Assignment No.5

Implicit Cursor

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)

```
CREATE TABLE ACCOUNT (
   NAME VARCHAR2(50),
   ACC_NO INT,
   STATUS VARCHAR2(20)
);
CREATE TABLE TRANSACTION (
   ACCOUNT_NO INT,
   TYPE VARCHAR(10),
   AMOUNT DECIMAL(10,2),
   DATE_OF_TRANSACTION DATE
);
INSERT INTO ACCOUNT (NAME, ACC_NO, STATUS) VALUES ('ATHARVA', 1,'INACTIVE');
INSERT INTO ACCOUNT (NAME, ACC_NO, STATUS) VALUES ('SOHAM',2,'INACTIVE');
INSERT INTO ACCOUNT (NAME, ACC_NO, STATUS) VALUES ('VEDANT', 3,'INACTIVE');
INSERT INTO ACCOUNT (NAME, ACC_NO, STATUS) VALUES ('HARI', 4, 'INACTIVE');
INSERT INTO ACCOUNT (NAME, ACC_NO, STATUS) VALUES ('MRNUANLINI',5, 'INACTIVE');
INSERT INTO TRANSACTION (ACCOUNT_NO, TYPE, AMOUNT, DATE_OF_TRANSACTION) VALUES (1, 'Withdrawal',
100.00, DATE '2022-02-21');
INSERT INTO TRANSACTION (ACCOUNT_NO, TYPE, AMOUNT, DATE_OF_TRANSACTION) VALUES (2, 'Deposit', 200.00,
DATE '2023-02-21');
INSERT INTO TRANSACTION (ACCOUNT_NO, TYPE, AMOUNT, DATE_OF_TRANSACTION) VALUES (3, 'Withdrawal',
50.00, DATE '2024-02-20');
INSERT INTO TRANSACTION (ACCOUNT_NO, TYPE, AMOUNT, DATE_OF_TRANSACTION) VALUES (4, 'Deposit', 300.00,
DATE '2024-02-19');
```

```
SQL> SELECT * FROM ACCOUNT;

NAME                           ACC_NO STATUS
------------------------------ ------ ------------------------------
ATHARVA                             1 INACTIVE
SOHAM                               2 INACTIVE
VEDANT                              3 INACTIVE
HARI                                4 INACTIVE
MRNUANLINI                          5 INACTIVE
```

```
SQL> SELECT * FROM TRANSACTION;

ACCOUNT_NO TYPE           AMOUNT DATE_OF_T
---------- ---------- ---------- ---------
         1 Withdrawal        100 21-FEB-22
         2 Deposit           200 21-FEB-23
         3 Withdrawal         50 20-FEB-24
         4 Deposit           300 19-FEB-24
```

```
SQL> DECLARE
  2     ACTIVE_COUNT INT := 0;
  3     INACTIVE_COUNT INT := 0;
  4  BEGIN
  5     FOR Account_stats IN (SELECT DISTINCT ACCOUNT_NO FROM TRANSACTION WHERE SYSDATE -
```

```
DATE_OF_TRANSACTION < 365) LOOP
  6       UPDATE ACCOUNT SET STATUS = 'ACTIVE' WHERE ACC_NO = Account_stats.ACCOUNT_NO;
  7       IF SQL%FOUND THEN
  8          ACTIVE_COUNT := ACTIVE_COUNT + SQL%ROWCOUNT;
  9       END IF;
 10     END LOOP;
 11     SELECT COUNT(NAME) INTO INACTIVE_COUNT FROM ACCOUNT WHERE STATUS='INACTIVE';
 12     DBMS_OUTPUT.PUT_LINE('Number of Active Accounts: ' || ACTIVE_COUNT);
 13     DBMS_OUTPUT.PUT_LINE('Number of Inactive Accounts: ' || INACTIVE_COUNT);
 14  END;
 15  /
Number of Active Accounts: 2
Number of Inactive Accounts: 3
PL/SQL procedure successfully completed.
```

```
SQL> SELECT * FROM ACCOUNT;

NAME                              ACC_NO STATUS
------------------------------ ---------- ------------------------------
ATHARVA                                 1 INACTIVE
SOHAM                                   2 INACTIVE
VEDANT                                  3 ACTIVE
HARI                                    4 ACTIVE
MRNUANLINI                              5 INACTIVE
```

EXPLICIT CURSOR:

