■STEP1. BASIC

Q001

- 다음과 같이 레코드를 정의하시오.

```
DEPTNO : 99
DNAME : DATABASE
LOC : SEOUL


PL/SQL 프로시저가 성공적으로 완료되었습니다.
```

```
DECLARE
   TYPE REC_DEPT IS RECORD(
      deptno NUMBER(2) NOT NULL := 99,
      dname DEPT.DNAME%TYPE,
      loc DEPT.LOC%TYPE
   );
   dept_rec REC_DEPT;
BEGIN
   dept_rec.deptno := 99;
   dept_rec.dname := 'DATABASE';
   dept_rec.loc := 'SEOUL';
   DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || dept_rec.deptno);
   DBMS_OUTPUT.PUT_LINE('DNAME : ' || dept_rec.dname);
   DBMS_OUTPUT.PUT_LINE('LOC : ' || dept_rec.loc);
END;
/
```

Q002

- 다음과 같이 DEPT테이블을 열과 행을 복사해 DEPT_RECORD테이블을 생성하시오.

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

```
CREATE TABLE DEPT_RECORD
    AS SELECT * FROM DEPT;

SELECT * FROM DEPT_RECORD;
```

Q003

1. 다음과 같이 REC_DEPT 레코드를 작성하시오.
2. dept_rec 레코드에
   deptno=99, dname=DATABASE , loc=SEOUL 값을 대입하시오.
3. DEPT_RECORD 테이블에 dept_rec 데이터를 삽입하시오.
   INSERT INTO DEPT_RECORD
   VALUES dept_rec;

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON
5	99	DATABASE	SEOUL

```
DECLARE
   TYPE REC_DEPT IS RECORD(
      deptno NUMBER(2) NOT NULL := 99,
      dname DEPT.DNAME%TYPE,
      loc DEPT.LOC%TYPE
   );
   dept_rec REC_DEPT;
BEGIN
   dept_rec.deptno := 99;
   dept_rec.dname := 'DATABASE';
   dept_rec.loc := 'SEOUL';

   INSERT INTO DEPT_RECORD
   VALUES dept_rec;
END;
/

SELECT * FROM DEPT_RECORD;
```


Q004

1. 다음과 같이 REC_DEPT 레코드를 작성하시오.
2. dept_rec 레코드에
   deptno=50, dname=DB , loc=SEOUL 값을 대입하시오.
3. DEPT_RECORD 테이블에 DEPTNO=99인 데이터의 dept_rec 데이터를 수정하시오.
   UPDATE DEPT_RECORD
   SET ROW = dept_rec
   WHERE DEPTNO = 99;

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON
5	50	DB	SEOUL

```
DECLARE
   TYPE REC_DEPT IS RECORD(
      deptno NUMBER(2) NOT NULL := 99,
      dname DEPT.DNAME%TYPE,
      loc DEPT.LOC%TYPE
   );
   dept_rec REC_DEPT;
BEGIN
   dept_rec.deptno := 50;
   dept_rec.dname := 'DB';
   dept_rec.loc := 'SEOUL';

   UPDATE DEPT_RECORD
      SET ROW = dept_rec
    WHERE DEPTNO = 99;
END;
/

SELECT * FROM DEPT_RECORD;
```


Q005

- 레코드에 다른 레코드를 포함하시오.

