAP(ENAME) || ' is ' || LOWER(JOB)
                    || '.' "EMPLOYEE'S JOB STATUS"
          FROM emp;
```

#hr로 접속해보자. hr계정이 풀려있어야 한다.

```
SQL> conn hr
```

→hr이 lock 되어있다면, 계정을 풀어준다.

```
SQL > conn system
SQL > ALTER user hr ACCOUNT UNLOCK;
SQL > ALTER user hr IDENTIFIED BY hr;
SQL > conn hr
```

 # DB 상의 data가 대소문자를 구별하므로 저장된 정확한 형태를 알지 못하는 경우 Function을 이용하여 data를 변형시킨 후 비교하기도 한다.

```
SQL> SELECT department_id, department_name
       FROM departments   WHERE department_name = 'SALES';

SQL> SELECT department_id, department_name
      FROM departments   WHERE UPPER(department_name) = 'SALES';
```

 # CHR, ASCII

```
SQL> SELECT CHR(79)||CHR(114)||CHR(97)||CHR(99)||CHR(108)||CHR(101) FROM dual;
SQL> SELECT ASCII('O'),ASCII('r'),ASCII('a') FROM dual;
```

 # 문자열 치환

```
SQL> SELECT REPLACE('Oracle DB System','DB','Database') FROM dual;
```

 # 문자열 일부 추출

```
SQL> SELECT SUBSTR('Oracle DB System',2,4) FROM dual;
```

 # 각 글자 단위로 변환 A->1, B->2, ...

```
SQL> SELECT TRANSLATE('Oracle DBMS','ABCD','1234') FROM dual;
```

 # 처음 나오는 위치?

```
SQL> SELECT INSTR('Oracle DBMS', 'a') FROM dual;
```

 # 길이

```
SQL> SELECT LENGTH('Oracle DBMS') FROM dual;
```

#SUBSTR: 문자열 일부 추출

```
SQL> SELECT department_name, SUBSTR(department_name, 1,3) FROM departments;
SQL> SELECT department_name, SUBSTR(department_name, 1) FROM departments;
SQL> SELECT department_name, SUBSTR(department_name, -5, 3) FROM departments;
```

 # LENGTH 는 날짜 값의 경우 문자열로 디스플레이되는 길이를 return한다.

```
SQL> SELECT LENGTH(last_name), LENGTH(hire_date) FROM employees;
SQL> SELECT last_name, hire_date FROM employees;
```

#INSTR 함수는 지정된 문자가 문자열 내에 없을 때는 0을 return한다.

```
SQL> SELECT department_name,
    INSTR(department_name, 'Sale'),
    INSTR(department_name, '%') FROM departments;
```

#SCOTT으로 접속해보자.

```
SQL> conn scott
```

 # Number data의 경우 character string으로 자동 변환된 후에 처리된다.

```
SQL> SELECT CONCAT(ename, sal) FROM emp;
SQL> SELECT INSTR(sal,'00') FROM emp;
SQL> SELECT LPAD(sal, 10, '₩') FROM emp;
SQL> SELECT LENGTH(sal) FROM emp;
```

DATE 함수

#날짜 더하기

```
SQL> SELECT ADD_MONTHS(sysdate, 5), ADD_MONTHS(sysdate, -5) FROM dual;
```

 # 마지막 날?

SQL> SELECT LAST_DAY('03/01/01'), LAST_DAY('03/02/01') FROM dual;

 # 사이에 몇 달이 ?

SQL> SELECT MONTHS_BETWEEN('03/01/01', '03/07/01') FROM dual;

 # 날짜 변환