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)

```
CREATE TABLE EMP (
   E_no INT PRIMARY KEY,
   Salary DECIMAL(10, 2)
);
CREATE TABLE increment_salary (
   E_no INT,
   Salary DECIMAL(10, 2)
);
INSERT INTO EMP (E_no, Salary) VALUES (1, 20000.00);
INSERT INTO EMP (E_no, Salary) VALUES (2, 80000.00);
INSERT INTO EMP (E_no, Salary) VALUES (3, 10000.00);
INSERT INTO EMP (E_no, Salary) VALUES (4, 70000.00);
INSERT INTO EMP (E_no, Salary) VALUES (5, 55000.00);
INSERT INTO EMP (E_no, Salary) VALUES (6, 40000.00);
INSERT INTO EMP (E_no, Salary) VALUES (7, 5000.00);
SQL> SELECT * FROM EMP;
    E_NO    SALARY
---------- ----------
       1    20000
       2    80000
       3    10000
       4    70000
       5    55000
       6    40000
```

```
      7      5000

7 rows selected.
SQL> DECLARE
  2     AVG_SAL DECIMAL(10, 2);
  3     emp INT;
  4  BEGIN
  5     SELECT AVG(Salary) INTO AVG_SAL FROM EMP;
  6
  7     FOR emp IN (SELECT E_no, Salary FROM EMP WHERE Salary < AVG_SAL) LOOP
  8        IF SQL%FOUND THEN
  9           INSERT INTO increment_salary (E_no, Salary) VALUES (emp.E_no, emp.Salary * 1.1);
 10        END IF;
 11     END LOOP;
 12  END;
 13  /
PL/SQL procedure successfully completed.
```

```
SQL> SELECT * FROM increment_salary;

      E_NO     SALARY
---------- ----------
         1      22000
         3      11000
         7       5500
```

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));
create table stud21(roll number(4) PRIMARY KEY, att number(4), status varchar(2));
create table d_stud(roll number(4), att number(4));

INSERT INTO STUD21 (roll,att,status) VALUES (1,76,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (2,99,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (3,88,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (4,46,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (5,96,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (6,86,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (7,56,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (8,66,'ND');
SELECT * FROM STUD21;
```

```
SQL> SELECT * FROM STUD21;

      ROLL        ATT STATUS
---------- ---------- ------------------------------
         1         76 ND
         2         99 ND
         3         88 ND
         4         46 ND
         5         96 ND
         6         86 ND
         7         56 ND
         8         66 ND
```

```
SQL> DECLARE
  2     CURSOR detained_cur IS
  3        SELECT roll, att FROM stud21 WHERE att < 75;
  4     v_roll stud21.roll%TYPE;
  5     v_att stud21.att%TYPE;
  6  BEGIN
  7     FOR detained_rec IN detained_cur LOOP
  8        v_roll := detained_rec.roll;
  9        v_att := detained_rec.att;
 10         UPDATE stud21 SET status = 'D' WHERE roll = v_roll;
 11         INSERT INTO d_stud VALUES (v_roll, v_att);
 12      END LOOP;
 13  END;
 14  /
PL/SQL procedure successfully completed.
```

```
SQL> SELECT * FROM STUD21;

      ROLL        ATT STATUS
---------- ---------- ------------------------------
         1         76 ND
         2         99 ND
         3         88 ND
         4         46 D
         5         96 ND
         6         86 ND
         7         56 D
         8         66 D
```

Parameterized Cursor

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.

```
-- Create N_RollCall table
CREATE TABLE N_RollCall (
   rollcall_id NUMBER PRIMARY KEY,
   rollcall_name VARCHAR2(100)
);
-- Create O_RollCall table
CREATE TABLE O_RollCall (
   rollcall_id NUMBER PRIMARY KEY,
   rollcall_name VARCHAR2(100)
);
-- Insert values into N_RollCall table
INSERT INTO N_RollCall (rollcall_id, rollcall_name) VALUES (1, 'Soham');
INSERT INTO N_RollCall (rollcall_id, rollcall_name) VALUES (2, 'Vedant');
INSERT INTO N_RollCall (rollcall_id, rollcall_name) VALUES (3, 'Mrunaline');
INSERT INTO O_RollCall (rollcall_id, rollcall_name) VALUES (4, 'Aarya');
INSERT INTO O_RollCall (rollcall_id, rollcall_name) VALUES (5, 'Hari');
INSERT INTO O_RollCall (rollcall_id, rollcall_name) VALUES (6, 'Atahrva');
SQL> DECLARE
  2     v_n_rollcall_id N_RollCall.rollcall_id%TYPE;
  3     v_n_rollcall_name N_RollCall.rollcall_name%TYPE;
  4     v_o_rollcall_id O_RollCall.rollcall_id%TYPE;
  5     CURSOR n_cursor IS SELECT rollcall_id, rollcall_name FROM N_RollCall;
```

