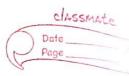
Code

```
mysql> drop database if exists A6;
Query OK, 2 rows affected (0.03 sec)
mysql> create database A6;
Query OK, 1 row affected (0.00 sec)
mysql> use A6;
Database changed
mysql>
mysql> CREATE TABLE O_rollCall(
       roll_no INT NOT NULL,
       name VARCHAR(20)
  ->);
;Query OK, 0 rows affected (0.03 sec)
mysql>
mysql> CREATE TABLE N_rollCall(
  -> roll_no INT NOT NULL,
  ->
       name VARCHAR(20)
  ->);
Query OK, 0 rows affected (0.05 sec)
mysql> INSERT INTO O_rollCall VALUES(1, 'Prathamesh');
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO O_rollCall VALUES(2, 'Aditya');
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO O_rollCall VALUES(3, 'Ram');
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO O_rollCall VALUES(4, 'Sooraj');
Query OK, 1 row affected (0.01 sec)
mysql> INSERT INTO O_rollCall VALUES(5, 'Shreya');
Query OK, 1 row affected (0.00 sec)
mysql>
mysql> INSERT INTO N_rollCall VALUES(1, 'Prathamesh');
Query OK, 1 row affected (0.01 sec)
mysql> INSERT INTO N_rollCall VALUES(3, 'Ram');
```

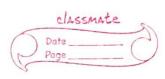
```
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO N_rollCall VALUES(5, 'Shreya');
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO N_rollCall VALUES(6, 'Swatej');
NSERT INTO N_rollCall VALUES(7, 'Yuvraj'); Query OK, 1 row affected (0.01 sec)
mysql> INSERT INTO N_rollCall VALUES(7, 'Yuvraj');
Query OK, 1 row affected (0.01 sec)
mysql> SELECT * FROM O_rollCall;
+----+
| roll_no | name |
+----+
    1 | Prathamesh |
    2 | Aditya
   3 | Ram
    4 | Sooraj
    5 | Shreya
+----+
5 rows in set (0.00 \text{ sec})
mysql> SELECT * FROM N_rollCall;
+----+
| roll_no | name |
    1 | Prathamesh |
    3 | Ram
    5 | Shreya
    6 | Swatej
    7 | Yuvraj
+----+
5 rows in set (0.00 \text{ sec})
mysql> delimiter $$
mysql> CREATE PROCEDURE MergeTables()
  -> BEGIN
  -> DECLARE roll INT;
  -> DECLARE finished INT DEFAULT 0;
  -> DECLARE cur1 cursor for select roll_no from O_rollCall;
  -> DECLARE CONTINUE handler for NOT found set finished = 1;
  -> OPEN cur1;
  ->
  -> loop1: loop
  -> IF finished = 1 THEN
  -> close cur1;
  -> leave loop1;
  -> END IF;
```

```
-> FETCH cur1 INTO roll;
  -> IF NOT EXISTS(select roll_no from N_rollCall where roll_no = roll) THEN
  -> INSERT INTO N_rollCall select * from O_rollCall where roll_no = roll;
  -> END IF;
  -> END loop loop1;
  -> END
  -> $$
Query OK, 0 rows affected (0.02 sec)
mysql> delimiter;
mysql> call MergeTables();
Query OK, 0 rows affected (0.01 sec)
mysql>
mysql>
mysql> SELECT * FROM O_rollCall;
+----+
| roll_no | name
+----+
    1 | Prathamesh |
    2 | Aditya
    3 | Ram
    4 | Sooraj
    5 | Shreya
5 rows in set (0.00 \text{ sec})
mysql> SELECT * FROM N_rollCall;
+----+
| roll no | name
    1 | Prathamesh |
    3 | Ram
    5 | Shreya
    6 | Swatej
    7 | Yuvraj
    2 | Aditya
    4 | Sooraj
+----+
7 rows in set (0.00 \text{ sec})
mysql>
```

