

8. 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 available in the newly created table N_Roll_Call with the data available in the table O_Roll_Call. If the data in the first table already exists in the second table then that data should be skipped.

Define Tables: Ensure both tables have been created with the same structure.

```
sql Copy code CREATE TABLE N_Roll_Call ( Roll_no INT, Name VARCHAR2(100), Class VARCHAR2(50) );
```

```
CREATE TABLE O_Roll_Call ( Roll_no INT PRIMARY KEY, Name VARCHAR2(100), Class VARCHAR2(50) );
```

PL/SQL Block with Parameterized Cursor:

```
sql Copy code DECLARE -- Define a parameterized cursor that selects records from N_Roll_Call
CURSOR c_new_data(p_roll_no INT) IS SELECT Roll_no, Name, Class FROM N_Roll_Call WHERE Roll_no = p_roll_no;
```

```
v_name N_Roll_Call.Name%TYPE; v_class N_Roll_Call.Class%TYPE;
v_exists INT;
```

```
BEGIN -- Loop through each record in N_Roll_Call
FOR rec IN (SELECT Roll_no FROM N_Roll_Call) LOOP -- Check if Roll_no from N_Roll_Call already exists in O_Roll_Call
SELECT COUNT(*) INTO v_exists FROM O_Roll_Call WHERE Roll_no = rec.Roll_no;
```

```
-- If the record doesn't exist in O_Roll_Call, insert it
IF v_exists = 0 THEN
-- Open the parameterized cursor
OPEN c_new_data(rec.Roll_no);
FETCH c_new_data INTO rec.Roll_no, v_name, v_class;
```

```
-- Insert the new data into O_Roll_Call
INSERT INTO O_Roll_Call (Roll_no, Name, Class) VALUES (rec.Roll_no, v_name, v_class);
```

```
-- Close the cursor after each insert
CLOSE c_new_data;
END IF;
END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE('Data merged successfully.');
```

```
EXCEPTION WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM); END;
```