Assignment 6

Q1) 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> create table acc_dets 176(id int , name varchar(20) , status varchar( 20 )) ;
SQL> insert into acc_dets176 values(1, 'Rohit', 'inactive');
1 row created.
SQL> insert into acc_dets176 values(2, 'Shantanu', 'active ');
1 row created.
SQL> insert into acc_dets176 values(3, 'Suyash', 'inactive') ;
1 row created.
SQL> select* from acc_dets176;
ID NAME                STATUS
1 Rohit       inactive
2 Shantanu          active
3 Suyash    inactive

SQL> declare·
2   total_rows int;
3    begin
4   update acc_dets176
5    set status = 'active'
6   where status= ' inactive';
7   if sql%notfound then
8   dbms_output.put_line (1 no accounts updated 1 );
9   elsif sql%found then
10   total_rows := sql%rowcount;
11   dbms_output . put_line{ ' accounts affected:' || total_rows ) ;
12   end if;
13   end;
14   /

PL/SQL procedure, successfully completed .
SQL> select* from acc_dets176;
ID NAME                 STATUS
---------- ---------- ----------
1 Rohit                 active
```

2 Shantanu active
3 Sahil active

Q2) 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.

```
SQL> create table emp176(e_no int, salary int) ;
Table created.
SQL> insert into emp176 values(1 , 12000);
1 row created .
SQL> insert into emp176 values(2, 24000) ;
1 row created.
SQL> insert into emp176 values(3, 360000);
1 row created.
SQL> create table inc_salary176(e_no int , salary int);
Table created.

SQL > set serveroutput on;
SQL> create procedure salary is
cursor cur1 is
select* from emp176;
temp emp176%rowtype;
avg_salary int;
temp2 int;
begin
select avg1(salary) into avg_salary from emp176;
open cur1;
loop
fetch cur1 into temp;
exit when cur1%notfound;
if(temp.salary < avg_salary) then
update emp176
set salary = temp.salary +(0.1*temp.salary)
where e_no = temp.e_no;
insert into inc_salary176 values(temp.e_no, temp.salary+(0.1*temp.salary));
end if;
end loop;
close cur1;
end;
/
```

Procedure created.

```
SQL> select* from emp176;
     E_NO       SALARY
-------------   ---------------
        1           12000
        2          24000
        3          360000

SQL> exec salary;
PL/SQL procedure successfully completed
SQL> select * from emp176;
     E_NO       SALARY
-------------   ---------------
        1           13200
        2          26400
        3           396000
SQL> select * from inc_salary176;
     E_NO       SALARY
-------------   ---------------
         1         13200
          3         396000
```

Q3) 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));

```
SQL> create table students176(roll int, att int, status varchar);
Table created.
SQL> create table d_stud176(roll int, att int);
Table created.
SQL> insert into students176 values(1, 68, '');
1 row created.
SQL> insert into students176 values(2, 89, '');
1 row created.
SQL> insert into students176 values(3, 35, '');
1 row created.
SQL> insert into students176 values(4, 71, '');
1 row created.
SQL> insert into students176 values(5, 99, '');
1 row created.
```

```
SQL> select * from students176;
    E_NO              ATT        S
--------------   -------------  -------
       1             68
       2             89
```

```
SQL> create procedure detention is
cursor cur1 is
select * from students176;
temp students176%rowtype;
begin
open cur1;
loop
fetch cur1 into temp;
exit when cur1%notfound;
if(temp.att<75) then
update students176
set status = 'D'
where roll = temp.roll;
insert into d_stud176 values(temp.roll, temp.att);
elsif(temp.att>75) then
update students176
set status = 'N'
where roll = temp.roll;
end if;
end loop;
close cur1;
end;
/
Procedure created.
SQL> exec detention;
PL/SQL procedure successfully completed.
SQL> select * from students176;
        ROLL  ATT    S
-----------------   ---------  -------
        1       68      D
        2       89      N
        3       35      D
        4       71      D
        5       99      N

SQL> select * from d_stud176;
        ROLL          ATT
```

----------1--------- -----68----------

3	35
4	71