```
SQL> SELECT TO_CHAR(NEW_TIME(TO_DATE('03:10:30','HH24:MI:SS'),
                        'EST','GMT'),'HH24:MI:SS') FROM dual;
```

 # 다음 일요일의 날짜는?

SQL> SELECT NEXT_DAY(sysdate, '일요일') FROM dual;

(* Date 값에 사용되는 언어를 바꿔준다.)

SQL> ALTER SESSION SET NLS_DATE_LANGUAGE='AMERICAN';

 # 다음 일요일 (위에서 날짜 포멧을 바꿔야 SUN이라는 문자열을 인식한다.)

SQL> SELECT NEXT_DAY(sysdate, 'SUN') FROM dual;

 # 다음 날짜가 어떻게 다른지 비교하시오

SQL> SELECT ROUND(TO_DATE('03/07/16'), 'YEAR') FROM dual;
SQL> SELECT TRUNC(TO_DATE('03/07/16'), 'YEAR') FROM dual;

SQL> SELECT ROUND(TO_DATE('03/07/16'), 'MONTH') FROM dual;
SQL> SELECT TRUNC(TO_DATE('03/07/16'), 'MONTH') FROM dual;

SQL> SELECT ROUND(TO_DATE('03/07/16'), 'DAY') FROM dual;
SQL> SELECT ROUND(TO_DATE('03/07/17'), 'DAY') FROM dual;
SQL> SELECT TRUNC(TO_DATE('03/07/16'), 'DAY') FROM dual;
SQL> SELECT TRUNC(TO_DATE('03/07/17'), 'DAY') FROM dual;

 # 오늘 날짜

SQL> SELECT sysdate FROM dual;

 # 오늘 현재 날짜 5일 뒤의 날짜와 5일 전의 날짜를 출력하시오.

SQL> SELECT sysdate + 5, sysdate - 5 FROM dual;

#이름이 KING 인 사원의 근무 일수를 출력하시오.

SQL> SELECT sysdate - hiredate FROM emp WHERE ename = 'KING';

 # 조회 결과 정수 부분을 근무일수가 되며 소수부분은 시간으로 나타낼 수 있다.
 # 만약 4479.45341 가 return이 되었다면 4479 + 0.45341 에서
 # 소수부분에 24를 곱하면, 24시간 * 0.45341 = 10.88184
 # 즉, 10시간 + 0.88184 시간이다. 소수부분에 60을 곱하면, 60분 * 0.88184 = 52.9104
 # 즉, 52분 + 0.9104 분이다. 결국, 약 4479 일 10시간 53분 정도라고 환산해 볼 수 있다.

 # 근무한 지 200000 시간이 넘은 사원을 출력하시오.
SQL> SELECT ename, (sysdate - hiredate) * 24 FROM emp
WHERE (sysdate - hiredate) * 24 > 200000;

 # 모든 사원에 대해 근무한 지 몇 주가 지났는지를 출력하시오.
SQL> SELECT (sysdate - hiredate)/7 weeks FROM emp;

변환 함수

 # 다음과 같이 여러 가지 형식을 사용하여 TO_CHAR 함수를 실습해 보시오.

 # FM은 빈공백이나 0을 없애라는 뜻.
SQL> SELECT sal, TO_CHAR(sal, '99,999'),TO_CHAR(sal, '099,999') ,
TO_CHAR(sal, 'FM99,999'),TO_CHAR(sal, '99,999.0')
FROM emp;

 # 음수일때 MINUS부호 MI 나 괄호 PR 붙이기
SQL> SELECT TO_CHAR(-12345, '99,999MI'), TO_CHAR(-12345, '99,999PR') FROM dual;
SQL> SELECT TO_CHAR(12345, '99,999MI'), TO_CHAR(12345, '99,999PR') FROM dual;

 # ROMAN 숫자 RN, rn
SQL> SELECT rownum, TO_CHAR(rownum, 'RN'), TO_CHAR(rownum, 'rn') FROM emp;

 # 부호 붙이기 S
SQL> SELECT TO_CHAR(-12345, 'S99,999'), TO_CHAR(12345, 'S99,999') FROM dual;

 # 16진수 변환
SQL> SELECT sal, TO_CHAR(sal, 'XXXX'), TO_CHAR(sal, 'xxxx') FROM emp;

 # 화폐단위

 # 현재 session의 Local Currency를 확인한 후 다음을 실습해 보시오.

SQL> SELECT value FROM nls_session_parameters WHERE parameter = 'NLS_CURRENCY';

SQL> ALTER SESSION SET NLS_CURRENCY='₩';

SQL> SELECT TO_CHAR(sal, '$99,999'), TO_CHAR(sal, 'L99,999') FROM emp;

 # TO_NUMBER 함수는 character string이 number와 +,?로만 구성된 경우

 # format을 주지 않고도 변환이 가능하다.

SQL> SELECT TO_NUMBER('1234') FROM dual;

SQL> SELECT TO_NUMBER('-1234') FROM dual;

SQL> SELECT TO_NUMBER('+1234') FROM dual;

SQL> SELECT TO_NUMBER('$123,456','$999,999') FROM dual;

DATE 변환

 # 현재의 시각을 출력하시오.

SQL> SELECT TO_CHAR(sysdate, 'HH24"시" MI"분" SS"초"') FROM dual;

SQL> SELECT TO_CHAR(sysdate, 'HHAM'), TO_CHAR(sysdate, 'HHPM') FROM dual;

 # 오늘이 올해의 몇 번째 날인지를 출력하시오.

SQL> SELECT TO_CHAR(sysdate, 'DDD"일"') FROM dual;

 # 오늘의 요일을 출력하시오.

SQL> SELECT TO_CHAR(sysdate, 'DAY DY') FROM dual;

