**[ IV ] 단일행함수**

**단일행함수들의 분류 : 숫자함수, 문자처리함수, 날짜함수, 형변환함수, NULL, etc**

**SQL함수는 단일행함수(INPUT1 -> OUTPUT 1)**

오직 단일 행에서만 적용가능하고 행 별로 하나의 결과를 리턴

문자, 숫자, 날짜 형 변환 함수 등

ex. SELECT ENAME, TO\_CHAR(HIREDATE, 'YY"년"') 입사년 FROM EMP; --단일행

SELECT ename, replace(ename, 'A','B') FROM emp; -- input1행 output1행

**그룹함수(INPUT n행 -> OUTPUT 1행)**

여러 개의 행을 조작하여 행의 그룹당 하나의 결과를 리턴

ex. SELECT SUM(SAL) FROM EMP; -- 그룹함수(input n행, output)

ex. SELECT DEPTNO, SUM(SAL) FROM EMP GROUP BY DEPTNO;

1. DUAL 테이블과 SQL 함수 분류
   1. DUAL 테이블의 구조

; DUAL은 테스트하기 위한 용도의 oracle 내장 객체, 한 행으로 출력된다. 주로, 연산, 함수적용내용 확인 시 활용. 산술연산결과를 한줄로 얻기 위해 오라클에서 제공하는 테이블

* + - DESC DUAL;를 통해 실행결과를 보면 DUAL 테이블은 DUMMY라는 단 하나의 컬럼으로 구성되어 있습니다. 이 컬럼에는 최대 길이는 1byte입니다
    - DUMMY 컬럼엔 과연 어떤 값이 저장되어 있는 것일까요?
      * SELECT \* FROM DUAL;
      * DESC DUAL;
      * SELECT 12\*4 FROM DUAL;
      * SELECT 12\*4 FROM EMP;
      * DUAL 테이블은 DUMMY라는 단 하나의 컬럼에 X라는 단 하나의 로우만을 저장하고 있으나 이 값은 아무런 의미가 없습니다.
      * 쿼리문의 수행 결과가 하나의 로우로 출력되도록 하기 위해서 단 하나의 로우를 구성하고 있을 뿐입니다. 그래서 다음의 함수 실습에 이용할 것입니다

1. 숫자 함수 : 숫자를 처리하는 함수. 절대값, 수학함수(sin, cos, tan…), 올림(ceil), 반올림(round), 내림(trunc), 나머지(mod) 등
   1. ABS(x) : 절대값
   2. COS(x) : COSINE 값을 반환
   3. EXP(x) : e(2.71828183……)의 x승
   4. FLOOR(x) : 소수점 아래를 버림
   5. LOG(x) : log값을 반환
   6. POWER(m,n) : m의 n승
   7. SIGN(n) : n<0이면 -1, n=0이면 0, n>0이면 1을 반환
   8. SIN(x) : SINE값 반환
   9. TAN(x) : tangent값 반환
   10. ROUND(x) : 특정 자릿수에서 반올림
   11. ROUND(데이터, 반올림할 소수점 자리수) ; 반올림할 자리수가 음수인 경우 -1은 십단위. -2는 백단위
   12. TRUNC(x, n):n으로 지정한 자리수 이하를 버림
   13. MOD(x, n) : x를 n으로 나눈 나머지값
       * SELECT -10, ABS(-10) FROM DUAL;
       * SELECT SAL, ABS(SAL) FROM EMP;
       * SELECT 34.5678, FLOOR(34.5678) FROM DUAL; -- 소수점 이하 버림
       * SELECT 34.5678, CEIL(34.5678) FROM DUAL; -- 소수점 이하 올림
       * SELECT 34.5678, ROUND(34.5678) FROM DUAL; --35
       * SELECT 34.5678, ROUND(34.5678, 2) FROM DUAL; --소수2번째자릿수에서 반올림
       * SELECT 34.5678, ROUND(34.5678, -1) FROM DUAL; -- 자릿수에 음수를 지정할 수 있는데 이럴 경우 소수점 이하가 아니라 반대쪽인 일단위, 십단위, 백단위 순으로 거슬러 올라가서 반올림
       * SELECT TRUNC(34.5678, 2), TRUNC(34.5678, -1), TRUNC(34.5678) FROM DUAL; 지정한 자리수에서 버림. TRUNC 함수의 두 번째 인자 값이 2이면 소수점 이하 세 번째 자리에서 버림 연산을 하여 소수점 이하 두 번째 자리까지 표시. 두 번째 인자 값이 0이거나 없는 경우에는 소수점자리에서 버림 연산을 하고 -1인 경우는 일의 자리에서 버림 연산
       * SELECT ENAME, TRUNC(SAL, -3) FROM EMP;
       * SELECT MOD (27, 2), MOD (27, 5), MOD (27, 7) FROM DUAL;
       * 사번이 홀수인 사람을 검색해 보시오
         + SELECT \* FROM EMP WHERE MOD(EMPNO, 2)=1;
       * 홀수달에 입사한 사람들

