**1. SQL 함수**

제공되는 함수들은 기본적인 Query문을 더욱 강력하게 해주고 데이터 값을 조작하는데 사용된다. 여러분은 단일 행 함수를 이용하여 문자,숫자,날짜 함수에 대해 살펴볼 뿐만 아니라 형을 전환하는 함수들에 대해서도 살펴본다. 또한 복수 행 함수를 이용하여 복수의 행 조합하여 그룹 당 하나의 결과를 출력하는 그룹 함수에 대해서 살펴본다.

**1.1 SQL함수의 특징 및 이점**

1. 데이터에 계산을 수행할 수 있다.
2. 개별적인 데이터 항목을 수정할 수 있다.
3. 행의 그룹에 대해 결과를 조작할 수 있다.
4. 출력을 위한 날짜와 숫자 형식을 조절할 수 있다.
5. 열의 자료형을 변환할 수 있다.

**1.2 단일 행 함수(Single Row Function)**

이 함수는 단일 행에 대해서만 적용 가능하고 행별로 하나의 결과를 RETURN한다..

|  |
| --- |
| **Function\_name** (column | expression [ ,arg1,arg2, . . . . ]) |

function\_name 함수 명

column 데이터 베이스의 Column Name

expression 어떤 문자 스트링이거나 계산된 표현식

arg1,arg2 함수에 의해 사용될 수 있는 인수

**1.2.1 단일 행 함수가 이용되는 곳**

1. 데이터에 대해 계산을 수행할 경우
2. 각각의 데이터 항목을 변경할 경우
3. 출력할 날짜 형식을 변경할 경우
4. Column Data Type을 변경할 경우

**1.2.2 단일 행 함수의 종류**

1. 문자형 함수 : 문자를 입력 받고 문자와 숫자 값 모두를 RETURN할 수 있다.
2. 숫자형 함수 : 숫자를 입력 받고 숫자를 RETURN한다.
3. 날짜형 함수 : 날짜형에 대해 수행하고 숫자를 RETURN하는 MONTHS\_BETWEEN 함수를 제외하고 모두 날짜 데이터형의 값을 RETURN한다.
4. 변환형 함수 : 어떤 데이터형의 값을 다른 데이터형으로 변환한다.
5. 일반적인 함수 : NVL, DECODE

**1.2.3 단일 행 함수의 특징**

1. 질의에서 RETURN되는 각각의 행에 대해 수행
2. 행별로 하나의 결과를 RETURN
3. 참조 시 사용한 데이터 형과 다른 데이터 형으로 결과를 RETURN할 수 있다,
4. 하나 이상의 인수를 필요로 한다.
5. SELECT,WHERE,ORDER BY절에서 사용할 수 있습니다.
6. 함수를 중첩할 수 있습니다.
7. 단일 행 함수들은 여러 LEVEL에 걸쳐 중첩 사용이 가능하다.
8. 중첩된 함수들은 가장 하위 LEVEL에서 가장 상위 LEVEL 순으로 진행된다.

**1.3 문자형 함수(Character Function)**

|  |  |  |
| --- | --- | --- |
| 종 류 | 함 수 | 사 용 목 적 |
| 변환 함수 | LOWER | 알파벳 값을 소문자로 변환 |
| UPPER | 알파벳 값을 대문자로 변환 |
| INITCAP | 첫번째 글자만 대문자로 변환 |
| 문자 조작 함수 | CONCAT | 두 문자열을 연결(합성) |
| SUBSTR | 문자열 중 특정 문자 또는 문자열의 일부분을 선택 |
| LENGTH | 문자열의 길이를 구함 |
| INSTR | 명명된 문자의 위치를 구함 |
| LPAD | 왼쪽 문자 자리 채움 |
| RPAD | 오른쪽 문자 자리 채움 |
| LTRIM | 왼쪽 문자를 지움 |
| RTRIM | 오른쪽 문자를 지움 |
| TRANSLATE | 특정 문자열을 대체 |
| REPLACE | 특정 문자열을 대신 |

**1.3.1 LOWER함수**

대소문자가 혼합되어 있거나 대문자인 문자열을 소문자로 변환 합니다.

|  |  |
| --- | --- |
| Syntax | LOWER( column | expression) |
| 사 용 예 | LOWER(‘MANAGER’) → manager |

문제1) EMP 테이블에서 scott의 정보를 사원번호,성명,담당업무(소문자로),부서번호를 출력하여라.

|  |
| --- |
| SQL> **SELECT empno,ename,LOWER(job),deptno**  2 **FROM emp**  3 **WHERE LOWER(ename) = 'scott';**  EMPNO ENAME LOWER(JOB DEPTNO  --------- ---------- --------- ---------  7788 SCOTT analyst 20 |

**1.3.2 UPPER 함수**

대문자가 혼합되어 있거나 소문자인 문자열을 대문자로 변환 합니다.

|  |  |
| --- | --- |
| Syntax | UPPER( column | expression) |
| 사 용 예 | UPPER(‘manager’) → MANAGER |

문제2) EMP 테이블에서 scott의 정보를 사원번호,성명,담당업무,부서번호를 출력하여라.

|  |
| --- |
| SQL> **SELECT empno,ename,job,deptno**  2 **FROM emp**  3 **WHERE ename = UPPER('scott');**  EMPNO ENAME JOB DEPTNO  --------- ---------- --------- ---------  7788 SCOTT ANALYST 20 |

**1.3.3 INITCAP 함수**

각 단어의 첫번째 문자를 대문자로 나머지 문자는 소문자로 변경합니다.

|  |  |
| --- | --- |
| Syntax | INITCAP( column | expression) |
| 사 용 예 | INITCAP(‘ORACLE SERVER’) → Oracle Server |

문제3) DEPT 테이블에서 첫 글자만 대문자로 변환하여 모든 정보를 출력하여라.

|  |
| --- |
| 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 |

**1.3.4 CONCAT 함수**

두 개의 문자열을 합성합니다. CONCAT는 두개의 매개변수만 사용 가능합니다.