	Assignment - 6
	FISSIGNMENT - 6
	Title - Cursors
	Date of Completion: 14/09/2020
_	100 Control of the co
	Date of Submission: 30/09/2020
	100 Hasin soli et sovieti i 1111
	Problem Statement:
	Write a PLISAL block of code using
	parameterized cursor, that will merge
	the data avaliable in newly created table
	N-EmpID with data avalible in the
	O-EmpID Duplicate data must be
	skipped
	+ + + + + + + + + + + + + + + + + + +
	Objective:
	To understand and implement types of
	cursors with PL/SQL block of code
	S.V. 1 4 17 /25 180 181
	Outcomes:
	i) Implement types of cursors
	Luc (3 3 8 1 0 3)
	Theory:
	ORACLE:
	PLISAL:
	* Cursors:
	A cursor is a temporary work area
	created in system memory when
	SQL statement is executed.



A cursor can hold more than one row but can process only one row at a time. Types of Cursors: > Implicit Cursors: 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. 2) Explicit cursors: They must be created when you are executing a select Statement that returns more than one rows. Only one row can be processed at a time. When you fetch a row the current you position moves to next row. Implicit Cursors:) / FOUND refurns 1 if DML statement affected more than I row.



	2) / NOTFOUND -
	returns true if DML statements affected
	no rows.
	3) 1. ISOPEN -
	always returns false for implicit
6	Cursor.
	4) / ROWCOUNT-
	returns number of rows affected by
	an insert, update delete statements.
91	Commercial Some whole
	Syntax:
	Good Lors
	declare
	declaration Statements
	begin
	execution statements
	if sql : not found then
	cles if Sql. / found then
	TOWNS YOUND - : - XAD PARTY IN THE
	endif;
	end; it is the state of
	Signal Francis
4	
4	
-	
-	
+	
+	

	Classmate Date Page
	Explicit Cursors:
4.4/534	CAPTICIT CUYSOTS.
	Syntax:
The second secon	
	declaration
	cursor name is select statement
	begin
	open coname is select statement
400	Open _name_;
Letri 9A	elete estatos loopo inten ano
	fetch = name into variables;
	exit when _ name/attribute
	end loop;
	close _name_
	endimoted molastic
	0000
0	CURSOR FOR 100P
	Sell burten & loc 2
	Syntax:
	and formed the contract to the series
	for recordindex in cursor_name
	10 op
	{ statements }
	end loop;

•	Parameterized cursor:	
	15 151 15 172 CO CO. SO.	
	Syntax:	
	- Cyron.	
_	declare	
	cursor cur-name (parameters)	
	is select-statements;	
	heain	
XXX	open cur-name (params);	
	000	
	fetch currame into variables	
	exit when curname - attribute;	
	¿ statements}	~
	end loop;	~
	close currame;	~
	end;	$\neg \vec{i}$
	as we helpswisign be simile	
•	MSQL Cursors Leson bevol	
	Syntax:	~
	The second that days sold	~
	create procedure name (params)	
	begin 16211 0. NOBLO	~
	declarations	~
	a declare cur-name cursor for	~
	select statements;	~
	open cur-name;	_~
	1009	~
	Fetch cur: name into variable;	~
	if cond then	
	? Statements }	
	end if;	~

	Page
	end loop loopli
	close. Cur-name
	end;
	Test Cases:
G	1 CH3CO
	input expected output Status
	imput expected confur states
	call procs (C) old Success
	2. new cool
	aslor ray of a source 2013 olds
	Ladate de son ano 4 new
	s new
0	a gratha de
	ENVENTION BICKS
•	Conclusion:
	. Thus we implemented cursors in
	Mysal stored procedure with
· ·	parameter.
	· We learnt about types of cursors
	cursor for loop & parameterized
	Cursom in PLISQL.
	elector at ons
	ac declare our nome curve so
	actnownsted tools
- 30	- 2/400 - 4/10 (NO:10)
	shows at a source of state
	asti Phaos 4
	Salara sole