```
SQL> select * from students176;
```

ROLL	ATT	S
1	78	
2	89	
3	35	
4	71	
5	99	

```
SQL> select * from d_stud176;
no rows selected

SQL> create procedure detention is
cursor cur1 is
select * from students176;
begin
for I in cur1 loop
if(i.att<75) then
update students176
set status = 'D'
where roll = i.roll;
insert into d_stud176 values(i.roll, i.att);
elsif(i.att>75) then
update students176
set status = 'N'
where roll = i.roll;
end if;
end loop;
end;
/
Procedure created.
SQL> exec detention;
PL/SQL procedure successfully completed.
SQL> select * from students176;
```

ROLL	ATT	S
1	68	D
2	89	N
3	35	D
4	71	D
5	99	N

Q4 . parameterized Cursor
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 table O_RollCall(
roll int,
name varchar(10)
);
create table N_RollCall(
roll int,
name varchar(10)
);

SQL> insert into O_RollCall values(1,'kunal');
1 row created.
SQL> insert into O_RollCall values(2,'vineet');
1 row created.
SQL> insert into O_RollCall values(3,'abhishek');
1 row created.
SQL> insert into O_RollCall values(4,'pratik');
1 row created.
SQL> insert into O_RollCall values(5,'moin');
1 row created.
SQL> insert into O_RollCall values(6,'aniket');
1 row created.
SQL> insert into O_RollCall values(7,'tanuja');
1 row created.
SQL> insert into N_RollCall values(2,'vineet');
1 row created.
SQL> insert into N_RollCall values(5,'pratik');
1 row created.

SQL> select * from O_RollCall;
ROLL NAME
---------- ----------
1 kunal
2 vineet
3 abhishek
4 pratik
5 moin
6 aniket
```

```
7 tanuja
7 rows selected.
SQL> select * from N_RollCall;
ROLL NAME
---------- ----------
2 vineet
5 pratik

declare
m_roll int;
m_name varchar(30);
temp int;
cursor old_cursor is select roll,name from O_RollCall;
cursor new_cursor(pi_roll int) is select roll from N_RollCall where roll = pi_roll;
begin
open old_cursor;
loop
fetch old_cursor into m_roll,m_name;

exit when old_cursor%notfound;
open new_cursor(m_roll);
fetch new_cursor into temp;
if new_cursor%notfound then
insert into N_RollCall values(m_roll,m_name);
end if;
close new_cursor;
end loop;
close old_cursor;
end;
/
PL/SQL procedure successfully completed.
```

Q5.parameterized Cursor
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)

```
SQL>create table empnew
(
e_no int,
d_no int,
salary float
);
Table created.
SQL> create table dept_salary
```

```
(
d_no int,
salary float
);
Table created.

SQL> insert into empnew values(101,401,750000);
1 row created.
SQL> insert into empnew values(102,401,120000);
1 row created.
SQL> insert into empnew values(103,402,500000);
1 row created.
SQL> insert into empnew values(104,403,400000);
1 row created.
SQL> insert into empnew values(105,402,800000);
1 row created.
SQL> insert into empnew values(106,404,50000);
1 row created.
SQL> insert into empnew values(107,405,100000);
1 row created.


SQL> select * from empnew;
E_NO D_NO SALARY
---------- ---------- ----------
101 401 750000
102 401 120000
103 402 500000
104 403 400000
105 402 800000
106 404 50000
107 405 100000
7 rows selected.

declare
lf_salary float;
cursor avg_sal(pi_d_no int) is select avg(salary) avg_sal from empnew where
d_no=pi_d_no;
begin
for li_d_no in (select distinct d_no from empnew)
loop
open avg_sal(li_d_no.d_no);
fetch avg_sal into lf_salary;
close avg_sal;
insert into dept_salary values(li_d_no.d_no,lf_salary);
end loop;
end;
```

/
PL/SQL procedure successfully completed.

SQL> select * from dept_salary;
D_NO SALARY
---------- ----------
401 435000
402 650000
403 400000
404 50000
405 100000

Q6) 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));

create table stud21
(roll number(4),
att number(4),
status varchar(1)
); Table created.
create table d_stud21
(roll number(4),
att number(4)
); Table created.

SQL> insert into stud21 values(1,89,NULL);
1 row created.
SQL> insert into stud21 values(2,69,NULL);
1 row created.
SQL> insert into stud21 values(3,82,NULL);
1 row created.
SQL> insert into stud21 values(4,74,NULL);
1 row created.
SQL> insert into stud21 values(5,99,NULL);
1 row created.
SQL> insert into stud21 values(6,45,NULL);

SQL> select * from stud21;
ROLL ATT S

---------- ---------- -
1 89
2 69
3 82
4 74
5 99
6 45
6 rows selected.

```
declare
ln_roll number(4);
ln_att number(4);
cursor att_cursor is select roll,att from stud21 where att<75;
begin
for stud_record in att_cursor
loop
update stud21 set status='D' where roll=stud_record.roll;
insert into d_stud21 values(stud_record.roll,stud_record.att);
end loop;
end;
```

PL/SQL procedure successfully completed.

```
SQL> select * from stud21;
```

ROLL ATT S
---------- ---------- -
1 89
2 69 D
3 82
4 74 D
5 99
6 45 D

6 rows selected.

```
SQL> select * from d_stud21;
```

ROLL ATT
---------- ----------
2 69
4 74
6 45