|  |  |
| --- | --- |
| Syntax | CONCAT( column1 | expression1, column2 | expression2) |
| 사 용 예 | INITCAP(‘ORACLE’ ,‘SERVER’) → ORACLESERVER |

문제4) 두개의 SELECT문이 있다. 결과의 차이점을 설명하여라

|  |
| --- |
| SQL> **col e\_name format a15**  SQL> **col e\_empno format a15**  SQL> **col e\_job format a15**  SQL> **SELECT empno,ename,job,CONCAT(empno,ename) e\_name,**  2 **CONCAT(ename,empno) e\_empno,**  3 **CONCAT(ename,job) e\_job**  4 **FROM emp**  5 **WHERE deptno = 10;**  EMPNO ENAME JOB E\_NAME E\_EMPNO E\_JOB  --------- ---------- --------- --------------- --------------- -------------  7839 KING PRESIDENT 7839KING KING7839 KINGPRESIDENT  7782 CLARK MANAGER 7782CLARK CLARK7782 CLARKMANAGER  7934 MILLER CLERK 7934MILLER MILLER7934 MILLERCLERK |
| SQL> **col no format 99**  SQL> **col d\_name format a18**  SQL> **col d\_deptno format a18**  SQL> **col d\_loc format a25**  SQL> **SELECT deptno no, dname, loc, CONCAT(deptno,dname) d\_name,**  2 **CONCAT(dname,deptno) d\_deptno, CONCAT(dname,loc) d\_loc**  3 **FROM dept;**    NO DNAME LOC D\_NAME D\_DEPTNO D\_LOC  --- ----------- --------- ------------- ---------------- ----------------------  10 ACCOUNTING NEW YORK 10ACCOUNTING ACCOUNTING 10 ACCOUNTING NEW YORK  20 RESEARCH DALLAS 20RESEARCH RESEARCH 20 RESEARCH DALLAS  30 SALES CHICAGO 30SALES SALES 30 SALES CHICAGO  40 OPERATIONS BOSTON 40OPERATIONS OPERATIONS 40 OPERATIONS BOSTON |

♣ 참고

Column의 데이터 타입이 varchar2, number, char의 차이로 varchar2와 number는 가변 길이, char는 고정 길이입니다.

**1.3.5 SUBSTR 함수**

지정된 길이만큼의 문자열을 추출합니다.

|  |  |
| --- | --- |
| Syntax | SUBSTR( column | expression, m [,n]) |
| 사 용 예 | SUBSTR(‘000101-3234232’, 8, 1) → 3 |

문제5) EMP 테이블에서 이름의 첫글자가 ‘K’ 보다 크고 ‘Y’보다 적은 사원의 정보를 사원번호, 이름, 업무, 급여, 부서번호를 출력하여라. 단 이름순으로 정렬하여라.

|  |
| --- |
| SQL> **SELECT empno,ename,job,sal,deptno**  2  **FROM emp**  3 **WHERE SUBSTR(ename,1,1) > 'K' AND SUBSTR(ename,1,1) < 'Y'**  4 **ORDER BY ename;**  EMPNO ENAME JOB SAL DEPTNO  --------- ---------- --------- --------- ---------  7654 MARTIN SALESMAN 1250 30  7934 MILLER CLERK 1300 10  7788 SCOTT ANALYST 3000 20  7369 SMITH CLERK 800 20  7844 TURNER SALESMAN 1500 30  7521 WARD SALESMAN 1250 30  6 rows selected. |

**1.3.6 LENGTH 함수**

문자열의 길이를 숫자 값으로 RETURN한다.

|  |  |
| --- | --- |
| Syntax | LENGTH( column | expression ) |
| 사 용 예 | INITCAP(‘000101-3234232’) → 14 |

문제6) EMP 테이블에서 20번 부서 중 이름의 길이 및 급여의 자릿수를 사원번호, 이름, 이름의 자릿수, 급여, 급여의 자릿수를 출력하여라.

|  |
| --- |
| 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  7788 SCOTT 5 3000 4  7876 ADAMS 5 1100 4 |

**1.3.7 INSTR 함수**

명명된 문자의 위치를 숫자 값으로 RETURN한다.

|  |  |
| --- | --- |
| Syntax | INSTR( column | expression, m[,n]) |
| 사 용 예 | INSTR(‘MILLER’, ‘L’, 1, 2) → 4 |

문제7) EMP 테이블에서 이름 중 ‘L’자의 위치를 출력하여라.

|  |
| --- |
| 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  . . . . . . . . . .  14 rows selected. |

문제8) 파일명을 입력을 입력받아 확장자가 없으면 .SQL을 붙여 출력하여라.

|  |
| --- |
| SET VERIFY OFF  SET SERVEROUTPUT ON  ACCEPT p\_filename PROMPT '파일명을 입력하시오 : '  DECLARE  v\_filename VARCHAR2(300) := '&p\_filename';  BEGIN  IF INSTR(v\_filename,'.',1,1) = 0 THEN  DBMS\_OUTPUT.PUT\_LINE('FILE NAME : ' || v\_filename || '.SQL');  ELSIF INSTR(v\_filename,'.',1,1) >= 1 THEN  DBMS\_OUTPUT.PUT\_LINE('FILE NAME : ' || v\_filename);  END IF;  END;  /  SET VERIFY ON  SET SERVEROUTPUT OFF |

**1.3.8 LPAD함수**

문자값을 우측부터 채웁니다.

|  |  |
| --- | --- |
| Syntax | LPAD(column | expression, n, ’string’) |
| 사 용 예 | LPAD(‘MILLER’, 10, ‘\*’) → \*\*\*\*MILLER |

문제9) 아래 두 문장의 결과를 보고 차이점을 설명하여라.

