Name:-Aniket Prajapati

SYCS--351

PRACTICAL NO:2

1. Write a PL/SQL program to display the sum of first 10 integers.

```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on Mon Aug 9 17:24:11 2021
Copyright (c) 1982, 2014, Oracle. All rights reserved.
SQL> connect system
Enter password:
Connected.
SQL> declare
 2 a int;
 3 b int :=0;
 4 begin
 5 for a in 1..10 loop
 6 b :=b+a;
 7 end loop;
 8 dbms_output.put_line(b);
 9 end;
10 /
PL/SQL procedure successfully completed.
```

2. Write a PL/SQL program to display the sum of first 10 odd numbers.

```
SQL> declare
2 a number :=0;
3 b number :=1;
4 c number :=0;
5 begin
6 while ac10 loop
7 dbms_output.put_line(b);
8 c :=b+c;
9 b :=b+2;
10 a :=a+1;
11 end loop;
12 dbms_output.put_line(c);
13 end;
14 /
1
1
3
5
7
9
11
11
13
15
17
19
100
PL/SQL procedure successfully completed.
```

3. Write a PL/SQL program to calculate the area of a circle and insert the area and radius in a table aoc (sno,radius,area) till radius is less than 10.

```
SQL> declare

2 a number;

3 radius number :=1;

4 area number;

5 pi number :=3.14;

6 begin

7 for a in 1..9 loop

8 radius := a;

9 area := radius * radius * pi;

10 insert into aco values(a,radius,area);

11 end loop;

12 end;

13 /

PL/SQL procedure successfully completed.
```

4. Write a PL/SQL program to reverse the number (234 as 432).

```
SQL> declare

2 a number :=234;

3 b number;

4 begin

5 b :=0;

6 while a>0 loop

7 b :=(b * 10) + mod(a,10);

8 a :=floor(a/10);

9 end loop;

10 dbms_output.put_line(b);

11 end;

12 /

432

PL/SQL procedure successfully completed.
```

5. Write a PL/SQL program to print the length of entered string.

```
Run SQL Command Line

SQL> declare
2 a varchar2(20) := '&a';
3 begin
4 dbms_output.put_line(length(a));
5 end;
6 /
Enter value for a: hello
old 2: a varchar2(20) := '&a';
new 2: a varchar2(20) := 'hello';

PL/SQL procedure successfully completed.
```

6. Write a PL/SQL program to count number of employees in dept 10, and if they are greater than 3 print the count otherwise do nothing.

```
Run SQL Command Line

SQL> declare
2 num number;
3 begin
4 select count(*) into num from aniket_emp351 where deptno = 10;
5 if num > 3 then
6 dbms_output.put_line(num || 'Employee in deptno 10');
7 end if;
8 end;
9 /

PL/SQL procedure successfully completed.
```

7. Write a PL/SQL block to find the factorial of a number.

```
Run SQL Command Line
SQL> declare
 2 a number :=&a;
 3 b number := 1;
 4 begin
 5 while a>0 loop
 6 b := b*a;
 7 a := a-1;
 8 end loop;
 9 dbms_output.put_line(b);
 10 end;
11 /
Enter value for a: 4
old 2: a number :=&a;
new 2: a number :=4;
24
PL/SQL procedure successfully completed.
```

8. Write a block to display the number from 1 to 10 using unconstraint loop.

```
SQL> declare
2 a number :=1;
3 begin
4 loop
5 if a>10 then
6 exit;
7 end if;
8 dbms_output.put_line(a);
9 a :=a+1;
10 end loop;
11 end;
12 /
1
2
3
4
5
6
6
7
8
9
10

PL/SQL procedure successfully completed.
```

9. Write a PL/SQL block using CASE statement to accept the owner name from the user.

The user name can be SYS, SYSTEM, HR or SCOTT.

If the owner name is SYS then print the result is 'The Owner is SYS'. If the owner name is SYSTEM then print the result is 'The Owner is SYSTEM'.