```
 6  BEGIN
 7    FOR n_rec IN n_cursor LOOP
 8      BEGIN
 9        SELECT rollcall_id INTO v_o_rollcall_id FROM O_RollCall WHERE rollcall_id = n_rec.rollcall_id;
10      EXCEPTION
11        WHEN NO_DATA_FOUND THEN
12          v_o_rollcall_id := NULL;
13      END;
14        INSERT INTO O_RollCall(rollcall_id, rollcall_name)
15        VALUES (n_rec.rollcall_id, n_rec.rollcall_name);
16      ELSE
17        DBMS_OUTPUT.PUT_LINE('Data with rollcall_id ' || n_rec.rollcall_id || ' already exists');
18      END IF;
19    END LOOP;
20  END;
21  /
```

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)

```
CREATE TABLE EMP(
        e_no INT,
        d_no INT ,
        Salary DECIMAL(11,2));
INSERT INTO EMP (e_no, d_no, Salary) VALUES (1, 101, 50000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (2, 101, 60000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (3, 102, 55000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (4, 102, 62000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (5, 103, 58000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (6, 103, 59000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (7, 104, 54000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (8, 104, 63000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (9, 105, 57000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (10, 105, 61000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (11, 106, 56000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (12, 106, 62000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (13, 107, 59000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (14, 107, 64000.00);
INSERT INTO EMP (e_no, d_no, Salary) VALUES (15, 108, 60000.00);
CREATE TABLE dept_salary (
  d_no NUMBER,
  Avg_salary NUMBER
);
SQL> DECLARE
 2    v_dno INT;
 3    v_avg_salary INT;
 4
 5    CURSOR c_dept_avg_salary IS
 6      SELECT d_no, AVG(salary) AS avg_salary
 7      FROM EMP
 8      GROUP BY d_no;
 9
10  BEGIN
11    FOR dept_rec IN c_dept_avg_salary LOOP
12      v_dno := dept_rec.d_no;
```

```
 13        v_avg_salary := dept_rec.avg_salary;
 14
 15        INSERT INTO dept_salary (d_no, Avg_salary)
 16        VALUES (v_dno, v_avg_salary);
 17     END LOOP;
 18     DBMS_OUTPUT.PUT_LINE('Department-wise average salary inserted into dept_salary table successfully.');
 19  EXCEPTION
 20     WHEN OTHERS THEN
 21        DBMS_OUTPUT.PUT_LINE('An error occurred');
 22  END;
 23  /
```

Department-wise average salary inserted into dept_salary table successfully.
PL/SQL procedure successfully completed.

```
SQL> SELECT DISTINCT(D_NO),avg_salary FROM dept_salary;

      D_NO AVG_SALARY
---------- ----------
       108      60000
       105      59000
       106      59000
       107      61500
       102      58500
       103      58500
       101      55000
       104      58500

8 rows selected.
```

EXPLICIT CURSOR: Cursor for loop

6. Write PL/SQL block using explicit cursor: Cursor FOR Loop 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) PRIMARY KEY, att number(4), status varchar(2));
create table d_stud(roll number(4), att number(4));
INSERT INTO STUD21 (roll,att,status) VALUES (1,76,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (2,99,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (3,88,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (4,46,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (5,96,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (6,86,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (7,56,'ND');
INSERT INTO STUD21 (roll,att,status) VALUES (8,66,'ND');
```

```
SQL> SELECT * FROM STUD21;

      ROLL        ATT STATUS
---------- ---------- ------------------------------
         1         76 ND
         2         99 ND
         3         88 ND
         4         46 ND
         5         96 ND
         6         86 ND
         7         56 ND
         8         66 ND
```

SQL> DECLARE

```
 2    s_roll stud21.roll%TYPE;
 3    CURSOR detained_stud IS SELECT roll FROM stud21 WHERE att < 75;
 4  BEGIN
 5    FOR detained_rec IN detained_stud LOOP
 6      s_roll := detained_rec.roll;
 7      UPDATE stud21 SET status = 'D' WHERE roll = s_roll;
 8      INSERT INTO d_stud (roll, att) VALUES (s_roll, (SELECT att FROM stud21 WHERE roll = s_roll));
 9    END LOOP;
10  END;
11  /
```

PL/SQL procedure successfully completed.

```
SQL> SELECT * FROM STUD21;

      ROLL        ATT STATUS
---------- ---------- ------------------------------
         1         76 ND
         2         99 ND
         3         88 ND
         4         46 D
         5         96 ND
         6         86 ND
         7         56 D
         8         66 D

8 rows selected.
```