|  |
| --- |
| 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 |
| SQL> **SELECT deptno,dname,LPAD(dname,20,'\*')**  2 **FROM dept;**  DEPTNO DNAME LPAD(DNAME,20,'\*')  --------- -------------- --------------------  10 ACCOUNTING \*\*\*\*\*\*ACCOUNTING  20 RESEARCH \*\*\*\*\*\*RESEARCH  30 SALES \*\*\*\*\*\*SALES  40 OPERATIONS \*\*\*\*\*\*OPERATIONS |

**1.3.9 RPAD함수**

문자값을 좌측부터 채웁니다.

|  |  |
| --- | --- |
| Syntax | RPAD(column | expression, n, ’string’) |
| 사 용 예 | RPAD(‘MILLER’, 10, ‘\*’) → MILLER\*\*\*\* |

문제10) 아래 두 문장의 결과를 보고 차이점을 설명하여라.

|  |
| --- |
| 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\*\*\*\*\*\* |
| SQL> **SELECT deptno,dname,RPAD(dname,20,'\*')**  2 **FROM dept;**  DEPTNO DNAME RPAD(DNAME,20,'\*')  --------- -------------- -----------------------------------------  10 ACCOUNTING ACCOUNTING \*\*\*\*\*\*  20 RESEARCH RESEARCH \*\*\*\*\*\*  30 SALES SALES \*\*\*\*\*\*  40 OPERATIONS OPERATIONS \*\*\*\*\*\* |

**1.3.10 LTRIM함수**

왼쪽 문자를 지우는 함수 입니다.

|  |  |
| --- | --- |
| Syntax | LTRIM(column1 | expression1, column1 | expression1) |
| 사 용 예 | LTRIM(‘MILLER’, ‘M’) → ILLER |

문제11) EMP 테이블에서 10번 부서에 대하여 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)  ---------- --------- --------- --------- ----------------  KING PRESIDENT PRESIDENT 5000 5000  BLAKE MANAGER MANAGER 2850 2850  CLARK MANAGER MANAGER 2450 2450  JONES MANAGER MANAGER 2975 2975  MARTIN SALESMAN SALESMAN 1250 250  ALLEN SALESMAN SALESMAN 1600 600  TURNER SALESMAN SALESMAN 1500 500  JAMES CLERK CLERK 950 950  WARD SALESMAN SALESMAN 1250 250  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  14 rows selected. |

**1.3.11 RTRIM함수**

오른쪽 문자를 지우는 함수 입니다.

|  |  |
| --- | --- |
| Syntax | RTRIM(column1 | expression1,column2 | expression2) |
| 사 용 예 | RTRIM(‘MILLER’, ‘R’) → MILLE |

문제12) EMP 테이블에서 10번 부서에 대하여 담당 업무 중 우측에 ‘T’를 삭제하고 급여 중 우측의 0을 삭제하여 출력하여라.

|  |
| --- |
| 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 |

**1.3.11 TRANSLATE 함수**

특정 문자열을 대체하는 함수 입니다. 즉 str1을 str2 문자로 대체하는 함수이다.

|  |  |
| --- | --- |
| Syntax | TRANSLATE(column1 | expression1, ‘string1’, ‘string2’) |
| 사 용 예 | TRANSLATE(‘MILLER’, ‘L’, ‘\*’) → MI\*\*ER |

문제13) EMP 테이블에서 성명을 소문자로 바꾸어 출력하여라.

|  |
| --- |
| SQL> **var u\_lower varchar2(10)**  SQL> **var n\_h varchar2(10)**  SQL> **col u\_lower format a10**  SQL> **col n\_h format a10**  SQL>  SQL> **SELECT empno,ename,TRANSLATE(ename,'ABCDEFGHIJKLMNOPQRSTUVWXYZ',**  2 **'abcdefghijklmnopqrstuvwxyz') u\_lower,**  3  **sal,TRANSLATE(sal,'0123456789',**  4  **'영일이삼사오육칠팔구') n\_h**  5 **FROM emp**  6  **WHERE deptno = 10;**  EMPNO ENAME U\_LOWER SAL N\_H  --------- ---------- ---------- --------- ----------  7839 KING king 5000 오영영영  7782 CLARK clark 2450 이사오영  7934 MILLER miller 1300 일삼영영 |

**1.3.11 REPLACE 함수**

특정 문자열을 대신하는 함수 입니다.

|  |  |
| --- | --- |
| Syntax | REPLACE(column1 | expression1, ‘string1’, ‘string2’) |
| 사 용 예 | REPLACE(‘JACK and JUE’, ‘J’, ‘BL’) → BLACK and BLUE |

문제14) EMP 테이블에서 JOB에 ‘A’를 ‘$’로 바꾸어 출력하여라.

|  |
| --- |
| SQL> **SELECT ename,job,REPLACE(job,'A','$'),sal**  2 **FROM emp;**  ENAME JOB REPLACE(J SAL  ---------- --------- --------- ---------  KING PRESIDENT PRESIDENT 5000  BLAKE MANAGER M$N$GER 2850  CLARK MANAGER M$N$GER 2450  JONES MANAGER M$N$GER 2975  MARTIN SALESMAN S$LESM$N 1250  . . . . . . .  14 rows selected. |

**1.4 숫자형 함수**

|  |  |
| --- | --- |
| 함 수 | 사 용 목 적 |
| ROUND | 숫자를 반올림 |
| TRUNC | 숫자를 절삭 |
| MOD | 나머지를 구함 |
| POWER | 거듭제곱 |
| SQRT | 제곱근 |
| SIGN | 양수, 음수,0인지를 구분 |
| CHR | ASCII값에 해당하는 문자를 구함 |

**1.4.1 ROUND 함수**

명시된 소수점으로 반올림하는 함수입니다. 숫자를 n자리까지 반올림한다. n이 양수이면 소수 자리를, 음수이면 정수 자리를 사사오입합니다. 생략할 수 있으며 Default는 0입니다.

