

■STEP1. BASIC

Q001

- SELECT INTO를 사용하여 DEPT테이블의 40번 부서 데이터를 조회하고 각 열의 결과값을 변수에 저장하고 출력하시오.

```
DEPTNO : 40  
DNAME : OPERATIONS  
LOC : BOSTON
```

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```
DECLARE
```

```
    V_DEPT_ROW DEPT%ROWTYPE;
```

```
BEGIN
```

```
    SELECT DEPTNO, DNAME, LOC INTO V_DEPT_ROW
```

```
        FROM DEPT
```

```
        WHERE DEPTNO = 40;
```

```
    DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || V_DEPT_ROW.DEPTNO);
```

```
    DBMS_OUTPUT.PUT_LINE('DNAME : ' || V_DEPT_ROW.DNAME);
```

```
    DBMS_OUTPUT.PUT_LINE('LOC : ' || V_DEPT_ROW.LOC);
```

```
END;
```

```
/
```

Q002

- 하나의 행만 조회되는 경우 번거롭지만, 여러행이 조회되는 행이 여러개일때 커서의 사용이 극대화 됨.

DEPTNO : 40
DNAME : OPERATIONS
LOC : BOSTON

PL/SQL 프로시저가 성공적으로 완료되었습니다.

DECLARE

-- 커서 데이터를 입력할 변수 선언
V_DEPT_ROW DEPT%ROWTYPE;

-- 명시적 커서 선언(Declaration)

CURSOR c1 IS

SELECT DEPTNO, DNAME, LOC
FROM DEPT
WHERE DEPTNO = 40;

BEGIN

-- 커서 열기(Open)

OPEN c1;

-- 커서로부터 읽어온 데이터 사용(Fetch)

FETCH c1 **INTO** V_DEPT_ROW;

DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || V_DEPT_ROW.DEPTNO);

DBMS_OUTPUT.PUT_LINE('DNAME : ' || V_DEPT_ROW.DNAME);

DBMS_OUTPUT.PUT_LINE('LOC : ' || V_DEPT_ROW.LOC);

-- 커서 닫기(Close)

CLOSE c1;

END;

/

Q003

- LOOP와 커서를 이용하여 여러행의 데이터를 커서에 저장해서 사용하기

```
DEPTNO : 10, DNAME : ACCOUNTING, LOC : NEW YORK  
DEPTNO : 20, DNAME : RESEARCH, LOC : DALLAS  
DEPTNO : 30, DNAME : SALES, LOC : CHICAGO  
DEPTNO : 40, DNAME : OPERATIONS, LOC : BOSTON
```

PL/SQL 프로시저가 성공적으로 완료되었습니다.

DB:DBIG
youtube.com/@DbdbigCode
Teacher Sally

DECLARE

```
-- 커서 데이터를 입력할 변수 선언  
V_DEPT_ROW DEPT%ROWTYPE;
```

```
-- 명시적 커서 선언(Declaration)
```

CURSOR c1 IS

```
    SELECT DEPTNO, DNAME, LOC  
    FROM DEPT;
```

BEGIN

```
-- 커서 열기(Open)
```

```
OPEN c1;
```

LOOP

```
-- 커서로부터 읽어온 데이터 사용(Fetch)
```

```
FETCH c1 INTO V_DEPT_ROW;
```

```
-- 커서의 모든 행을 읽어오기 위해 %NOTFOUND 속성 지정
```

```
EXIT WHEN c1%NOTFOUND;
```

```
DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || V_DEPT_ROW.DEPTNO  
                      || ', DNAME : ' || V_DEPT_ROW.DNAME  
                      || ', LOC : ' || V_DEPT_ROW.LOC);
```