- SELECT \* FROM EMP WHERE MOD(TO\_CHAR(HIREDATE, 'MM'),2) = 1;

* + - SELECT EXP(2) FROM DUAL; -- e의 2승
    - SELECT POWER(2,3) FROM DUAL; -- 2의 3승

1. 문자처리함수
   1. LOWER(str) : 소문자로
   2. UPPER(str) : 대문자로
   3. INITCAP(str) : 첫 글자만 대문자로 나머지 글자는 소문자로
   4. CONCAT(str1, str2) : 문자 연결
   5. SUBSTR(str, 시작할위치, 추출할갯수) : 문자를 잘라 추출 (한글은 1byte), 시작할위치는 인덱스 아님

시작할 위치가 음수이면 끝에서부터 자리수를 센다

* 1. SUBSTRB(str, 시작할위치,추출할갯수) : 문자를 잘라 추출 (한글은 3byte)
  2. LENGTH(str) : 문자길이 (한글은 1byte)
  3. LENGTHB(str) : 문자길이 (한글은 2~3byte)
  4. INSTR(str, 찾을글자, 시작위치, 몇번째발견 검색된 횟수) str에서 찾을글자를 몇번째 발견하는지
  5. LPAD, RPAD : 입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환한다
  6. TRIM : 잘라내고 남은 문자를 표시한다
  7. CHR : ASCII코드 값으로 SELECT CHR(65) FROM DUAL;
  8. ASCII : ASCII코드값을 문자로
  9. REPLACE : 문자열에서 특정문자를 변경
     + SELECT 'Welcome to Oracle', UPPER('Welcome to Oracle') FROM DUAL;
       - 문자가 변환되는 결과를 명료하게 살펴보기 위해서 특정 테이블에 대한 컬럼에 대해서 함수를 적용하는 것은 나중에 하기로 하고 우선 문자 상수에 대해서 적용해보도록 한다
     + SELECT 'Welcome to Oracle', LOWER('Welcome to Oracle') FROM DUAL;
     + SELECT LOWER(job), LOWER(ename), UPPER(job) FROM emp; -- 소문자로
     + SELECT 'WELCOME TO ORACLE', INITCAP('WELCOME TO ORACLE') FROM DUAL;
     + SELECT EMPNO, ENAME, JOB FROM EMP WHERE LOWER(JOB)='manager';
       - 굳이 job이 소문자 ‘manager’로 검색해야 한다면
     + SELECT CONCAT(CONCAT('TEST','1234'),'5678') FROM DUAL;
     + XXX는 JOB이다

- SELECT CONCAT(CONCAT(ENAME, '는 '),CONCAT(JOB,'이다')) TITLE FROM EMP;

* + - SELECT INITCAP(ename) FROM emp; -- 첫글자만 대문자로
    - SELECT ename || job FROM EMP; -- 이름+업무
    - SELECT CONCAT(ename, job) FROM EMP;
    - SELECT LENGTH('Oracle'), LENGTH('오라클') FROM DUAL;
    - SELECT LENGTH('Oracle'), LENGTHB('오라클') FROM DUAL;

