Database Systems
Laboratory 10
CURSOR

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

CURSOR

Cursor

Cursor

Cursor is the work area which Oracle reserves for internal processing of SQL statements. The data which is stored in the cursor is called Active Data Set

Cursor is used when you have a SELECT statement that returns more than one row from the database

A cursor is basically a set of rows that you access one at a time

A cursor acts logically as a pointer into a result set

Cursor Attributes

Attribute	Description
%ISOPEN	Returns TRUE if cursor is open
%FOUND	Returns TRUE if record was fetched successfully
%NOTFOUND	Returns TRUE if record was not fetched
%ROWCOUNT	Returns the number of records processed from the cursor

Cursor Operations

Cursor Operations

- Declare the cursor:
  - *CURSOR cursor_name IS select_query;*
- Open the cursor:
  - *OPEN cursor_name;*
- Fetch the data rows:
  - LOOP
    FETCH cursor_name INTO variable_name(s);
    EXIT WHEN cursor_name%NOTFOUND;
    END LOOP
- Close the cursor:
  - *CLOSE cursor_name;*

%TYPE and %ROWTYPE

%TYPE

It provides a column type

Variablename tablename.column%TYPE;

Ex: cno Customer.custno%TYPE;

%ROWTYPE

It provides a record type

variablename tablename%ROWTYPE;

Ex: vcust Customer%ROWTYPE;

Implicit Cursor

Implicit Cursor

- It is automatically created & handled by Oracle
- It reserves an area in main memory to populate the data
- It releases the memory area after the processing

- Implicit cursor handles all DMLs, i.e. INSERT, UPDATE & DELETE operations, which affect multiple rows and SELECT statement which returns exactly one row

- You have no control over an implicit cursor

- Implicit cursor attributes are: SQL%ISOPEN, SQL%FOUND, SQL%NOTFOUND, SQL%ROWCOUNT

Implicit Cursor...

An HRD manager has decided to raise the salary of employees by 15%. Write a PL/SQL block to accept an employee number and update the salary of that employee. Display the appropriate message based on the existence of the record in the EMP table

```
BEGIN
        UPDATE EMP SET salary=salary*1.15
                WHERE emp_no=&emp_no;
        IF SQL%FOUND THEN
           DBMS_OUTPUT.PUT_LINE('MODIFIED');
        END IF;
        IF SQL%NOTFOUND THEN
           DBMS_OUTPUT.PUT_LINE('NOT MODIFIED');
        END IF;
END;
```

Implicit Cursor...

An HRD manager has decided to raise the salary of employees working in department 30 by 15%. Write the PL/SQL block to display the number of records updated

```
DECLARE
        rowaff NUMBER(4);
BEGIN
      UPDATE EMP SET salary=salary*1.15 WHERE DEPT=30;
      rowaff : =SQL%ROWCOUNT;
      IF rowaff>0 THEN
         DBMS_OUTPUT.PUT_LINE(rowaff||' Employee
               records modified');
      ELSE
           DBMS_OUTPUT.PUT_LINE('There is no employee
                 working for dept. 30');
      END IF;
END;
```

Explicit Cursor

Explicit Cursor

- The cursors which are declared by user are called Explicit Cursors
- When a SELECT statement returns more than one row of results, we have to use explicit cursor
- We cannot use explicit cursor for DML statements

- Different attributes of explicit cursors are:
  - Cursorname%ROWCOUNT
  - Cursorname%FOUND
  - Cursorname%NOTFOUND
  - Cursorname%ISOPEN

Steps of Explicit Cursor

Steps of Explicit Cursor

- **Declare it**:
  - CURSOR cursorname IS SELECT statement;
- **Open it**:
  - OPEN cursorname;
- **Fetch rows from it**:
  - FETCH cursorname INTO variableset/ recordname;
- **Close it**:
  - CLOSE cursorname;

Explicit Cursor...

An HRD manager has decided to raise the salary of all the employees in department 30 by 5%. Whenever any such raise is given to the employee, a record for the same is maintained in the Emp_raise table. Write a PL/SQL block to update the salary of each employee & insert into the Emp_raise table

```
DECLARE
      CURSOR CSR IS SELECT e_id, salary FROM EMP
            WHERE dept=30;
      eid1 EMP.e_id%TYPE;
      sal1 EMP.salary%TYPE;
BEGIN
      OPEN CSR;
      IFCSR%ISOPEN THEN
        LOOP
            FETCH CSR INTO eid1,sal1;
            EXIT WHEN CSR%NOTFOUND;
```

Explicit Cursor...

```
...
   ...
      ...
        IFCSR%FOUND THEN
          UPDATE EMP SET salary=salary*1.05 WHERE e_id=eid1;
          INSERT INTO Emp_raise VALUES(eid1, SYSDATE,
                    sal1*0.05);
        END IF;
        END LOOP;
      COMMIT;
   ELSE
          DBMS_OUTPUT.PUT_LINE('Unable to open the cursor');
   END IF;
   CLOSE CSR;
END;
```

Explicit Cursor...

