**ASSIGNMENT NO.05**

Q.1 Write a PL/SQL block for following programs.

1. Write a PL/SQL block to find given number is even or odd.

2. Write a PL/SQL block to print the given number is positive or negative.

3. Write a PL/SQL block to find maximum numbers among the 3 numbers.

4. Write a PL/SQL block to print multiplication table for given number.

5. Write a PL/SQL block to print sum of digit for a given number.

6. Write a PL/SQL block to print reverse number for a given number.

7. Write a PL/SQL block to print Fibonacci series upto given term.

8. Write a PL/SQL block to print given number is prime or not.

9. Write a PL/SQL block to print given number is Armstrong or not .

10. Write a PL/SQL block to print given number is palindrome or not.

11. Write a PL/SQL block to find out factorial of a given number.

12. Write a PL/SQL block to print string in a reverse order.

13. Write a PL/SQL block to print the given number is perfect or not.

14. Write a PL/SQL block to print a divisor of a given number.

15. Write a PL/SQL block to print prime number in between range.

16. Write a PL/SQL, block to display all armstrong numbers between 0 to 500.

Q.2 Exercise on following queries.

1. Create table emp with fields(empno ,ename ,sal, doj, dept\_no, noe).

2. Insert 5 records in the table emp.

3. Write a PL/SQL block to check whether the salary of `Asmita'>5000 or not.

4. Write a PL/SQL block to accept it that department.

5. Write a PL/SQL block to print the name of employee having maximum salary & print the salary.

6. Write a PL/SQL block to accept employee name & print the date of joining.

7. Write a PL/SQL block to print the employee number having salary<=accepted salary.

**Q.1 Write a PL/SQL block for following programs.**

**1. Write a PL/SQL block to find given number is even or odd.**

SQL> declare

2 n number(4);

3 r number(4);

4 begin

5 n := '&n';

6 r := mod(n, 2);

7 if(r = 0)

8 then

9 dbms\_output.put\_line(n || ' is Even no');

10 else

11 dbms\_output.put\_line(n || ' is Odd no');

12 end if;

13 end;

14 /

**Output:**

Enter value for n: 67

old 5: n := '&n';

new 5: n := '67';

67 is Odd no

PL/SQL procedure successfully completed.

**2. Write a PL/SQL block to print the given number is positive or negative.**

SQL> declare

2 num1 NUMBER:=&get\_num;

3 BEGIN

4 IF num1 < 0 THEN DBMS\_OUTPUT.PUT\_LINE('The number' || num1 || 'is a negative number');

5 ELSIF

6 num1 = 0 THEN DBMS\_OUTPUT.PUT\_LINE ('The number ' || num1 || ' is equal to zero');

7 ELSE

8 DBMS\_OUTPUT.PUT\_LINE ('The number ' || num1 || ' is a positive number');

9 END IF;

10 END;

11

12 /

**Output:-**

Enter value for get\_num: -9

old 2: num1 NUMBER:=&get\_num;

new 2: num1 NUMBER:=-9;

The number-9is a negative number

PL/SQL procedure successfully completed.

**3. Write a PL/SQL block to find maximum numbers among the 3 numbers.**

SQL> declare

2 a number;

3 b number;

4 c number;

5 begin

6 a := '&a';

7 b := '&b';

8 c := '&c';

9 dbms\_output.put\_line('a=' || a || 'b=' || b || 'c=' || c);

10 if a > b AND a > c

11 then dbms\_output.put\_line('a is greatest');

13 else

14 if b > a AND b > c

15 then

16 dbms\_output.put\_line('b is greatest');

17 else

18 dbms\_output.put\_line('c is greatest');

19 end if;

20 end if;

21 end;

22 /

**Output:-**

Enter value for a: 5

old 6: a := '&a';

new 6: a := '5';

Enter value for b: 48

old 7: b := '&b';

new 7: b := '48';

Enter value for c: 15

old 8: c := '&c';

new 8: c := '15';

a=5b=48c=15

b is greatest

PL/SQL procedure successfully completed.

**4. Write a PL/SQL block to print multiplication table for given number.**

SQL> declare

2 n number;

3 i number;

4 begin

5 n := & n;

6 for i in 1..10

7 loop

8 dbms\_output.put\_line(n || 'x' || i || '=' || n \* i);

9 end loop;

10 end;

11

12 /

**Output:-**

Enter value for n: 5

old 5: n := & n;

new 5: n := 5;

5x1=5

5x2=10

5x3=15

5x4=20

5x5=25

5x6=30

5x7=35

5x8=40

5x9=45

5x10=50

PL/SQL procedure successfully completed.