B는 메모리를 차지하는 바이트수

* + - SELECT **SUBSTR**(HIREDATE, 1, 2) 년도, **SUBSTR**(HIREDATE, 4, 2) 달 FROM EMP;
      * 사원들의 입사년도 만 출력하려면 어떻게 해야 할까요?
      * SUBSTR 함수를 이용해서 입사일을 저장하고 있는 HIREDATE 컬럼에서 첫 글자부터 2개를 추출하면 됩니다. 입사한 달만 출력하려면 HIREDATE 컬럼에서 네번째 글자부터 2개를 추출하면 됩니다
    - 탄탄다지기
      * 9월에 입사한 사원을 출력해 보시오
        + SELECT \* FROM EMP WHERE SUBSTR(HIREDATE, 4, 2)='09'
      * 83년도에 입사한 직원을 알아내기 위해 SUBSTR 함수를 이용하여 HIREDATE 컬럼에서 첫 글자부터 2개를 추출하여 그 값이 83인지를 체크하는 방법으로도 구해 보도록 하세요.
        + SELECT \* FROM EMP WHERE SUBSTR(HIREDATE, 1,2)='87';
      * 다음은 이름이 E로 끝나는 사원을 검색해 보도록 합시다. SUBSTR 함수를 이용하여 ENAME 컬럼의 마지막 문자 한개만 추출해서 이름이 E로 끝나는 사원을 검색해 보도록 하시오.
        + SELECT \* FROM EMP WHERE SUBSTR(ENAME, -1,1)='E';
      * “사번, 이름, 직책(이름의 문자열 수 만큼만 표시)”의 형식의 데이터를 화면에 LIST

7369 SMITH CLERK

7499 ALLEN SALES

7521 WARD SALE

7566 JONES MANAG

* SELECT EMPNO, ENAME, SUBSTR(JOB, 1, LENGTH(ENAME)) FROM EMP;
* Hiredate의 경우, 데이터 type이 날짜 형식이지만, 자동 형변환 되어 문자열 처리 함수를 문자열을 추출하여 처리할 수 있다
  + SELECT SUBSTR(HIREDATE ,1, LENGTH(HIREDATE)) FROM EMP;
  + SELECT \* FROM EMP WHERE SUBSTR(HIREDATE, 1, 2)=’80’;
* EMP 테이블에서 사원이름을 2자리만 출력되고 나머지는 \*로 숨기도록 출력
  + SELECT RPAD(SUBSTR(ENAME,1,2),LENGTH(ENAME), '\*') FROM EMP;
* 1982년 12월1일 ~1982년12월31일 입사한 사람을 이름과 입사일로 출력;날짜형이문자형으로 자동 형변환
  + SELECT ENAME, HIREDATE FROM EMP WHERE SUBSTR(HIREDATE, 1, 5) = ‘82/12’;
  + SELECT ENAME, HIREDATE FROM EMP WHERE TO\_CHAR(HIREDATE,'YY/MM') = '82/12';
    - SELECT SUBSTR('Welcome To Oracle', 3, 4), SUBSTRB('Welcome To Oracle', 3, 4) FROM DUAL;
    - SELECT SUBSTR('Welcome to Oracle', 4, 3) FROM DUAL; --4번째부터 3글자 추출
    - SELECT SUBSTR('Welcome to Oracle', -4, 3) FROM DUAL; --시작위치를 음수 값으로 줄 경우 문자열의 뒤쪽부터 세어서 시작위치를 잡는다
    - SELECT **SUBSTR**('웰컴투오라클', 3, 4), **SUBSTRB**('웰컴투오라클', 3, 4) FROM DUAL;
      * SUBSTRB 함수는 문자열을 추출하기 위해서 문자를 셀 때 문자의 개수가 아닌 그 문자가 메모리에 저장되는 바이트 수로 셉니다.
      * 영문자는 1자는 메모리에 1바이트로 저장되기 때문에 SUBSTR 함수와 SUBSTRB 함수 어떤 함수를 사용하여도 결과가 동일합니다. 한글 1자는 2바이트를 차지하기 때문에 SUBSTR 함수와 SUBSTRB 함수의 결과가 달라집니다
    - SELECT INSTR('ABCDABCDABCD', 'B') FROM DUAL;
    - SELECT INSTR('WELCOME TO ORACLE', 'O') FROM DUAL;
      * INSTR 함수는 대상 문자열이나 칼럼에서 특정 문자가 나타나는 위치를 알려줍니다. 문자열 ‘Welcome to Oracle'에 'O'가 저장된 위치 5출력. 첫번째 문자 위치는 1로
    - INSTR(S1, S2, n, k) ; S1 문자열 중 n번째 문자부터 시작하여 찾고자 하는 문자열 S2가 k번째 나타나는 문자 위치
    - SELECT INSTR('ABCDABCDABCD', 'B', 2, 2) FROM DUAL;

