Name: Vinayak Madan Shete

Roll No.: SYCOC303 Course Name: Project Based Learning - II

Div: C Batch: C4 Course Code: BCE4409

==========

Problem Statement:

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)

Output:

```
SQL> select *from BankAccounts;

    ACC_NO NAME       STATUS
---------- ---------- ----------
       101 ABC        INACTIVE
       102 DEF        ACTIVE
       103 GHI        INACTIVE
       104 JKL        ACTIVE
       105 MNO        INACTIVE
       106 PQR        INACTIVE
       107 STU        INACTIVE
       108 VWX        ACTIVE
       109 YZA        INACTIVE
       110 BCD        ACTIVE

10 rows selected.
```

```
SQL> set serveroutput on;
SQL> DECLARE
  2  rows_affec number;
  3  BEGIN
  4  update BankAccounts set Status='ACTIVE' where Status='INACTIVE';
  5  --Implicit cursor has been created
  6  if SQL%FOUND then
  7  dbms_output.put_line('All Inactive Accounts have been Activated!');
  8  rows_affec:=SQL%ROWCOUNT;
  9  dbms_output.put_line('Total no. of rows affected are: '||rows_affec);
 10  end if;
 11
 12  if SQl%NOTFOUND then
 13  dbms_output.put_line('No Inactivate Accounts found!');
 14  end if;
 15  END;
 16  /
All Inactive Accounts have been Activated!
Total no. of rows affected are: 6

PL/SQL procedure successfully completed.

SQL> select *from BankAccounts;

    ACC_NO NAME       STATUS
---------- ---------- ----------
       101 ABC        ACTIVE
       102 DEF        ACTIVE
       103 GHI        ACTIVE
       104 JKL        ACTIVE
       105 MNO        ACTIVE
       106 PQR        ACTIVE
       107 STU        ACTIVE
       108 VWX        ACTIVE
       109 YZA        ACTIVE
       110 BCD        ACTIVE

10 rows selected.

SQL>
```

==========

EXPLICIT CURSOR:

1. Organization has decided to increase the salary of employees by 10% of existing salary, who arehaving salary less than average salary of organization, Whenever such salary updates takes place, arecord for the same is maintained in the increment_salary table.

EMP (E_no , Salary)

increment_salary(E_no , Salary)

Output:

```
SQL> select *from EMP;

      E_NO     SALARY
---------- ----------
       101      60000
       102      20000
       103      60300
       104      31000
       105      82000
       106      10000
       107      42000
       108      55000
       109      36000
       110      12000

10 rows selected.

SQL> select *from increment_salary;

no rows selected

SQL>
```

```
SQL> DECLARE
  2  Cursor cur_inc_sal is select *from EMP where Salary < All(select avg(Salary) from EMP);
  3  meno EMP.E_no%type;
  4  msal EMP.Salary%type;
  5  musal EMP.Salary%type;
  6  BEGIN
  7  open cur_inc_sal;
  8  if cur_inc_sal%isopen then
  9  loop
 10  fetch cur_inc_sal into meno,msal;
 11  exit when cur_inc_sal%notfound;
 12  if cur_inc_sal%found then
 13  musal:=msal+(msal*0.1);
 14  update EMP set Salary=musal where E_no=meno;
 15  insert into increment_salary values(meno,musal);
 16  end if;
 17  end loop;
 18  end if;
 19  close cur_inc_sal;
 20  END;
 21  /

PL/SQL procedure successfully completed.

SQL>
```

```
SQL> select *from EMP;

      E_NO     SALARY
---------- ----------
       101      60000
       102      22000
       103      60300
       104      34100
       105      82000
       106      11000
       107      42000
       108      55000
       109      39600
       110      13200

10 rows selected.

SQL> select *from increment_salary;

      E_NO     SALARY
---------- ----------
       102      22000
       104      34100
       106      11000
       109      39600
       110      13200

SQL>
```

2. 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));

Output:

```
SQL> select *from stud21;

      ROLL        ATT S
---------- ---------- -
       101         75
       102         90
       103         74
       104         85
       105         63
       106         89
       107         71
       108         62

8 rows selected.

SQL> select *from d_stud;

no rows selected

SQL>
```

```
SQL> DECLARE
  2  Cursor crsr_att is select roll, att,status from stud21 where att<75;
  3  mroll stud21.roll%type;
  4  matt stud21.att%type;
  5  mstatus stud21.status%type;
  6  BEGIN
  7  open crsr_att;
  8  if crsr_att%isopen then
  9  loop
 10  fetch crsr_att into mroll,matt,mstatus;
 11  exit when crsr_att%notfound;
 12  if crsr_att%found then
 13  update stud21 set status='D' where roll=mroll;
 14  insert into d_stud values(mroll,matt);
 15  end if;
 16  end loop;
 17  end if;
 18  close crsr_att;
 19  END;
 20  /

PL/SQL procedure successfully completed.
```

```
PL/SQL procedure successfully completed.

SQL> select *from stud21;

      ROLL        ATT S
---------- ---------- -
       101         75
       102         90
       103         74 D
       104         85
       105         63 D
       106         89
       107         71 D
       108         62 D

8 rows selected.

SQL> select *from d_stud;

      ROLL        ATT
---------- ----------
       103         74
       105         63
       107         71
       108         62

SQL>
```