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

문제15) 다음의 결과를 분석하여라.

|  |
| --- |
| SQL> **SELECT ROUND(4567.678),ROUND(4567.678,0),**  2 **ROUND(4567.678,2),ROUND(4567.678,-2)**  3 **FROM dual;**  ROUND(4567.678) ROUND(4567.678,0) ROUND(4567.678,2) ROUND(4567.678,-2)  --------------- ----------------- ----------------- ------------------  4568 4568 4567.68 4600 |

♣ 참고

DUAL 테이블은 SYS User가 Owner이며 모든 사용자가 사용할 수 있도록 권한을 부여하였다. Dummy라는 하나의 Column과 X값을 가지는 하나의 행을 포함합니다. DUAL 테이블은 오직 하나의 값을 출력하고자 할 때 유용합니다. 예를 들어 데이터를 가진 테이블에서 파생되지 않은 상수, 의사열, 표현식의 값인 경우 입니다. 즉 임의의 값을 알고자 할 경우 유용하게 사용할 수 있다. 위 SELECT문장에서 dual이 아닌 dept를 사용하면 결과는 어떻게 될까?

**1.4.2 TRUNC 함수**

명시된 숫자를 절삭하는 함수입니다. 숫자를 n자리까지 절삭한다. n이 양수이면 소수 자리를, 음수이면 정수 자리를 절삭합니다. 생략할 수 있으며 Default는 0입니다.

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

문제16) 다음의 결과를 분석하여라.

|  |
| --- |
| SQL> **SELECT TRUNC(4567.678),TRUNC(4567.678,0),**  2 **TRUNC(4567.678,2),TRUNC(4567.678,-2)**  3 **FROM dual;**  TRUNC(4567.678) TRUNC(4567.678,0) TRUNC(4567.678,2) TRUNC(4567.678,-2)  --------------- ----------------- ----------------- ------------------  4567 4567 4567.67 4500 |

**1.4.3 MOD 함수**

숫자의 나머지를 구하는 함수입니다.

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

문제17) EMP 테이블에서 급여를 30으로 나눈 나머지를 구하여 출력하여라.

|  |
| --- |
| SQL> **SELECT sal, MOD(sal,30)**  2 **FROM emp**  3 **WHERE deptno = 10;**  SAL MOD(SAL,30)  --------- -----------  5000 20  2450 20  1300 10 |

**1.4.4 POWER 함수**

거듭제곱을 구하는 함수 입니다.

|  |  |
| --- | --- |
| Syntax | POWER(column1 | expression1 , n) |
| 사 용 예 | POWER(2, 3) → 8 |

**1.4.5 SQRT 함수**

제곱근을 구하는 함수 입니다.

|  |  |
| --- | --- |
| Syntax | SQRT( column1 | expression1 ) |
| 사 용 예 | SQRT(4) → 2 |

**1.4.6 SIGN 함수**

주어진 숫자가 양수인지 음수인지 또는 0인지를 구하는 함수 입니다.

|  |  |
| --- | --- |
| Syntax | SIGN( column1 | expression1 ) |
| 사 용 예 | SIGN(100) → 1 |

**1.4.7 CHR 함수**

ASCII Code값에 해당하는 문자를 구하는 함수 입니다.

|  |  |
| --- | --- |
| Syntax | CHR( column1 | expression1 ) |
| 사 용 예 | CHR(65) → A |

문제18) EMP 테이블에서 20번 부서 중 이름과 담당 업무를 연결하여 출력하여라. 단 담당 업무를 한 줄 아래로 출력하여라

|  |
| --- |
| SQL> **SELECT empno,ename,job,ename || CHR(10) || job**  2 **FROM emp**  3 **WHERE deptno = 20;**  EMPNO ENAME JOB ENAME||CHR(10)||JOB  --------- ---------- --------- --------------------  7566 JONES MANAGER JONES  MANAGER  7902 FORD ANALYST FORD  ANALYST  7369 SMITH CLERK SMITH  CLERK  7788 SCOTT ANALYST SCOTT  ANALYST  7876 ADAMS CLERK ADAMS  CLERK |

**1.5 날짜형 함수**

**1.5.1 오라클 날짜 형식**

1. 오라클은 세기,년,월,일,시,분,초를 내부 숫자(7 Byte) 형식으로 날짜를 저장 합니다.
2. Default Date Type은 DD-MON-YY(변경 가능)입니다.
3. 오라클 날짜의 범위는 B.C 4712년 1월 1일부터 A.D 9999년 12월 31일 사이입니다.
4. SYSDATE는 오라클이 설치되어 있는 서버의 현재 날짜와 시간을 RETURN하는 함수 입니다.

**1.5.2 날짜 연산**

1. 날짜에서 숫자를 더하거나 빼어 날짜 결과를 출력
2. 날짜 사이의 일수를 알기 위해서 두개의 날짜를 뺍니다.
3. 시간을 24로 나누어서 시간을 날짜에 더합니다.

|  |  |  |
| --- | --- | --- |
| 날 짜 연 산 | 결 과 | 설 명 |
| Date + Number | Date | 일수를 날짜에 더합니다. |
| Date - Number | Date | 날짜에서 일수를 뺍니다. |
| Date - Date | 일수 | 어떤 날짜에서 다른 날짜를 뺍니다 |
| Date + Number / 24 | Date | 시간을 날짜에 더합니다. |

문제19) EMP 테이블에서 현재까지 근무일 수가 몇주 몇일 인가를 출력하여라. 단 근무 일수가 많은 사람 순으로 출력하여라.

|  |
| --- |
| 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;**  ENAME HIREDATE SYSDATE Total Days WEEKS DAYS  ---------- ---------------- ---------------- ---------- --------- ---------  SMITH 17-DEC-80 01-MAR-99 6648.5677 949 6  ALLEN 20-FEB-81 01-MAR-99 6583.5677 940 4  WARD 22-FEB-81 01-MAR-99 6581.5677 940 2  JONES 02-APR-81 01-MAR-99 6542.5677 934 5  BLAKE 01-MAY-81 01-MAR-99 6513.5677 930 4  CLARK 09-JUN-81 01-MAR-99 6474.5677 924 7  TURNER 08-SEP-81 01-MAR-99 6383.5677 911 7  MARTIN 28-SEP-81 01-MAR-99 6363.5677 909 1  KING 17-NOV-81 01-MAR-99 6313.5677 901 7  JAMES 03-DEC-81 01-MAR-99 6297.5677 899 5  FORD 03-DEC-81 01-MAR-99 6297.5677 899 5  . . . . . . . . . .  14 rows selected. |

