Exp-6

Date

Aim: Write a PL/SQL to add two numbers

PL/SQL:

```
declare
    a number:=30;
    b number:=13;
    c number;
begin
    c:=a+b;
    dbms_output.put_line('sum of'||a||'and'||b||
        'is'||c);
end;
```

Exp-7

Date

Aim: Write a PL/SQL to check whether the given number is even or odd

PL/SQL:

```
declare
   a number:='&a';
begin
   if(mod(a,2)=0) then
      dbms_output.put_line(a || ' is even');
   else
      dbms_output.put_line(a || 'is odd');
   end if;
end;
```

Expt-8

Date

Aim: Write a PL/SQL to generate natural number using various loops.

PL/SQL:

```
declare
    a number := &a;
    i number := 1;
begin
    if (a>0) then
        for i in 1..a
        loop
        dbms_output.put_line(i);
        end loop;
    end if;
end;
```

Expt-9

Date

Aim: Write a PL/SQL to find the roots of quadratic equation.

PLSQL:

```
declare
   a number:=&a;
   b number:=&b;
   c number:=&c;
   d number;
   r1 number;
   r2 number;
begin
   d:=b*b-4*a*c;
   if (d>0) then
      r1:=(-b+sqrt(d))/(2*a);
      r2:=(-b-sqrt(d))/(2*a);
      dbms_output.put_line('the roots are'||r1||'and'||r2||);
   else if(d=0) then
      r1:=-b/(2*a);
      dbms_output.put_line('the roots are'||r1||'and'||r1||);
   else
      r1:=-b/(2*a);
      r2:=sqrt(-d)/(2*a);
      dbms_output.put_line('the roots are'||r1||'and'||'+i'
            ||r2||r1||'-i'||r2);
   end if;
end;
```

Expt-10

Date

Aim: Write a PL/SQL to check whether the given number is prime or not.

PL/SQL:

```
declare
    n number := &n;
    i number := 1;
    cnt number := 0;
begin
    for i in 1..n
      loop
      if (mod (n,i)=0 ) then
          cnt := cnt+1;
      end if;
      end loop;
   if (cnt = 2) then
     dbms_output.put_line(n || 'is prime');
   else
     dbms_output.put_line(n || ' is not prime');
   end if;
end;
```

Expt-11

Date

Aim: Write a PL/SQL to generate prime numbers upto m.

PL/SQL:

```
declare
   m number:=&m;
   j number;
   n number;
   c number;
   i number;
begin
    j:=2
    while(j<=m)
      loop
        n:=j;
        i:=1;
        c:=0;
        while(i<=n)
          loop
            if(mod(n,i)=0) then
               c:=c+1;
            end if;
            i:=i+1;
          end loop;
        if(c=2) then
          dbms_output.put_line(n);
        end if;
     end loop;
end;
```

Expt-12

Date

Aim: Write a PL/SQL to generate mathematical product table for a given number.

PL/SQL:

```
declare
    n number:=&n;
    i number:=1;
begin
    for i in 1..10
      loop
      dbms_output.put_line(n||'*'||i||'='||n*i);
      end loop;
end;
```

Expt-13

Date

Aim: Write a PL/SQL to check whether the given number is palindrome or not.

PL/SQL:

```
declare
   n number:=&n;
   rem number:=0;
   tot number:=0;
   k number:=0;
begin
   k:=n;
   while(n>0)
     loop
     rem:=mod(n,10);
     sum:=sum*10+rem;
     n:=trunc(n/10);
    end loop;
   if (sum=k) then
     dbms_output.put_line(k||'is palindrome');
   else
     dbms_output.put_line(k||'is not palindrome');
   end if;
end;
```

Expt-14

Date

Aim: Write a PL/SQL to check whether the given string is Palindrome or not.

PL/SQL:

```
declare
    g varchar2(20);
    r varchar2(20);
    i number(4);
begin
    g:='&g';
    for i in reverse 1..length(g)
    loop
        r:=r||substr(g,i,1);
    end loop;
    dbms_output.put_line('reverse string is' ||r);
    if r=g then
        dbms_output.put_line('string is palindrome');
    else
        dbms_output.put_line('string is not palindrome');
    end if;
end;
```

Expt-15

Date

Aim: Write a PL/SQL to find the date of birth of a given programmer.

