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

To accomplish this task in MySQL, we can use a stored procedure with a parameterized cursor to merge data from one table (N\_RollCall) into another table (O\_RollCall) while skipping existing data. We’ll iterate through the records of N\_RollCall and insert them into O\_RollCall only if they do not already exist.

1. Create the Tables

First, let’s create the N\_RollCall and O\_RollCall tables with similar structure:

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

Let’s insert some sample data into the O\_RollCall table:

mysql> -- Insert common data into O\_RollCall

mysql> INSERT INTO O\_RollCall (student\_id, student\_name, birth\_date)

-> VALUES

-> (1, 'Shivanna', '1995-08-15'),

-> (3, 'Cheluva', '1990-12-10');

Query OK, 2 rows affected (0.17 sec)

Records: 2 Duplicates: 0 Warnings: 0

Let’s insert some sample data into the N\_RollCall table, including records that are common with O\_RollCall:

mysql> -- Insert sample records into N\_RollCall

mysql> 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');

Query OK, 5 rows affected (0.21 sec)

Records: 5 Duplicates: 0 Warnings: 0

3. Define the Stored Procedure

Next, let’s 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:

mysql> CALL merge\_rollcall\_data();

Query OK, 0 rows affected (0.87 sec)

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:

mysql> -- Select all records from O\_RollCall

mysql> 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)