Tech Mahindra

Cursors

Copyright © 2013 Tech Mahindra. All rights reserved.

Loops

- Write a program to print the numbers in the format

```
1   1   1
2   2   2
3   3   3
```

```
BEGIN
   DBMS_OUTPUT.NEW_LINE;
   FOR i IN 1 .. 3
   LOOP
      FOR j IN 1 .. 3
      LOOP
         DBMS_OUTPUT.PUT(i||' ');
      END LOOP;
      DBMS_OUTPUT.NEW_LINE;
   END LOOP;
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Conditional vs. Iterative loop

- To display first 10 numbers

```
SET SERVEROUTPUT ON;
DECLARE
   I NUMBER := 1;
BEGIN
  WHILE( I<=10)
  LOOP
   DBMS_OUTPUT.PUT_LINE(I);
   I:= I+1;
  END LOOP;
END;
/
```

```
SET SERVEROUTPUT ON;
BEGIN
  FOR I IN 1 .. 10
  LOOP
   DBMS_OUTPUT.PUT_LINE(I);
  END LOOP;
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Exercise

1. Write a program to validate the string to accept only lower case letters without numbers and special characters.

2. Write a program check the given number is prime or not

Copyright © 2013 Tech Mahindra. All rights reserved.

Cursors

- Oracle uses work area to execute SQL commands and store processing information. PL/SQL allows you to access this area through a name using a cursor.
- Whenever you issue a SQL statement, the Oracle Server opens an area of memory in which the command is parsed and executed. This area is called a cursor.
- When you execute a SQL statement from PL/SQL, the Oracle RDBMS assigns a private work area for that statement. This work area contains information about the SQL statement and the set of data returned or affected by that statement.
- The PL/SQL cursor is a mechanism by which you can name that work area and manipulate the information within it. In its simplest form, you can think of a cursor as a pointer into a table in the database.

Copyright © 2013 Tech Mahindra. All rights reserved.

Cursor types

- Two types of Cursors
  - Implicit cursor
    - PL/SQL declares a cursor implicitly for all SQL data manipulation statements, including queries that return only one row. However, for queries that return more than one row, you must declare an explicit cursor or use a cursor FOR loop. The name of the implicit cursor is SQL. You can directly use the cursor without any declaration.
  - Explicit cursor
    - The set of rows returned by a query can consist of zero, one or multiple rows, depending on how many rows meet your search criteria. When a query returns multiple rows, you can explicitly declare a cursor to process the rows.
    - The set of rows returned by a multiple-row query is called the active set. *It is manipulated just like a file in programming languages.*

Copyright © 2013 Tech Mahindra. All rights reserved.

Example to use implicit cursor

- Write a PL/SQL program to update a value in the table

```
SET SERVEROUTPUT ON;
DECLARE
  v_empno  payroll.empno%type;
  v_basic  payroll.basic%type;
BEGIN
   v_empno := &empno;
   v_basic := &basic;
  UPDATE payroll SET BASIC = v_basic
  WHERE empno = v_empno;
  DBMS_OUTPUT.PUT_LINE('Record updated');
  COMMIT;
END;
/
```

The above program displays the output “Record updated” either the record updated or not. To see the status of this updation, we can use implicit cursor (SQL).

Copyright © 2013 Tech Mahindra. All rights reserved.

Cursor - Attributes

Attribute	Data type	Significance	Recommended time to use
FOUND	BOOLEAN	TRUE if most recent fetch found a row in the table; otherwise FALSE	After opening and fetching from the cursor but before closing it (will be NULL before first fetch)
NOTFOUND	BOOLEAN	This the just logical inverse of FOUND	Same as above
ROWCOUNT	NUMBER	Number of Rows fetched so far	Same as above (except it will be zero before the first fetch)
ISOPEN	BOOLEAN	TRUE if the cursor is already opened otherwise returns FALSE	Just before opening the cursor

Copyright © 2013 Tech Mahindra. All rights reserved.