**1.5.3 날짜 함수**

날짜 함수는 오라클 날짜에 대해 연산을 합니다. 모든 날짜 함수는 숫자값을 RETURN하는데 MONTHS\_BETWEEN을 제외하고는 DATE형을 RETURN합니다

|  |  |
| --- | --- |
| 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. |

**1.5.4 MONTHS\_BETWEEN 함수**

1. 날짜와 날짜 사이의 월수를 계산합니다
2. 결과는 음수 또는 양수가 될 수 있습니다.
3. 결과의 비정수 부분을 월의 부분을 나타냅니다.

|  |  |
| --- | --- |
| Syntax | MONTHS\_BETWEEN(date1, date2) |
| 사 용 예 | MONTHS\_BETWEEN(sysdate,hiredate) → 212.04794 |

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

문제20) EMP 테이블에서 10번 부서 중 현재까지의 근무 월수를 계산하여 출력하여라.

|  |
| --- |
| 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;**  ENAME HIREDATE SYSDATE M\_BETWEEN T\_BETWEEN  ---------- ------------------ ------------------ --------- ---------  CLARK 09-JUN-81 10-FEB-99 212.04812 212  KING 17-NOV-81 10-FEB-99 206.79005 206  MILLER 23-JAN-82 10-FEB-99 204.5965 204 |

**1.5.5 ADD\_MONTHS 함수**

1. 날짜에 월을 더합니다(ADD\_MONTHS(hiredate,10))
2. 날짜에 월을 뺍니다(ADD\_MONTHS(hiredate,-10))
3. 결과의 날짜형입니다.

|  |  |
| --- | --- |
| Syntax | ADD\_MONTHS(date1, n) |
| 사 용 예 | ADD\_MONTHS(hiredate,5) → 23-JUN-82 |

문제21) EMP 테이블에서 10번 부서 중 입사 일자로부터 5개월이 지난 후 날짜를 계산하여 출력하여라.

|  |
| --- |
| SQL> **SELECT ename,hiredate,ADD\_MONTHS(hiredate,5) a\_month**  2 **FROM emp**  3 **WHERE deptno = 10**  4 **ORDER BY hiredate DESC;**  ENAME HIREDATE A\_MONTH  ---------- ------------------ ------------------  MILLER 23-JAN-82 23-JUN-82  KING 17-NOV-81 17-APR-82  CLARK 09-JUN-81 09-NOV-81 |

**1.5.6 NEXT\_DAY 함수**

1. 명시된 요일의 돌아오는 날짜를 계산 합니다.
2. 요일이 아니라 숫자도 기술 가능(SUNDAY:1, MONDAY:2, . . . . .)
3. NLS\_LANG이 KOREAN\_KOREA.KO16KSC5601로 되어 있으면 한글도 사용 가능(일요일,월요일,화요일, . . . . . )

|  |  |
| --- | --- |
| Syntax | NEXT\_DAY(date1, ‘string’ | n ) |
| 사 용 예 | NEXT\_DAY(hiredate,’FRIDAY’) → 29-JAN-82  NEXT\_DAY(hiredate,’금요일’) → 29-JAN-82 |

문제22) EMP 테이블에서 10번 부서 중 입사 일자로부터 돌아오는 금요일을 계산하여 출력하여라.

|  |
| --- |
| SQL> **SELECT ename,hiredate,NEXT\_DAY(hiredate,'FRIDAY') n\_day,**  2  **NEXT\_DAY(hiredate,6) n\_6,NEXT\_DAY(hiredate,7) n\_7**  3 **FROM emp**  4 **WHERE deptno = 10**  5 **ORDER BY hiredate DESC;**  ENAME HIREDATE N\_DAY N\_6 N\_7  ---------- ------------------ ------------------ ------------------ ----------  MILLER 23-JAN-82 29-JAN-82 29-JAN-82 30-JAN-82  KING 17-NOV-81 20-NOV-81 20-NOV-81 21-NOV-81  CLARK 09-JUN-81 12-JUN-81 12-JUN-81 13-JUN-81 |

**1.5.7 LAST\_DAY 함수**

1. 월의 마지막 날짜를 계산
2. 윤년, 평년은 자동 계산

|  |  |
| --- | --- |
| Syntax | LAST\_DAY(date1) |
| 사 용 예 | LAST\_DAY(hiredate) → 30-NOV-81 |

문제23) EMP 테이블에서 입사한 달의 근무 일수를 계산하여 출력하여라. 단 토요일과 일요일도 근무 일수에 포함한다.

|  |
| --- |
| SQL> **SELECT empno,ename,hiredate,LAST\_DAY(hiredate) l\_last,**  2 **LAST\_DAY(hiredate) - hiredate l\_day**  3 **FROM emp**  4 **ORDER BY LAST\_DAY(hiredate) - hiredate DESC;**  EMPNO ENAME HIREDATE L\_LAST L\_DAY  --------- ---------- ------------------ ------------------ ---------  7698 BLAKE 01-MAY-81 31-MAY-81 30  7566 JONES 02-APR-81 30-APR-81 28  7900 JAMES 03-DEC-81 31-DEC-81 28  7902 FORD 03-DEC-81 31-DEC-81 28  7844 TURNER 08-SEP-81 30-SEP-81 22  7788 SCOTT 09-DEC-82 31-DEC-82 22  7782 CLARK 09-JUN-81 30-JUN-81 21  7876 ADAMS 12-JAN-83 31-JAN-83 19  7369 SMITH 17-DEC-80 31-DEC-80 14  7839 KING 17-NOV-81 30-NOV-81 13  . . . . . . . . . .  14 rows selected. |

