**1. 매개변수와 CURSOR**

CURSOR가 열릴 때 CURSOR로 매개변수 값을 전달하고, CURSOR가 실행될 때 질의에서 그 값이 사용될 수 있습니다.이것은 각 경우마다 다른 활성 셋(set)을 생성하는 블록에서 여러 번 명시적 CURSOR를 열고 닫을 수 있음을 의미합니다. CURSOR선언 시 각각 형식적인(formal) 매개변수는 OPEN문장에서 실제 해당 매개변수를 가져야 합니다. 매개변수 데이터형은 스칼라 변수의 데이터형과 동일하지만 크기는 주지 않습니다. 매개변수 명은 CURSOR의 질의 표현식에서 참조하기 위한 것입니다.

**1.1 Syntax**

|  |
| --- |
| **CURSOR** cursor\_name [(parameter\_name1 datatype, . . . .)]  **IS**  select\_statement; |

cursor\_name 앞에 선언된 CURSOR대한 PL/SQL식별자 입니다.

parameter\_name 매개변수 이름입니다. 매개변수는 아래의 구문을 따름니다.

|  |
| --- |
| Cursor\_parameter\_name [IN] datatype [{ := | DEFAULT} expression] |

datatype 매개변수의 스칼라 데이터형 입니다.

select\_statement INTO절이 없는 SELECT 문장을 기술합니다.

♣ 참고

매개변수 표기법은 더 많은 기능성을 제공하지 않습니다, 단순히 입력 값을 명확하고 쉽게 지정할 수 있도록 해 줍니다. 이것은 동일한 CURSOR가 반복적으로 참조될 때 특히 유용합니다.

문제1) DEPT 테이블의 내용을 조회하는 SCRIPT을 작성하여라. 단 매개변수를 이용하여라

|  |
| --- |
| SET SERVEROUTPUT ON  DECLARE  CURSOR dept\_cursor (v\_deptno NUMBER) IS  SELECT \*  FROM dept  WHERE deptno = v\_deptno;  BEGIN  DBMS\_OUTPUT.PUT\_LINE('부서번호 부 서 명 위 치');  DBMS\_OUTPUT.PUT\_LINE('-------- ------------- ------------');  FOR dept\_record IN dept\_cursor(10) LOOP  DBMS\_OUTPUT.PUT\_LINE(LPAD(dept\_record.deptno,2) || ' '  || RPAD(dept\_record.dname,15) || RPAD(dept\_record.loc,12));  END LOOP;  FOR dept\_record IN dept\_cursor(20) LOOP  DBMS\_OUTPUT.PUT\_LINE(LPAD(dept\_record.deptno,2) || ' '  || RPAD(dept\_record.dname,15) || RPAD(dept\_record.loc,12));  END LOOP;  END;  /  SET SERVEROUTPUT OFF |

문제2) EMP 테이블에서 부서번호와 업무를 입력 받아 사원번호,이름,급여를 출력하는 SCRIPT를 작성하여라. 단 매개변수를 이용하여라.

|  |
| --- |
| SET VERIFY OFF  SET SERVEROUTPUT ON  ACCEPT p\_deptno PROMPT ' 부서번호를 입력하시오 : '  ACCEPT p\_job PROMPT ' 담당업무를 입력하시오 : '  DECLARE  v\_sal\_total NUMBER(10,2) := 0;  CURSOR emp\_cursor(v\_deptno emp.deptno%TYPE,  v\_job VARCHAR2) IS  SELECT empno,ename,sal  FROM emp  WHERE deptno = v\_deptno AND job = v\_job  ORDER BY empno;  BEGIN  DBMS\_OUTPUT.PUT\_LINE('사번 이 름 급 여');  DBMS\_OUTPUT.PUT\_LINE('---- ---------- ----------------');  FOR emp\_record IN emp\_cursor(&p\_deptno,UPPER('&p\_job')) LOOP  v\_sal\_total := v\_sal\_total + emp\_record.sal;  DBMS\_OUTPUT.PUT\_LINE(RPAD(emp\_record.empno,6) ||  RPAD(emp\_record.ename,12) ||  LPAD(TO\_CHAR(emp\_record.sal,'$99,999,990.00'),16));  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('----------------------------------');  DBMS\_OUTPUT.PUT\_LINE(RPAD(TO\_CHAR(&p\_deptno),2) || '번 부서의 합 ' ||  LPAD(TO\_CHAR(v\_sal\_total,'$99,999,990.00'),16));  END;  /  SET VERIFY ON  SET SERVEROUTPUT OFF |

문제3) EMP 테이블에서 부서번호와 업무를 입력 받아 사원번호,이름,급여를 출력하는 SCRIPT를 작성하여라.