Modify - Above Code

```
SET SERVEROUTPUT ON;
DECLARE
  v_empno     payroll.empno%type;
  v_basic     payroll.basic%type;
  not_updated EXCEPTION;
BEGIN
   v_empno := &empno;
   v_basic := &basic;
  UPDATE payroll SET BASIC = v_basic
  WHERE empno = v_empno;
  IF SQL%NOTFOUND THEN
     RAISE not_updated;
  END IF;
  DBMS_OUTPUT.PUT_LINE('Record updated  '||SQL%ROWCOUNT);
  COMMIT;
EXCEPTION
  WHEN not_updated THEN
   DBMS_OUTPUT.PUT_LINE('Employee not found..');
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Explicit cursor functions

- Can process beyond the first row returned by the query, row by row.
- Keep track of which row is currently being processed.
- Allow the programmer to manually control them in the PL/SQL block.

Note: The fetch for an implicit cursor is an array fetch, and the existence of a second row still raises the TOO_MANY_ROWS exception. Furthermore, you can use explicit cursors to perform multiple fetches and to re-execute parsed queries in the work area.

Example : SELECT sal INTO V_SAL FROM emp WHERE ENAME LIKE ‘E%’;

Above PL/SQL statement returns more than one salary value, which can not be accommodated in a single variable. So, it fires an EXCEPTION TOO_MANY_ROWS.

To process such PL/SQL statements, we are going to use Cursors.

Copyright © 2013 Tech Mahindra. All rights reserved.

Handling Explicit Cursor

- Explicit cursor is a name used to refer to an area where you can place multiple rows retrieved by SELECT
- **STEPS**
  - The following are the required steps to process an explicit cursor.
  - Declare the cursor in declare section
  - Open the cursor using OPEN
  - Fetch rows from the cursor FETCH
  - Repeat above Fetch operation until Last Record.
  - Close the cursor after the process is over using CLOSE

Copyright © 2013 Tech Mahindra. All rights reserved.

How does a cursor work?

- Let us understand the cursor, with C File.
  - Declare File Pointer using FILE *fp
  - Open the file using fp = FOPEN(“filename”,”mode”)
  - Read the content of the file using FREAD()
  - Repeat FREAD() operation until EOF
  - Close the file using FCLOSE()

Copyright © 2013 Tech Mahindra. All rights reserved.

Declare a Cursor

- A cursor is declared in DECLARE section using CURSOR statement.
  - **Syntax**: CURSOR <cursorname> IS <SELECT command>;

Example

- **CURSOR emp_cur IS**
  SELECT empno, ename, job, sal FROM emp WHERE empno >= 7521;

- **Note**: No data is actually retrieved at the time of cursor declaration. Data is placed in the cursor when cursor is opened.

Copyright © 2013 Tech Mahindra. All rights reserved.

OPEN

- ***Opening a cursor using OPEN command***

OPEN statement is used to execute the query associated with the cursor and place the result into cursor.

Syntax: OPEN cursor_name;

Example

Open emp_cur;

When a cursor is opened, all the rows retrieved by SELECT, given at the time of cursor declaration, are placed in the cursor.

Copyright © 2013 Tech Mahindra. All rights reserved.

Fetching rows from a cursor using FETCH command

- For each column in the cursor there should be a corresponding variable in FETCH statement.
- FETCH statement is to be repeatedly executed to fetch all the rows of the cursor.

Closing a Cursor using CLOSE command

CLOSE statement is used to close after the cursor is processed.

Syntax: CLOSE cursor_name

Example

CLOSE emp_cur;

Copyright © 2013 Tech Mahindra. All rights reserved.

Program to test the cursor

