22-03-2021

Exp-9 PL or SQL

Aim : To study the various basic PL/SQL operations on the database.

Queries:

1. Write a PL/SQL coding for addition of two numbers.

```
declare
a number: =5;
b number: = 10;
c number;
begin
c: = a+b;
dbms_output.put_line ('Sum of two numbers : =
' || c);
end;
/
```

2. Using PL/SQL general syntax for if condition, declare two variables b and c and print the maximum among them.

```
DECLARE
  b number;
  c number;
BEGIN
  DBMS_OUTPUT.PUT_LINE ('Enter a number')
    b := &NUMBER;
  DBMS_OUTPUT.PUT_LINE ('Enter another number')
```

```
    C := &NUMBER;

IF b<c THEN
   DBMS_OUTPUT.PUT_LINE ("'||c||' is greater
   than '||b||"");

ELSE
   dbms_output.put_line ("'||b||' is greater
   than '||c||"");
ENDIF
END;
/
```

3. Using PL/SQL, get a number and print if it is less than 5 or greater than.

```
DECLARE
   N   NUMBER;
BEGIN
   dbms_output.put_line ('Enter a number')
      N := &NUMBER;
IF N>5 THEN
   dbms_output.put_line ("'||N||'
   is greater than 5');
ELSE
   dbms_output.put_line ("'||N||' is lesser
   than 5');
ENDIF;
END;
/
```

4. Using PL/SQL, general syntax for nested if, get three numbers as input and print which is maximum

```
DECLARE
    a NUMBER;
    b NUMBER;
    c NUMBER;
BEGIN
dbms_output.Put_line ('Enter 3 numbers :=');
    a := &NUMBER;
    b := &NUMBER,
    c := &NUMIZER

IF  a> b
    AND  a>c  THEN
    dbms_output.Put_line('Greatest number is ' || a);

ELSE
    IF  b > a
        AND  b>c  THEN
        dbms_output. Put_line('Greatest number is'
        || b );
    ELSE
        dbms_output. Put_line ('Greatest number is'
        || c );
        ENDIF;
    ENDIF;
    END;
/
```

5. Using PQL/SQL general syntax for looping statement, find sum of odd numbers till a given value and print it.

```
DECLARE
  num NUMBER(3);
  sum1 NUMBER(4) := 0;
BEGIN
  i NUMBER(3) := 1;

BEGIN
  dbms_output.Put_line('Enter the number:')
  num := &NUMBER;
  FOR i in 1 .. num LOOP
    IF mod(i,2) = 1 THEN
      sum1 = sum1+i;
    ENDIF;
  END LOOP;
  dbms_output.put_line('Sum of odd numbers:'
  || sum1);
END;
/
```

6. Using PL/SQL, using general syntax of while loop find the sum of odd numbers till given value and print it.

```
DECLARE
  num NUMBER(3);
  sum1 NUMBER(4) := 0;
  i NUMBER(3) := 1;
BEGIN
```

```
WHILE   i <= num   LOOP
    IF  mod(i, 2) = 1  THEN
        sum1 = sum1 + i;
    ENDIF
    i := i+1;
END LOOP;
dbms_output.put_line (' Sum of odd numbers is '
|| sum1);
END;
/
```

2. ~~TRIGGER~~

8. Find factorial of a number using Function.

```
declare
n number(4):= &n;
fact number(8);
begin
  fact = Factorial(n);
dbms_output.put_line ('Factorial of' || n || 'is'
||Fact);
end;
/
<Main Program>

FUNCTION  Factorial ( n  number)
  RETURN  number
  IS
    f  number;
  BEGIN
      IF  n = 0  then
        f := 1;
      ELSE
        f := n * Factorial(n-1);
  ENDIF
```

```
RETURN f;
END FACTORIAL;
```

9. a Write a procedure using positional parameters.

```
CREATE OR REPLACE PROCEDURE Call ( pA Varchar2,
pA pB NUMBER, pc Boolean, pD Date) AS
BEGIN
   NULL;
END Call;
/
DECLARE
    V1   Varchar2(10);
    ~~V2  Varchar2(7~~
    V2   number(7,6);
    V3   boolean;
    V4   Date;
BEGIN
   Call (V1, V2, V3, V4);
END;
/
```

~~7.~~ ~~TRIGGER~~

9. b Write a procedure using notational parameters.

```
set serveroutput on;
create or replace procedure Grade (n IN
number, grade OUT Varchar) as
begin
if   n>90  then
   grade = 'A';
else if  n>80  then
```

```
    grade = 'B';

  else if n >70 then
  grade = 'C';

  else if n > 60 then
    grade = "D";
  else
    grade = "E";
  end if;
  end;
  /
declare
  n number;
  grade varchar(1);
begin
  n:= 91;
  Grade(n, grade);
  dbms_output.put_line('Grade: ' || grade);
  end;
  /
Output:
   Grade: A
```