|  |
| --- |
| SET VERIFY OFF  SET SERVEROUTPUT ON  ACCEPT p\_deptno PROMPT ' 부서번호를 입력하시오 : '  ACCEPT p\_job PROMPT ' 담당업무를 입력하시오 : '  DECLARE  TYPE emp\_record\_type IS RECORD(  v\_empno emp.empno%TYPE,  v\_ename emp.ename%TYPE,  v\_sal emp.sal%TYPE);  emp\_record emp\_record\_type;  v\_sal\_total NUMBER(10,2) := 0;  CURSOR emp\_cursor(v\_deptno emp.deptno%TYPE,  v\_job VARCHAR2) IS  SELECT empno,ename,sal  FROM emp  WHERE deptno = v\_deptno AND job = v\_job  ORDER BY empno;  BEGIN  DBMS\_OUTPUT.PUT\_LINE('사번 이 름 급 여');  DBMS\_OUTPUT.PUT\_LINE('---- ---------- ----------------');  OPEN emp\_cursor(&p\_deptno,UPPER('&p\_job'));  LOOP  FETCH emp\_cursor INTO emp\_record;  EXIT WHEN emp\_cursor%NOTFOUND;  v\_sal\_total := v\_sal\_total + emp\_record.v\_sal;  DBMS\_OUTPUT.PUT\_LINE(RPAD(emp\_record.v\_empno,6) ||  RPAD(emp\_record.v\_ename,12) ||  LPAD(TO\_CHAR(emp\_record.v\_sal,'$99,999,990.00'),16));  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('----------------------------------');  DBMS\_OUTPUT.PUT\_LINE(RPAD(TO\_CHAR(&p\_deptno),2) || '번 부서의 합 ' ||  LPAD(TO\_CHAR(v\_sal\_total,'$99,999,990.00'),16));  CLOSE emp\_cursor;  END;  /  SET VERIFY ON  SET SERVEROUTPUT OFF |

**1.2 FOR UPDATE절**

행을 갱신하거나 삭제하기 전에 행을 잠글 수 있습니다. CURSOR가 열릴 때 영향을 미치는 행을 잠그기 위해 CURSOR질의에서 FOR UPDATE절을 추가합니다. ORACLE SERVER은 TRANSACTION이 종료할 때 잠금(locking)을 해제하기 때문에 FOR UPDATE가 사용된다면, 명시적 CURSOR에서 인출(fetch)한 후에 바로 COMMIT해서는 안됩니다. FOR UPDATE절은, ORDER BY절이 있다 해도, SELECT문자에서 마지막 절이 됩니다. 다중 테이블을 질의할 때, 특정 테이블에 대해서만 행을 잠그기 위해 FOR UPDATE절을 사용할 수 있습니다. 테이블의 행은 FOR UPDATE절이 그 테이블의 열을 참조할 때만 잠겨집니다. 독점적인(exclusive) 행 잠금(locking)은 FOR UPDATE절이 사용될 때, OPEN하기 전에 결과 셋(set)에 행해집니다.

**1.2.1 Syntax**

|  |
| --- |
| **CURSOR** cursor\_name IS  **SELECT** . . . . .  **FOR UPDATE** [**NOWAIT**] [**OF** column1[,column2,. . . .]]; |

♣ 주의

ORACLE SERVER은 SELECT FOR UPDATE에서 필요로 하는 행의 잠금을 얻을 수 없다면, 막연하게 기다립니다. SELECT FOR UPDATE문장에서 NOWAIT절을 사용할 수 있고 루프에서 잠금을 얻는데 실패하여 생기는 오류 코드를 테스트할 수 있습니다. 그러므로 PL/SQL블록을 종료하기 전에 CURSOR OPEN을 n번 다시 시도할 수 있습니다. 대형 테이블이라면, 테이블의 모든 행을 잠그기 위해 LOCK TABLE문장을 사용함으로써 더 나은 성능을 얻을 수도 있습니다. 그러나 LOCK TABLE을 사용할 때, WHERE CURRENT OF절을 사용할 수 없고 WHERE column = identifier를 사용해야 합니다. FOR UPDATE OF절이 열을 참조 하는 것이 필수적 이지는 않지만 이것은 더 쉽게 읽고 유지할 수 있게 하기 위해 추천되어집니다.

**1.3 WHERE CURRENT OF절**

명시적 CURSOR에서 현재 참조할 때 WHRE CURRENT OF절을 사용합니다. 이를 통해서 명시적으로 ROWID를 참조하지 않고 현재 처리 중인 행을 갱신하고 삭제할 수 있게 해 줍니다. 행을 OPEN시에 잠기게 하기 위해서 CURSOR의 SELECT 문에서 FOR UPDATE절을 포함해야 합니다. CURSOR에서 일정 조건에 따라 행을 갱신 할 수 있고 또한 FETCH문장에 의해 가장 최근에 프로세스된 행을 참조하기 위해 WHERE CURRENT OF cursor\_name절이 있는 DELETE또는 UPDATE문장을 쓸 수 있습니다. WHERE CURRENT OF절을 사용할 때 참조되는 CURSOR는 CURSOR질의에서 FOR UPDATE절을 포함해야 하고 존재해야 합니다. 그렇지 않으면 에러를 일으키게 됩니다. 이 절은 ROWID pseudocolumn을 명시적으로 참조할 필요 없이 현재 처리된 행에 대해 갱신과 삭제를 할 수 있도록 해 줍니다.