**1.5.8 ROUND 함수**

1. 명시된 형식으로 반올림 합니다.
2. 날짜를 가장 가까운 년도 또는 월로 반올림할 수 있습니다.
3. fmt에 명시된 단위에 대해 반올림한 날짜를 계산
4. fmt가 생략되면 날짜를 가장 가까운 날짜로 반올림한다.

|  |  |
| --- | --- |
| Syntax | ROUND(date1 [,fmt] ) |
| 사 용 예 | ROUND(‘25-JUN-99’,’MONTH’) → 01-AUG-99  ROUND(‘25-JUN-98’,’YEAR’) → 01-JAN-99 |

**1.5.9 TRUNC 함수**

1. 명시된 형식으로 절삭 합니다.
2. 날짜를 가장 가까운 년도 또는 월로 절삭할 수 있습니다.
3. fmt에 명시된 단위에 대해 절삭한 날짜를 계산
4. fmt가 생략되면 날짜를 가장 가까운 날짜로 절삭한다.

|  |  |
| --- | --- |
| Syntax | TRUNC(date1 [,fmt] ) |
| 사 용 예 | TRUNC(‘25-JUN-99’,’MONTH’) → 01-JUN-99  TRUNC(‘25-JUN-98’,’YEAR’) → 01-JAN-98 |

문제24) EMP 테이블에서 10번 부서 중 입사한 달의 ROUND과 TRUNC 함수를 비교합니다.

|  |
| --- |
| SQL> **SELECT ename,hiredate,ROUND(hiredate,'MONTH') m\_round,**  2 **TRUNC(hiredate,'MONTH') m\_trunc, ROUND(hiredate,'YEAR') y\_round,**  3 **TRUNC(hiredate,'YEAR') y\_trunc**  4 **FROM emp**  5 **WHERE deptno = 10**  6 **ORDER BY hiredate DESC;**  ENAME HIREDATE M\_ROUND M\_TRUNC Y\_ROUND Y\_TRUNC  ---------- ----------- ----------- ---------- ----------- -----------  MILLER 23-JAN-82 01-FEB-82 01-JAN-82 01-JAN-82 01-JAN-82  KING 17-NOV-81 01-DEC-81 01-NOV-81 01-JAN-82 01-JAN-81  CLARK 09-JUN-81 01-JUN-81 01-JUN-81 01-JAN-81 01-JAN-81 |

**1.6 변환 함수**

**1.6.1 데이터의 형 변환**

오라클 서버는 어떤 일정한 데이터형의 데이터를 사용해야 하는 곳에, 그것과 다른 데이터형의 데이터를 사용할 수 있게 합니다. 이것은 오라클 서버가 자동적으로 데이터형을 변환할 수 있을 때 허용됩니다. 이 데이터형 변환은 오라클 서버에 의해서 암시적으로 행해지거나 또는 사용자에 의해서 명시적으로 행해질 수 있습니다.

**가) 암시적인 데이터형 변환**

값 할당(assignment)시,오라클 서버는 다음을 자동으로 변환할 수 있습니다.

|  |  |
| --- | --- |
| FROM | TO |
| VARCHAR2 or CHAR | NUMBER |
| VARCHAR2 or CHAR | DATE |
| NUMBER | VARCHAR2 |
| DATE | VARCHAR2 |

♣ 참고

오라클 서버가 값 할당(assignment) 문장에서 사용된 값의 데이터형을 목표(target)값의 데이터형으로 변환할 수 있을 경우에 할당(assignment) 문장은 올바로 수행됩니다. 또한 CHAR가 NUMBER로의 변환은 문자열이 적절한 숫자로 나타낼수 있을 경우에만 가능하고 CHAR가 DATE로의 변환은 문자열이 Default Date Type와 같을 경우에만 성공합니다.

☞ Guidelines

비록 암시적 데이터형 변환을 이용할 수 있더라도,SQL문장의 안정성을 위해서 명시적 데이터형 변환을 할 것을 권장합니다.

**나) 명시적인 데이터형 변환**

SQL은 변환 함수를 통하여 어떤 데이터형의 값을 다른 데이터형의 값으로 변환하기 위하여 아래의 함수를 제공 합니다.

|  |  |
| --- | --- |
| 함 수 | 사 용 목 적 |
| TO\_CHAR | 숫자나 문자값을 지정한 형식의 VARCHAR2문자열로 변환 합니다. |
| TO\_NUMBER | 숫자를 포함하는 문자열을 숫자로 변환 합니다. |
| TO\_DATE | 날짜를 나타내는 문자열을 명시된 날짜로 변환 합니다. |

**1.6.2 TO\_CHAR 함수**

숫자,날짜,문자열을 지정한 형식의 VARCHAR2 문자열로 변환하는 함수입니다.

**1) 날짜 형식을 변환**

|  |  |
| --- | --- |
| Syntax | TO\_CHAR( date, ‘fmt’ ) |
| 사 용 예 | TO\_CHAR(hiredate, ‘YY/MM/DD’) → 81/11/17 |

**가) 특정 형식으로 날짜를 출력**

이전의 모든 날짜 형식은 DD-MON-YY형식이었다. TO\_CHAR함수는 이러한 형식의 날짜를 명시한 날짜 형식으로 변환하여 출력할 수 있다.

