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

1. **Borrower(Rollin, Name, DateofIssue, NameofBook, Status)**
2. **Fine(Roll_no,Date,Amt)**

**Accept roll_no &name of book from user.
Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.
If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per
After submiting the book, status will change from I to R.
If condition of fine is true, then details will be stored into fine table.**

output:

```
create or replace procedure proc18(  mroll in number,
                         mname in varchar2
                                             )
                  is
                      di borrower.dateofissue%TYPE;
                         dor DATE:=SYSDATE;
                          fine number;
                         difference  number;

                   begin
                   select dateofissue into di from borrower where rollin =mroll and
nameofbook=mname;


                  difference:= dor - di;
                   if difference < 15 then
                        dbms_output.put_line('Book is returned');
                       insert into fine(roll_no,dateofreturn, amt) values(mroll,dor,0);
                        update borrower set status ='return' where rollin =mroll and
nameofbook=mname;
                   elsif difference<=30 then
                        fine:=(difference-15)*5;
                       dbms_output.put_line('Book is returned');
                        insert into fine(roll_no,dateofreturn, amt) values(mroll,dor,fine);
                        update borrower set status ='return' where rollin =mroll and
nameofbook=mname;
                   else
                        fine:= 15*5+((difference-30)*50);
                       dbms_output.put_line('Book is returned');
                        insert into fine(roll_no,dateofreturn, amt) values(mroll,dor,fine);
                        update borrower set status ='return' where rollin =mroll and
nameofbook=mname;
                   end if;
              end;
```

```
  /


declare
          mroll number(10);
           mname varchar2(50);
   begin
           mroll:=&mroll;
          mname :='&name';
          proc10(mroll,mname);
     end;
   /

Enter value for mroll: 71
old   5:            mroll:=&mroll;
new   5:             mroll:=71;
Enter value for name: java
old   6:            mname :='&name';
new   6:             mname :='java';
Book is returned

PL/SQL procedure successfully completed.

SQL> select * from fine;

   ROLL_NO DATEOFRET        AMT
---------- --------- ----------
        71 28-MAR-24      13097
```

2.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".

```
Create or replace procedure proc_acc2
   (
      cacct_no in number,
      cacct_status out varchar2,
      cacc_amt out varchar2,
      cacc_name out varchar2
   )
   is
  begin
      select acc_status, acc_name, acc_amt
     into cacct_status, cacc_name, cacc_amt
     from acct_details
     where acc_no = cacct_no;

     if cacct_status = 'active' then
        dbms_output.put_line('Active account details are inserted in active_acc_details table');
        Insert into active_acc_details values(cacct_no, cacc_name, cacc_amt);
    End if;
 exception
     When no_data_found then
        dbms_output.put_line('Account not found.');
        When others then
       dbms_output.put_lineE('Error occurred’);
  end;
  /
```

Procedure created.

```
 declare
       cacct_no number;
      cacct_status varchar2(10);
      cacct_amt number(10);
       cacc_name varchar2(100);
   begin
       cacct_no := &acct_no;
       proc_acc2(cacct_no, cacct_status, cacct_amt, cacc_name);
   end;
  /

Enter value for acct_no: 121
old   7:     cacct_no := &acct_no;
new   7:     cacct_no := 121;
Active account details are inserted in active_acc_details table

PL/SQL procedure successfully completed.

SQL> select * from active_acc_details;

  ACCT_ID ACCT_NAME                                      ACCT_AMT
---------- -------------------------------------------------- ----------
      121 anjali                                            10000
```

3. 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
Write a PL/SQL block for using procedure created with above requirement.
Stud_Marks(name, total_marks)
Result(Roll,Name, Class)

```
create or replace procedure pro_grade(smarks in number ,sroll in number,sname in varchar)
    is
    Begin
              if smarks>=990 and smarks <=1500 then
           insert into result values(sroll,sname,'distinction');
             elsif smarks>=900 and smarks<=989 then
            insert into result values(sroll,sname,'first_class');
              elsif smarks>=825 and smarks<=899 then
            insert into result values(sroll,sname,'higher second class');
             else
             dbms_output.put_line('failed');
            end if;
  end;
/
```

Procedure created.

```
declare
    smarks number(10);
    sroll number(10);
    sname number(50);
begin
    sroll := &roll_no;

    select total_marks, name into smarks, sname
    form stud_marks
    where roll_no = sroll;

    pro_grade(smarks, sroll, sname);
end;
/
```

PL/SQL procedure successfully completed.

```
  ROLL NAME       CLASS
---------- ---------- ----------
        71 anjali     distinctio
                      n
```