-- 2째문자부터 2번째로 나오는 'B'의 위치 ==> 6

* + - SELECT INSTR('WELCOME TO ORACLE', 'O', 6, 2) FROM DUAL;
      * 6번째 위치부터 ‘O’를 두번째 발견하는 곳의 위니 12 반환
    - SELECT INSTR('데이터베이스', '이', 3, 1), INSTRB('데이터베이스', '이', 3, 1) FROM DUAL;
      * INSTRB 함수 역시 SUBSTRB 함수에서와 마찬가지로 문자의 위치를 알아내기 위한 기준으로 바이트합니다. 영문자는 1 글자가 1 바이트이므로 INSTR 함수와 INSTRB 함수의 결과가 동일합니다. 한글의 경우는 1글자가 2바이트
    - 탄탄 다지기
      * 이름의 세 번째 자리가 R인 사원을 검색하기 위해서 이름의 세 번째 자리가 R로 끝나는 사원을 검색하기 위해서 와일드카드 \_ 와 LIKE 연산자를 사용하여 다음과 같이 표현할 수 있습니다.
      * SELECT ENAME FROM EMP WHERE ENAME LIKE '\_\_R%';
      * SELECT ENAME FROM EMP WHERE INSTR(ENAME,'R',3,1)=3;
      * SELECT ENAME FROM EMP WHERE SUBSTR(ENAME, 3,1)='R';
    - SELECT LPAD('Oracle', 20, '#') FROM DUAL;
      * LPAD(LEFT PADDING) 함수는 칼럼이나 대상 문자열을 명시된 자릿수에서 오른쪽에 나타내고, 남은 왼쪽 자리를 특정 기호로 채움
    - SELECT RPAD('Oracle', 20, '#') FROM DUAL;
      * RPAD(RIGHT PADDING) 함수는 반대로 칼럼이나 대상 문자열을 명시된 자릿수에서 왼쪽에 나타내고, 남은 오른쪽 자리를 특정 기호로 채움
    - SELECT LTRIM(' Oracle ') FROM DUAL; --왼쪽공백삭제
    - SELECT RTRIM(' Oracle ') FROM DUAL; --오른쪽 공백 삭제
    - SELECT TRIM('a' FROM 'aaaaOracleaaaa') FROM DUAL;
      * TRIM 함수는 칼럼이나 대상 문자열에서 특정 문자가 첫 번째 글자이거나 마지막 글자이면 잘라내고 남은 문자열만 반환
    - SELECT TRIM(' Oracle ') FROM DUAL; --공백삭제
    - SELECT RPAD(SUBSTR(ENAME, 1,1), LENGTH(ENAME), '\*') FROM EMP;

-- 이름의 첫글자만 나오고 나머지는 \*로 출력

1. 날짜함수
   1. SYSDATE 시스템 저장된 현재 날짜를 반환한다.
   2. MONTHS\_BETWEEN 두 날짜 사이가 몇 개월인지를 반환한다
   3. ADD\_MONTHS 특정 날짜에 개월 수를 더한다
   4. NEXT\_DAY 특정 날짜에서 최초로 도래하는 인자로 받은 요일의 날짜를 반환한다
   5. LAST\_DAY 해당 달의 마지막 날짜를 반환한다
   6. ROUND 인자로 받은 날짜를 특정 기준으로 반올림한다
   7. TRUNC 인자로 받은 날짜를 특정 기준으로 버린다
      * SELECT SYSDATEFROM DUAL; --시스템에 저장된 현재 날짜를 반환
      * SELECT SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일 FROM DUAL;
      * SELECT TO\_CHAR(SYSDATE-1, 'YY-MM-DD DY HH24:MI:SS') 어제,

TO\_CHAR(SYSDATE, 'YY-MM-DD DY HH24:MI:SS') 오늘,

TO\_CHAR(SYSDATE+1, 'YY-MM-DD DY HH24:MI:SS') 내일 FROM DUAL;

