Page No.
Date

Assignment - 06

* Title: Cursors (All Types : Implicit, Explicit, cursor FOR LOOP, Parameterized cursor)

* Date of completion: 14/09/2020

* Date of Submission: 30/09/2020

* Problem statement:
  Write a PL/SQL block of code using parameterized cursor, that will merge the data available in newly created table N_EmpId with data available in the table O_EmpId. If the data in the first table already exists in second table then that data should be skipped.

* Learning Objectives:
  - To understand & implement types of cursors with PL/SQL block code.

* Learning Outcomes:
  Students will be able to:
  - Implement PL/SQL block code
  - Implement types of cursors.

* S/W & H/W Requirements:
  Windows 10(64 bit), i5 processor, MySQL

Teacher's Sign.: _____

Page No.
Date

* Theory:

ORACLE:

PL/SQL:

* Cursors

- A cursor is a temporary work area created in system memory when a SQL statement is executed.
- A cursor can hold more than one row, but can process only one row at a time.
- The set of rows a cursor handle holds is called active set.

There are 2 types of cursors in PL/SQL:

i) Implicit Cursor:

These are created by default when DML statements like insert, update and delete statements are executed. They are also created when select statements that returns just one row is executed.

ii) Explicit Cursors:

- They must be created when you are executing a select statement that returns more than one row.
- Only one row can be processed at a time.
- When you fetch a row the current row position moves to next row.

Teacher's Sign.:

Page No.
Date

- **Implicit Cursors:**

In PL/SQL you can refer to the most recent implicit cursor as the SQL cursor

It has the following attributes:

(i) %FOUND - returns true if DML statements affected more than one row.

(ii) %NOTFOUND - returns true if DML statements affected no rows.

(iii) %ISOPEN - always returns false for implicit cursor.

(iv) %ROWCOUNT - returns number of rows affected by an insert, update, delete statements

Syntax:

```
declare
    declaration statements
begin
    execution statements
    if sql%notfound then
        execution statements
    elsif sql%found then
        execution statements
    end if;
end;
```

- **Explicit cursors:**

- program defined cursors for gaining more control over the context area.

Teacher's Sign:

Page No.
Date

Syntax:

Creating an explicit cursor

CURSOR cursor_name IS select_statement;

Syntax for using cursor:

```
declare
   declaration statements
   cursor cursor_name is select_statement;
begin
   open cursor_name;   (opening the cursor)
   loop
   fetch cursor_name into variables;
         exit when cursor_name %. attribute;
   end loop;
   close cursor_name;
end;
```

- CURSOR FOR LOOP

  - you would use a cursor for loop when you want to fetch and process every record in a cursor. It will terminate when all of records in the cursor have been fetched.

Syntax:

```
for recordindex in cursor_name
   loop
         { . . . statements}
end loop;
```

Teacher's Sign.: _____

Here,

- record_index
  - index of record
- cursor_name
  - name of the cursor that you wish to fetch records from.

- **Parameterized Cursor:**
  - a cursor can have parameters in pl/sql

Syntax:

```
declare
    cursor cur_name (parameter list)
        is select_statement;
begin
    open cur_name (parameters);
    loop
        fetch cur_name into variables
        exit when cur_name %: attribute;
    { ---- statements ---- }
end loop;
close cur_name;
```

- **MySQL Cursors:**
  - MySQL: cursor is read-only, non scrollable and asensitive

Teacher's Sign.: _____

Page No.
Date

Syntax:

```
create procedure procedure_name (parameters)
begin
declare
    { ... declarations ... }
declare cursor_name cursor for select
                              statement;
open cursor_name;
loop1 : loop;
        fetch cursor_name into variable;
        if condition then
            { statements }
        end if;
        end loop loop1;
        close cursor name;
        end;
```

- Test cases:

Input	Expected Output	Result
call proc1()	1 "Varun Karwa" (N) 2 "Tejas Dahad" (Duplicate) 3 "Prachi Wagh" (N) 4 "Bhavika Chuttar" (N)	Success.

- Conclusion:
  - Thus, we implemented cursors in MySQL stored procedure with parameter
  - We learnt about types of cursors, cursor for loop and parameterized cursor in PL/SQL.

Teacher's Sign: _____