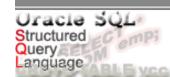


Writing Explicit Cursors

1. 커서 및 레코드

```
DECLARE
   CURSOR emp cursor IS
      SELECT employee id, last name FROM employees
      WHERE department id =30;
   emp record emp cursor%ROWTYPE;
BEGIN
   OPEN emp cursor;
   LOOP
      FETCH emp cursor INTO emp record;
      EXIT WHEN emp cursor%NOTFOUND;
      INSERT INTO temp_list (empid, empname)
      VLAUES (emp record.employee id, emp record.last name);
   END LOOP;
   COMMIT;
   CLOSE emp cursor;
END;
```

☑ 값을 PL/SQL 레코드로 인출(FETCH)하여 활성 집합(Active set)의 행을 쉽게 처리



2. 커서 FOR 루프

```
2-1. 커서 FOR 早里 (1)

FOR record_name IN cursor_name LOOP

statement1;
statement2;
• • • •
END LOOP; 3. CLOSE
```

- ☑ 커서 FOR 루프는 명시적 커서를 처리하기 위한 <mark>단축 방법</mark>
- ☑ 열기, 인출, 닫기가 암시적으로 발생
- ☑ 레코드가 암시적으로 선언
- ☑ 레코드의 범위는 루프 내에서만 가능
- 🔟 필요한 경우 반복 실행 중에 커서 속성을 테스트 가능
- ☑ 필요한 경우 FOR 문에서 커서 이름 뒤에 커서의 매개변수를 괄호로 묶어 제공할 수 있음
- □ 커서 조작을 수동으로 처리해야 하는 경우에는 커서 FOR 루프를 사용하면 안됨

2-2. 커서 FOR 루프 (2)

```
SET SERVEROUTPUT ON
DECLARE
  CURSOR emp_cursor IS
    SELECT ename, deptno
   FROM emp;
BEGIN
  FOR emp record IN emp cursor LOOP
  -- 암시적인 커서 열기, 인출 발생
    IF emp record.deptno = 30 THEN
     DBMS OUTPUT.PUT LINE ('Employee ' ||
       emp record.ename || ' works in the Sales Dept.');
    END IF:
  END LOOP; -- 암시적인 커서 닫기 발생
END;
```

2-3. 서브쿼리를 사용하는 커서 FOR 루프

```
BEGIN

FOR emp_record IN (SELECT empno, ename
FROM emp) LOOP

-- 암시적인 커서 열기, 인출 발생

IF emp_record.deptno = 30 THEN

DBMS_OUTPUT.PUT_LINE ('Employee' ||
emp_record.ename || 'works in the Sales Dept.');
END IF;
END LOOP; -- 암시적인 커서 닫기 발생
END;
```

☑ PL/SQL을 사용하면 서브쿼리를 치환할 수 있으므로 커서를 선언하지 않아 도 됨

mbo



3. 매개변수(Parameter) 사용 커서

```
CURSOR cursor_name
   [(parameter_name datatype, • • •)]
IS
   select_statement;

DECLARE
   CURSOR emp_cursor
   (v_deptno NUMBER, v_job VARCHAR2) IS
        SELECT empno, ename
        FROM emp
        WHERE deptno = v_deptno
        AND job = v_job;
BEGIN
   OPEN emp_cursor(10, 'CLEAR');
   • • •
```

- 꾀 커서가 열리고 질의가 실행되면 커서에 매개변수 값을 전달
- ☑ 블록에서 명시적 커서를 여러 번 열고 닫아 매번 다른 활성 집합(Active set)을 반환
- ☑ 데이터 유형은 스칼라 변수의 데이터 유형과 동일하지만 크기는 지정 하지 않음



4. FOR UPDATE 절

```
SELECT •••
FROM •••
FOR UPDATE [OF column_reference] [NOWAIT];

DECLARE
    CURSOR emp_cursor IS
        SELECT empno, ename, sal
        FROM emp
    WHERE deptno = 30
        FOR UPDATE OF sal NOWAIT;
```

- ▣ 명시적 잠금을 사용하여 트랜잭션 기간 동안 액세스 거부 가능
- ☑ 갱신 또는 삭제 전에 행을 잠금
- ☑ FOR UPDATE 절은 ORDER BY가 있는 경우에도 SELECT 문의 마지막 절
- ☑ NOWAIT 절을 사용하여 루프에서 잠글 수 없어 반환되는 오류 코드 검사
- ☑ FOR UPDATE OF 절이 반드시 열을 참조해야 하는 것은 아니지만 가독성 및 유지 관리 향상을 위해 열을 참조하는 것이 좋음

5. WHERE CURRENT OF 절

```
WHERE CURRENT OF cursor;
```

```
DECLARE
   CURSOR sal cursor IS
      SELECT salary
      FROM employees
      WHERE department id = 30
      FOR UPDATE OF salary NOWAIT;
BEGIN
   FOR emp record IN sal cursor LOOP
      UPDATE employees
             salary = emp record.salary * 1.10
      SET
      WHERE CURRENT OF sal cursor;
   END LOOP;
   COMMIT;
END;
```

- ☑ 커서를 사용하여 현재 행 갱신(UPDATE) 및 삭제(DELETE)
- ☑ 커서 질의에 FOR UPDATE 절을 포함시켜 먼저 행을 잠금
- ☑ WHERE CURRENT OF 절을 사용하여 명시적 커서에서 현재 행을 참조,,,,



6. 서브쿼리 포함 커서

```
CURSOR my_cursor IS

SELECT t1.deptno, t1.dname, t2 STAFF

FROM dept t1, (SELECT deptno,

count(*) STAFF

FROM emp

GROUP BY deptno) t2

WHERE t1.deptno = t2.deptno

AND t2.STAFF >= 5;
```

- ▣ 커서에 서브쿼리 또는 상호관련(Correlated) 서브쿼리 사용 가능
- □ 다른 SQL 데이터 조작문 내의 질의로서 대개 괄호로 묶어 표시하고 값 또 는 값 집합을 명령문에 제공
- ☑ 대개 SELECT 문의 WHERE 절에 사용하며 FROM 절에서 해당 질의의 임시 데이터 소스 생성 가능