**1.3.1 Syntax**

|  |
| --- |
| DECLARE  . . . . .  CURSOR cursor\_name IS  SELECT . . . . .  FOR UPDATE [NOWAIT] [OF column1[,column2,. . . .]];  BEGIN  OPEN cursor\_name;  LOOP  . . . . .  UPDATE . . . . . .  WHERE CURRENT OF cursor\_name;  . . . . .  END LOOP;  COMMIT;  CLOSE cursor\_name;  END; |

문제4) EMP 테이블에서 다음의 조건을 만족하는 SCRIPT를 작성하여라. 10번부서는 급여의 25%, 20번부서는 급여의 15%, 30번부서는 급여의 20%을 인상하고 10번 부서는 20, 20번 부서는 30번,30번 부서는 10번 부서로 바꾸어라..

|  |
| --- |
| DECLARE  CURSOR emp\_cursor IS  SELECT sal,deptno  FROM emp  ORDER BY deptno  FOR UPDATE OF sal,deptno;  BEGIN  FOR emp\_record IN emp\_cursor LOOP  IF emp\_record.deptno = 10 THEN  UPDATE emp  SET sal = TRUNC(emp\_record.sal \* 1.25,-1), deptno = 20  WHERE CURRENT OF emp\_cursor;  ELSIF emp\_record.deptno = 20 THEN  UPDATE emp  SET sal = TRUNC(emp\_record.sal \* 1.15,-1), deptno = 30  WHERE CURRENT OF emp\_cursor;  ELSIF emp\_record.deptno = 30 THEN  UPDATE emp  SET sal = TRUNC(emp\_record.sal \* 1.20,-1), deptno = 10  WHERE CURRENT OF emp\_cursor;  END IF;  END LOOP;  COMMIT;  END;  / |

**1.4 SUBQUERY**

SUBQUERY은 다른SQL데이터 조작 문장 속에 있는 질의(일반적으로 괄호로 둘러쌈)입니다. SUBQUERY는 수행되면 값 또는 값의 집합을 RETURN합니다. SUBQUERY은 SELECT문장의 WHERE절에서 주로 사용됩니다. 또한 FROM절에서도 사용될 수 있습니다. SUBQUERY 또는 상호 관련 질의(correlated subquery)가 사용됩니다.

문제5) CURSOR에서 SUBQUERY를 이용하여 부서번호,부서명,인원수를 출력하여라.

|  |
| --- |
| SET SERVEROUTPUT ON  DECLARE  v\_cnt NUMBER;  CURSOR dept\_cursor IS  SELECT d.deptno,d.dname  FROM dept d  WHERE 5 <= (SELECT count(\*)  FROM emp  WHERE deptno = d.deptno);  BEGIN  DBMS\_OUTPUT.PUT\_LINE('부서번호 부 서 명 인원수');  DBMS\_OUTPUT.PUT\_LINE('-------- ------------- ------');  FOR dept\_record IN dept\_cursor LOOP  SELECT COUNT(\*)  INTO v\_cnt  FROM emp  WHERE deptno = dept\_record.deptno;  DBMS\_OUTPUT.PUT\_LINE(LPAD(dept\_record.deptno,2) || ' ' ||  RPAD(dept\_record.dname,15) || LPAD(v\_cnt,4));  END LOOP;  END;  /  SET SERVEROUTPUT OFF |

**◈ 연 습 문 제 ◈**

1. EMP\_MESS(E\_MESS VARXHAR2(100)) 테이블을 생성한다.

2. DEPT TABLE과 EMP TABLE을 각각 CURSOR로 OPEN하여 EMP\_MESS TABLE에 다음과 같이 등록하여라.

|  |
| --- |
| KING - ACCOUNTING  CLARK - ACCOUNTING  MILLER - ACCOUNTING  JONES - RESEARCH  FORD - RESEARCH |

3. CURSOR 처리에서 FOR UPDATE와 WHERE CURRENT OF의 절을 설명하여라.

4. EMP TABLE에서 10번부서는 급여의 10%를 가산하고, 20번부서는 20%,30번부서는 15%를 가산하여라.

5. EMP TABLE에서 급여가 1000~2000사이의 사람만 현재의 급여에 30%를 가산하여라.