Write a PL/SQL block that will displaying the name, dept and salary of the first 3 employees getting the highest salary

```
DECLARE
    CURSOR CSR IS SELECT e_name,dept,salary FROM EMP
            ORDER BY salary DESC;
    vname EMP.e_name%TYPE;
    vdept EMP.dept%TYPE;
    vsal EMP.salary%TYPE;
BEGIN
    OPEN CSR;
    LOOP
        FETCH CSR INTO vname,vdept,vsal;
        EXIT WHEN CSR%ROWCOUNT=4 OR CSR%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(vname||' '||vdept||' '||vsal);
    END LOOP;
    CLOSE CSR;
END;
```

Cursor FOR Loop

Cursor FOR Loop

In cursor FOR loop, you do not have to explicitly open and close the cursor. It is automatically done by FOR loop

```
FOR variable IN cursorname
        LOOP
                Statements
        END LOOP
```

Cursor FOR Loop...

An HRD manager has decided to raise the salary of all the employees in department 30 by 5%. Whenever any such raise is given to the employee, a record for the same is maintained in the Emp_raise table. Write a PL/SQL block to update the salary of each employee & insert into the Emp_raise table

```
DECLARE
      CURSOR CSR IS SELECT e_id,salary FROM EMP
            WHERE dept=30;
BEGIN
      FOR i IN CSR
          LOOP
               UPDATE EMP SET salary=i.salary*1.05
                    WHERE e_id=i.e_id;
               INSERT INTO Emp_raise VALUES(i.e_id, SYSDATE,
                    i.salary*0.05);
          END LOOP;
      COMMIT;
END;
```

Parameterized Cursors

Parameterized Cursors

Parameters in cursors are useful when a cursor is required to be opened based on different set of parameter values

The parameter makes the cursor more reusable

A cursor with parameter can be opened and closed several times. Each time a new active set is loaded in the memory and the pointer is placed at first

CURSOR cursorname (parametername type, parametername type) IS SELECT statement;

Parameterized Cursors...

Display the name and designation of employees corresponding to the inputted deptno

```
DECLARE
    vname EMPLOYEE.ename%TYPE;
    vdesg EMPLOYEE.designation%TYPE;
    did NUMBER(2);
    CURSOR empcr(deptno EMPLOYEE.dno%TYPE) IS SELECT
      ename, designation FROM EMPLOYEE WHERE dno=deptno;
BEGIN
    did: =&did;
    OPEN empcr(did);
    DBMS_OUTPUT.PUT_LINE('Employee in dept.'||did);
    LOOP
        FETCH empcr INTO vname, vdesg;
        EXIT WHEN empcr%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(vname||' '||vdesg);
    END LOOP;
    CLOSE empcr;
END;
```

FOR UPDATE Clause

FOR UPDATE Clause

It is used to lock rows before updating or deleting of records

It is added in the cursor query to lock the affected records when cursor is opened

Oracle server releases the lock at the end of the transaction

It is used to ensure that the record is not changed by another user before the update or delete

CURSOR cursorname IS SELECT columns FROM tables WHERE *cond* FOR UPDATE [OF columns] [NOWAIT];

WHERE CURRENT OF Clause

WHERE CURRENT OF Clause

It is used for referencing the current row of the active set retrieved by the explicit cursor

It allows to apply updates & deletes to the row currently being accessed without referencing ROWID

UPDATE tablename SET clause WHERE CURRENT OF cursorname;

DELETE FROM tablename WHERE CURRENT OF cursorname;

WHERE CURRENT OF clause references the cursor and changes only to the last fetched row

Explicit Cursor

Write the PL/SQL block which increases the salary of the employees of the dept 10 & 30 with the locking of the records

```
DECLARE
      vrec EMPLOYEE%ROWTYPE;
      CURSOR cremp IS SELECT * FROM EMPLOYEE WHERE
          deptno IN (10,30) ORDER BY deptno FOR UPDATE OF
               deptno, salary NOWAIT;
BEGIN
      OPEN cremp;
      LOOP
            FETCH cremp INTO vrec;
            EXIT WHEN cremp%NOTFOUND;
            UPDATE EMPLOYEE SET salary=vrec.salary*1.1
                  WHERE CURRENT OF cremp;
      END LOOP;
      CLOSE cremp;
END;
```

Disadvantages of Cursors

Disadvantages of Cursors

It uses much more network bandwidth

It allocates resources at the server

If the cursor is not properly closed, the resources will not be freed until the SQL session itself is closed