* + - SELECT SYSDATE, SYSDATE+14 반납일 FROM DUAL;
      * 날짜 연산도 가능. 날짜 형 데이터에 숫자를 더하면(날짜+숫자) 그 날짜로부터 그 기간만큼 지난 날짜를 계산합니다. 날짜 형 데이터에 숫자를 빼면(날짜-숫자) 그 날짜로부터 그 기간만큼 이전 날짜를 구합니다
    - SELECT ENAME, SYSDATE, HIREDATE, TRUNC(MONTHS\_BETWEEN (SYSDATE, HIREDATE)) FROM EMP;
    - SELECT ADD\_MONTHS(SYSDATE,2) FROM DUAL;
    - SELECT SYSDATE, NEXT\_DAY(SYSDATE, '수요일') FROM DUAL;
      * 다음은 오늘을 기준으로 최초로 도래하는 수요일은 언제인지 알아보는 예제입니다.
    - SELECT LAST\_DAY(SYSDATE) FROM DUAL;
    - SELECT HIREDATE, ROUND (HIREDATE, 'MONTH') FROM EMP;
      * ROUND 함수의 포멧 모델로 MONTH를 지정하였기에 특정 날짜(DATE)를 달(MONTH)을 기준으로 반올림한 날짜를 구함. 일을 기준으로 16일보다 적으면 이번 달 1일을 크면 다음달 1일을 반환. 6월 9일 반올림하며 6월 1일이 되고 11월 17일은 반올림하여 12월 1이 됩니다. 또한 1월 23일을 반올림하면 2월 1일 됩니다. 입사일을 달을 기준으로 반올림한 예제입니다
    - SELECT HIREDATE, ROUND (HIREDATE, 'YEAR') FROM EMP;
    - SELECT HIREDATE, ROUND (HIREDATE, 'DAY') FROM EMP;
    - SELECT SYSDATE, ROUND (SYSDATE, 'DAY') FROM DUAL; -- 일요일로 반올림
    - 탄탄다지기 : 날짜에서 날짜를 빼는 연산도 사용할 수 있는데 이는 현실에서 많이 사용되는 연산입니다. 예를 들면 나는 태어나서 얼마나 살았는지, 애인과 만난 지 며칠이 흘렀는지 등을 날짜 - 날짜 연산으로 구할 수 있습니다. 이 때 주의할 점은 현재에 가까운 날짜가 앞에 와야 하며 뒤에 올 경우 음수 값이 출력 됩니다. 각 사원들의 현재까지의 근무 일수를 구해봅시다
      * SELECT ENAME, TRUNC(SYSDATE-HIREDATE) FROM EMP;
    - SELECT ENAME, HIREDATE, LAST\_DAY(HIREDATE) 첫월급날 FROM EMP; -- 월급날이 말일
    - SELECT HIREDATE, ROUND(HIREDATE+15, 'MONTH') 첫월급날 FROM EMP; -- 월급날이1일
    - SELECT HIREDATE, ROUND(HIREDATE+14, 'MONTH')+1 첫월급날 FROM EMP; -- 월급날이 2일
    - SELECT HIREDATE, ROUND(HIREDATE+13, 'MONTH')+2 첫월급날 FROM EMP; -- 월급날이 3일
    - SELECT HIREDATE, ROUND(HIREDATE+12, 'MONTH')+3 첫월급날 FROM EMP; -- 월급날이 4일
    - SELECT HIREDATE, ROUND(HIREDATE+11, 'MONTH')+4 첫월급날 FROM EMP; -- 월급날이 5일
    - SELECT HIREDATE, ROUND(HIREDATE+10, 'MONTH')+5 첫월급날 FROM EMP; -- 월급날이 6일
    - SELECT HIREDATE, ROUND(HIREDATE+9, 'MONTH')+6 첫월급날 FROM EMP; -- 월급날이 7일
    - SELECT HIREDATE, ROUND(HIREDATE+8, 'MONTH')+7 첫월급날 FROM EMP; -- 월급날이 8일
    - SELECT HIREDATE, ROUND(HIREDATE+7, 'MONTH')+8 첫월급날 FROM EMP; -- 월급날이 9일
    - SELECT HIREDATE, ROUND(HIREDATE+6, 'MONTH')+9 첫월급날 FROM EMP; -- 월급날이 10일