**5. Write a PL/SQL block to print sum of digit for a given number.**

SQL> declare

2 n number;

3 i number;

4 s number := 0;

5 r number;

6 d number;

7

8 begin

9 n := & n;

10 d := n;

11 while n > 0

12 loop r := mod(n, 10);

13 s := s + r;

14 n := trunc(n / 10);

15 end loop;

16 dbms\_output.put\_line(d || '' || 'Sum of Digit Is' || '' || s);

17 end;

18 /

**Output:-**

Enter value for n: 7345

old 9: n := & n;

new 9: n := 7345;

7345Sum of Digit Is19

PL/SQL procedure successfully completed.

**6. Write a PL/SQL block to print reverse number for a given number.**

SQL> declare

2 n number;

3 i number;

4 rev number := 0;

5 r number;

6 begin

7 n := & n;

8 while n > 0

9 loop

10 r := mod(n, 10);

11 rev :=(rev \* 10) + r;

12 n := trunc(n / 10);

13 end loop;

14 dbms\_output.put\_line('reverse is ' || rev);

15 end;

16 /

**Output:-**

Enter value for n: 6789

old 7: n := & n;

new 7: n := 6789;

reverse is 9876

PL/SQL procedure successfully completed.

**7. Write a PL/SQL block to print Fibonacci series upto given term.**

SQL> declare

2 a number(3) := 1;

3 b number(3) := 1;

4 c number(3);

5 n number(3):=&n;

6 begin

7 Dbms\_output.put\_line('the fibinocci series is:');

8 while a <= n

9 loop

10 dbms\_output.put\_line(a);

11 c := a + b;

12 a := b;

13 b := c;

14 end loop;

15 end;

16 /

**Output:-**

Enter value for n: 100

old 5: n number(3):=&n;

new 5: n number(3):=100;

the fibinocci series is:

1

1

2

3

5

8

13

21

34

55

89

PL/SQL procedure successfully completed.

**8. Write a PL/SQL block to print given number is prime or not.**

SQL> declare

2 n number;

3 i number;

4 flag number;

5 begin

6 i := 2;

7 flag := 1;

8 n:=&n;

9 for i in 2..n / 2

10 loop

11 if mod(n, i) = 0

12 then

13 flag := 0;

14 exit;

15 end if;

16 end loop;

17

18 if flag = 1

19 then

20 dbms\_output.put\_line('prime');

21 else

22 dbms\_output.put\_line('not prime');

23 end if;

24 end;

25

26 /

**Output:-**

Enter value for n: 17

old 8: n:=&n;

new 8: n:=17;

prime

PL/SQL procedure successfully completed.

**9. Write a PL/SQL block to print given number is Armstrong or not .**

SQL> DECLARE

2 n number(3);

3 s number(3) := 0;

4 t number(3);

5 BEGIN

6 n := & n;

7 t := n;

8 while t > 0 loop

9 s := s + power((t mod 10), 3);

10 t := trunc(t / 10);

11 end loop;

12

13 if(s = n) then dbms\_output.put\_line(

14 'The Given Number' || n || ' is an Armstrong Number'

15 );

16 else dbms\_output.put\_line(

17 'The Given Number ' || n || ' is Not an Armstrong Number'

18 );

19 end if;

20 END;

21 /

**Output:-**

Enter value for n: 153

old 6: n := & n;

new 6: n := 153;

The Given Number153 is an Armstrong Number

PL/SQL procedure successfully completed.

**10. Write a PL/SQL block to print given number is palindrome or not.**

SQL> Declare

2 a number(4);

3 b number(4);

4 n number(4);

5 Begin

6 a := & a;

7 b := 0;

8 n := a;

9 while n <> 0

10 loop

11 b := 10 \* b + n mod 10;

12 n := n / 10;

13 end loop;

14 if a = b

15 then

16 dbms\_output.put\_line('given number is palindromial number ');

17 else

18 dbms\_output.put\_line('given number is not palindromial number ');

19 end if;

20 end;

21 /

**Output:-**

Enter value for a: 131

old 6: a := & a;

new 6: a := 131;

given number is palindromial number

PL/SQL procedure successfully completed.

**11. Write a PL/SQL block to find out factorial of a given number.**

SQL> declare

2 n number;

3 fac number := 1;

4 i number;

5

6 begin

7 n := & n;

8

9 for i in 1..n

10 loop

11 fac := fac \* i;

12 end loop;

13

14 dbms\_output.put\_line('factorial=' || fac);

15 end;

16 /

**Output:-**

Enter value for n: 5

old 7: n := & n;

new 7: n := 5;

factorial=120

PL/SQL procedure successfully completed.

**12. Write a PL/SQL block to print string in a reverse order.**

SQL> declare

2 str1 varchar2(50) := '&str';

3 str2 varchar2(50);

4 len number;

5 i number;

6

7 begin

8 len := length(str1);

9

10 for i in reverse 1..len

11 loop

12 str2 := str2 || substr(str1, i, 1);

13 end loop;

14

15 dbms\_output.put\_line('Reverse of String is :'||str2);

16 end;

17 /

**Output:-**

Enter value for str: seema

old 2: str1 varchar2(50) := '&str';

new 2: str1 varchar2(50) := 'seema';

Reverse of String is :amees

PL/SQL procedure successfully completed.