If the owner name is HR then print the result is 'The Owner is HR'. If the owner name is SCOTT then print the result is 'The Owner is SCOTT'. Otherwise print 'Invalid Choice'.

```
Run SQL Command Line
SQL> declare
 2 name varchar2(20) := '&name';
 3 begin
 4 case name
 5 when 'sys' then dbms_output.put_line('the owner is '||name);
 6 when 'system' then dbms_output.put_line('the owner is '||name);
 7 when 'hr' then dbms_output.put_line('the owner is '||name);
 8 when 'scott' then dbms_output.put_line('the owner is '||name);
10 dbms_output.put_line('Invalid Choise');
11 end case;
12 end;
13 /
Enter value for name: sys
     2: name varchar2(20) := '&name';
     2: name varchar2(20) := 'sys';
the owner is sys
PL/SQL procedure successfully completed.
```

10. Write a PL/SQL block to find factorial of a number which is accepted by the user and store it under the table fac(num,fact).

```
Run SQL Command Line
PL/SQL procedure successfully completed.
SQL> create table fac(a number,b number);
Table created.
SQL> declare
 2 a number := &a;
 3 b number := 1;
 4 c number;
 5 begin
 6 c := a;
    while c>0 loop
 8 b := b*c;
 9 c := c-1;
10 end loop;
11 insert into fac values(a,b);
12 end;
13 /
Enter value for a: 5
old 2: a number := &a;
     2: a number := 5;
PL/SQL procedure successfully completed.
```

11. Write a PL/SQL to read a number and check whether it is greater than 100 or not and print appropriate message.

```
Run SQL Command Line
SQL> declare
 2 a number :=&a;
 3 begin
 4 if a>100 then
 5 dbms_output.put_line('the given number is greater than 100');
 7 dbms_output.put_line('the given number is smaller than 100');
 8 end if;
 9 end;
10 /
Enter value for a: 52
   2: a number :=&a;
old
     2: a number :=52;
the given number is smaller than 100
PL/SQL procedure successfully completed.
```

12. Write a PL/SQL to read the salary of an employee 7900 and display the appropriate message if it lies in the range of 1000 and 5000.

```
🚣 Oracle SQL*Plus
File Edit Search Options Help
SQL*Plus: Release 10.2.0.3.0 - Production on Mon Aug 9 21:20:40 2021
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.
Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
SQL> set serveroutput on
SQL> declare
 2 sint;
 3 begin
 4 select sal into s from emp where empno=7900;
   if (s >=1000) and (s <=5000) then
 6 dbms_output.put_line('the salary lies in the range od 1000 tp 5000');
 8 dbms_output.put_line('the salary does not lies in the range od 1000 tp 5000');
 9
    end if;
10 end;
11 /
the salary does not lies in the range od 1000 tp 5000
PL/SQL procedure successfully completed.
```

13. Write a PL/SQL to swap two numbers and display the swapped numbers.

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> declare
 2 a number := &a;
 3 b number := &b;
 4 temp number;
 5 begin
 6 temp := a;
 7
    a := b;
 8 b := temp;
 9 dbms_output.put_line(a|| ' ' ||b);
10 end;
11 /
Enter value for a: 6
old 2: a number := &a;
    2: a number := 6;
Enter value for b: 1
old
     3: b number := &b;
new
      3: b number := 1;
PL/SQL procedure successfully completed.
```

14. Write a PL/SQL block to update the salary of the employee with 1000 when total number of employees in a particular department is greater than 3.

```
🚣 Oracle SQL*Plus
File Edit Search Options Help
SQL> declare
 2 deptn number;
 3 total number;
 4 begin
 5 select count(*) into total from emp where deptno = 10;
 6 if total > 3 then
 7 update emp
 8 set sal = sal +1000;
 9 select count(*) into total from emp where deptno = 20;
10 if total > 3 then
11 update emp
12 set sal = sal +1000;
13 select count(*) into total from emp where deptno = 10;
14 if total > 3 then
15 update emp
16 set sal = sal +1000;
17 end if;
18 end if;
19 end if;
20 end;
21 /
PL/SQL procedure successfully completed.
```

15. Write a PL/SQL block to delete the records of the table employee by accepting the table name from the user.