    - SELECT HIREDATE, ROUND(HIREDATE+5, 'MONTH')+10 첫월급날 FROM EMP; -- 월급날이 11일
    - SELECT HIREDATE, ROUND(HIREDATE+4, 'MONTH')+11 첫월급날 FROM EMP; -- 월급날이 12일
    - SELECT HIREDATE, ROUND(HIREDATE+3, 'MONTH')+12 첫월급날 FROM EMP; -- 월급날이 13일
    - SELECT HIREDATE, ROUND(HIREDATE+2, 'MONTH')+13 첫월급날 FROM EMP; -- 월급날이 14일
    - SELECT HIREDATE, ROUND(HIREDATE+1, 'MONTH')+14 첫월급날 FROM EMP; -- 월급날이 15일
    - SELECT HIREDATE, ROUND(HIREDATE 'MONTH')+15 첫월급날 FROM EMP; -- 월급날이 16일
    - SELECT HIREDATE, ROUND(HIREDATE-1, 'MONTH')+16 첫월급날 FROM EMP; -- 월급날이 17일
    - SELECT HIREDATE, ROUND(HIREDATE-2, 'MONTH')+17 첫월급날 FROM EMP; -- 월급날이 18일
    - SELECT HIREDATE, ROUND(HIREDATE-3, 'MONTH')+18 첫월급날 FROM EMP; -- 월급날이 19일
    - SELECT HIREDATE, ROUND(HIREDATE-4, 'MONTH')+19 첫월급날 FROM EMP; -- 월급날이 20일
    - SELECT HIREDATE, ROUND(HIREDATE-5, 'MONTH')+20 첫월급날 FROM EMP; -- 월급날이 21일
    - SELECT HIREDATE, ROUND(HIREDATE-6, 'MONTH')+21 첫월급날 FROM EMP; -- 월급날이 22일
    - SELECT HIREDATE, ROUND(HIREDATE-7, 'MONTH')+22 첫월급날 FROM EMP; -- 월급날이 23일
    - SELECT HIREDATE, ROUND(HIREDATE-8, 'MONTH')+23 첫월급날 FROM EMP; -- 월급날이 24일
    - SELECT HIREDATE, ROUND(HIREDATE-9, 'MONTH')+24 첫월급날 FROM EMP; -- 월급날이 25일
    - SELECT HIREDATE, ROUND(HIREDATE-10, 'MONTH')+25 첫월급날 FROM EMP; -- 월급날이 26일
    - SELECT HIREDATE, ROUND(HIREDATE-11, 'MONTH')+26 첫월급날 FROM EMP; -- 월급날이 27일
    - SELECT HIREDATE, ROUND(HIREDATE-12, 'MONTH')+27 첫월급날 FROM EMP; -- 월급날이 28일
    - SELECT HIREDATE, TRUNC(HIREDATE, 'MONTH') FROM EMP; -- 무조건 그달의 1일로
      * 입사일을 달을 기준으로 절삭한 예제
    - SELECT ENAME, SYSDATE, HIREDATE, MONTHS\_BETWEEN (SYSDATE, HIREDATE) FROM EMP;
    - SELECT ENAME, SYSDATE, HIREDATE, TRUNC(MONTHS\_BETWEEN (SYSDATE, HIREDATE)) FROM EMP -- 각 직원이 현재까지 근무한 개월 수를 구하는 예제(소수점까지 아래는 소수점을 제외하고 절삭)
    - SELECT ENAME, HIREDATE, ADD\_MONTHS(HIREDATE, 6) FROM EMP;
      * 입사날짜에서 6개월을 추가하는 예제
    - SELECT HIREDATE, LAST\_DAY(HIREDATE) FROM EMP; --입사한 달의 마지막 날
    - SELECT ename, trunc(SYSDATE-hiredate) 근무일 FROM emp;

