Page No.
Date

DBMSL 31147

Assignment 6

Date of Completion: 15/9/20 Date of Submission: 4/10/20

Title: Cursor

Problem Statement:
Write a PL/SQL block of code using parameterized cursor, chat will merge the data available in the newly created table N_EmpId with data available in the table O_Empid. If the data in the 1st table already exists in the 2nd table, then that data should be skipped.

Learning Objectives:
1) Understand the cursors in SQL.
2) Understanding different types of cursors (parameterized).
3) Learn about data merging.

Learning Outcomes:
1) Implement procedure for merging two table with the help of cursor.
2) Learn parameterized cursors.

~~Theory~~ S/W and H/W requirements
Windows/ubuntu,
MYSQL command line client

Theory:
A cursor in SQL is a temporary work area created in system memory when a SQL statement is executed.
MySQL cursor handle a result set inside a stored procedure, you a cursor. A cursor allows you to iterate a set of rows returned by a query and process each row individually

Declare cursor :

Declare cursor_name CURSOR FOR SELECT_STATEMENT;

Next open the cursor. The open statement initializes the result set for the cursor, therefore you must call the open statement

Open cursor_name;

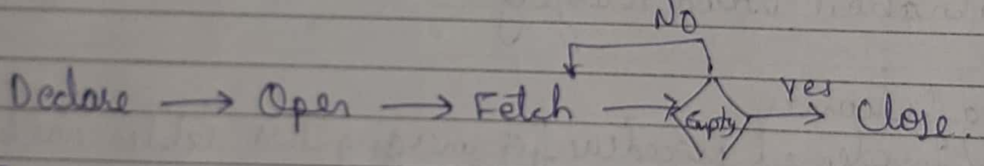
Then Fetch statement to retrieve the next row pointed by cursor.

FETCH cursor_name Into variable_list;

Finally deactivate the cursor and realse the memory associated with it CLOSE operation

CLOSE cursor_name

To declare a NOT FOUND. SET FINISHED = 1



Program Listing

```
CREATE PROCEDURE mergeTable()
BEGIN
DECLARE n_id INT;
DECLARE n_name VARCHAR(100);
DECLARE o_id INT;
DECLARE o_name INT;
CURSOR n_cursor(n INT) is SELECT * FROM n_empid where id=n;
CURSOR o_cursor is SELECT * FROM o_empid;
OPEN o_cursor
loop
FETCH o_cursor INTO n_id, n_name;
EXIT WHEN o_cursor %notfound;
OPEN n_cursor(n_id);
FETCH n_cursor INTO o_id, o_name;
```

```
if n_cursor%notfound THEN
    INSERT INTO nempid.VALUES (n_id,n_name);
endif.
 close n_cursor;
 end loop;
 close o_cursor;
 end;
```

Test Cases:

1) O_empId Table

id	name
1	Pradyumna
2	Sanket
3	Sahil
4	Shubham

2) N_empId Table

id	name
1	Pradyuma
2	Sanket
5	John
6	Eddard.

