

DBMS Assignment A6

```
mysql> create table RollCall(Rno integer(13),Name varchar(20),Marks integer(5));
Query OK, 0 rows affected, 2 warnings (2.06 sec)
```

```
mysql> insert into RollCall values(1,"Dhruvil",100);
Query OK, 1 row affected (0.22 sec)
```

```
mysql> insert into RollCall values(2,"Soham",99);
Query OK, 1 row affected (0.12 sec)
```

```
mysql> insert into RollCall values(3,"Gaurav",90);
Query OK, 1 row affected (0.09 sec)
```

```
mysql> insert into RollCall values(4,"Sudesh",88);
Query OK, 1 row affected (0.08 sec)
```

```
mysql> insert into RollCall values(5,"Aabha",40);
Query OK, 1 row affected (0.09 sec)
```

```
mysql> insert into RollCall values(6,"Shreyas",84);
Query OK, 1 row affected (0.14 sec)
```

```
mysql> insert into RollCall values(7,"Krupa",80);
Query OK, 1 row affected (0.08 sec)
```

```
mysql> select * from RollCall;
```

Rno	Name	Marks
1	Dhruvil	100
2	Soham	99
3	Gaurav	90
4	Sudesh	88
5	Aabha	40
6	Shreyas	84
7	Krupa	80

```
7 rows in set (0.00 sec)
```

```
mysql> delimiter $$
```

```
mysql> create procedure pass_stud()
```

```
-> begin
```

```
-> declare mrk int;
```

```
-> declare nm varchar(10);
```

```
-> declare v_finish int default 0;
```

```
-> declare cursor1 cursor for select Name, Marks from RollCall where Marks>45;
```

```
-> declare continue handler for not found set v_finish=1;
```

```
-> open cursor1;
```

```
-> get_stud:loop
```

```

-> fetch cursor1 into nm,mrk;
-> select nm,mrk;
-> if v_finish=1 then
-> leave get_stud;
-> end if;
-> end loop get_stud;
-> end$$

```

```
mysql> call pass_stud();
```

```

+-----+-----+
| nm      | mrk    |
+-----+-----+
| Dhruvil | 100    |
+-----+-----+
1 row in set (0.00 sec)

```

```

+-----+-----+
| nm      | mrk    |
+-----+-----+
| Soham   | 99     |
+-----+-----+
1 row in set (0.00 sec)

```

```

+-----+-----+
| nm      | mrk    |
+-----+-----+
| Gaurav  | 90     |
+-----+-----+
1 row in set (0.01 sec)

```

```

+-----+-----+
| nm      | mrk    |
+-----+-----+
| Sudesh  | 88     |
+-----+-----+
1 row in set (0.01 sec)

```

```

+-----+-----+
| nm      | mrk    |
+-----+-----+
| Shreyas | 84     |
+-----+-----+
1 row in set (0.02 sec)

```

```

+-----+-----+
| nm      | mrk    |
+-----+-----+

```

Krupa	80
+-----+	+-----+

1 row in set (0.02 sec)

Query OK, 0 rows affected (0.05 sec)

MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student: Dhruvil Shah	Class: TE Comp 1
Semester/Year: 5th/2020	Roll No: 047
Date of Performance:	Date of Submission:
Examined By:	Experiment No: Part A-06

GROUP: A ASSIGNMENT NO: 06

AIM: Write a PL/SQL block of code for Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor)

PROBLEM STATEMENT:

Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped. Frame the separate problem statement for writing PL/SQL block to implement all types

OBJECTIVES:

- To study cursor programming.
- To learn various cursors attributes.
- To learn different cursor operations.

PRE - REQUISITES:

Interactive SQL commands, PL/SQL programming, use of oracle 11g database Editor.

APPARATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative
- Front End :- Oracle Editor
- Back end: Oracle 11g

CURSOR SYNTAX:

A cursor is a temporary work area created in the system memory when a SQL statement is executed. A cursor contains information on a select statement and the rows of data accessed by it.

This temporary work area is used to store the data retrieved from the database, and manipulate this data. A cursor can hold more than one row, but can process only one row at a time. The set of rows the cursor holds is called the *active* set.

Working with an explicit cursor includes the following steps –

- Declaring the cursor for initializing the memory
- Opening the cursor for allocating the memory
- Fetching the cursor for retrieving the data
- Closing the cursor to release the allocated memory

Declaring the Cursor

Declaring the cursor defines the cursor with a name and the associated SELECT statement. For example –

```
CURSOR c_cusers IS  
    SELECT id, name, address FROM cusers;
```

Opening the Cursor

Opening the cursor allocates the memory for the cursor and makes it ready for fetching the rows returned by the SQL statement into it. For example, we will open the above defined cursor as follows –

```
OPEN c_cusers;
```

Fetching the Cursor

Fetching the cursor involves accessing one row at a time. For example, we will fetch rows from the above-opened cursor as follows –

```
FETCH c_cusers INTO c_id, c_name, c_addr;
```

Closing the Cursor

Closing the cursor means releasing the allocated memory. For example, we will close the above-opened cursor as follows –

```
CLOSE c_cusers;
```

CONCLUSION:

QUESTIONS:

1. What are different types of cursor. Explain each type with syntax.
2. What are the different attributes of cursor?
3. What is the parameterized cursor?

Q1 What are the different types of cursor. Explain each type with a syntax.

Ans The different type of cursors are :

Implicit cursor:

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed when there is no explicit cursor for the statement. Whenever a DML statement is issued, an implicit cursor is associated with this statement.

Explicit cursor:

Programmers are allowed to create named context area to execute their DML operations to get more control over it. The explicit cursor should be defined in the declaration section of the PL/SQL block, and it is created for the 'SELECT' statement that needs to be used in the code.

Syntax:

DECLARE

CURSOR <cursor name> IS <select statement>
<cursor variable declaration>

BEGIN

OPEN <cursor name>;

FETCH <cursor name> INTO <cursor-variable>;

;

;

CLOSE <cursor name>;

END;

Q2 What are the different attributes of the cursor ?

Cursor Attribute	Description
% FOUND	It returns the Boolean result 'True' if the most recent fetch operation fetched a record successfully, else it will return 'False'.
% NOT FOUND	This works oppositely to % FOUND. it will return 'True' if the most recent fetch operation could not be able to fetch any record.
% ISOPEN	It returns boolean result 'True' if the given cursor is already opened else 'False'.
% ROWCOUNT	It returns the numerical value. It gives the actual count of records that got affected by the DML activity.

Q3 What is parameterized cursor?

Ans Oracle allows to pass parameters to cursors that can be used to provide condition with WHERE clause. If parameters are passed to the cursor, the cursor is called parameterized cursor.

Syntax:

```
CURSOR CURSOR_NAME (Variable_name Datatype) IS  
SELECT - - - - -
```