 # 사원들 입사 요일을 출력하시오.

SQL> ALTER SESSION SET NLS_DATE_LANGUAGE='AMERICAN';

SQL> SELECT TO_CHAR(HIREDATE, 'DAY"x"') FROM emp;

SQL> ALTER SESSION SET NLS_DATE_LANGUAGE='KOREAN';

SQL> SELECT TO_CHAR(HIREDATE, 'DAY"x"') FROM emp;

(* 이 때, 요일의 이름은 고정된 9자리 string으로 변환되어 출력된다.)

 # 오늘이 올해의 몇 번째 주의 몇 번째 날인지를 출력하시오.

SQL> SELECT TO_CHAR(sysdate, 'WW"주" D"일"') FROM dual;

#사원들의 입사일을 출력하시오.

SQL> SELECT TO_CHAR(HIREDATE, 'BC YYYY Q MM DD') FROM emp;

SQL> SELECT TO_CHAR(HIREDATE, 'AD YY Q MON DD') FROM emp;

 # 사원들 입사일의 월을 알파벳 전체 이름으로 출력하시오.

SQL> SELECT TO_CHAR(HIREDATE, 'MONTH"월"') FROM emp;

(* 이 때, 월의 이름은 고정된 9자리 string으로 변환되어 출력된다.)

SQL> SELECT TO_CHAR(HIREDATE, 'FMMONTH"월"') FROM emp;

(* 이 때, 월의 이름은 공백문자를 제거한 가변길이 string으로 변환되어 출력된다.)

 # TO_DATE 함수를 사용하여 character string을 date 값으로 변환하시오.

SQL> SELECT TO_DATE('1966, 2, 8', 'YYYY, MM, DD') FROM dual;

SQL> SELECT TO_DATE('April', 'Month', 'NLS_DATE_LANGUAGE = American') FROM dual;

SQL> SELECT TO_DATE('03', 'YY') FROM dual;

SQL> SELECT TO_DATE('03', 'MM') FROM dual;

SQL> SELECT TO_DATE('03', 'DD') FROM dual;

 # 변환 함수를 사용하지 않으면 날짜 값은 NLS_DATE_FORMAT에 맞춰 출력된다.

 # value 라는 column을 20 문자폭으로 출력하도록 설정한다.

SQL> COL value FORMAT A20

SQL> SELECT value FROM NLS_SESSION_PARAMETERS

WHERE parameter = 'NLS_DATE_FORMAT';

SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD Month YYYY';

SQL> SELECT sysdate FROM dual;

SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'RR/MM/DD';

SQL> SELECT sysdate FROM dual;

 # 다음을 실습해 보시오.

SQL> SELECT TO_CHAR(sysdate, '"오늘의 날짜" yyyy month dd') FROM dual;

(* Format에 임의의 character string을 추가할 때는 이중 인용 부호(")로 묶어준다.)

SQL> SELECT TO_CHAR(sysdate, 'CC YEAR MONTH DD') FROM dual;

SQL> SELECT TO_CHAR(sysdate, 'J') FROM dual;

SQL> SELECT TO_CHAR(sysdate, 'CCSP YEAR MONTH DAY') FROM dual;

SQL> SELECT TO_CHAR(sysdate, 'FMCCSP YEAR MONTH DAY') FROM dual;

SQL> SELECT TO_CHAR(hiredate, 'QSPTH') FROM emp;

 # TO_CHAR 함수에서 character string이 NLS_DATE_FORMAT에 맞춰져 있을 때는format을 주지 않아도 변환이 가능하다.

SQL> SELECT TO_DATE('66/02/08') FROM dual;
SQL> SELECT sysdate - TO_DATE('66/02/08') FROM dual;

RR 포멧의 경우 Y2K를 고려한 2자리 연도 표기
 # 다음 문장들을 실행하여 결과를 확인하시오.
SQL> SELECT TO_CHAR(TO_DATE('30/01/01','YY/MM/DD'),'YYYY') FROM dual ;
(* 결과가 2030으로 나타난다.)
SQL> SELECT TO_CHAR(TO_DATE('30/01/01','RR/MM/DD'),'YYYY') FROM dual;
(* 결과가 2030으로 나타난다.)
SQL> SELECT TO_CHAR(TO_DATE('80/01/01','YY/MM/DD'),'YYYY') FROM dual;
(* 결과가 2080으로 나타난다.)
SQL> SELECT TO_CHAR(TO_DATE('80/01/01','RR/MM/DD'),'YYYY') FROM dual;
(* 결과가 1980으로 나타난다.)

CASE & DECODE

==========

CASE는 IF-THEN-ELSE와 비슷한 logic을 제공한다.

각 부서별로 실적에 따라 급여를 다르게 인상하고자 한다.
10번과 20번 부서는 각각 10%, 20% 인상을 하고
나머지 부서는 동결할 경우의 급여를 CASE를 써서 출력하시오.
SQL> SELECT ename, CASE deptno WHEN 10 THEN sal * 1.1
  WHEN 20 THEN sal * 1.2
  ELSE sal END new_sal
 FROM emp;

 # DECODE
 # 각 부서별로 실적에 따라 급여를 다르게 인상하고자 한다.
 # 10번과 20번 부서는 각각 10%, 20% 인상을 하고
 # 나머지 부서는 동결할 경우의 급여를 DECODE 함수를 써서 출력하시오.
SQL> SELECT DECODE(deptno, 10, sal * 1.1, 20, sal * 1.2, sal)
          FROM emp;

 # CLERK은 sal 10%인상, Manager는 15%인상, president는 20%인상

#CASE