```
File Edit Search Options Help

SQL> declare
2 name varchar2(20) := '&name';
3 begin
4 delete from emp where ename =name;
5 end;
6 /
Enter value for name: james
old 2: name varchar2(20) := '&name';
new 2: name varchar2(20) := 'james';

PL/SQL procedure successfully completed.
```

16. Write a PL/SQL to check whether the character entered is a vowel or not.

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> declare
 2 ch char :='&ch';
 3 begin
 4 if ch in ('A','E','I','O','U')then
 5 dbms output.put line('the character is a vowel');
     dbms_output.put_line('the character is not vowel');
    end if;
 9 end;
10 /
Enter value for ch: h
      2: ch char :='&ch';
old
     2: ch char :='h';
new
the character is not vowel
PL/SQL procedure successfully completed.
```

17. Write a PL/SQL to check whether a number is even or odd.

```
🚵 Oracle SQL*Plus
File Edit Search Options Help
SQL> declare
 2 a number :=&a;
  3 begin
 4 if mod(a,2) = 0 then
    dbms_output.put_line('the given number is even');
     dbms_output.put_line('the given number in odd');
  7
     end if;
    end;
10 /
Enter value for a: 7
old
      2: a number :=&a;
      2: a number :=7;
new
the given number in odd
PL/SQL procedure successfully completed.
```

18. Write a PL/SQL block using case statement to print the salary as high if it is greater then 10000, moderate if it is between 5000 and 10000 and low if it is less than 5000. The salary has been taken as an input of a specific employee whose empid is accepted by the user.

```
🚣 Oracle SQL*Plus
File Edit Search Options Help
SQL> declare
    a number;
 3
    s number := &s;
    begin
 5
    select sal into a from emp where empno=s;
 6
    when a > 10000 then dbms_output.put_line('salary is High');
    when a > 5000 and a < 10000 then dbms_output.put_line('salary is medium');
 9
    when a < 5000 then dbms_output.put_line('Salary is law');
 10 else
 11 dbms_output.put_line('Inavalid EmpNo');
12 end case;
13 end;
14 /
Enter value for s: 7782
     3: s number := &s;
    3: s number := 7782;
new
Salary is law
PL/SQL procedure successfully completed.
```

19. Write a PL/SQL block using case statement to perform addition, subtraction, multiplication and division for the individual choices a ,s ,m, d.The division can only take place if the divisor is greater than 0 else error message should be printed.

```
Run SQL Command Line
                                                                                          SQL> declare
 2 f_no int :=&fno;
 3 l_no int :=&lno;
 4 sel varchar2(5):='&sel';
 5 begin
 6 case sel
 7 when 'a' then dbms_output.put_line('f_no + l_no');
 8 when 's' then dbms_output.put_line('f_no - l_no');
 9 when 'm' then dbms_output.put_line('f_no * l_no');
 10 when 'd' then
11 if l_no >0 then
12 dbms_output.put_line(f_no / l_no);
14 dbms_output.put_line('cannot divide by zero');
15 end if;
16 end case;
17 end;
Enter value for fno: 8
    2: f_no int :=&fno;
new 2: f_no int :=8;
Enter value for lno: 2
    3: l_no int :=&lno;
old
    3: 1 no int :=2;
Enter value for sel: d
old 4: sel varchar2(5):='&sel';
new 4: sel varchar2(5):='d';
PL/SQL procedure successfully completed.
```

20. Write a PL/SQL block to print the numbers from 1 to 10 using While and For Loop.

Run SQL Command Line

```
SQL> declare
2 a number;
3 begin
4 for a in 1..10 loop
5 dbms_output.put_line(a);
6 end loop;
7 end;
8 /
1
2
3
4
5
6
7
8
9
10
PL/SQL procedure successfully completed.
```

21. Write a PL/SQL block to print the Fibonacci series up to 10.

```
Run SQL Command Line

SQL > declare
2  first number :=0;
3  second number :=1;
4  temp number;
5  n number := 5;
6  i number;
7  begin
8  dbms_output.put_line('Fibonacci series: ');
9  dbms_output.put_line(first);
10  dbms_output.put_line(second);
11  for i in 2..n loop
12  temp := first + second;
13  first := second;
14  second := temp;
15  dbms_output.put_line(temp);
16  end loop;
17  end;
18  /
Fibonacci series:
0
PL/SQL procedure successfully completed.
```