1. REC_DEPT , REC_EMP 레코드를 작성하시오.
2. EMP, DEPT 테이블을 JOIN 하여
   EMPNO가 7788이고
   EMP, DEPT테이블의 DEPTNO 가 같은 데이터를 검색하시오.
3. 다음과 같이 출력하시오.

```
>>Query Run In:질의 결과 2
EMPNO : 7788
ENAME : SCOTT
DEPTNO : 20
DNAME : RESEARCH
LOC : DALLAS


PL/SQL 프로시저가 성공적으로 완료되었습니다.
```

```
DECLARE
   TYPE REC_DEPT IS RECORD(
      deptno DEPT.DEPTNO%TYPE,
      dname DEPT.DNAME%TYPE,
      loc DEPT.LOC%TYPE
   );
   TYPE REC_EMP IS RECORD(
      empno EMP.EMPNO%TYPE,
      ename EMP.ENAME%TYPE,
      dinfo REC_DEPT
   );
   emp_rec REC_EMP;
BEGIN
   SELECT E.EMPNO, E.ENAME, D.DEPTNO, D.DNAME, D.LOC
     INTO emp_rec.empno, emp_rec.ename,
          emp_rec.dinfo.deptno,
          emp_rec.dinfo.dname,
          emp_rec.dinfo.loc
     FROM EMP E, DEPT D
    WHERE E.DEPTNO = D.DEPTNO
      AND E.EMPNO = 7788;

   DBMS_OUTPUT.PUT_LINE('EMPNO : ' || emp_rec.empno);
   DBMS_OUTPUT.PUT_LINE('ENAME : ' || emp_rec.ename);
   DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || emp_rec.dinfo.deptno);
   DBMS_OUTPUT.PUT_LINE('DNAME : ' || emp_rec.dinfo.dname);
   DBMS_OUTPUT.PUT_LINE('LOC : ' || emp_rec.dinfo.loc);
END;
/
```

Q006

- 연관배열을 이용하여 다음과 같이 출력하시오.
- TYPE ITAB_EX , 자료형 VARCHAR2(20) , 인덱스형 PLS_INTEGER

```
text_arr(1) : 1st data
text_arr(2) : 2nd data
text_arr(3) : 3rd data
text_arr(4) : 4th data


PL/SQL 프로시저가 성공적으로 완료되었습니다.
```

```
DECLARE
   TYPE ITAB_EX IS TABLE OF VARCHAR2(20)
      INDEX BY PLS_INTEGER;

   text_arr ITAB_EX;

BEGIN
   text_arr(1) := '1st data';
   text_arr(2) := '2nd data';
   text_arr(3) := '3rd data';
   text_arr(4) := '4th data';

   DBMS_OUTPUT.PUT_LINE('text_arr(1) : ' || text_arr(1));
   DBMS_OUTPUT.PUT_LINE('text_arr(2) : ' || text_arr(2));
   DBMS_OUTPUT.PUT_LINE('text_arr(3) : ' || text_arr(3));
   DBMS_OUTPUT.PUT_LINE('text_arr(4) : ' || text_arr(4));
END;
/
```

Q007

- DEPT 테이블에서 DEPTNO, DNAME 값을 조회해 연관배열을 이용하여
- 다음과 같이 출력하시오.

```
10 : ACCOUNTING
20 : RESEARCH
30 : SALES
40 : OPERATIONS


PL/SQL 프로시저가 성공적으로 완료되었습니다.
```

```
DECLARE
   TYPE REC_DEPT IS RECORD(
      deptno DEPT.DEPTNO%TYPE,
      dname DEPT.DNAME%TYPE
   );

   TYPE ITAB_DEPT IS TABLE OF REC_DEPT
      INDEX BY PLS_INTEGER;

   dept_arr ITAB_DEPT;
   idx PLS_INTEGER := 0;

BEGIN
   FOR i IN (SELECT DEPTNO, DNAME FROM DEPT) LOOP
      idx := idx + 1;
      dept_arr(idx).deptno := i.DEPTNO;
      dept_arr(idx).dname := i.DNAME;

      DBMS_OUTPUT.PUT_LINE(
         dept_arr(idx).deptno || ' : ' || dept_arr(idx).dname);
   END LOOP;
END;
/
```

Q008

- SELECT * FROM DEPT 를 %ROWTYPE 와 연관배열을 이용하여
- 다음과 같이 출력하시오.

```
10 : ACCOUNTING : NEW YORK
20 : RESEARCH : DALLAS
30 : SALES : CHICAGO
40 : OPERATIONS : BOSTON


PL/SQL 프로시저가 성공적으로 완료되었습니다.
```

```
DECLARE
   TYPE ITAB_DEPT IS TABLE OF DEPT%ROWTYPE
      INDEX BY PLS_INTEGER;

   dept_arr ITAB_DEPT;
   idx PLS_INTEGER := 0;

BEGIN
   FOR i IN(SELECT * FROM DEPT) LOOP
      idx := idx + 1;
      dept_arr(idx).deptno := i.DEPTNO;
      dept_arr(idx).dname := i.DNAME;
      dept_arr(idx).loc := i.LOC;

      DBMS_OUTPUT.PUT_LINE(
      dept_arr(idx).deptno || ' : ' ||
      dept_arr(idx).dname || ' : ' ||
      dept_arr(idx).loc);
   END LOOP;
END;
/
```

