LAB6

Aniket sambher

Reg no-190905466

Section -A

Roll no-58

1.Write a PL/SQL block to display the GPA of given student.

```
DECLARE
rollin StudentTable.RollNo%Type;
temp StudentTable.GPA%Type;
begin
rollin:=&RollNumber;
select GPA into temp from StudentTable where RollNo=rollin;
dbms_output.put_line('GPA is' || temp);
end;
/
```

2.Write a PL/SQL block to display the letter grade(0-4: F; 4-5: E; 5-6: D; 6-7: C; 7-8: B; 8-9: A; 9-10: A+} of given student

```
SET SERVEROUTPUT ON
DECLARE
rollin StudentTable.RollNo%Type;
temp StudentTable.GPA%Type;
BEGIN
rollin:=&RollNumber;
select GPA into temp from StudentTable where RollNo=rollin;
IF (temp>=0 AND temp<=4) THEN
        DBMS_OUTPUT.PUT_LINE('F');
ELSIF (temp>=4 AND temp<=5) THEN
        DBMS_OUTPUT.PUT_LINE('E');
ELSIF (temp>=5 AND temp<=6) THEN
```

```
            DBMS_OUTPUT.PUT_LINE('D');
ELSIF (temp>=6 AND temp<=7) THEN
            DBMS_OUTPUT.PUT_LINE('C');
ELSIF (temp>=7 AND temp<=8) THEN
            DBMS_OUTPUT.PUT_LINE('B');
ELSIF (temp>=8 AND temp<=9) THEN
            DBMS_OUTPUT.PUT_LINE('A');
ELSIF (temp>=9 AND temp<=10) THEN
            DBMS_OUTPUT.PUT_LINE('A+');
ELSE
            DBMS_OUTPUT.PUT_LINE('No such grade');
END IF;
END;
/
```

3.Input the date of issue and date of return for a book. Calculate and display the fine with the appropriate message using a PL/SQL block.

```
DECLARE
issuedate DATE;
returndate DATE;
noDays number(8);
BEGIN
issuedate := SYSDATE-31;
returndate := SYSDATE;
noDays := returndate-issuedate;
IF noDays <= 7 THEN
   DBMS_OUTPUT.PUT_LINE('Nil');
ELSIF noDays > 7 AND noDays <=15 THEN
   DBMS_OUTPUT.PUT_LINE(1*(noDays-7));
ELSIF noDays > 15 AND noDays <=30 THEN
   DBMS_OUTPUT.PUT_LINE((1*8)+(2*(noDays-15)));
ELSIF noDays > 30 THEN
```

```
   DBMS_OUTPUT.PUT_LINE((1*8)+(2*15)+(5*(noDays-30)));
END IF;
END;
/
```

4.Write a PL/SQL block to print the letter grade of all the students(RollNo: 1 - 5).

```
SET SERVEROUTPUT ON
DECLARE
x NUMBER := 1;
temp StudentTable.GPA%Type;
BEGIN
WHILE x <=5 LOOP
select GPA into temp from StudentTable where RollNo=x;
IF (temp>=0 AND temp<=4) THEN
            DBMS_OUTPUT.PUT_LINE('F');
ELSIF (temp>=4 AND temp<=5) THEN
            DBMS_OUTPUT.PUT_LINE('E');
ELSIF (temp>=5 AND temp<=6) THEN
            DBMS_OUTPUT.PUT_LINE('D');
ELSIF (temp>=6 AND temp<=7) THEN
            DBMS_OUTPUT.PUT_LINE('C');
ELSIF (temp>=7 AND temp<=8) THEN
            DBMS_OUTPUT.PUT_LINE('B');
ELSIF (temp>=8 AND temp<=9) THEN
            DBMS_OUTPUT.PUT_LINE('A');
ELSIF (temp>=9 AND temp<=10) THEN
            DBMS_OUTPUT.PUT_LINE('A+');
ELSE
            DBMS_OUTPUT.PUT_LINE('No such grade');
END IF;
X := x + 1;
```

```
END LOOP;

END;

/
```

5.Alter StudentTable by appending an additional column LetterGrade Varchar2(2). Then write a PL/SQL block to update the table with letter grade of each student.

```
alter table studenttable add lettergrade varchar2(2);

SET SERVEROUTPUT ON

DECLARE

x NUMBER := 1;

temp StudentTable.GPA%Type;

grade varchar(2);

BEGIN

WHILE x <=5 LOOP

select GPA into temp from StudentTable where RollNo=x;

IF (temp>=0 AND temp<=4) THEN

            grade:='F';

ELSIF (temp>=4 AND temp<=5) THEN

            grade:='E';

ELSIF (temp>=5 AND temp<=6) THEN

            grade:='D';

ELSIF (temp>=6 AND temp<=7) THEN

            grade:='C';

ELSIF (temp>=7 AND temp<=8) THEN

            grade:='B';

ELSIF (temp>=8 AND temp<=9) THEN

            grade:='A';

ELSIF (temp>=9 AND temp<=10) THEN

            grade:='A+';

 END IF;
```

```
update studenttable set LetterGrade=grade where RollNo=x;
X := x + 1;
END LOOP;
END;
/
```