☞ Guidelines

1. 포맷(fmt) 모델은 단일 인용 부호로 둘러 싸여 있어야 하고 대소문자를 구분한다.
2. 포맷(fmt) 모델은 어떤 타당한 날짜 형식도 포함 가능하다.
3. 추가된 공백을 제거하거나 앞부분의0을 없애기 위해서 “fm”요소를 사용한다.
4. SQL\*Plus COLUMN명령어로 문자 필드 결과의 출력 폭의 크기를 조절할 수 있다.(DEFAULT는 80)

**나) 날짜 형식 모델**

|  |  |
| --- | --- |
| 구 성 요 소 | 설 명 |
| SCC or CC | 세기;BC날짜에는 \_S를 붙입니다. |
| Years in dates YYYY or SYYYY | 년;BC날짜에는 \_S를 붙입니다. |
| YYY or YY or Y | 년의 마지막3,2또는1자리 수 |
| Y,YYY | 콤마가 있는 년 |
| IYYY,IYY,IY,I | ISO표준에 바탕을 둔4,3,2또는 1자리 수 |
| SYESR or YEAR | 문자고 표현된 년;BC날짜에는 \_S를 붙입니다. |
| BC or AD | BC/AD지시자 |
| B.C or A.D | .이 있는 BC/AD지시자 |
| Q | 년의 4분의1 |
| MM | 두자리 값의 월 |
| MONTH | 9자리를 위해 공백을 추가한 월 이름 |
| MON | 세 자리의 약어로 된 월 이름 |
| RM | 로마 숫자 월 |
| WW or W | 년이나 월의 주 |
| DDD or DD or D | 년,월 또는 주의 일 |
| DAY | 9자리를 위해 공백을 추가한 요일 이름 |
| DY | 세 자리 약어로된 요일 이름 |
| J | Julian day;BC4713년12월 31일 이후의 요일 수 |

**다) 시간 형식**

1. 시간 요소는 날짜의 시간 부분을 형식화(HH24:MI:SS AM → 15:34:32 PM)
2. 문자열에 이중 인용 부호를 사용하여 문자열을 추가(DD “of” MONTH→10 of OCTOBER)
3. 숫자 접미사는 숫자를 문자로 변환(ddspth → fourteenth)
4. 시간 형식의 종류

|  |  |
| --- | --- |
| 요 소 | 설 명 |
| AM or PM | 정오 지시자 |
| A.M or P.M | .이 있는 정오 지시자 |
| HH or HH12 or HH24 | 하루 중 시간(1-12, 0-23) |
| MI | 분(0-59) |
| SS | 초(0-59) |
| SSSSS | 자정 이후의 초(0-86399) |

**라) 기타 형식**

|  |  |
| --- | --- |
| 요 소 | 설 명 |
| / . , | 사용 문자가 결과에 다시 나타난다. |
| “of the” | 인용 부호내의 문자가 결과에 출력 |

**마) 숫자에 영향을 주는 접미사**

|  |  |
| --- | --- |
| 요 소 | 설 명 |
| TH | 서수(DDTH → 4TH) |
| SP | 명시한 수(DDSP → FOUR) |
| SPTH or THSP | 명시한 서수(DDSPTH → FOURTH) |

문제25) EMP 테이블에서 10번 부서 중 입사 일자를 ‘1 May 1981’와 ‘1998년 1월 1일’의 형태로 출력하여라

|  |
| --- |
| SQL> **var t\_hiredate varchar2(30)**  SQL> **var t\_kor varchar2(20)**  SQL> **col t\_hiredate format a30**  SQL> **col t\_kor format a20**  SQL> **SELECT ename,hiredate,TO\_CHAR(hiredate, 'fmDD Month YYYY') t\_hiredate,**  2 **TO\_CHAR(hiredate, 'YYYY"년" MM"월" DD"일"') t\_kor**  3  **FROM emp**  4 **WHERE deptno = 10**  5 **ORDER BY hiredate DESC;**  ENAME HIREDATE T\_HIREDATE T\_KOR  ---------- ------------------ ------------------------------ --------------------  MILLER 23-JAN-82 23 January 1982 1982년 01월 23일  KING 17-NOV-81 17 November 1981 1981년 11월 17일  CLARK 09-JUN-81 9 June 1981 1981년 06월 09일 |

**2) 숫자 형식을 변환**

TO\_CHAR함수를 사용하여 숫자 값을 문자로 출력하기 위해 사용한다.

|  |  |
| --- | --- |
| Syntax | TO\_CHAR( number, ‘fmt’ ) |
| 사 용 예 | TO\_CHAR(sal, ‘$999,999’) → $3,000 |

**가) 숫자를 가진 TO\_CHAR함수**

1. 숫자 값을 문자로 변환할 때 즉 NUMBER형을 VARCHAR2로 전환할 때
2. 이 기법은 연결(Concatenation) 시에 유용

☞ Guidelines

1. 형식에 의해 제공되는 자릿수를 초과하는 숫자에 대해서는 “#”을 출력
2. 지정된 소수 값을 형식에서 제공하는 소수점 자리로 반올림 한다.

**나) 숫자 형식 모델**

|  |  |  |  |
| --- | --- | --- | --- |
| 요 소 | 설 명 | 예 | 결 과 |
| 9 | 9의 수는 출력 폭을 결정 | 999999 | 1234 |
| 0 | 무효의 0을 출력 | 099999 | 001234 |
| $ | 달러 기호 | $999999 | $1234 |
| L | 지역 화패 기호 | L999999 | \1234 |
| . | 명시한 위치에 소수점 | 999999.99 | 1234.00 |
| , | 명시한 위치에 콤마 | 999,999 | 1,234 |
| MI | 우측에 마이너스 기호(음수 값) | 999999MI | 1234- |
| PR | 음수를 “()”로 묶는다 | 999999PR | <1234> |
| EEEE | 과학적인 부호 표기 | 99.999EEEE | 1.234E+03 |
| V | 10을 n번 곱합니다. | 9999V99 | 123400 |
| B | 0을 0이 아닌 공백으로 출력 | B9999.99 | 1234.00 |

문제26) EMP 테이블에서 부서 20중 급여 앞에 $를 삽입하고 3자리마다 ,를 출력하여라

|  |
| --- |
| SQL> **SELECT empno,ename,job,sal,TO\_CHAR(sal,'$999,999')**  2 **FROM emp**  3 **WHERE deptno = 20**  4 **ORDER BY sal DESC;**  EMPNO ENAME JOB SAL TO\_CHAR(S  --------- ---------- --------- --------- ---------  7902 FORD ANALYST 3000 $3,000  7788 SCOTT ANALYST 3000 $3,000  7566 JONES MANAGER 2975 $2,975  7876 ADAMS CLERK 1100 $1,100  7369 SMITH CLERK 800 $800 |