22. Write a PL/SQL block to calculate the area of a circle till radius less than 10. 23. Write a PL/SQL block to display the number of employees when the deptno is inputted by the user.

```
Run SQL Command Line
SQL> declare
 2 pi float := 3.14;
 3 radius number := &radius;
 4 result number;
 5 begin
 6 dbms output.put line('Area of circle: ');
 7 if radius<10 then
 8 result := pi*radius*radius;
 9 dbms_output.put_line(result);
 10 end if;
11 end;
12 /
Enter value for radius: 7
old 3: radius number := &radius;
new 3: radius number := 7;
Area of circle:
153.86
PL/SQL procedure successfully completed.
```

23. Write a PL/SQL block to display the number of employees when the deptno is inputted by the user.

```
🚣 Oracle SQL*Plus
File Edit Search Options Help
SQL> declare
  2 e number;
  3 d number :=&d;
  4
    c int :=0;
    begin
    for i in (select empno from emp where deptno = d)loop
    c :=c+1;
    end loop;
  9
    dbms_output.put_line('There are: ' || c || 'employee in deptno: ' ||d);
 10 end;
 11
Enter value for d: 10
      3: d number :=&d;
old
      3: d number :=10;
There are: 3employee in deptno: 10
PL/SQL procedure successfully completed.
```

24. Write a PL/SQL block to print greatest among three numbers.

```
Select Run SQL Command Line
                                                                                           SQL> declare
 2 a number := &a;
 3 b number := &b;
 4 c number := &c;
 5 begin
 6 if(a>b) and (a>c) then
 7 dbms_output.put_line('Number ' || a || ' is greater');
 8 elsif (b>a) and (b>c) then
 9 dbms_output.put_line('Number ' || b || ' is greater');
10 else
11 dbms_output.put_line('Number ' || c || ' is greater');
12 end if;
13 end;
14 /
Enter value for a: 16
    2: a number := &a;
     2: a number := 16;
Enter value for b: 7
    3: b number := &b;
    3: b number := 7;
Enter value for c: 34
old 4: c number := &c;
new 4: c number := 34;
Number 34 is greater
PL/SQL procedure successfully completed.
```

25. Write a PL/SQL block to display the appropriate day of the week according to the choice made by the user.

```
Select Run SQL Command Line
SOL> declare
  2 dnum number :=&dnum;
 3 begin
 4 case dnum
 5 when 1 then dbms_output.put_line('sunday');
  6 when 2 then dbms output.put line('Monday');
  7 when 3 then dbms_output.put_line('Tuesday');
  8 when 4 then dbms_output.put_line('Wednesday');
 9 when 5 then dbms_output.put_line('Thursday');
 10 when 6 then dbms_output.put_line('Friday');
11 when 7 then dbms_output.put_line('Satarday')
 12 else dbms output.put line('Not a number of day in week');
13 end case;
14 end;
15 /
Enter value for dnum: 7
      2: dnum number :=&dnum;
      2: dnum number :=7;
Satarday
PL/SQL procedure successfully completed.
```

26. Create a PL/SQL block that has four sections. Each section should output a statement. Use labels and the Goto command to output the section messages in the following order:

Section 3

Section 2

Section 1

Section 4

```
Select Run SQL Command Line
SQL> Begin
 2 Goto section3;
 3 <<section1>>
 4 dbms_output.put_line('section 1');
 5 Goto section4;
 6 <<section2>>
 7 dbms_output.put_line('section 2');
 8 Goto section1;
 9 <<section3>>
 10 dbms output.put line('section 3');
 11 Goto section2;
12 <<section4>>
13 dbms_output.put_line('section 4');
14 end;
15 /
section 3
section 2
section 1
section 4
PL/SQL procedure successfully completed.
```