**13. Write a PL/SQL block to print the given number is perfect or not**

SQL> declare

2 n number;

3 i number;

4 tot number;

5

6 begin

7 n := & n;

8 tot := 0;

9 for i in 1..n / 2

10 loop

11 if(n mod i = 0) then

12 tot := tot + i;

13 end if;

14 end loop;

15

16 if (n = tot) then

17 dbms\_output.put\_line('Perfect no');

18 else

19 dbms\_output.put\_line('Not a Perfect no');

20 end if;

21 end;

22 /

**Output:-**

Enter value for n: 6

old 7: n := & n;

new 7: n := 6;

Perfect no

PL/SQL procedure successfully completed.

**14. Write a PL/SQL block to print a divisor of a given number.**

SQL> declare

2 n number;

3 i number;

4 r number;

5

6 begin

7 n := '&n';

8 i := 1;

9 r := 0;

10

11 while(n >= i)

12 loop

13 r := n mod i;

14 if(r = 0)

15 then

16 dbms\_output.put\_line(i);

17 end if;

18 i := i + 1;

19 end loop;

20 end;

21 /

**Output:-**

Enter value for n: 28

old 7: n := '&n';

new 7: n := '28';

1

2

4

7

14

28

PL/SQL procedure successfully completed.

**15. Write a PL/SQL block to print prime number in between range.**

**16. Write a PL/SQL, block to display all armstrong numbers between 0 to 500.**

SQL> DECLARE

2 n number;

3 s number;

4 t number;

5 r number;

6 m number;

7

8 BEGIN

9 n := 1;

10 dbms\_output.put\_line('\*\*\*\* Armstrong Number Between 1 TO 500\*\*\*\*\*\*\*');

11

12 while n <= 500

13 loop

14 t := n;

15 m := t;

16 s := 0;

17 r := 0;

18 while t > 0 loop

19 r := mod(t, 10);

20 s := s+r\*r\*r;

21 t:=trunc(t/10);

22 end loop;

23

24 if(s=m) then

25 dbms\_output.put\_line(s||' Is Armstrong Number ');

26 end if;

27 n:=n+1;

28

29 end loop;

30 END;

31 /

**Output:-**

\*\*\*\* Armstrong Number Between 1 TO 500\*\*\*\*\*\*\*

1 Is Armstrong Number

153 Is Armstrong Number

370 Is Armstrong Number

371 Is Armstrong Number

407 Is Armstrong Number

PL/SQL procedure successfully completed.

**Q.2 Exercise on following queries.**

**1. Create table emp with fields(empno ,ename ,sal, doj, dept\_no, noe).**

SQL> create table Employees(Empno number(3),ename varchar2(20),

2 sal number(7,2),

3 DOJ date,

4 Dptno varchar2(4),

5 Noe number(5)

6 );

Table created.

SQL> desc Employees;

Name Null? Type

----------------------------------------- -------- ----------------------------

EMPNO NUMBER(3)

ENAME VARCHAR2(20)

SAL NUMBER(7,2)

DOJ DATE

DPTNO VARCHAR2(4)

NOE NUMBER(5)

**2. Insert 5 records in the table emp.**

SQL> insert into Employees values(1,'Mina',8000,'15-May-2017','D1',3);

1 row created.

SQL> insert into Employees values(2,'Snehal',5000,'15-Jun-2019','D2',5);

1 row created.

SQL> insert into Employees values(3,'Akku',5000,'20-Jul-2011','D3',7);

1 row created.

SQL> insert into Employees values(4,'Panu',2000,'20-Sep-2015','D4',10);

1 row created.

SQL> insert into Employees values(5,'Guddi',9000,'20-Jan-2017','D5',2);

1 row created.

SQL> insert into Employees values(6,'Asmita',6000,'27-Jan-2017','D3',7);

1 row created.

**3. Write a PL/SQL block to check whether the salary of `Asmita'>5000 or not.**

SQL> declare

2 salary Employees.sal%TYPE := 0;

3 BEGIN

4 SELECT sal into salary FROM Employees WHERE ename = 'Asmita' and sal > 5000;

5

6 if salary > 0 then

7 dbms\_output.put\_line('Yes Employee exists');

8 else

9 dbms\_output.put\_line('No Employee doesnt exists');

10 end if;

11

12 END;

13 /

**Output:-**

Yes Employee exists

PL/SQL procedure successfully completed.

**5. Write a PL/SQL block to print the name of employee having maximum salary & print the salary.**

**6. Write a PL/SQL block to accept employee name & print the date of joining.**

**7. Write a PL/SQL block to print the employee number having salary<=accepted salary.**