Question 6

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.**

Solution

1. **Create the Tables First, N_RollCall and O_RollCall tables.**

```
CREATE DATABASE ROLLCALL;


USE ROLLCALL;


CREATE TABLE N_RollCall (
  student_id INT PRIMARY KEY,
  student_name VARCHAR(255),
  birth_date DATE
);


CREATE TABLE O_RollCall (
  student_id INT PRIMARY KEY,
  student_name VARCHAR(255),
  birth_date DATE
);
INSERT INTO O_RollCall (student_id, student_name, birth_date)
  VALUES
    (1, 'Amit', '1995-08-15'),
    (3, 'Chinmay', '1990-12-10');
```

Insert some sample data into the N_RollCall table, including records that are common with O_RollCall

```
INSERT INTO N_RollCall (student_id, student_name, birth_date)
  VALUES
     (1, 'Amit', '1995-08-15'),  -- Common record with O_RollCall
     (2, 'Nishmitha', '1998-03-22'),
     (3, 'Chinmay', '1990-12-10'), -- Common record with O_RollCall
     (4, 'Devang', '2000-05-18'),
     (5, 'Manish', '1997-09-03');
```

----

2. Define the Stored Procedure and define the merge_rollcall_data stored procedure to merge records from N_RollCall into O_RollCall, skipping existing records:

```
DELIMITER //
CREATE PROCEDURE merge_rollcall_data()
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE n_id INT;
  DECLARE n_name VARCHAR(255);
  DECLARE n_birth_date DATE;

  -- Declare cursor for N_RollCall table
  DECLARE n_cursor CURSOR FOR
    SELECT student_id, student_name, birth_date
    FROM N_RollCall;

  -- Declare handler for cursor
  DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET done = TRUE;

  -- Open the cursor
  OPEN n_cursor;

  -- Start looping through cursor results
  cursor_loop: LOOP
    -- Fetch data from cursor into variables
    FETCH n_cursor INTO n_id, n_name, n_birth_date;

    -- Check if no more rows to fetch
    IF done THEN
      LEAVE cursor_loop;
    END IF;

    -- Check if the data already exists in O_RollCall
    IF NOT EXISTS (
      SELECT 1
      FROM O_RollCall
      WHERE student_id = n_id
    ) THEN
      -- Insert the record into O_RollCall
      INSERT INTO O_RollCall (student_id, student_name, birth_date)
      VALUES (n_id, n_name, n_birth_date);
    END IF;
  END LOOP;

  -- Close the cursor
  CLOSE n_cursor;
```

```
END//


DELIMITER ;
```

- ➢ The stored procedure merge_rollcall_data uses a cursor (n_cursor) to iterate through the records of the N_RollCall table.
- ➢ Inside the cursor loop (cursor_loop), each record (n_id, n_name, n_date) from N_RollCall is fetched and checked against the O_RollCall table.
- ➢ If the record does not already exist in O_RollCall (checked using NOT EXISTS), it is inserted into O_RollCall.
- ➢ The cursor loop continues until all records from N_RollCall have been processed.
- ➢ The cursor is then closed (CLOSE n_cursor).

---

4. Execute the Stored Procedure

Finally, execute the merge_rollcall_data stored procedure to merge records from N_RollCall into O_RollCall while skipping existing records:

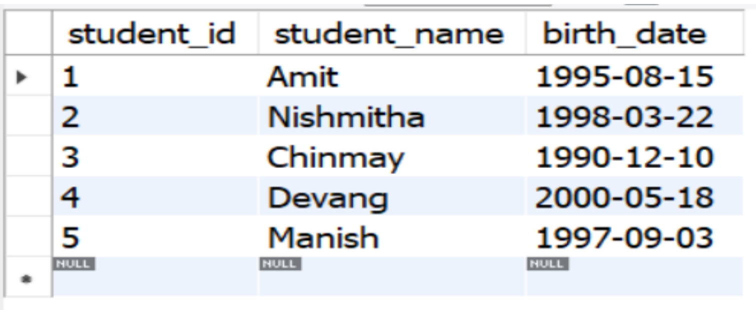
```
CALL merge_rollcall_data( );
```

---

5. Verify Records in O_RollCall

After executing the procedure, verify the records in the O_RollCall table to confirm that new records from N_RollCall have been inserted, while existing common records have been skipped:

```
SELECT * FROM O_RollCall;
```



	student_id	student_name	birth_date
▸	1	Amit	1995-08-15
	2	Nishmitha	1998-03-22
	3	Chinmay	1990-12-10
	4	Devang	2000-05-18
	5	Manish	1997-09-03
*	NULL	NULL	NULL