**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**

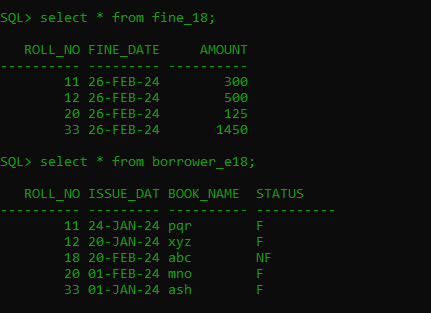
**amount will be Rs 5per day.**

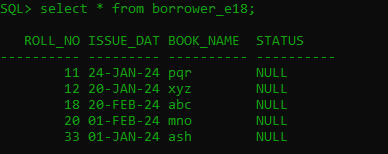
**If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per**

**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 code:-**

CREATE OR REPLACE PROCEDURE student\_18(mroll in number,

mbook in varchar)

is

datediff number(10);

mdate date;

mfine number;

invalid\_input EXCEPTION;

begin

select issue\_date into mdate from borrower\_e18 where roll\_no=mroll and book\_name=mbook;

datediff:=to\_date(sysdate)-to\_date(mdate);

-- Check for overdue

IF datediff < 0 THEN

RAISE invalid\_input; -- Negative date difference indicates invalid input

end if;

IF datediff < 15 then

dbms\_output.put\_line('No Fine');

update borrower\_e18 set status='NF' where roll\_no=mroll and book\_name=mbook;

ELSIF datediff > 15 and datediff < 30 then

mfine:=(datediff\*5);

dbms\_output.put\_line('Fine is: '||mfine);

update borrower\_e18 set status='F' where roll\_no=mroll and book\_name=mbook;

insert into fine\_18 values(mroll,sysdate,mfine);

ELSIF datediff > 30 then

mfine:=((datediff-30)\*50+(30\*5));

dbms\_output.put\_line('Fine is: '||mfine);

update borrower\_e18 set status='F' where roll\_no=mroll and book\_name=mbook;

insert into fine\_18 values(mroll,sysdate,mfine);

End if;

Exception

WHEN invalid\_input THEN

dbms\_output.put\_line('Invalid input: Book return date is after current date.');

End student\_18 ;

/

**Procedure Calling code:-**

DECLARE

mroll borrower\_e18.roll\_no%TYPE;

mbook borrower\_e18.book\_name%TYPE;

BEGIN

mroll := &roll\_no;

mbook := '&book\_name';

student\_18(mroll, mbook);

EXCEPTION

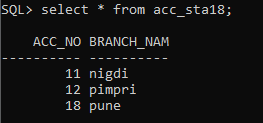
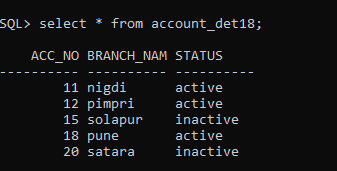
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;

/

**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 CODE:-

create or replace function acc\_18(macc\_no in number,mbranch\_name in varchar)

return number

is

mstatus varchar(15);

begin

select status into mstatus from account\_det18 where acc\_no=macc\_no and branch\_name = mbranch\_name;

if mstatus='active' then

insert into acc\_sta18 values(macc\_no,mbranch\_name);

return 1;

else

return 0;

end if;

end;

/

FUNCTION CALL:-

declare

macc\_no number;

mbranch\_name varchar(15);

ch number;

begin

macc\_no := &acc\_no;

mbranch\_name := '&branch\_name';

ch := acc\_18(macc\_no,mbranch\_name);

if ch=1 then

dbms\_output.put\_line('THE ACCOUNT IS ACTIVE');

else

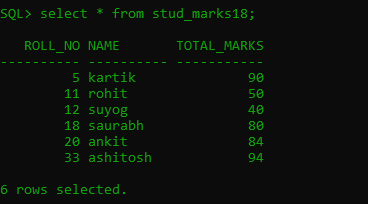
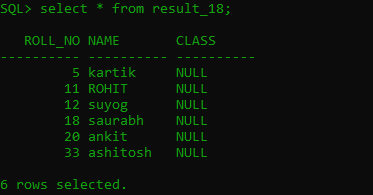
dbms\_output.put\_line('THE ACCOUNT IS INACTIVE');

end if;

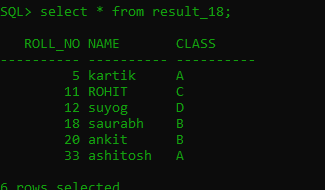
end;

/

**Que3. 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).**



**AFTER CODE RUN:-**

****

**Procedure code:-**

Create or Replace Procedure proc\_grade(mmarks in

number, mroll in number) is

Begin

if mmarks>=90 and mmarks <=100 then

update result\_18 set class= 'A' where roll\_no=mroll;

elsif mmarks>=60 and mmarks<=89 then

update result\_18 set class='B' where roll\_no=mroll;

elsif mmarks>=50 and mmarks<60 then

update result\_18 set class='C' where roll\_no=mroll;

else

update result\_18 set class='D' where roll\_no=mroll;

end if;

End;

**/**

**Procedure Call Code:-**

Declare

mmarks number(4);

mroll number(4);

Begin

mroll:=&roll;

select roll\_no,total\_marks into mroll,mmarks from stud\_marks18

where roll\_no=mroll ;

proc\_grade(mmarks, mroll );

End;