Assignment 5

Q1) Write a PL/SQL stored Procedure for following requirements and call the procedure in appropriate PL/SQL block.

```
SQL> create or replace procedure cal_fine_176(mroll in number) is
fine number;
sdate date;
issuedate date;
noofdays number;
Begin
select dateofissue into issuedate from Book_Borrower_176 where roll_no=mroll;
select sysdate into sdate from dual;
noofdays:=to_date(sdate)-to_date(issuedate);
 if noofdays>15 and noofdays<30 then
fine:=(noofdays-15)*5;
 update Book_Borrower_176 set status='I' where roll_no=mroll;
 insert into fine_176 values(mroll, sdate, fine);
 elsif noofdays>30 then
 fine:=75+(noofdays-30)*50;
 update Book_Borrower_176 set status='I' where roll_no=mroll;
 insert into fine_176 values(mroll, sdate, fine);
 else
 update Book_Borrower_176 set status='I' where roll_no=mroll;
 end if;
 End;
 /

Procedure created.

SQL> Declare
  mmroll number;
  Begin
  mmroll:=&mmroll;
  cal_fine_176(mmroll);
  End;
  /
Enter value for mmroll: 2
old   4: mmroll:=&mmroll;
new   4: mmroll:=2;

SQL> Declare
  mmroll number;
  Begin
  mmroll:=&mmroll;
  cal_fine_176(mmroll);
  End;
  /
Enter value for mmroll: 3
old   4: mmroll:=&mmroll;
new   4: mmroll:=3;

PL/SQL procedure successfully completed.

SQL> Declare
  mmroll number;
  Begin
  mmroll:=&mmroll;
  cal_fine_176(mmroll);
  End;
  /
Enter value for mmroll: 4
old   4: mmroll:=&mmroll;
new   4: mmroll:=4;

PL/SQL procedure successfully completed.

SQL> select * from fine_176;

      ROLL DATEOFRET     AMOUNT
---------- --------- ----------
         2 27-APR-23        725
         3 27-APR-23       1875
         4 27-APR-23         10
```

Q2) Write a stored function in PL/SQL for given requirement and use the same in PL/SQL block.

Account no. and branch name will be accepted from user. The same will be searched in table acct_details. If status of account is active then display appropriate message and also store the account details in active_acc_details table, otherwise display message on screen “account is inactive”.

```
SQL> create table acct_details(acc_no int, B_name varchar(20), status varchar(20));
Table created.
SQL> insert into acct_details values(11111, 'Akurdi', 'A');
```

```
1 row created.
SQL> insert into acct_details values(12345, 'Nigdi', 'IA');
1 row created.
SQL> insert into acct_details values(22222, 'Akurdi', 'A');
1 row created.
SQL> insert into acct_details values(23456, 'Ravet', 'IA');
1 row created.
SQL> create table active_acc_details(account_no int);
Table created.
SQL> select * from acct_details;
    ACC_NO B_NAME               STATUS
---------- -------------------- ----------
     11111 Akurdi               A
     12345 Nigdi                IA
     22222 Akurdi               A
     23456 Ravet                IA
SQL> create or replace function active_acc_176(macc_no in number, mb_name in char)
return number is
  mstatus char(10);
  Begin
  select status into mstatus from acct_details where acc_no=macc_no and
B_name=mb_name;
  if mstatus = 'A' then
  return 1;
  else
  return 0;
  end if;
 End;
 /
Function created.
SQL> Declare
  mmacc_no number(10);
  mmb_name char(20);
  mstatus int;
  Begin
  mmacc_no:=&mmacc_no;
  mmb_name:=&mmb_name;
  mstatus:=active_acc_176(mmacc_no, mmb_name);
  if mstatus=1 then
 insert into active_acc_details values(mmacc_no);
 end if;
 End;
 /
Enter value for mmacc_no: 11111
old   6: mmacc_no:=&mmacc_no;
```

```
new   6: mmacc_no:=11111;
Enter value for mmb_name: 'Akurdi'
old   7: mmb_name:=&mmb_name;
new   7: mmb_name:='Akurdi';

PL/SQL procedure successfully completed.

SQL> Declare
  mmacc_no number(10);
  mmb_name char(20);
  mstatus int;
  Begin
  mmacc_no:=&mmacc_no;
  mmb_name:=&mmb_name;
  mstatus:=active_acc_176(mmacc_no, mmb_name);
  if mstatus=1 then
 insert into active_acc_details values(mmacc_no);
 end if;
 End;
 /
Enter value for mmacc_no: 12345
old   6: mmacc_no:=&mmacc_no;
new   6: mmacc_no:=12345;
Enter value for mmb_name: 'Nigdi'
old   7: mmb_name:=&mmb_name;
new   7: mmb_name:='Nigdi';

PL/SQL procedure successfully completed.

SQL> Declare
  mmacc_no number(10);
  mmb_name char(20);
  mstatus int;
  Begin
  mmacc_no:=&mmacc_no;
  mmb_name:=&mmb_name;
  mstatus:=active_acc_176(mmacc_no, mmb_name);
  if mstatus=1 then
 insert into active_acc_details values(mmacc_no);
 end if;
 End;
 /
Enter value for mmacc_no: 22222
old   6: mmacc_no:=&mmacc_no;
new   6: mmacc_no:=65432;
```

```
Enter value for mmb_name: 'Ravet'
old   7: mmb_name:=&mmb_name;
new   7: mmb_name:='Ravet';
PL/SQL procedure successfully completed.

SQL> select * from active_acc_details;
ACCOUNT_NO
----------
    11111
    22222
```