```
END LOOP;
```

```
-- 커서 닫기(Close)
```

```
CLOSE c1;
```

```
END;
```

```
/
```

Q004

- 커서와 FOR LOOP를 사용하여 다음과 같이 출력하시오.

>>Query Run In:질의 결과

DEPTNO : 10, DNAME : ACCOUNTING, LOC : NEW YORK

DEPTNO : 20, DNAME : RESEARCH, LOC : DALLAS

DEPTNO : 30, DNAME : SALES, LOC : CHICAGO

DEPTNO : 40, DNAME : OPERATIONS, LOC : BOSTON

PL/SQL 프로시저가 성공적으로 완료되었습니다.

	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

-- 명시적 커서 선언(Declaration)

CURSOR c1 IS

SELECT DEPTNO, DNAME, LOC

FROM DEPT;

BEGIN

-- 커서 FOR LOOP 시작 (자동 Open, Fetch, Close)

FOR c1_rec IN c1 LOOP

DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || c1_rec.DEPTNO
|| ', DNAME : ' || c1_rec.DNAME
|| ', LOC : ' || c1_rec.LOC);

END LOOP;

END;

/

SELECT * FROM DEPT_RECORD;

Q005

- DEPT 테이블의 부서번호가 10 또는 20번일때 파라미터를 사용하는 커서를 다음과 같이 작성하시오.

10번 부서 - DEPTNO : 10, DNAME : ACCOUNTING, LOC : NEW YORK

20번 부서 - DEPTNO : 20, DNAME : RESEARCH, LOC : DALLAS

PL/SQL 프로시저가 성공적으로 완료되었습니다.

DECLARE

-- 커서 데이터를 입력할 변수 선언

V_DEPT_ROW DEPT%ROWTYPE;

-- 명시적 커서 선언(Declaration)

CURSOR c1 (p_deptno DEPT.DEPTNO%TYPE) IS

SELECT DEPTNO, DNAME, LOC

FROM DEPT

WHERE DEPTNO = p_deptno;

BEGIN

-- 10번 부서 처리를 위해 커서 사용

OPEN c1 (10);

LOOP

FETCH c1 INTO V_DEPT_ROW;

EXIT WHEN c1%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('10번 부서 - DEPTNO : ' || V_DEPT_ROW.DEPTNO
|| ', DNAME : ' || V_DEPT_ROW.DNAME
|| ', LOC : ' || V_DEPT_ROW.LOC);

END LOOP;

CLOSE c1;

-- 20번 부서 처리를 위해 커서 사용

OPEN c1 (20);

LOOP

FETCH c1 INTO V_DEPT_ROW;

EXIT WHEN c1%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('20번 부서 - DEPTNO : ' || V_DEPT_ROW.DEPTNO
|| ', DNAME : ' || V_DEPT_ROW.DNAME
|| ', LOC : ' || V_DEPT_ROW.LOC);

END LOOP;

CLOSE c1;

END;

/

Q006

- 커서에 사용할 파라미터를 입력받으시오.

DB:DBIG
youtube.com/@DbdbigCode
Teacher Sally

이전:DECLARE

-- 사용자가 입력한 부서 번호를 저장하는 변수선언

v_deptno DEPT.DEPTNO%TYPE;

-- 명시적 커서 선언(Declaration)

CURSOR c1 (p_deptno DEPT.DEPTNO%TYPE) IS

SELECT DEPTNO, DNAME, LOC

FROM DEPT

WHERE DEPTNO = p_deptno;

BEGIN

-- INPUT_DEPTNO에 부서 번호 입력받고 v_deptno에 대입

v_deptno := &INPUT_DEPTNO;

-- 커서 FOR LOOP 시작. c1 커서에 v_deptno를 대입

FOR c1_rec IN c1(v_deptno) LOOP

DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || c1_rec.DEPTNO

|| ', DNAME : ' || c1_rec.DNAME

|| ', LOC : ' || c1_rec.LOC);

END LOOP;

END;

신규:DECLARE

-- 사용자가 입력한 부서 번호를 저장하는 변수선언

v_deptno DEPT.DEPTNO%TYPE;

-- 명시적 커서 선언(Declaration)

CURSOR c1 (p_deptno DEPT.DEPTNO%TYPE) IS

SELECT DEPTNO, DNAME, LOC

FROM DEPT

WHERE DEPTNO = p_deptno;

BEGIN

-- INPUT_DEPTNO에 부서 번호 입력받고 v_deptno에 대입

v_deptno := 10;

-- 커서 FOR LOOP 시작. c1 커서에 v_deptno를 대입

FOR c1_rec IN c1(v_deptno) LOOP

DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || c1_rec.DEPTNO

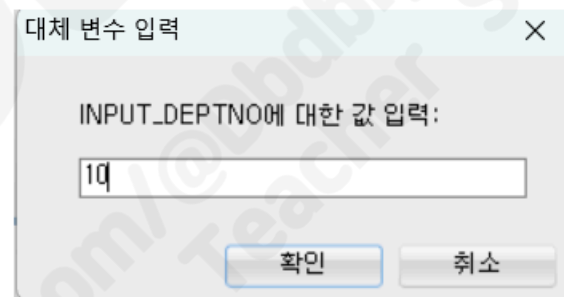
|| ', DNAME : ' || c1_rec.DNAME

|| ', LOC : ' || c1_rec.LOC);

END LOOP;

END;

DEPTNO : 10, DNAME : ACCOUNTING, LOC : NEW YORK



PL/SQL 프로시저가 성공적으로 완료되었습니다.

DECLARE

-- 사용자가 입력한 부서 번호를 저장하는 변수선언

v_deptno DEPT.DEPTNO%TYPE;

-- 명시적 커서 선언(Declaration)

CURSOR c1 (p_deptno DEPT.DEPTNO%TYPE) IS

SELECT DEPTNO, DNAME, LOC

FROM DEPT

WHERE DEPTNO = p_deptno;

BEGIN

-- INPUT_DEPTNO에 부서 번호 입력받고 v_deptno에 대입

v_deptno := &INPUT_DEPTNO;

-- 커서 FOR LOOP 시작. c1 커서에 v_deptno를 대입

FOR c1_rec IN c1(v_deptno) LOOP

DBMS_OUTPUT.PUT_LINE('DEPTNO : ' || c1_rec.DEPTNO

|| ', DNAME : ' || c1_rec.DNAME

|| ', LOC : ' || c1_rec.LOC);

END LOOP;

END;

/

Q007

- 묵시적 커서의 속성을 다음과 같이 사용해 보시오 .

갱신된 행의 수 : 0

갱신 대상 행 존재 여부 : false

커서의 OPEN 여부 : false

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```

BEGIN
  UPDATE DEPT SET DNAME='DATABASE'
  WHERE DEPTNO = 50;

  DBMS_OUTPUT.PUT_LINE('갱신된 행의 수 : ' || SQL%ROWCOUNT);

  IF (SQL%FOUND) THEN
    DBMS_OUTPUT.PUT_LINE('갱신 대상 행 존재 여부 : true');
  ELSE
    DBMS_OUTPUT.PUT_LINE('갱신 대상 행 존재 여부 : false');
  END IF;

  IF (SQL%ISOPEN) THEN
    DBMS_OUTPUT.PUT_LINE('커서의 OPEN 여부 : true');
  ELSE
    DBMS_OUTPUT.PUT_LINE('커서의 OPEN 여부 : false');
  END IF;

END;
/

