pratical 6

Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized
Cursor)
Write a PL/SQL block of code using parameterized Cursor that will merge the
data availablein
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.
Note: Instructor will frame the problem statement for writing PL/SQL block
using all types of
Cursors in line with above statement.

```
-- Create the N_RollCall table (new data)
CREATE TABLE N_RollCall (
Roll INT PRIMARY KEY,
Name VARCHAR(50)
);

-- Create the O_RollCall table (old data)
CREATE TABLE O_RollCall (
Roll INT PRIMARY KEY,
Name VARCHAR(50)
);



-- Insert data into N_RollCall (new records)
INSERT INTO N_RollCall (Roll, Name) VALUES (1, 'Alice'), (2, 'Bob'), (3,
'Charlie');

-- Insert data into O_RollCall (existing records)
INSERT INTO O_RollCall (Roll, Name) VALUES (2, 'Bob'), (4, 'David');



DELIMITER $$

CREATE PROCEDURE merge_RollCall()
BEGIN
DECLARE v_Roll INT;
DECLARE v_Name VARCHAR(50);
DECLARE done INT DEFAULT 0;
```

```sql
-- Declare a cursor to iterate through N_RollCall
DECLARE cur CURSOR FOR
SELECT Roll, Name FROM N_RollCall;

-- Declare a handler for when the cursor reaches the end
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

-- Open the cursor
OPEN cur;

-- Loop through each row in N_RollCall
read_loop: LOOP
-- Fetch the current row from the cursor
FETCH cur INTO v_Roll, v_Name;

-- Exit the loop if the cursor is done
IF done THEN
LEAVE read_loop;
END IF;

-- Check if the record already exists in O_RollCall
IF NOT EXISTS (SELECT 1 FROM O_RollCall WHERE Roll = v_Roll) THEN
-- If the record doesn't exist, insert it into O_RollCall
INSERT INTO O_RollCall (Roll, Name) VALUES (v_Roll, v_Name);
END IF;
END LOOP;

-- Close the cursor
CLOSE cur;

END$$

DELIMITER ;



-- Call the procedure to merge the data
CALL merge_RollCall();



-- Verify the merged data in O_RollCall
SELECT * FROM O_RollCall;
```