1. 형변환함수 ; 오라클을 사용하다 보면 숫자, 문자, 날짜의 데이터 형을 다른 데이터형으로 변환해야 하는 경우가 생신다. 이럴때 사용하는 함수가 형변환 함수이다.
   1. TO\_CHAR : 날짜형 혹은 숫자형을 문자형으로 변환
      * TO\_CHAR(날짜데이터, ‘출력형식’);

SELECT SYSDATE, TO\_CHAR(SYSDATE, 'YYYY-MM-DD') FROM DUAL;

* + - * YYYY 년도 4자리 표현 / YY 년도 2자리 표현
      * MM 월을 숫자로 표현 / MON 월을 알파벳으로 표현 / DD 일
      * DAY 요일 표현 / DY 요일을 약어로 표현
      * AM또는 PM / A.M또는 P.M.. 오전 오후 시각 표시
      * HH12(12시간표현) HH24(24시간 표현) / MI 분표현 / SS 초표현
      * 사원들의 입사일을 출력하되 요일까지 출력 : SELECT HIREDATE, TO\_CHAR (HIREDATE, 'YYYY/MM/DD DAY') FROM EMP
      * 사원들의 입사일을 출력하되 년도는 2자리, 월은 문자로 출력하고 요일은 약어로 출력 : SELECT HIREDATE, TO\_CHAR (HIREDATE, 'YY/MON/DD DY') FROM EMP
      * SELECT TO\_CHAR(SYSDATE, 'YYYY/MM/DD, HH24:MI:SS') FROM DUAL;
      * SELECT TO\_CHAR(SYSDATE, 'YYYY-MM-DD DY AM HH:MI:SS') FROM DUAL;
    - TO\_CHAT(숫자데이터, ‘출력형식’);
      * 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
      * SELECT ENAME, SAL, TO\_CHAR (SAL, 'L999,999') FROM EMP;
      * SELECT TO\_CHAR (123456, '000000000'), TO\_CHAR (123456, '999,999,999') FROM DUAL;
      * SELECT TO\_CHAR(12345678, 'L99,999,999') FROM DUAL;
      * SELECT TO\_CHAR(12345678, 'L9,999,999') FROM DUAL;
  1. TO\_DATE : 문자형을 날짜형으로 변환
     + TO\_DATE(‘문자’, ‘FORMAT’);
       - 날짜 형으로 변환하기에 앞서 날짜 형에 대해서 살펴보도록 합시다. 날짜 형은 세기, 년도, 월, 일. 시간, 분. 초와 같이 날짜와 시간에 대한 정보를 저장합니다.
       - 오라클에서 기본 날짜 형식은 ‘YY/MM/DD'형식으로 ’년/월/일‘ 예를 들면 ’16/03/11‘ 식으로 나타냅니다.
       - 만일 년도를 4자리로 출력하려면 ‘YYYY/MM/DD' 형식으로 지정합니다.

SELECT ENAME, HIREDATE FROM EMP

WHERE HIREDATE=TO\_DATE('19810220','YYYYMMDD');

SELECT ENAME, HIREDATE FROM EMP

WHERE HIREDATE=TO\_DATE(19810220,'YYYYMMDD');

* + - * 올해 며칠이 지났는지 현재 날짜에서 새해날짜를 뺀 결과를 출력하는 예제
      * SELECT TRUNC(SYSDATE-TO\_DATE('2018/01/01', 'YYYY/MM/DD')) FROM DUAL;
      * SELECT TRUNC(SYSDATE-TRUNC(SYSDATE,'YEAR')) FROM DUAL;
  1. TO\_NUMBER : 문자형을 숫자형으로 변환
     + SELECT TO\_NUMBER('20,000', '99,999') - TO\_NUMBER('10,000', '99,999') FROM DUAL; -- TO\_NUMBER 함수는 특정 데이터를 숫자형으로 변환해 주는 함수입니다. 위와 같이 '20,000'을 '10,000'의 차이를 알아보기 위해서 빼기를 해 봅시다. 산술 연산을 하려면 문자형을 숫자형으로 변환한 후에 실행해야 합니다