**1.6.3 TO\_NUMBER 함수**

숫자를 포함하는 문자열을 숫자로 변환 합니다.

|  |  |
| --- | --- |
| Syntax | TO\_NUMBER( char ) |
| 사 용 예 | TO\_NUMBER(‘1234’) → 1234 |

**1.6.4 TO\_DATE 함수**

날짜를 나타내는 문자열을 명시된 날짜로 변환 합니다.

|  |  |
| --- | --- |
| Syntax | TO\_DATE( char [, ‘fmt’ ] ) |
| 사 용 예 | TO\_DATE(‘19990220181030’,‘YYYYMMDDHH24MISS’) →1999/02/20 18:10:30 |

문제27) February 22, 1981에 입사한 사원의 정보를 이름, 업무, 입사일자를 출력하여라.

|  |
| --- |
| SQL> **SELECT ename,job,TO\_CHAR(hiredate, 'Month DD, YYYY') t\_hire**  2  **FROM emp**  3 **WHERE hiredate = TO\_DATE('February 22, 1981','Month DD, YYYY');**  ENAME JOB T\_HIRE  ---------- --------- -------------------------------------------------  WARD SALESMAN February 22, 1981 |

**1.7 기타 함수**

**1.7.1 ECODE 함수**

CASE나 IF-THEN-ELSE-END IF문장의 조건적 조회를 가능하게 함

|  |  |
| --- | --- |
| Syntax | DECODE(col | expr,search1,result1[,search2,result2,..][,default]) |
| 사 용 예 | DECODE(deptno, 10, sal\*1.1, 20, sal\*1.5, 30, sal\*1.2, sal) |

문제28) EMP 테이블에서 JOB이 ANALYST이면 급여 증가는 10%이고 JOB이 CLERK이면 급여 증가는 15%이고 JOB이 MANAGER이면 급여 증가는 20%입니다. 다른 업무에 대해서는 급여 증가가 없습니다. 사원번호, 이름, 업무, 급여, 증가된 급여를 출력하여라.

|  |
| --- |
| SQL> **SELECT empno,ename,job,sal,DECODE(job,'ANALYST', sal\*1.1,**  2 **'CLERK', sal\*1.15,'MANAGER', sal\*1.2, sal) d\_sal**  3 **FROM emp**  4 **ORDER BY sal DESC;**  EMPNO ENAME JOB SAL D\_SAL  --------- ---------- --------- --------- ---------  7839 KING PRESIDENT 5000 5000  7902 FORD ANALYST 3000 3300  . . . . . . . . . .  14 rows selected. |

**1.8 중첩 함수**

1. 단일행 함수는 여러 LEVEL에 걸쳐 중첩 가능
2. 중첩 함수는 가장 하위 LEVEL에서 상위 LEVEL순으로 진행

|  |  |
| --- | --- |
| Syntax | F3( F2( F1(col,arg1), arg2), arg3) |
| 사 용 예 | NVL(TO\_CHAR(mgr), ‘No Manager’) |

문제28) 다음의 결과를 분석하여 보아라.

|  |
| --- |
| SQL> **col t\_rpad format a20**  SQL> **col r\_r format a20**  SQL> **SELECT deptno,dname,RPAD(dname,20,'\*') t\_rpad,**  2 **RPAD(RTRIM(dname),20,'\*') r\_r,loc**  3 **FROM dept;**    DEPTNO DNAME T\_RPAD R\_R LOC  --------- -------------- -------------------- -------------------- -------------  10 ACCOUNTING ACCOUNTING \*\*\*\*\*\* ACCOUNTING\*\*\*\*\*\*\*\*\*\* NEW YORK  20 RESEARCH RESEARCH \*\*\*\*\*\* RESEARCH\*\*\*\*\*\*\*\*\*\*\*\* DALLAS  30 SALES SALES \*\*\*\*\*\* SALES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CHICAGO  40 OPERATIONS OPERATIONS \*\*\*\*\*\* OPERATIONS\*\*\*\*\*\*\*\*\*\* BOSTON |

**◈ 연 습 문 제 ◈**

1. 현재 날짜를 출력하고 열 레이블은 Current Date로 출력하는 SELECT 문장을 기술하시오.

2. EMP 테이블에서 현재 급여에 15%가 증가된 급여를 사원번호,이름,업무,급여,증가된 급여(New Salary),증가액(Increase)를 출력하는 SELECT 문장을 기술하시오.

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

4. EMP 테이블에서 이름,입사일, 입사일로부터 현재까지의 월수,급여, 입사일부터 현재까지의 급여의 총계를 출력하는 SELECT 문장을 기술하시오.

5. EMP 테이블에서 다음의 결과가 출력되도록 작성하시오.

|  |
| --- |
| Dream Salary  ------------------------------------------------------------  KING earns $5,000.00 monthly but wants $15,000.00  BLAKE earns $2,850.00 monthly but wants $8,550.00  CLARK earns $2,450.00 monthly but wants $7,350.00  . . . . . . . . . .  14 rows selected |

6. EMP 테이블에서 모든 사원의 이름과 급여(15자리로 출력 좌측의 빈곳은 “\*”로 대치)를 출력하는 SELECT 문장을 기술하시오.

7. EMP 테이블에서 모든 사원의 정보를 이름,업무,입사일,입사한 요일을 출력하는 SELECT 문장을 기술하시오.

8. EMP 테이블에서 이름의 길이가 6자 이상인 사원의 정보를 이름,이름의 글자수,업무를 출력하는 SELECT 문장을 기술하시오.

9. EMP 테이블에서 모든 사원의 정보를 이름,업무,급여,보너스,급여+보너스를 출력하는 SELECT 문장을 기술하시오.