

## EXPERIMENT - 6

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.

### 1. Create the Tables

```
CREATE DATABASE ROLLCALL;

USE ROLLCALL;

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

-- Create O_RollCall table with common data
CREATE TABLE O_RollCall (
    student_id INT PRIMARY KEY,
    student_name VARCHAR(255),
    birth_date DATE
);
```

### 2. Add Sample Records to both tables

```
-- Insert common data into O_RollCall
INSERT INTO O_RollCall (student_id, student_name, birth_date)
VALUES
    (1, 'Shivanna', '1995-08-15'),
    (3, 'Cheluva', '1990-12-10');
```

```
Records: 2  Duplicates: 0  Warnings: 0
```

## EXPERIMENT - 6

```
-- Insert sample records into N_RollCall
INSERT INTO N_RollCall (student_id, student_name, birth_date)
VALUES
  (1, 'Shivanna', '1995-08-15'), -- Common record with O_RollCall
  (2, 'Bhadramma', '1998-03-22'),
  (3, 'Cheluva', '1990-12-10'), -- Common record with O_RollCall
  (4, 'Devendra', '2000-05-18'),
  (5, 'Eshwar', '1997-09-03');
```

```
Records: 5 Duplicates: 0 Warnings: 0
```

### 3. Define the Stored Procedure

```
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;
```

## EXPERIMENT - 6

```
-- 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//
```

## EXPERIMENT - 6

DELIMITER;

4. Execute the Stored Procedure

```
CALL merge_rollcall_data();
```

5. Verify Records in O\_RollCall

```
-- Select all records from O_RollCall  
SELECT * FROM O_RollCall;
```

```
+-----+-----+-----+  
| student_id | student_name | birth_date |  
+-----+-----+-----+  
|          1 | Shivanna     | 1995-08-15 | ← Common record, not duplicated  
|          2 | Bhadramma    | 1998-03-22 | ← New record from N_RollCall  
|          3 | Cheluva      | 1990-12-10 | ← Common record, not duplicated  
|          4 | Devendra     | 2000-05-18 | ← New record from N_RollCall  
|          5 | Eshwar       | 1997-09-03 | ← New record from N_RollCall  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```