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)

```
SQL> select * from acc;

   ACC_ID LAST_TRAN STATUS
---------- --------- --------------------------------------------------
         1 01-OCT-23 Active
         2 11-AUG-22 Inactive
         3 28-JUN-22 Inactive
         4 04-DEC-23 active
```

```
  Declare
            inactive_count number :=0;
 Begin
          for acc_rec in(select acc_id from acc where last_transc_date< SYSDATE- 365) loop
                 update acc set status='Active' where acc_id = acc_rec.acc_id;
                  if SQL%FOUND then
                        inactive_count := inactive_count + SQL%ROWCOUNT;
                  end if;
          end loop;
          if inactive_count =0 then
                 dbms_output.put_line('No inactive account found');
          elsif inactive_count =1 then
                 dbms_output.put_line('1 account has been reactivated');
         else
                 dbms_output.put_line(inactive_count ||'account has been reactivated');
          END IF;
  end;
  /
```

Output:

```
2account has been reactivated

PL/SQL procedure successfully completed.
```

2.Organization has decided to increase the salary of employees by 10% of existing salary, who are having salary less than average salary of organization, Whenever such salary updates takes place, a record for the same is maintained in the increment_salary table.
EMP (E_no, Salary)
increment_salary(E_no, Salary)

```
SQL> select * from Emp;

      E_NO     SALARY
---------- ----------
         1      55000
         2      60000
         3      49500
         4      70000
         5      60500
```

```
 Declare
             avg_salary number;
              Cursor emp_cursor is select e_no, salary from emp;
    Begin
              select avg(salary) into avg_salary from emp;
              for emp_rec in emp_cursor loop
                     if emp_rec.salary < avg_salary then
                             insert into increment_salary values(emp_rec.E_no, emp_rec.salary,
emp_rec.salary * 1.10, SYSDATE);
                     update emp set salary = emp_rec.salary * 1.10 where E_no = emp_rec.E_no ;
                     end if;
             end loop;
             dbms_output.put_line('salary updates completed successfully');
   end;
   /
```

Output:

```
salary updates completed successfully

PL/SQL procedure successfully completed.

 select * from increment_salary;

      E_NO OLD_SALARY NEW_SALARY INCREMENT
---------- ---------- ---------- ---------
         1      50000      55000 25-FEB-24
         3      45000      49500 25-FEB-24
         5      55000      60500 25-FEB-24
```

3.Write PL/SQL block using explicit cursor for following requirements:
College has decided to mark all those students detained (D) who are having attendance less than 75%. Whenever such update takes place, a record for the same is maintained in the D_Stud table.
create table stud21(roll number(4), att number(4), status varchar(1));
create table d_stud(roll number(4), att number(4));

```
select * from stud21;

      ROLL        ATT S
---------- ---------- -
         1         70
         2         80
         3         65
         4         90
         5         55
Declare
          cursor stud_cursor is select roll ,att from stud21 where att<75;
  Begin
          open stud_cursor;
          for stud_rec in stud_cursor loop
                 update stud21 set status='D' where roll= stud_rec.roll;
                insert into d_stud values(stud_rec.roll, stud_rec.att);
          end loop;
          close stud_cursor;
         dbms_output.put_line('Detention updates completed successfully');
  end;
  /
```

Output:

```
Detention updates completed successfully.

PL/SQL procedure successfully completed.

SQL> select * from d_stud;

      ROLL        ATT
---------- ----------
         1         70
         3         65
         5         55
```

4.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.
parameterized Cursor

```
 select * from N_RollCall;

ROLL_CALL_ID STUDENT_ID A
------------ ---------- -
           1        101 P
           1        102 A
           2        101 P
           2        102 P

DECLARE
   CURSOR merge_cursor (p_roll_call_id n_rollcall.roll_call_id%TYPE) IS
      SELECT roll_call_id, student_id, attendance_status
      FROM n_rollcall
      WHERE roll_call_id = p_roll_call_id;

   v_count NUMBER;
BEGIN
   FOR merge_rec IN merge_cursor(1) LOOP
      SELECT COUNT(*)
      INTO v_count
      FROM o_rollcall
      WHERE roll_call_id = merge_rec.roll_call_id
      AND student_id = merge_rec.student_id;

      IF v_count = 0 THEN
         INSERT INTO o_rollcall (roll_call_id, student_id, attendance_status)
         VALUES (merge_rec.roll_call_id, merge_rec.student_id,
merge_rec.attendance_status);
      ELSE
         DBMS_OUTPUT.PUT_LINE('Data already exists for roll_call_id ' ||
merge_rec.roll_call_id || ' and student_id ' || merge_rec.student_id || '. Skipping...');
      END IF;
   END LOOP;

   COMMIT;

   DBMS_OUTPUT.PUT_LINE('Merge completed successfully.');
END;
/
```

Output:

```
Merge completed successfully.
```

```
PL/SQL procedure successfully completed.

SQL> select * from O_RollCall;

ROLL_CALL_ID STUDENT_ID A
------------ ---------- -
           1        101 P
           1        102 A
```

5.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)

```
 select * from emp;

      E_NO       D_NO     SALARY
---------- ---------- ----------
         1        101      50000
         2        101      60000
         3        102      45000
         4        102      70000
         5        103      55000
```

```
DECLARE
   v_d_no EMP.d_no%TYPE;
   v_avg_salary NUMBER;

   CURSOR dept_cursor (p_d_no EMP.d_no%TYPE) IS
      SELECT Salary
      FROM EMP
      WHERE d_no = p_d_no;

   v_total_salary NUMBER := 0;
   v_employee_count NUMBER := 0;
BEGIN
   FOR dept_rec IN (SELECT DISTINCT d_no FROM EMP) LOOP
      v_total_salary := 0;
      v_employee_count := 0;

      FOR emp_rec IN dept_cursor(dept_rec.d_no) LOOP
         v_total_salary := v_total_salary + emp_rec.Salary;
         v_employee_count := v_employee_count + 1;
      END LOOP;

      IF v_employee_count > 0 THEN
         v_avg_salary := v_total_salary / v_employee_count;
```

```
        ELSE
            v_avg_salary := 0;
        END IF;

        INSERT INTO dept_salary (d_no, Avg_salary)
        VALUES (dept_rec.d_no, v_avg_salary);
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Department-wise average salary inserted successfully.');
END;
/
```

Output:

```
PL/SQL procedure successfully completed.

SQL> select * from dept_salary;

      D_NO AVG_SALARY
---------- ----------
       101      55000
       102      57500
       103      55000
```

6.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)

```
DECLARE
    CURSOR stud_cursor IS
        SELECT roll, att
        FROM stud21
        WHERE att < 75;

    v_roll stud21.roll%TYPE;
    v_att stud21.att%TYPE;
BEGIN
    FOR stud_rec IN stud_cursor LOOP
        UPDATE stud21
        SET status = 'D'
        WHERE roll = stud_rec.roll;

        INSERT INTO d_stud (roll, att)
        VALUES (stud_rec.roll, stud_rec.att);
    END LOOP;
```

```
   COMMIT;

   DBMS_OUTPUT.PUT_LINE('Detention updates completed successfully.');
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
/
```

Output:

```
PL/SQL procedure successfully completed.

SQL>
SQL> select * from d_stud;

      ROLL        ATT
---------- ----------
         1         70
         3         65
         5         55
         1         70
         3         65
         5         55

6 rows selected.
```