1. NULL을 다른 값으로 변환하는 NVL 함수
   * + SELECT ENAME, SAL, COMM, SAL\*12+COMM, NVL(COMM, 0), SAL\*12+NVL(COMM, 0) FROM EMP ORDER BY JOB;
     + 탄탄다지기 ; 모든 사원은 자신의 상관(MANAGER)이 있다. 하지만 EMP 테이블에 유일하게 상관이 없는 로우가 있는데 그 사원의 MGR 칼럼 값이 NULL이다. 상관이 없는 사원만 출력하되 MGR 칼럼 값 NULL 대신 CEO로 출력해 봅시다
       - SELECT EMPNO, ENAME, NVL(MGR, 9999) MANAGER FROM EMP WHERE MGR IS NULL;
       - SELECT EMPNO, ENAME, NVL(MGR, 'CEO') 상사 FROM EMP WHERE MGR IS NULL; -- 에러
       - SELECT W.EMPNO, W.ENAME, NVL(M.ENAME, 'CEO') MANAGER FROM EMP W, EMP M WHERE W.MGR=M.EMPNO(+) AND W.MGR IS NULL;
2. 그외,
   1. 사원테이블에서 부서번호가 20인 사원의 사원번호, 사원이름, 직업, 급여를 출력하되 급여 앞에 $를 붙이고 3자리마다 ,를 출력
      * SELECT empno, ename, job, sal, TO\_CHAR(sal, ‘$999,9999’) FROM EMP WHERE deptno=20 ORDER BY sal DESC;
   2. EXTRACT ; 입력값인 날짜 데이터 타입에서 지정된 값을 추출하는 함수.
      * SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;
      * SELECT ENAME, HIREDATE, EXTRACT(MONTH FROM HIREDATE) FROM EMP WHERE DEPTNO=10;
   3. 사원테이블의 계층화
      * LEVEL : 계층적 질의문에서 검색된 결과에 대한 계층별로 레벨 표시
      * START WITH : 계층적인 출력 형식을 표현하기 위한 최상위 행
      * CONNECT BY PRIOR : 계층 관계의 데이터를 지정하는 컬럼
      * SELECT LPAD(' ', 3\*LEVEL-3)|| ENAME, LEVEL, EMPNO, MGR, DEPTNO FROM EMP START WITH MGR IS NULL CONNECT BY PRIOR EMPNO=MGR;
      * SELECT LPAD(' ', 3\*LEVEL-3)|| ENAME, LEVEL, EMPNO, MGR, DEPTNO FROM EMP START WITH ENAME=’KING’ CONNECT BY PRIOR EMPNO=MGR;
      * SELECT LPAD(' ', 3\*LEVEL-3)|| ENAME, LEVEL, EMPNO, MGR, DEPTNO FROM EMP WHERE ENAME!=’FORD’ START WITH ENAME=’KING’ CONNECT BY PRIOR EMPNO=MGR;
        + 사원계층구조에서 ‘FORD’가 노드 제외된 SQL 문장(58PT)

-- <총 연습문제>

-- 1. 현재 날짜를 출력하고 TITLE에 “Current Date”로 출력하는 SELECT 문장을 기술하시오.

-- 2. EMP 테이블에서 현재 급여에 15%가 증가된 급여를 계산하여,

-- 사원번호,이름,업무,급여,증가된 급여(New Salary),증가액(Increase)를 출력하는 SELECT 문장

--3. 이름, 입사일, 입사일로부터 6개월 후 돌아오는 월요일 구하여 출력하는 SELECT 문장을 기술하시오.

--4. 이름, 입사일, 입사일로부터 현재까지의 개월수, 급여, 입사일부터 현재까지의 받은 급여의 총계를 출력

--5. 모든 사원의 이름과 급여(15자리로 출력 좌측의 빈 곳은 “\*”로 대치)를 출력

--6. 모든 사원의 정보를 이름,업무,입사일,입사한 요일을 출력하는 SELECT 문장을 기술하시오.

--7. 이름의 길이가 6자 이상인 사원의 정보를 이름,이름의 글자수,업무를 출력

--8. 모든 사원의 정보를 이름, 업무, 급여, 보너스, 급여+보너스를 출력

-- 9.사원 테이블의 사원명에서 2번째 문자부터 3개의 문자를 추출하시오.

--10. 사원 테이블에서 입사일이 12월인 사원의 사번, 사원명, 입사일을 검색하시오.

-- 시스템에 따라 DATEFORMAT 다를 수 있으므로 아래의 방법도 알아보자

--11. 다음과 같은 결과를 검색할 수 있는 SQL 문장을 작성하시오

--EMPNO ENAME 급여

--7369 SMITH \*\*\*\*\*\*\*800

--7499 ALLEN \*\*\*\*\*\*1600

--7521 WARD \*\*\*\*\*\*1250

--…….

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

-- EMPNO ENAME 입사일

-- 7369 SMITH 1980-12-17

-- ….

--13. 사원 테이블에서 부서 번호가 20인 사원의 사번, 이름, 직무, 급여를 출력하시오.

--(급여는 앞에 $를 삽입하고3자리마다 ,를 출력한다)