```
SET SERVEROUTPUT ON   -- SQL*plus Environment command
DECLARE
   CURSOR emp_cur IS SELECT empno, ename, job, sal FROM emp
   WHERE empno     >= 7521;
   v_emp_rec  emp_cur%ROWTYPE;
BEGIN
   /* open the cursor */
   OPEN emp_cur;
   /* fetch a record from cursor */
   FETCH emp_cur INTO v_emp_rec;
   DBMS_OUTPUT.PUT_LINE(v_emp_rec.empno || v_emp_rec.ename||
v_emp_rec.sal);
   -- closing the cursor
   CLOSE emp_cur;
END;
/
```

Analysis:

This program reads and prints only one record from cursor

Copyright © 2013 Tech Mahindra. All rights reserved.

Program to read each and every record from the cursor

```
SET SERVEROUTPUT ON
DECLARE
   Cursor emp_cur is Select empno, ename, job, sal FROM emp WHERE empno >= 7521;
   v_emp_rec  emp_cur%ROWTYPE;
BEGIN
   /* open the cursor */
   OPEN emp_cur;
   /* fetch all the records of the cursor one by one */
   LOOP
      FETCH emp_cur INTO v_emp_rec;
      /* Exit loop if reached end of cursor
      NOTFOUND is the cursor attribute */
      EXIT WHEN emp_cur%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE (v_emp_rec.empno || v_emp_rec.ename||
v_emp_rec.sal);
   END LOOP;
   CLOSE emp_cur; -- closing the cursor
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Passing parameters to cursors

```
SET SERVEROUTPUT ON
DECLARE
   -- p_empno is the formal parameter
   CURSOR emp_cur(p_empno NUMBER)
   IS SELECT empno, ename, job, sal FROM emp WHERE empno >= p_empno;
   v_emp_rec  emp_cur%ROWTYPE;
   v_eno      emp.empno%TYPE;
BEGIN
   /* input the employee number */
   v_eno  := &empno;
   /* open the cursor */
   OPEN emp_cur(v_eno);  -- Actual argument
   /* fetch all the records of the cursor one by one */
   LOOP
      FETCH emp_cur INTO v_emp_rec;
      --Exit loop if reached end of cursor NOTFOUND is the cursor attribute
      EXIT WHEN emp_cur%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE (emp_rec.empno || emp_rec.ename|| emp_rec.sal);
   END LOOP;
   -- closing the cursor
   CLOSE emp_cur;
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Declaring Multiple Cursors

```
SET SERVEROUTPUT ON
DECLARE
   CURSOR dept_cur IS SELECT * FROM dept;
   CURSOR emp_cur(p_deptno IN NUMBER) IS
   SELECT empno,ename,sal FROM emp WHERE deptno=p_deptno;
   dept_rec   dept_cur%ROWTYPE;
   emp_rec    emp_cur%ROWTYPE;
BEGIN
   OPEN dept_cur;
   LOOP
      FETCH dept_cur INTO dept_rec;
      EXIT WHEN dept_cur%NOTFOUND:
      DBMS_OUTPUT.PUT_LINE('Employees working Under ' || dept_rec.deptno);
      OPEN emp_cur(dept_rec.deptno);
      LOOP
         FETCH emp_cur INTO emp_rec;
         EXIT WHEN emp_cur%NOTFOUND;
         DBMS_OUTPUT.PUT_LINE(emp_rec.empno || emp_rec.ename||emp_rec.job);
      END LOOP;
      CLOSE emp_cur;
   END LOOP:
   CLOSE dept_cur;
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Using Cursor Attributes

```
SET SERVEROUTPUT ON
DECLARE
   CURSOR emp_cur IS SELECT empno, ename, job, sal FROM emp WHERE empno >= 7521;
   emp_rec  emp_cur%ROWTYPE;
BEGIN
   If NOT emp_cur%ISOPEN THEN
      OPEN emp_cur;
   END IF;
   LOOP
      FETCH emp_cur INTO emp_rec;
      IF emp_cur%FOUND THEN
         DBMS_OUTPUT.PUT_LINE('No of rows effected ' || emp_cur%ROWCOUNT);
      ELSE
         DBMS_OUTPT.PUT_LINE('END Of file ');
      END IF;
      EXIT WHEN emp_cur%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE (emp_rec.empno || emp_rec.ename|| emp_rec.sal);
   END LOOP;
   CLOSE emp_cur;
END;
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Cursor For Loop

In order to process a cursor, you can use cursor FOR loop to automate the following steps.

- Opening cursor
- Fetching rows from the cursor
- Terminating loop when all rows in the cursor are processed
- Closing cursor

Syntax:

