PRACTICAL_NO:4

Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory.

Suggested Problem statement: Consider Tables:

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

- Accept Roll_no and NameofBook 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 and 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.
- Also handles the exception by named exception handler or user define exception handler

PLSQL CODE:-

```
declare
r int;
bn varchar(25);
DateofIssue date;
Return_Date date;
d int;
f float;
begin
r:=&r;
bn:='&bn';
select DateofIssue into DateofIssue from borrower where  ROLL_NO=r and NameofBook =bn
and  STATUS='I';
select sysdate into Return_Date from dual;
d:=Return_Date-DateofIssue;
```

```
dbms_output.put_line('TOTAL DAYS ARE!!'||d);
if d<15 then
dbms_output.put_line('NO FINE');
update borrower set STATUS='R' where ROLL_NO=r and NameofBook=bn;
elsif d>15 and  d<30 then
f:=d*5;
update borrower set STATUS='R' where ROLL_NO=r and NameofBook=bn;
else
f:=d*50;
update borrower set STATUS='R' where ROLL_NO=r and NameofBook=bn;
end if;
if f>0 then
insert into fine values(r,Return_Date,f);
end if;
exception
when NO_DATA_FOUND then
dbms_output.put_line('DATA NOT AVAILABLE IN TABLE');
end;
/
```

OUTPUT:-

```
SQL> create table Borrower(Roll_no int, Name varchar(25), DateofIssue Date, NameofBook varchar(25), Status varchar(1));

Table created.

SQL> create table Fine(Roll_no int,Return_Date date,Amt float);

Table created.


SQL> insert into borrower values(28,'OMKAR','16-JUL-24','DBMS','I');
SQL> select * from borrower;


  ROLL_NO            NAME          DATEOFISS NAMEOFBOOK               S
---------- ------------------------ --------- ------------------------- -
        28           OMKAR                    16-JUL-24 DBMS                    I


SQL> @PDB4.sql
Enter value for r: 28
old   9: r:=&r;
new   9: r:=28;
Enter value for bn: DBMS
old  10: bn:='&bn';
new  10: bn:='DBMS';
TOTAL DAYS ARE!! 5
NO FINE

PL/SQL procedure successfully completed.

SQL> @PDB4.sql
Enter value for r: 28
old   9: r:=&r;
new   9: r:=28;
Enter value for bn: DBMS
old  10: bn:='&bn';
new  10: bn:='DBMS';
TOTAL DAYS ARE!! 25

PL/SQL procedure successfully completed.
```

```
SQL> select * from borrower;

   ROLL_NO NAME                     DATEOFISS NAMEOFBOOK                S
---------- ------------------------ --------- ------------------------- -
        28 OMKAR                    16-JUL-24 DBMS                      R

SQL> select * from fine;

   ROLL_NO RETURN_DA        AMT
---------- --------- ----------
        28 09-AUG-24        125

SQL> @PDB4.sql
Enter value for r: 28
old   9: r:=&r;
new   9: r:=28;
Enter value for bn: DBMS
old  10: bn:='&bn';
new  10: bn:='DBMS';
TOTAL DAYS ARE!!116

PL/SQL procedure successfully completed.

SQL> select * from borrower;
```

```
  ROLL_NO NAME                     DATEOFISS NAMEOFBOOK                S
---------- ------------------------ --------- ------------------------ -
        28 OMKAR                    16-APR-24 DBMS                     R

SQL> select * from fine;

  ROLL_NO RETURN_DA        AMT
---------- --------- ----------
        28 09-AUG-24       5800

SQL> @PDB4.sql
Enter value for r: 28
old   9: r:=&r;
new   9: r:=28;
Enter value for bn: DBMS
old  10: bn:='&bn';
new  10: bn:='DBMS';
DATA NOT AVAILABLE IN TABLE

PL/SQL procedure successfully completed.
```

Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 5 to 9. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns, radius and area.

```
DECLARE
        r NUMBER;
        a NUMBER;
        pi NUMBER := 3.14;
BEGIN
        FOR r IN 5 .. 9 LOOP
                a := pi*r*r;
                dbms_output.put_line(a);
                INSERT INTO AREA VALUES (r,a);
        END LOOP;

END;
/
```

<u>OUTPUT:-</u>

```
SQL*Plus: Release 21.0.0.0.0 - Production on Fri Aug 9 21:08:02 2024
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Fri Aug 09 2024 20:59:43 -07:00

Connected to:
```

```
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> create table AREA (radius float,area float);

Table created.

SQL> set serveroutput on
SQL> @ar.sql
78.5
113.04
153.86
200.96
254.34

PL/SQL procedure successfully completed.

SQL> select * from AREA;

    RADIUS       AREA
---------- ----------
         5       78.5
         6     113.04
         7     153.86
         8     200.96
         9     254.34

SQL>
```