Parameterized Cursor

1. 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:

```
SQL> select *from O_RollCall;

      ROLL NAM
---------- ---
       101 ABC
       102 DEF
       103 GHI
       104 JKL
       105 MNO
       106 PQR
       107 STU
       108 VWX

8 rows selected.

SQL> select *from N_RollCall;

      ROLL NAM
---------- ---
       102 DEF
       105 MNO
       106 PQR

SQL>
```

```
SQL> set serveroutput on;
SQL> DECLARE
  2  cursor crsr_RollCall is select * from O_RollCall;
  3  cursor crsr_chk(str_name varchar2) is select roll from N_RollCall where name = str_name;
  4  str_roll N_RollCall.roll%type;
  5  str_name N_RollCall.name%type;
  6  v number(10);
  7  BEGIN
  8  open crsr_RollCall;
  9  loop
 10  fetch crsr_RollCall into str_roll,str_name;
 11  exit When crsr_RollCall%NOTFOUND;
 12  open crsr_chk(str_name);
 13  fetch crsr_chk into V;
 14  if crsr_chk%FOUND then
 15  dbms_output.put_line('================================================================================');
 16  dbms_output.put_line('Student' || ' ' || str_name || ' ' || 'exist! No need to insert it again..');
 17  dbms_output.put_line('================================================================================');
 18  else
 19  dbms_output.put_line('================================================================================');
 20  dbms_output.put_line('Student'|| str_name || ' does not exist. Inserting in N_RollCall table');
 21  dbms_output.put_line('================================================================================');
 22  insert into N_RollCall values(str_roll,str_name);
 23  end if;
 24  close crsr_chk;
 25  end loop;
 26  close crsr_RollCall;
 27  END;
 28  /
```

```
 28  /
================================================================================
StudentABC does not exist. Inserting in N_RollCall table
================================================================================
================================================================================
Student DEF exist! No need to insert it again..
================================================================================
================================================================================
StudentGHI does not exist. Inserting in N_RollCall table
================================================================================
================================================================================
StudentJKL does not exist. Inserting in N_RollCall table
================================================================================
================================================================================
Student MNO exist! No need to insert it again..
================================================================================
================================================================================
Student PQR exist! No need to insert it again..
================================================================================
================================================================================
StudentSTU does not exist. Inserting in N_RollCall table
================================================================================
================================================================================
StudentVWX does not exist. Inserting in N_RollCall table
================================================================================

PL/SQL procedure successfully completed.

SQL>
```

2. 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)

Output:

```
SQL> select *from EMP1;

      E_NO       D_NO     SALARY
---------- ---------- ----------
       101          1      30000
       102          2      60000
       103          3      36000
       104          3       3000
       105          1      40000
       106          2      55000
       107          3      65000
       108          1      90300
       109          2      62000
       110          1      78300

10 rows selected.

SQL> select *from dept_salary;

no rows selected
```

```
SQL> DECLARE
  2  Cursor sal_cur(dept_no number) IS select d_no, AVG(Salary) AS avg_salary from EMP1 where d_no=dept_no GROUP BY d_no;
  3  mdno number;
  4  mavgsal number;
  5
  6  BEGIN
  7  mdno:=&mdno;
  8  open sal_cur(mdno);
  9  loop
 10  fetch sal_cur into mdno,mavgsal;
 11  exit when sal_cur%NOTFOUND;
 12  insert into dept_salary values(mdno,mavgsal);
 13  end loop;
 14  END;
 15  /
Enter value for mdno: 1
old   7: mdno:=&mdno;
new   7: mdno:=1;

PL/SQL procedure successfully completed.

SQL> select *from dept_salary;

      D_NO AVG_SALARY
---------- ----------
         1      59650

SQL>
```

```
SQL> DECLARE
  2  Cursor sal_cur(dept_no number) IS select d_no, AVG(Salary) AS avg_salary from EMP1 where d_no=dept_no GROUP BY d_no;
  3  mdno number;
  4  mavgsal number;
  5
  6  BEGIN
  7  mdno:=&mdno;
  8  open sal_cur(mdno);
  9  loop
 10  fetch sal_cur into mdno,mavgsal;
 11  exit when sal_cur%NOTFOUND;
 12  insert into dept_salary values(mdno,mavgsal);
 13  end loop;
 14  END;
 15  /
Enter value for mdno: 2
old   7: mdno:=&mdno;
new   7: mdno:=2;

PL/SQL procedure successfully completed.

SQL> /
Enter value for mdno: 3
old   7: mdno:=&mdno;
new   7: mdno:=3;

PL/SQL procedure successfully completed.
```

```
SQL> select *from dept_salary;

      D_NO AVG_SALARY
---------- ----------
         1      59650
         2      59000
         3 34666.6667

SQL>
```

==========

EXPLICIT CURSOR: Cursor for loop

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), att number(4), status varchar(1));

create table d_stud(roll number(4), att number(4));

Output:

```
SQL> select *from stud21;

      ROLL        ATT S
---------- ---------- -
       101         75
       102         90
       103         74
       104         85
       105         63
       106         89
       107         71
       108         62

8 rows selected.

SQL> select *from d_stud;

no rows selected

SQL>
```

```
SQL> DECLARE
  2  Cursor crsr_att is select roll, att,status from stud21 where att<75;
  3  BEGIN
  4  for demo IN crsr_att
  5  loop
  6  update stud21 set status='D' where roll=demo.roll;
  7  insert into d_stud values(demo.roll,demo.att);
  8  end loop;
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL> select *from stud21;

      ROLL        ATT S
---------- ---------- -
       101         75
       102         90
       103         74 D
       104         85
       105         63 D
       106         89
       107         71 D
       108         62 D

8 rows selected.

SQL> select *from d_stud;

      ROLL        ATT
---------- ----------
       103         74
       105         63
       107         71
       108         62

SQL>
```

==========