1. The bank manager has decided to activate all those accounts which were previously marked as inactive for performing no transaction in last 365 days. Write a PL/SQ block (using implicit cursor) to update the status of account, display an approximate message based on the no. of rows affected by the update. (Use of %FOUND, %NOTFOUND, %ROWCOUNT)
2. 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. output:
3. Write the PL/SQL block for following requirements using parameterized Cursor: Consider table EMP(e\_no, d\_no, Salary), department wise average salary should be inserted into new table dept\_salary(d\_no, Avg\_salary)

a) PL/SQL Block to Activate Inactive Accounts:

```sql

DECLARE

v\_rows\_updated NUMBER := 0;

BEGIN

-- Update the status of inactive accounts

UPDATE accounts

SET status = 'Active'

WHERE last\_transaction\_date < SYSDATE - 365;

v\_rows\_updated := SQL%ROWCOUNT;

DBMS\_OUTPUT.PUT\_LINE('Approximately ' || v\_rows\_updated || ' accounts have been activated.');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;

/

```

In this PL/SQL block, we use an implicit cursor in the form of an `UPDATE` statement to activate accounts that were previously marked as inactive due to no transactions in the last 365 days. We display a message indicating the approximate number of accounts that have been activated.

b) PL/SQL Block to Merge Data from N\_RollCall to O\_RollCall:

```sql

DECLARE

CURSOR n\_rollcall\_cur IS

SELECT \* FROM n\_rollcall;

v\_n\_rollcall\_rec n\_rollcall%ROWTYPE;

v\_exists NUMBER;

BEGIN

FOR v\_n\_rollcall\_rec IN n\_rollcall\_cur LOOP

-- Check if the data already exists in o\_rollcall

SELECT COUNT(\*) INTO v\_exists

FROM o\_rollcall

WHERE e\_no = v\_n\_rollcall\_rec.e\_no

AND d\_no = v\_n\_rollcall\_rec.d\_no;

-- Insert the data if it doesn't exist in o\_rollcall

IF v\_exists = 0 THEN

INSERT INTO o\_rollcall (e\_no, d\_no, salary)

VALUES (v\_n\_rollcall\_rec.e\_no, v\_n\_rollcall\_rec.d\_no, v\_n\_rollcall\_rec.salary);

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Data from N\_RollCall has been merged into O\_RollCall.');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;

/

```

In this PL/SQL block, we use a parameterized cursor to fetch data from the N\_RollCall table and merge it into the O\_RollCall table. We check if the data already exists in O\_RollCall before inserting it.

c) PL/SQL Block to Calculate and Insert Department-Wise Average Salary:

```sql

DECLARE

CURSOR dept\_salary\_cur IS

SELECT d\_no, AVG(salary) AS avg\_salary

FROM emp

GROUP BY d\_no;

v\_dept\_salary\_rec dept\_salary%ROWTYPE;

BEGIN

FOR v\_dept\_salary\_rec IN dept\_salary\_cur LOOP

-- Insert the department-wise average salary into dept\_salary table

INSERT INTO dept\_salary (d\_no, avg\_salary)

VALUES (v\_dept\_salary\_rec.d\_no, v\_dept\_salary\_rec.avg\_salary);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Department-wise average salaries have been inserted into dept\_salary.');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;

/

```

In this PL/SQL block, we use a parameterized cursor to calculate the average salary for each department and insert the department-wise average salary into the `dept\_salary` table.