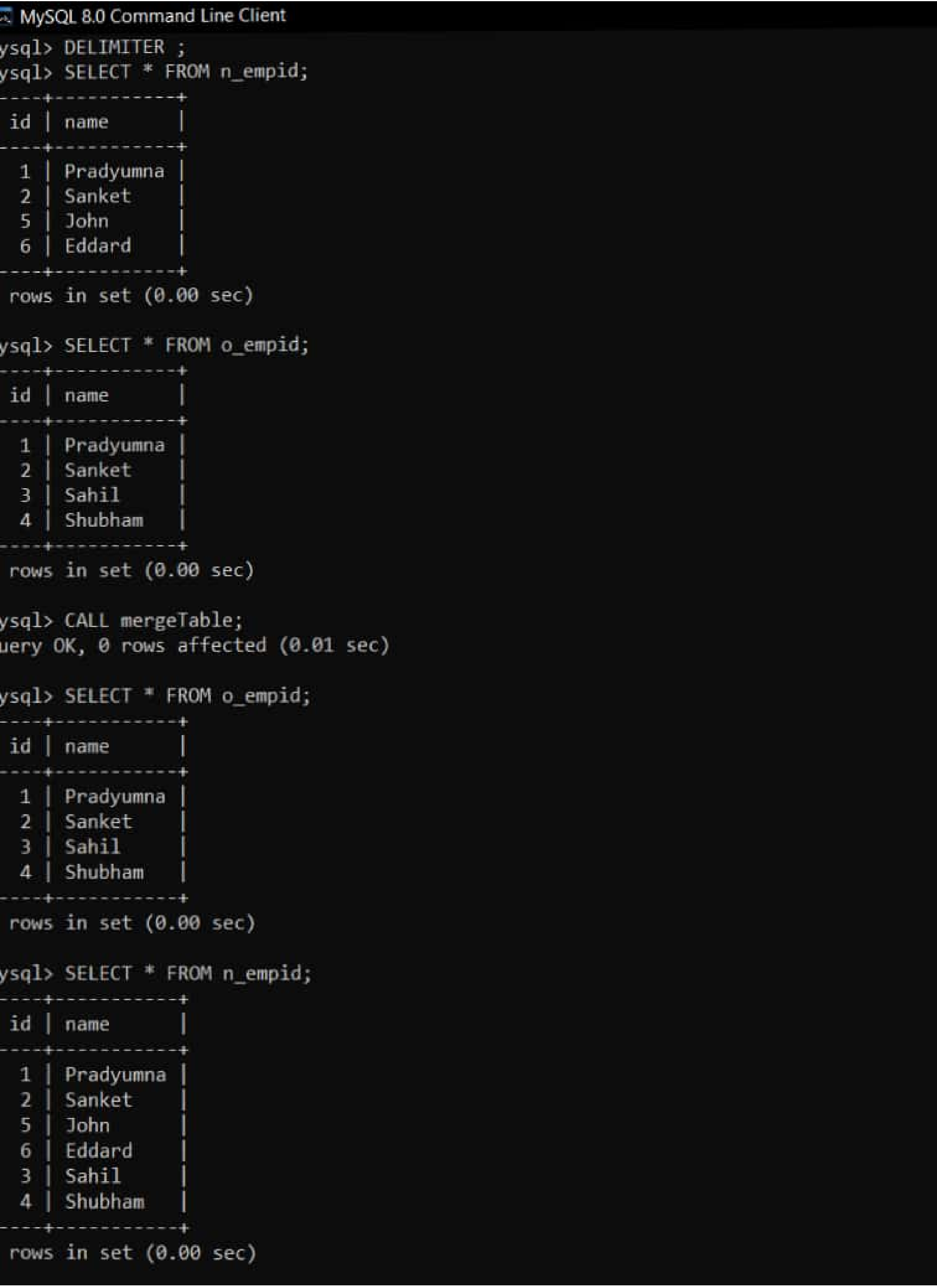
```
CALL mergeTable();
```

N_empid

id	name
1	Pradyumna
2	Sanket
5	John
6	Eddard
3	Sahil
4	Shubham.

Conclusion:

Thus the concept of cursor was understood and implemented for given schema and merged two table.



```
ysql> DELIMITER ;
ysql> SELECT * FROM n_empid;
----+-----------+
 id | name      |
----+-----------+
  1 | Pradyumna |
  2 | Sanket    |
  5 | John      |
  6 | Eddard    |
----+-----------+
 rows in set (0.00 sec)

ysql> SELECT * FROM o_empid;
----+-----------+
 id | name      |
----+-----------+
  1 | Pradyumna |
  2 | Sanket    |
  3 | Sahil     |
  4 | Shubham   |
----+-----------+
 rows in set (0.00 sec)

ysql> CALL mergeTable;
uery OK, 0 rows affected (0.01 sec)

ysql> SELECT * FROM o_empid;
----+-----------+
 id | name      |
----+-----------+
  1 | Pradyumna |
  2 | Sanket    |
  3 | Sahil     |
  4 | Shubham   |
----+-----------+
 rows in set (0.00 sec)

ysql> SELECT * FROM n_empid;
----+-----------+
 id | name      |
----+-----------+
  1 | Pradyumna |
  2 | Sanket    |
  5 | John      |
  6 | Eddard    |
  3 | Sahil     |
  4 | Shubham   |
----+-----------+
 rows in set (0.00 sec)
```