27. Write a PL/SQL block to check whether the entered year is a leap year or not.

```
Select Run SQL Command Line
SQL> declare
 2 year number := &year;
 3 begin
 4 if mod(year,4)=0 and mod(year,100)!=0 and mod(year,400)=0 then
 5 dbms_output.put_line(year || 'is a leap year');
 6 else
 7 dbms_output.put_line(year || 'is not a leap year');
 8 end if;
 9 end;
10 /
Enter value for year: 2015
old 2: year number := &year;
new 2: year number := 2015;
2015is not a leap year
PL/SQL procedure successfully completed.
```

```
SQL> declare
2 a int :=0;
3 begin
4 loop
5 dbms_output.put_line(a);
6 if a>=10 then
7 exit;
8 end if;
9 a:= a+1;
10 end loop;
11 end;
12 /
0
1
2
3
4
4
5
6
6
7
8
9
10

PL/SQL procedure successfully completed.
```

29. Write a PL/SQL block to accept job from EMP table.

Give the following raise in the salary: -

By 9% if job is clerk.

By 8% if job is manager.

By 7% if job is salesman.

Update the salary of the EMP table.

```
SQL> declare
  2 a varchar2(10):='&a';
  3 begin
  4 case a
 5 when 'CLERK'then
6 update emp
  7 set sal=+(sal*(9/100)) where job=a;
8 dbms_output.put_line('salary is updated');
     when 'MANAGER' then
 10 update emp
 11 set sal=sal+(sal*(8/100)) where job=a;
 12 dbms_output.put_line('salary is updated');
 13 when 'SALESMAN' then
 14 update emp
 15 set sal=sal+(sal*(7/100)) where job=a;
 16 dbms_output.put_line('salary is updated');
 17 end case;
 18 end;
 19 /
Enter value for a: MANAGER
      2: a varchar2(10):='&a';
      2: a varchar2(10):='MANAGER';
PL/SQL procedure successfully completed.
SQL>
```

30. Write a PL/SQL block to get the details of marks (rollno, marks1, marks2, grade) out of 100 for marks1 and marks2 respectively.

Display the grade in table marks using if statement as specified below

If stud_percent > 70 then grade is 'A'

If stud_percent > 60 and <70 then grade is 'B' else give grade 'C'.

```
SQL> declare
  2 roll number := &roll;
  3 m1 number := &m1;
  4 m2 number := &m2;
  5 grade char;
  6 stud_percent number;
  7 begin
  8 stud_percent := ((m1+m2)/200) * 100;
 9 if stud_percent > 70 then
 10 insert into mark values(roll,m1,m2,'A');
 11 elsif stud_percent >60 and stud_percent > 70 then
 12 insert into mark values(roll, m1, m2, 'B');
 13 else
 14 insert into mark values(roll,m1,m2,'C');
 15 end if;
 16 end;
 17 /
Enter value for roll: 7
      2: roll number := &roll;
      2: roll number := 7;
Enter value for m1: 80
old
      3: m1 number := &m1;
      3: m1 number := 80;
Enter value for m2: 90
old 4: m2 number := &m2;
new
      4: m2 number := 90;
PL/SQL procedure successfully completed.
```

31. Write a PL/SQL block to book a ticket for a movie. The tickets are of two type's deluxe rows (D) and Ordinary rows (O).

While booking the ticket the customer may ask 'D' or 'O' and number of ticket. For deluxe the rate is 350 and for ordinary 200.

Find the total amount that the customer will pay and number of tickets (using case statement).

```
SQL> declare
  2 t_type char(1) := '&t_type';
  3 no_tickets number := '&no_tickets';
 4 begin
 5 case t_type
6 when 'd' then dbms_output.put_line('Total price: ' || no_tickets*350);
7 when 'o' then dbms_output.put_line('Total price: ' || no_tickets*200);
 9 dbms_output.put_line('Not a valid ticket type');
 10 end case;
 11 end;
12 /
Enter value for t_type: o
new 2: t_type char(1) := 'o';
Enter value for no_tickets: 2
     3: no_tickets number := '&no_tickets';
      3: no_tickets number := '2';
PL/SQL procedure successfully completed.
```