6.Write a PL/SQL block to find the student with max. GPA without using aggregate function.

```
DECLARE
RNo StudentTable.RollNo%TYPE;
Gpa1 StudentTable.GPA%TYPE;
high NUMBER :=0;
j NUMBER :=0;
BEGIN
FOR i IN 1..5 LOOP
SELECT GPA INTO Gpa1 FROM StudentTable WHERE RollNo=i;
IF Gpa1>high THEN
high:=Gpa1;
j:=i;
END IF;
END LOOP;
DBMS_OUTPUT.PUT_LINE('RollNo of Student with Highest GPA is '||j);
END;
/
```

7.Implement lab exercise 4 using GOTO.

```
SET SERVEROUTPUT ON
DECLARE
x NUMBER := 1;
temp StudentTable.GPA%Type;
BEGIN
<<start_loop>>
```

```
select GPA into temp from StudentTable where RollNo=x;
IF (temp>=0 AND temp<=4) THEN
        DBMS_OUTPUT.PUT_LINE('F');
ELSIF (temp>=4 AND temp<=5) THEN
        DBMS_OUTPUT.PUT_LINE('E');
ELSIF (temp>=5 AND temp<=6) THEN
        DBMS_OUTPUT.PUT_LINE('D');
ELSIF (temp>=6 AND temp<=7) THEN
        DBMS_OUTPUT.PUT_LINE('C');
ELSIF (temp>=7 AND temp<=8) THEN
        DBMS_OUTPUT.PUT_LINE('B');
ELSIF (temp>=8 AND temp<=9) THEN
        DBMS_OUTPUT.PUT_LINE('A');
ELSIF (temp>=9 AND temp<=10) THEN
        DBMS_OUTPUT.PUT_LINE('A+');
ELSE
        DBMS_OUTPUT.PUT_LINE('No such grade');
END IF;
X := x + 1;
IF x<=5 then
GOTO start_loop;
END IF;
END;
/
```

PTO

8.

Based on the University database schema, write a PL/SQL block to display the details of the Instructor whose name is supplied by the user. Use exceptions to show appropriate error message for the following cases:

a. Multiple instructors with the same name

b. No instructor for the given name

```
DECLARE
instName instructor.name%TYPE;
instRec instructor%ROWTYPE;
BEGIN
instName := '&nameInst';
SELECT * INTO instRec FROM instructor where instructor.name = instName;
DBMS_OUTPUT.PUT_LINE('Instructor name:'||instRec.name);
DBMS_OUTPUT.PUT_LINE('Instructor ID:'||instRec.ID);
DBMS_OUTPUT.PUT_LINE('Instructor dept name:'||instRec.dept_name);
DBMS_OUTPUT.PUT_LINE('Instructor salary:'||instRec.salary);
EXCEPTION
   WHEN TOO_MANY_ROWS THEN DBMS_OUTPUT.PUT_LINE('Multiple instructors with same name.');
   WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('No instructors found.');
END;
/
```

9.Extend lab exercise 5 to validate the GPA value used to find letter grade. If it is outside the range, 0 – 10, display an error message, ‘Out of Range’ via an exception handler

```
DECLARE
RNo StudentTable.RollNo%TYPE;
Gpa1 StudentTable.GPA%TYPE;
i NUMBER := 1;
LetterG StudentTable.LetterGrade%TYPE;
outOfRange Exception;
BEGIN
```

```
WHILE i<7
LOOP
   Rno := i;
   SELECT GPA INTO Gpa1 FROM StudentTable WHERE RollNo=RNo;
   IF Gpa1 >= 9 AND Gpa1 <=10 THEN
      LetterG:='A+';
   ELSIF Gpa1 >= 8 AND Gpa1 <9 THEN
      LetterG:='A';
   ELSIF Gpa1 >= 7 AND Gpa1 <8 THEN
      LetterG:='B';
   ELSIF Gpa1 >= 6 AND Gpa1 <7 THEN
      LetterG:='C';
   ELSIF Gpa1 >= 5 AND Gpa1 <6 THEN
      LetterG:='D';
   ELSIF Gpa1 >= 4 AND Gpa1 <5 THEN
      LetterG:='E';
   ELSIF Gpa1 >= 0 AND Gpa1 <4 THEN
      LetterG:='E';
   ELSE
   RAISE outOfRange;
   END IF;
   UPDATE StudentTable set LetterGrade=LetterG where RollNo=RNo;
   i := i+1;
END LOOP;
EXCEPTION
   WHEN outOfRange THEN DBMS_OUTPUT.PUT_LINE('GPA Out of range');
END;
/
```