Q009

- 다음과 같이 컬렉션메서드를 이용하여 출력하시오.

```
text_arr.COUNT : 4
text_arr.FIRST : 1
text_arr.LAST : 50
text_arr.PRIOR(50) : 3
text_arr.NEXT(50) :


PL/SQL 프로시저가 성공적으로 완료되었습니다.
```

```
DECLARE
   TYPE ITAB_EX IS TABLE OF VARCHAR2(20)
      INDEX BY PLS_INTEGER;

   text_arr ITAB_EX;

BEGIN
   text_arr(1) := '1st data';
   text_arr(2) := '2nd data';
   text_arr(3) := '3rd data';
   text_arr(50) := '50th data';

   DBMS_OUTPUT.PUT_LINE('text_arr.COUNT : ' || text_arr.COUNT);
   DBMS_OUTPUT.PUT_LINE('text_arr.FIRST : ' || text_arr.FIRST);
   DBMS_OUTPUT.PUT_LINE('text_arr.LAST : ' || text_arr.LAST);
   DBMS_OUTPUT.PUT_LINE('text_arr.PRIOR(50) : ' || text_arr.PRIOR(50));
   DBMS_OUTPUT.PUT_LINE('text_arr.NEXT(50) : ' || text_arr.NEXT(50));

END;
/
```

■STEP2. EX

EX001

- 다음과 같이 PL/SQL 문을 작성하시오.

1. EMP 테이블과 같은 열구조를 가지는 빈테이블 EMP_RECORD를 생성하는 SQL문을 작성하시오.
2. EMP_RECORD 테이블에 레코드를 사용하여 새로운 사원정보를 다음과 같이 삽입하는 PL/SQL 프로그램을 작성하시오.

```
Table EMP_RECORD이(가) 생성되었습니다.

PL/SQL 프로시저가 성공적으로 완료되었습니다.
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1111	TEST_USER	TEST_JOB	(null)	18/03/01	3000	(null)	40

EX002

- EMP 테이블을 구성하는 모든열을 저장할 수 있는 레코드를 활용하여 연관배열을 작성하시오. 그리고 저장된 연관배열의 내용을 다음과 같이 출력하시오.

7839 : KING : PRESIDENT : : 81/11/17 : 5000 : : 10
7698 : BLAKE : MANAGER : 7839 : 81/05/01 : 2850 : : 30
7782 : CLARK : MANAGER : 7839 : 81/05/09 : 2450 : : 10
7566 : JONES : MANAGER : 7839 : 81/04/01 : 2975 : : 20
7654 : MARTIN : SALESMAN : 7698 : 81/09/10 : 1250 : 1400 : 30
7499 : ALLEN : SALESMAN : 7698 : 81/02/11 : 1600 : 300 : 30
7844 : TURNER : SALESMAN : 7698 : 81/08/21 : 1500 : 0 : 30
7900 : JAMES : CLERK : 7698 : 81/12/11 : 950 : : 30
7521 : WARD : SALESMAN : 7698 : 81/02/23 : 1250 : 500 : 30
7902 : FORD : ANALYST : 7566 : 81/12/11 : 3000 : : 20
7369 : SMITH : CLERK : 7902 : 80/12/09 : 800 : : 20
7788 : SCOTT : ANALYST : 7566 : 82/12/22 : 3000 : : 20
7876 : ADAMS : CLERK : 7788 : 83/01/15 : 1100 : : 20
7934 : MILLER : CLERK : 7782 : 82/01/11 : 1300 : : 10

PL/SQL 프로시저가 성공적으로 완료되었습니다.