**Assignment 5**
**SYCOA302 Ritish Shelke**

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 day.
 After submitting the book, status will change from I to R.
 If condition of fine is true, then details will be stored into fine table.

**Procedure:**

```
create or replace procedure proc_library(mroll in number, mbook in varchar, mfine out
number,issue_date out date)
is
        no_of_days number(5);
        today_date date;
begin
        select date_of_issue into issue_date from borrower where roll_no=mroll and
book_name=mbook;

        select sysdate into today_date from dual;

        no_of_days:=trunc(today_date)-issue_date;
-- step 2 calculate mfine amount
   IF no_of_days > 30 THEN
      mfine := 75 + 50 * (no_of_days - 30);
   ELSIF no_of_days > 15 THEN
      mfine:= 5 * (no_of_days - 15);
   ELSE
      mfine := 0;
   END IF;

dbms_output.put_line('Fine  amount to pay : '||mfine);
dbms_output.put_line('Date of book issue : '||issue_date);

-- step 3 change status
update borrower set status='R' where roll_no=mroll;

-- step 4  if fine applicable insert into fine table
IF mfine > 0 THEN
        insert into fine (roll_no,f_date,amt) values (mroll,today_date,mfine);
END IF;
end;
```

**pl/sql block:**

```
declare
        mroll borrower.roll_no%type;
        mbook borrower.book_name%type;
        mfine fine.amt%type;
        issue_date borrower.date_of_issue%type;

begin
        mroll := &mroll;
        mbook := '&mbook';

        select roll_no, book_name into mroll,mbook from borrower where roll_no = mroll and
book_name=mbook;
        proc_library(mroll,mbook,mfine,issue_date);
end;
```

**output:**

```
Enter value for mroll: 302
old   8:        mroll := &mroll;
new   8:        mroll := 302;
Enter value for mbook: SE BY PRESSMAN
old   9:        mbook := '&mbook';
new   9:        mbook := 'SE BY PRESSMAN';
Fine  amount to pay : 525
Date of book issue : 01-APR-23

PL/SQL procedure successfully completed.
```

```
SQL> select * from fine;

   ROLL_NO F_DATE           AMT
---------- --------- ----------
       276 28-APR-23         15
       302 01-APR-23        525
       302 28-APR-23       1475
```

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”

**function:**

```
create or replace function fun_bank(macc number,mbname varchar)
return number
is
mstatus varchar(10);
begin
        select acc_status into mstatus from acc_details where acc_no=macc and b_name=mbname;
```

```
        if mstatus='active' then
                return 1;
        elsif mstatus='inactive' then
                return 0;
        end if;
end;
```

**pl/sql block:**

```
declare
        macc acc_details.acc_no%type;
        mbname acc_details.b_name%type;
        ch number(1);
begin
        macc := &macc;
        mbname := '&mbname';
        select acc_no, b_name into macc,mbname from acc_details where acc_no=macc and
b_name=mbname;
        ch := fun_bank(macc,mbname);

        if ch=1 then
                dbms_output.put_line('Account is active');
                insert into active_acc_details values(macc,mbname);
        elsif ch=0 then
                dbms_output.put_line('Account is not active');
        end if;
end;
```

```
SQL> select * from acc_details;

    ACC_NO B_NAME               ACC_STATUS
---------- -------------------- ----------
       101 deccan               active
       102 kothrud              inactive
       110 akurdi               active
       201 pimpri               active
       202 deccan               inactive

SQL> select * from active_acc_details;

no rows selected
```

```
Enter value for macc: 101
old   7:        macc := &macc;
new   7:        macc := 101;
Enter value for mbname: deccan
old   8:        mbname := '&mbname';
new   8:        mbname := 'deccan';
Account is active

PL/SQL procedure successfully completed.

SQL> select * from active_acc_details;

    ACC_NO B_NAME
---------- --------------------
       101 deccan
```

```
Enter value for macc: 102
old   7:        macc := &macc;
new   7:        macc := 102;
Enter value for mbname: kothrud
old   8:        mbname := '&mbname';
new   8:        mbname := 'kothrud';
Account is not active

PL/SQL procedure successfully completed.
```

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)

**Procedure:**

```
create or replace procedure proc_marks(mroll number, mname varchar,mmarks number)
is begin
        if mmarks<=1500 and mmarks >=990 then
                insert into result values (mroll,mname,'DIST') ;
        elsif mmarks<=989 and mmarks>=900 then
                insert into result values(mroll,mname,'FC');
        elsif mmarks<=899 and mmarks>=825 then
                insert into result values(mroll,mname,'HSC');
        end if;
end;
```

**pl/sql block:**

```
declare
        mroll result.roll%type;
        mname stud_marks.name%type;
        mmarks stud_marks.total_marks%type;
begin
        mroll := &mroll;
        mname := '&mname';
        mmarks := &mmarks;
        select name,total_marks into mname,mmarks from stud_marks where name=mname and
total_marks=mmarks;
        proc_marks(mroll,mname,mmarks);
end;
```

```
SQL> select * from stud_marks;

NAME                 TOTAL_MARKS
-------------------- -----------
ritish                      1325
saurabh                      850
pranav                       989
rahul                        850
viraj                       1000

SQL> select * from result;

no rows selected
```

```
Enter value for mroll: 302
old   7:        mroll := &mroll;
new   7:        mroll := 302;
Enter value for mname: ritish
old   8:        mname := '&mname';
new   8:        mname := 'ritish';
Enter value for mmarks: 1325
old   9:        mmarks := &mmarks;
new   9:        mmarks := 1325;

PL/SQL procedure successfully completed.
```

```
SQL> select * from result;

      ROLL NAME                 CLASS
---------- -------------------- ----------
       302 ritish               DIST
       276 saurabh              HSC
```