```
FOR  rowtype_variable IN cursor_name (parameters)
LOOP
        Statements;
END LOOP;
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Example

```
SET SERVEROUTPUT ON
DECLARE
   CURSOR emp_cursor IS SELECT empno,ename FROM emp;
BEGIN
    FOR record_variable IN emp_cursor
    LOOP
       DBMS_OUTPUT.PUT_LINE(record_variable.empno || record_variable.ename);
    END LOOP;
END;
/
```

```
SET SERVEROUTPUT ON
DECLARE
   CURSOR emp_cursor(p_empno NUMBER) IS SELECT empno,ename FROM emp
    WHERE empno <= p_empno;
BEGIN
    FOR record_variable IN emp_cursor(&empno)
    LOOP
       DBMS_OUTPUT.PUT_LINE(record_variable.empno || record_variable.ename);
    END LOOP;
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

For Update Of and Current Of

•By default, Oracle locks rows while updating and deleting rows. But it is possible to override default locking by using FOR UPDATE.

•FOR UPDATE clause can be used with SELECT while declaring cursor to lock all records retrieved by cursor to make sure they are not changed before we update or delete them. As Oracle automatically locks rows for you, FOR UPDATE clause is required only when you want to lock rows before the update or delete – at the time of opening cursor.

•CURRENT OF clause can be used to refer to the current row in the cursor.

Note: FOR UPDATE must be given if you want to use CURRENT OF clause to refer to current row in the cursor.

Copyright © 2013 Tech Mahindra. All rights reserved.

Program

- Write a program to give the grades to different students

Rollno	Sname	Sub1	Sub2	Total	Average	Grade
1	Kris	56	61	117	58.5	
2	Martin	67	71	138	69	
3	Joshi	51	53	104	52	
4	Manjari	89	61	150	75	
5	Mohd	45	81	126	63	

For giving the grades, we have to follow the following sequence of steps:

1. Read each student information
2. Calculate the grade
3. update the row

Above 3 steps we have to repeat for each and every student. So, for each student, we have to read the existing values and update the grade. So, 2 times we are interacting with the DB.

So, for 5 students, 10 times we have to interact with DB. Whenever number of interactions are more it degrade the performance.

Copyright © 2013 Tech Mahindra. All rights reserved.

Using Cursor to update the grades

```
DECLARE
  CURSOR stu_cur(p_sid NUMBER) IS
  SELECT ROLLNO,AVERAGE,GRADE FROM STUDENT
  WHERE rollno >= p_sid;
BEGIN
   FOR STU_REC IN STU_CUR(&ROLLNO)
   LOOP
     IF STU_REC.AVERAGE > 70 THEN
        STU_REC.GRADE := 'A';
     ELSIF STU_REC.AVERAGE > 60 THEN
        STU_REC.GRADE := 'B';
     ELSIF STU_REC.AVERAGE > 50 THEN
        STU_REC.GRADE := 'C';
     ELSE
         STU_REC.GRADE := 'F';
     END IF;
    UPDATE STUDENT SET GRADE = STU_REC.GRADE
    WHERE ROLLNO = STU_REC.ROLLNO;
   END LOOP;
 COMMIT;
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Performance issues

- In the previous program there are certain performance issues:
  - In the statement
    - UPDATE STUDENT SET GRADE = STU_REC.GRADE
      WHERE ROLLNO = STU_REC.ROLLNO;

Whenever we are executing the above statement, everytime it has to scan the total table for updation.