```

Q008

- 예외가 발생하는 PL/SQL을 작성하시오.

```
DECLARE
```

```
*
```

```
오류 발생 행: 1:
```

```
ORA-06502: PL/SQL: numeric or value error: character to number conversion error
```

```
ORA-06512: at line 4
```

<https://docs.oracle.com/error-help/db/ora-06502/>

More Details :

<https://docs.oracle.com/error-help/db/ora-06502/>

<https://docs.oracle.com/error-help/db/ora-06512/>

```

DECLARE
    v_wrong NUMBER;
BEGIN
    SELECT DNAME INTO v_wrong
    FROM DEPT
    WHERE DEPTNO = 10;
END;
/

```

Q009

- 위의 예제에 예외처리를 하시오.
예외 처리 : 수치 또는 값 오류 발생

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```

DECLARE
    v_wrong NUMBER;
BEGIN
    SELECT DNAME INTO v_wrong
    FROM DEPT
    WHERE DEPTNO = 10;
EXCEPTION
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('예외 처리 : 수치 또는 값 오류 발생');
END;
/

```

Q0010

- 예외발생 후 코드 실행여부를 확인하시오.

예외 처리 : 수치 또는 값 오류 발생

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```
DECLARE
```

```
    v_wrong NUMBER;
```

```
BEGIN
```

```
    SELECT DNAME INTO v_wrong
```

```
    FROM DEPT
```

```
    WHERE DEPTNO = 10;
```

```
    DBMS_OUTPUT.PUT_LINE('예외가 발생하면 다음 문장은 실행되지 않습니다');
```

```
EXCEPTION
```

```
    WHEN VALUE_ERROR THEN
```

```
        DBMS_OUTPUT.PUT_LINE('예외 처리 : 수치 또는 값 오류 발생');
```

```
END;
```

```
/
```

Q0011

- 사전정의된 예외를 사용하시오 .