```
SQL> SELECT ename, job, sal, CASE job WHEN 'CLERK'   THEN   1.10*sal
                             WHEN 'MANAGER' THEN   1.15*sal
                             WHEN 'PRESIDENT'    THEN   1.20*sal
                             ELSE sal END      REVISED_SALARY
          FROM     emp;
```

 # DECODE

```
SQL> SELECT ename, job, sal, DECODE(job, 'CLERK',   1.10*sal,
                                        'MANAGER', 1.15*sal,
                                        'PRESIDENT',    1.20*sal,
                                        sal)   REVISED_SALARY
          FROM emp;
```

기타함수

 # GREATES, LEAST: 리스트 중 가장 큰 것과 가장 작은것

```
SQL> SELECT ename, job, dname, GREATEST(ename, job, dname) G,
                          LEAST(ename, job, dname) L
          FROM emp, dept where emp.deptno = dept.deptno;
```

 # VSIZE: 저장공간의 크기를 알아보자.

```
SQL> SELECT ename, hiredate, sal   FROM emp;
SQL> SELECT VSIZE(ename), VSIZE(hiredate), VSIZE(sal) FROM emp;
```

 # 사용자의 이름

```
SQL> SELECT user FROM dual;
SQL> CONN hr
SQL> SELECT user FROM dual;
```

NVL, NVL2, NULLIF, COALESCE

```
SQL> conn scott
```

 # NVL: NULL이면 2의 값 리턴

```
SQL> SELECT NVL(TO_CHAR(MGR), 'No Manager') FROM emp;
```

 # NVL2: NULL이 아니면 2, NULL이면 3 선택

SQL> SELECT NVL2(TO_CHAR(MGR), 'Manager exists','No Manager') FROM emp;

 # NULLIF: 두 값이 같으면 NULL 리턴

SQL> SELECT ename, job, NULLIF(ename, GREATEST(ename, job)) FROM emp;

 # COALESCE: 처음 NULL이 아닌값 리턴

SQL> SELECT ename, comm, mgr, COALESCE(COMM, mgr,-1000) FROM emp;

기타

#아래 두 문장의 실행 결과를 비교해 보시오.

SQL> SELECT ename, job FROM emp WHERE INSTR(job, 'ANA') > 0;

SQL> SELECT ename, job FROM emp WHERE job LIKE '%ANA%';

#사원의 이름과 매니저 사번을 출력하시오. 단, 매니저가 없는 사원의 경우 'TOP'이'라고 출력하시오.

SQL> SELECT empno, NVL(mgr, 'TOP') manager FROM emp;

SQL> SELECT empno, NVL(TO_CHAR(mgr), 'TOP') manager FROM emp;

(무엇이 잘못되었나?)

 # 매월 1,3주 토요일은 휴무이다. 현재 월의 휴무 일을 출력하시오.

```
SQL> ALTER SESSION SET NLS_DATE_LANGUAGE='KOREAN';
SQL> SELECT DECODE(TO_CHAR(TRUNC(sysdate,'MONTH'), 'D'),
                   7,TRUNC(sysdate,'MONTH'),
                   NEXT_DAY(TRUNC(sysdate,'MONTH'),'토요일')) FIRST,
                   DECODE(TO_CHAR(TRUNC(sysdate,'MONTH'), 'D'),
                   7,TRUNC(sysdate,'MONTH')+14,
                   NEXT_DAY(TRUNC(sysdate,'MONTH'),'토요일')+14) THIRD
     FROM dual;
```