0	Name: Rushikesh Kazbhazi Palve Date: Page No: 1/20 1
	Assignment No.A6
	DOP:- 28-09-2021 DOS:- 30-11-2021
	Tello: Charactic Atherens Tootist tudicit
510	Title: - Cuesocs: (All types: Implicit, Explicit, Cuesock FOR Loop, Poxameterized
THE RESERVE OF THE PARTY OF THE	
	ond for allocated on the oracle server
pour no	w door the chies bodologicale of colour
	Problem Definition:
	TO troined to the content of
	Write a pr/sqr block of code using parameterized
	cuesoa, that will merge the data available
	in the newly created table N-Emptd with the
	data available in the table atmptd. If the
	data in the effect table already exist in the
	second table then the data should be stipped.
.23	toppliest circuse the every part statemen
1- 51	
	olobjectives pi- and in this is to
(10120301)	1 Undewtand the types of cursons.
hamaya.	in Understand how to use cursons with privar
(Newsell)	block. 536500 D populso de
	30 is palage de cueros
301	पाठ कितानता तक most रहाक क्षेत्रीतिक के न
	Outcomes:- source paids
	- Incold the latest the second
	After completion of the assignment students will
	be able to recent to the leasest
	1 Undecitand the roncept of newows
	1 Use cuesoes with Pi/sqs block.
The same of the same of	

ROLL NO. 31258

9

exception. When this happens, exerction in the nuceent block terminates and rontrol is passed to the exception section. Unless you deliberately plan to handle this srenacio, use of the implicit cuesor is a doctaration of faith. You are saying, "I trust that query to always roturn a single row!"

It may stell be that today, with the rucent data, the query will only setuen a single sow. If the nation of the data ever rhanges, however, you may find the SELECT statement which towners the recent indicated a single sow now setums several. Your program will raise an exception. Pechaps this is what you will want. On the other hand, pechaps the presence of additional serveds is innonsequential and should be ignored. With the implicit query you rannot easily handle these different possibilities. With an explicit query, your program will be protected against changes in data and will rontinue to fetch rows without raising exceptions.

Conclusion :-

We successfully understood the roncept of Cursors, cursor types. Also, used russors with pl/sql block.

Scanned by TapScanner

```
OUTPUT :-
/*
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.
*/
mysql> CREATE TABLE O_Employee(
    -> EmpId INT PRIMARY KEY,
    -> EName VARCHAR(20),
    -> DId VARCHAR(20),
    -> DName VARCHAR(15)
    -> );
mysql> INSERT INTO O_Employee VALUES
    -> (1, 'Vidyut', 10, 'Computer'),
    -> (2, 'Pratap', 20, 'IT'),
    -> (3, 'Kailash', 30, 'E&TC'),
    -> (4, 'Mukund', 40, 'Mechanical'),
    -> (5, 'Girish', 50, 'Civil'),
    -> (6, 'Neeraj', 60, 'Electrical'),
    -> (7, 'Prashant', 30, 'E&TC'),
    -> (8, 'Raj', 10, 'Computer'),
    -> (9, 'Hari', 20, 'IT'),
    -> (10, 'Aditya', 40, 'Mechanical');
mysql> CREATE TABLE N_Employee(
    -> EmpId INT PRIMARY KEY,
    -> EName VARCHAR(20),
    -> DId VARCHAR(20),
    -> DName VARCHAR(15)
    -> );
mysql> # Procedure Without Cursor
mysql> DELIMITER $$
mysql> CREATE PROCEDURE Copy(
    -> IN eID INT
    -> )
    -> BEGIN
    -> DECLARE flag BOOLEAN;
    -> DECLARE EXIT HANDLER FOR SQLEXCEPTION SELECT 'ENTRY NOT FOUND' AS EXCEPTION;
    -> SET flag = FALSE;
    -> IF NOT EXISTS(SELECT * FROM O_Employee WHERE EmpId = eID) THEN
    -> SIGNAL SQLSTATE '45000';
    -> END IF;
    -> IF NOT EXISTS(SELECT * FROM N_Employee WHERE EmpId = eID ) THEN
    -> SET flag = TRUE;
    -> INSERT INTO N_Employee
    -> SELECT * FROM O_Employee
    -> WHERE EmpId = eID;
    -> END IF;
    -> IF NOT flag THEN
    -> SELECT 'Record Already Exists' AS MESSAGE;
    -> ELSE
    -> SELECT * FROM N_Employee;
                                                        Scanned by TapScanner
```

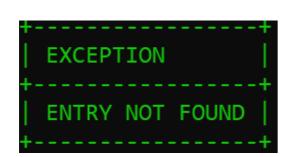
```
-> END IF;
-> END $$
```

mysql> # Procedure With Cursor

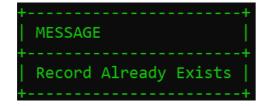
```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE NewCopy(
    -> IN eID INT
    -> )
    -> BEGIN
    -> DECLARE flag BOOLEAN;
    -> DECLARE C1 CURSOR FOR SELECT EmpId FROM O_Employee WHERE EmpId = eID;
    -> DECLARE EXIT HANDLER FOR NOT FOUND SELECT 'ENTRY NOT FOUND' AS EXCEPTION;
    -> OPEN C1;
    -> FETCH C1 INTO eID;
    -> SET flag = FALSE;
    -> IF NOT EXISTS(SELECT * FROM N_Employee WHERE EmpId = eID ) THEN
    -> SET flag = TRUE;
    -> INSERT INTO N_Employee
    -> SELECT * FROM O_Employee
    -> WHERE EmpId = eID;
    -> END IF;
    -> IF NOT flag THEN
    -> SELECT 'Record Already Exists' AS MESSAGE;
    -> ELSE
    -> SELECT * FROM N_Employee;
    -> END IF;
    -> CLOSE C1;
    -> END $$
      mysql> DELIMITER ;
      mysql> CALL Copy(1);
      mysql> CALL Copy(2);
      mysql> CALL Copy(3);
      mysql> CALL Copy(4);
      mysql> CALL Copy(5);
```

```
EmpId |
       EName
                 DId
                         DName
       Vidyut
                  10
                         Computer
       Pratap
                  20
                         IT
       Kailash
                 30
                         E&TC
       Mukund
                  40
                         Mechanical
                  50
       Girish
                         Civil
```

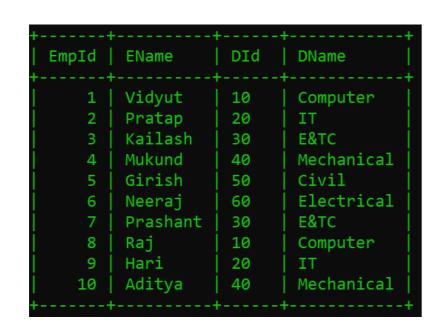
mysql> CALL Copy(11);



mysql> CALL NewCopy(1);



```
mysql> CALL NewCopy(6);
mysql> CALL NewCopy(7);
mysql> CALL NewCopy(8);
mysql> CALL NewCopy(9);
mysql> CALL NewCopy(10);
```



mysql> CALL NewCopy(11);