예외 처리 : 수치 또는 값 오류 발생

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```

DECLARE
    v_wrong NUMBER;
BEGIN
    SELECT DNAME INTO v_wrong
    FROM DEPT
    WHERE DEPTNO = 10;

    DBMS_OUTPUT.PUT_LINE('예외가 발생하면 다음 문장은 실행되지 않습니다');

EXCEPTION
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('예외 처리 : 요구보다 많은 행 추출 오류 발생');
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('예외 처리 : 수치 또는 값 오류 발생');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('예외 처리 : 사전 정의 외 오류 발생');
END;
/

```

Q0012

- 오류코드와 오류케시지를 사용하시오 .

예외 처리 : 사전 정의 외 오류 발생

SQLCODE : -6502

SQLERRM : ORA-06502: PL/SQL: numeric or value error: character to number conversion €

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```

DECLARE
    v_wrong NUMBER;
BEGIN
    SELECT DNAME INTO v_wrong
    FROM DEPT
    WHERE DEPTNO = 10;

    DBMS_OUTPUT.PUT_LINE('예외가 발생하면 다음 문장은 실행되지 않습니다');

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('예외 처리 : 사전 정의 외 오류 발생');
        DBMS_OUTPUT.PUT_LINE('SQLCODE : ' || TO_CHAR(SQLCODE));
        DBMS_OUTPUT.PUT_LINE('SQLERRM : ' || SQLERRM);
END;
/

```

■STEP2. EX

EX001

- 명시적 커서를 사용하여 EMP 테이블의 전체 데이터 조회한 후 커서안에 데이터가 다음과 같이 출력되도록 PL/SQL을 작성하시오.
1. LOOP를 사용하는 방식
 2. FOR LOOP를 사용하는 방식

EMPNO : 7839, ENAME : KING, JOB : PRESIDENT, SAL : 5000, DEPTNO : 10
EMPNO : 7698, ENAME : BLAKE, JOB : MANAGER, SAL : 2850, DEPTNO : 30
EMPNO : 7782, ENAME : CLARK, JOB : MANAGER, SAL : 2450, DEPTNO : 10
EMPNO : 7566, ENAME : JONES, JOB : MANAGER, SAL : 2975, DEPTNO : 20
EMPNO : 7654, ENAME : MARTIN, JOB : SALESMAN, SAL : 1250, DEPTNO : 30
EMPNO : 7499, ENAME : ALLEN, JOB : SALESMAN, SAL : 1600, DEPTNO : 30
EMPNO : 7844, ENAME : TURNER, JOB : SALESMAN, SAL : 1500, DEPTNO : 30
EMPNO : 7900, ENAME : JAMES, JOB : CLERK, SAL : 950, DEPTNO : 30
EMPNO : 7521, ENAME : WARD, JOB : SALESMAN, SAL : 1250, DEPTNO : 30
EMPNO : 7902, ENAME : FORD, JOB : ANALYST, SAL : 3000, DEPTNO : 20
EMPNO : 7369, ENAME : SMITH, JOB : CLERK, SAL : 800, DEPTNO : 20
EMPNO : 7788, ENAME : SCOTT, JOB : ANALYST, SAL : 3000, DEPTNO : 20
EMPNO : 7876, ENAME : ADAMS, JOB : CLERK, SAL : 1100, DEPTNO : 20
EMPNO : 7934, ENAME : MILLER, JOB : CLERK, SAL : 1300, DEPTNO : 10

PL/SQL 프로시저가 성공적으로 완료되었습니다.

EX002

- PL/SQL 문의 실행중 발생하는 예외를 다음결과와 같이 처리하는 예외처리부를 작성하시오.

오류가 발생하였습니다.[2024년12월22일 15시12분31초]

SQLCODE : -1841

SQLERRM : ORA-01841: (full) year must be between -4713 and +9999, and not be 0

PL/SQL 프로시저가 성공적으로 완료되었습니다.
