7

Advanced Explicit Cursor Concepts

Objectives

After completing this lesson, you should be able to do the following:

- Write a cursor that uses parameters
- Determine when a FOR UPDATE clause in a cursor is required
- Determine when to use the WHERE CURRENT OF clause
- Write a cursor that uses a subquery

Cursors with Parameters

Syntax:

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

- Pass parameter values to a cursor when the cursor is opened and the query is executed.
- Open an explicit cursor several times with a different active set each time.

```
OPEN cursor_name(parameter_value,.....) ;
```

Cursors with Parameters

Pass the department number and job title to the WHERE clause, in the cursor SELECT statement.

```
SET SERVEROUTPUT ON
DECLARE
    CURSOR emp_cursor (p_deptno NUMBER, p_job VARCHAR2) IS
        SELECT employee_id, last_name    FROM employees
        WHERE department_id = p_deptno   AND job_id = p_job;
    emp_c    emp_cursor%rowtype;
BEGIN
     OPEN emp_cursor (80, 'SA_REP');
     LOOP
         FETCH emp_cursor INTO  emp_c;
         EXIT WHEN  emp_cursor%NOTFOUND;
         DBMS_OUTPUT.PUT_LINE ('ROWS : '||emp_cursor%rowcount);
    END LOOP;
    CLOSE emp_cursor;
     OPEN emp_cursor (60, 'IT_PROG');
          . . .
END;
```

```
SET SERVEROUTPUT ON
DECLARE
   CURSOR dept_c IS SELECT * FROM departments WHERE department_id < 60;
   CURSOR emp_c (p_deptno NUMBER) IS
        SELECT employee_id, last_name   FROM employees
        WHERE department_id = p_deptno;
   dept_r   dept_c%rowtype;
   emp_r    emp_c%rowtype;
BEGIN
     OPEN dept_c;
     LOOP
       FETCH dept_c INTO dept_r;
       EXIT WHEN dept_c%NOTFOUND;
       DBMS_OUTPUT.PUT_LINE (dept_r.department_id||' '||dept_r.department_name);
           OPEN emp_c (dept_r.department_id);
            LOOP
               FETCH emp_c INTO  emp_r;
               EXIT WHEN  emp_c%NOTFOUND;
               DBMS_OUTPUT.PUT_LINE ('NV : '||emp_r.employee_id || '-'||emp_r.last_name);
           END LOOP;
           CLOSE emp_c;
      END LOOP;
      CLOSE dept_c;
END;
```

```
SET SERVEROUTPUT ON
DECLARE
   CURSOR dept_c IS SELECT * FROM departments
                            WHERE department_id < 60;
   CURSOR emp_c (p_deptno NUMBER) IS
        SELECT employee_id, last_name   FROM employees
        WHERE department_id = p_deptno;
BEGIN
  FOR dept_r IN dept_c
     LOOP
          DBMS_OUTPUT.PUT_LINE (dept_r.department_id||' -  '||
                                        dept_r.department_name);
          FOR emp_r IN emp_c (dept_r.department_id)
          LOOP
                DBMS_OUTPUT.PUT_LINE ('NV : '||emp_r.employee_id || '-'||
                                              emp_r.last_name);
          END LOOP;
     END LOOP;
END;
```

```
SET SERVEROUTPUT ON
BEGIN
  FOR dept_r IN (SELECT * FROM departments
                 WHERE department_id<60)
    LOOP
      DBMS_OUTPUT.PUT_LINE (dept_r.department_id||' -  '||
                            dept_r.department_name);
      FOR emp_r IN (SELECT employee_id, last_name
                    FROM employees
                    WHERE department_id =dept_r.department_id)
      LOOP
        DBMS_OUTPUT.PUT_LINE ('NV : '||emp_r.employee_id || '-'||
                              emp_r.last_name);
      END LOOP;
    END LOOP;
END;
```

The FOR UPDATE Clause

Syntax:

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

- Use explicit locking to deny access for the duration of a transaction.
- Lock the rows *before* the update or delete.

The FOR UPDATE Clause

Retrieve the employees who work in department 80 and update their salary.

```
DECLARE
   CURSOR emp_cursor IS
            SELECT employee_id, last_name, department_name
            FROM employees,departments
            WHERE employees.department_id =
                  departments.department_id
            AND employees.department_id = 80
            FOR UPDATE OF salary NOWAIT;
```

The WHERE CURRENT OF Clause

Syntax:

```
WHERE CURRENT OF cursor ;
```

- Use cursors to update or delete the current row.
- Include the FOR UPDATE clause in the cursor query to lock the rows first.
- Use the WHERE CURRENT OF clause to reference the current row from an explicit cursor.

The WHERE CURRENT OF Clause

```
SET SERVEROUTPUT ON
DECLARE
CURSOR sal_cursor IS
    SELECT department_id, employee_id emp_id, last_name, salary
    FROM employees    WHERE department_id = 20
    FOR UPDATE OF salary NOWAIT;
BEGIN
    FOR emp_r IN sal_cursor      LOOP
       DBMS_OUTPUT.PUT_LINE (emp_r.emp_id||'-'||emp_r.salary);
       IF  emp_r.salary  >  5000  THEN
             UPDATE employees  SET salary = emp_r.salary * 1.10
             WHERE CURRENT OF sal_cursor;
       END IF;
    END LOOP;
    COMMIT;
END;
/
SELECT department_id, employee_id emp_id, last_name, salary
FROM employees      WHERE department_id = 20;
```

Cursors with Subqueries

```
SET SERVEROUTPUT ON
DECLARE
  CURSOR my_cursor IS
     SELECT t1.department_id, t1.department_name, t2.staff
     FROM departments t1,
           (SELECT department_id dept_id, COUNT(*) AS STAFF
            FROM employees GROUP BY department_id) t2
            WHERE t1.department_id = t2.dept_id AND t2.staff  >=  3;
BEGIN
   FOR c1 IN my_cursor
   LOOP
      DBMS_OUTPUT.PUT_LINE (c1.department_name||'-'|| c1.staff);
   END LOOP;
END;
/
```

Summary

In this lesson, you should have learned to:

- Return different active sets using cursors with parameters.
- Define cursors with subqueries and correlated subqueries.
- Manipulate explicit cursors with commands using the:
  - FOR UPDATE clause
  - WHERE CURRENT OF clause

Practice 7 Overview

This practice covers the following topics:

- Declaring and using explicit cursors with parameters
- Using a FOR UPDATE cursor