  - For Rollno = 1, it has to perform 1 scan operation before updating grade
  - For Rollno = 2, it has to perform 2 scan operations before updating grade
  - For Rollno = 3, it has to perform 3 scan operations before updating grade

And so on….
So, performance degrades when no of records are more to update

To overcome this problem we can use FOR UPDATE OF and CURRENT OF Key words.

Copyright © 2013 Tech Mahindra. All rights reserved.

Modified Logic

```
DECLARE
  CURSOR stu_cur(p_sid NUMBER) IS
  SELECT ROLLNO,AVERAGE,GRADE FROM STUDENT
  WHERE rollno >= p_sid FOR UPDATE OF grade;
BEGIN
   FOR STU_REC IN STU_CUR(&ROLLNO)
   LOOP
     IF STU_REC.AVERAGE > 70 THEN
        STU_REC.GRADE := 'A';
     ELSIF STU_REC.AVERAGE > 60 THEN
        STU_REC.GRADE := 'B';
     ELSIF STU_REC.AVERAGE > 50 THEN
        STU_REC.GRADE := 'C';
     ELSE
         STU_REC.GRADE := 'F';
     END IF;
    UPDATE STUDENT SET GRADE = STU_REC.GRADE
    WHERE CURRENT OF stu_cur;
   END LOOP;
 COMMIT;
END;
/
```

Copyright © 2013 Tech Mahindra. All rights reserved.

Exercise 1 - Passbook Table

sno	trndate	trntype	amount	balance
1	sysdate	D	30000	30000
2	sysdate	W	5000	25000
3	sysdate	D	30000	55000
4	sysdate	W	3000	50000
5	sysdate	D	30000	80000
6	sysdate	W	5000	75000
7	sysdate	D	30000	105000
8	sysdate	W	5000	100000
9	sysdate	D	30000	130000
10	sysdate	W	5000	125000

In the above table, if we modify the amount in one particular transaction, Write a program to modify the corresponding balances.

Handle suitable exceptions, to accept only a positive amount and in no case, balance should become negative.

Copyright © 2013 Tech Mahindra. All rights reserved.

Exercise 2 Inventory

- Consider Supermarket. Maintaining various items with their stock details and the day to day transactions

- ITEM_MASTER

Itemno	ITEMNAME	STOCK
1	COKE	65
2	PEN	40
3	SHAMPOO	50

ITEM_TRANSACTION

TRANSACTION_NO	TRANSACTION_DATE	ITEMNO	TRNTYPE	QUNATITY	BALANCE
1	SYSDATE	2	I	5	30
2	SYSDATE	1	R	3	70
3	SYSDATE	2	R	10	40
4	SYSDATE	3	I	6	50
5	SYSDATE	1	I	10	60
6	SYSDATE	1	R	5	65

Copyright © 2013 Tech Mahindra. All rights reserved.

Programs

- 1. Write a program to maintain various items in the master table
  2. Write a program to record a transaction in the transaction table and update the stock in the master table
- 3. Write a program to update the balances in the transaction table, when there is a change in either TRNTYPE or Quantity or in both and also update the stock in the master table

Copyright © 2013 Tech Mahindra. All rights reserved.

Exercise 3 Generate timesheet report

- In TECHM, everyday employee filling their timesheets. DB Developer designed a Table to record employees timesheet details as

EMPNO	ENAME	START_DATE	END_DATE	WORK_STATUS
16752	KRIS	01-NOV-2013	10-NOV-2013	WLWWWWWLWW
7862	JOSHI	01-NOV-2013	10-NOV-2013	WLWWWWWLLL
56782	BABU	01-NOV-2013	10-NOV-2013	LLWWWLWLWW

W : Worked L Leave

Write a PL/SQL program to generate a report as follows:

```
EMPNO      ENAME          Work_day          Work_Status
  16752      KRIS         01-NOV-2013           Worked
                          02-NOV-2013           Leave
                          03-NOV-2013           Worked
                             -------
    7862      JOSHI        01-NOV-2013           Worked
                           02-NOV-2013           Leave
                             -------             -----
```

Copyright © 2013 Tech Mahindra. All rights reserved.