PL/SQL:

```
declare
    pn programmer.pname%.type:='&pn';
    db programmer.dob%.type;
begin
    select dob into db from programmer where pname=pn;
    dbms_output.put_line('Date of birth is '||db);
exception
    when no_data_found then
        dbms_output.put_line('No Data');
end;
```

Expt-16

Date

Aim: Write a PL/SQL to display the names and date of birth of programmers.

PL/SQL:

```
declare
    cursor s is select * from programmer;
    t s%.rowtype;
begin
    open s;
    loop
        fetch s into t;
        exit when s%.not found;
        dbms_output.put_line ('Pname' ||t.pname|| ' '
                ||'Date of Birth' || t.dob);
    end loop;
    Close s;
end;
```

Expt-17

Date

Aim: Write a PL/SQL to find the titles of projects done by a given programmer.

PL/SQL:

```
declare
    pn software.pname%type := 'Tulasi';
    cursor s is select * from software where pname=pn;
    t s%rowtype;
begin
    open s;
    loop
        fetch s into t;
        exit when s%notfound;
        dbms_output.put_line(t.title);
    end loop;
    close s;
end;
```

Expt-18

Date

Aim: Write a PL/SQL to find the name of programmer for a given project.

PL/SQL:

```
declare
    t1 software.title%type:='&t';
    pn software.pname%type;
begin
    select pname into pn from software where title=t;
    dbms_output.put_line('Name '||pn);
exception
    when no_data_found then
    dbms_output.put_line('No Data');
end;
```

Expt-19

Date

Aim: Write a PL/SQL to calculate area and perimeter of radii present in the table radius and insert the radius, area and perimeter into another table circle.

PL/SQL:

```
declare
    cursor s is select * from radius1;
    t s%.rowtype;
begin
    open s;
    loop
        fetch s into t;
        exit when s%.not found;
        insert into circle1 values(t.radius, 3.14 *
            t.radius * t.radius, 2 * 3.14 * t.radius);
    end loop;
end;
```

Expt-20

Date

Aim: Write a procedure to get the date of birth for a given programmer.

PL/SQL:

a)
```
create or replace procedure db(p in programmer.pname
%type) as
    d programmer.dob%type;
begin
    select dob into d from programmer where pname
    =p;
    dbms_output.put_line('Dob is '||d);
exception
    when no_data found then
    dbms_output.put_line('No data');
end;
```

b)
```
declare
    x programmer.pname%type:=&x;
begin
    db(x);
end;
```

Expt-21

Date

Aim: Write a function to return the date of birth for a given programmer.

PL/SQL:

a)
```
Create or replace function getdb (p in programmer.pname
%type) return date as
        d programmer.dob%type;
begin
        select dob into d from programmer where pname=p;
        return(d);
end;
```

b)
```
declare
    x programmer.pname%type:=&x;
    r programmer.dob%type;
begin
    r:=getdb(x);
    dbms_output.put_line('Date of birth is ' ||r);
end;
```

Expt-22.

Date

Aim: Write a procedure to display the names of programmer studied in a given institute.

PL/SQL:

a)
```
create or replace procedure getname(sp in studies.splace
%.type) as
      cursor s is select * from studies where splace=sp;
      t s%.rowtype;
begin
     open s;
     loop
        fetch s into t;
        exit when s%.notfound;
        dbms_output.put_line(t.pname);
        end loop;
     Close s;
end;
```

b)
```
declare
     x studies.splace%.type:=&x;
begin
     getname(x);
end;
```

Expt-23

Date

Aim: Write a function to calculate the total developme-nt cost for a given programmer using cursors.

PL/SQL:

a)
```
create or replace function gettot (p in software.pname
%type) return number as
        r number;
        cursor s is select * from software where pname=p;
        t s%rowtype;
    begin
        r:=0;
        open s;
        loop
            fetch s into t;
            exit when s%notfound;
            r:=r+t.dcost;
        end loop;
        close s;
        return(r);
    end;
```

b)
```
declare
        res number;
        P software.pname%type:='Anand';
    begin
        res:=gettot(p);
        dbms_output.put_line('Total development cost is'
            ||res);
    end;
```