Q3) Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scored
by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks
899 and 825 category is Higher Second Class

```
SQL>create table Result_176(Roll int, Name varchar(20), Class varchar(20));
Table created.
SQL>create table Stud_Marks_176(name varchar(20), total_marks int);
Table created.
SQL> create or replace procedure proc_grade(mroll in int, mname in varchar, mtotal_marks
in int )
   Begin
  select total_marks into mtotal_marks from Stud_Marks_176 where roll=mroll;
  if mtotal_marks<=1500 and mtotal_marks>=990 then
  insert into Result_176 values(mroll, mname, 'Distinction');
  elsif mtotal_marks<=989 and mtotal_marks>=900 then
  insert into Result_176 values(mroll, mname, 'FirstClass');
 elsif mtotal_marks<=899 and mtotal_marks>=825 then
 insert into Result_176 values(mroll, mname, 'HighSecondClass');
 else
 insert into Result_176 values(mroll, mname, 'Fail');
 end if;
 End;
 /
Procedure created.

SQL> Declare
  roll int;
  name varchar(20);
  Begin
  roll:=&roll;
  name:=&name;
  proc_grade(roll, name);
  End;
  /
Enter value for roll: 21
old   5: roll:=&roll;
new   5: roll:=21;
Enter value for name: 'ABC'
old   6: name:=&name;
new   6: name:='ABC';
PL/SQL procedure successfully completed.

SQL> Declare
  roll int;
  name varchar(20);
  Begin
  roll:=&roll;
  name:=&name;
  proc_grade(roll, name);
  End;
  /

Enter value for roll: 22
old   5: roll:=&roll;
new   5: roll:=22;
Enter value for name: 'XYZ'
old   6: name:=&name;
new   6: name:='XYZ';

PL/SQL procedure successfully completed.



SQL> Declare
  roll int;
  name varchar(20);
  Begin
  roll:=&roll;
  name:=&name;
  proc_grade(roll, name);
  End;
  /
Enter value for roll: 33
old   5: roll:=&roll;
new   5: roll:=33;
```

```
Enter value for name: 'PQR'
old   6: name:=&name;
new   6: name:='PQR';

PL/SQL procedure successfully completed.

SQL> select * from Result_176;

      ROLL NAME                 CLASS
---------- -------------------- --------------------
         1ABC              Distinction
         2XYZ        HigherSecondClass
         3 PQR               FirstClass
```