PART –II

1) Get the details for a given programmer

```
Declare
      name
programmer.pname%type:='& name';
      p1 programmer.prof1%type;
      p2 programmer.prof2%type;
      p3 programmer.salary%type;
begin
      select prof1,prof2,salary INTO
            p1,p2,p3 from programmer
                        where pname=name;
      dbms_output.put_line('name of
person:'||name);
      dbms_output.put_line('language
known1:'||p1);
      dbms_output.put_line('language
known2:'||p2);
      dbms_output.put_line('salary
paid:'||p3);
end;
```

2) Insert the new tuple into the table programmer

```
declare
      name
programmer.pname%type:='&name';
      db programmer.dob%type:='&db';
      dj programmer.doj%type:='&dj';
      ge programmer.gender%type:='&ge';
      p1 programmer.prof1%type:='&p1';
      p2 programmer.prof2%type:='&p2';
      s programmer.salary%type:=&s;
begin
      insert into programmer
values(name,db,dj,ge,p1,p2,s);
```

```
    dbms_output.put_line('inserted');
commit;
end;
```

3) Delete an unwanted tuple from the table programmer

```
declare
    name programmer.pname%type:='& name';
begin
    delete from programmer where pname=name;
    dbms_output.put_line('deleted');
end;
```

4) Create a table maths(a,b,a+b,a-b,a*b,a/b). Read two numbers a& b and insert a tuple into maths in the form (a,b,a+b,a- b,a*b,a/b).

```
declare
    a number:=& a;
    b number:=& b;
begin
    if(b=0) then
        insert into maths values(a,b,a+b,a-b,a*b,null);
    else
        insert into maths values(a,b,a+b,a-b,a*b,a/b);
    end if;
    commit;
end;
```

5) Display programmer name and his software details

```
declare
    cursor s is select * from software;
    x  s%rowtype;
begin
    open s;
    loop
        fetch s into x;
        exit when s%notfound;

dbms_output.put_line(‘pname:’||x.pname||   ‘title:’ ||x.title||’devd:’||x.dev_d);
    end loop;
dbms_output.put_line(s%rowcount);
close s;
end;
create table males(pname
 ,dob,doj,gender,prof1,prof2,sal)
```

6) Get the details of male programmers and insert into another table males

```
declare
```

```
    cursor s is select * from programmer
where gender='m';
    x  s%rowtype;
begin
        open s;
    loop
        fetch s into x;
        exit when s%notfound;
        insert into males
values(x.pname,x.dob,x.doj,x.gender,
                x.prof1,x.prof2,x.salary);
    end loop;
commit;
close s;
    end;
```

7) Radius:

```
declare
```

```
    cursor s is select radi from radius;
    xs%rowtype;
begin
    open s;
    loop
        fetch s into x;
        exit when s%notfound;
        insert into circle values(x.radi,3.14*x.radi*x.radi,2*3.14*x.radi);
    end loop;
    close s;
end;
```

8) Get nth highest paid programmer details

```
declare
```

```
    cursor s is select * from programmer
order by salary desc;
    t  s%rowtype;
    n number:='&n';
    c number:=0;
    x number;
    temp number:=0;
begin
    open s;
    loop
        fetch s into t;
        if(temp!=t.salary) then
            c:=c+1;
        end if;
        temp:=t.salary;
        if (c=n) then
            x:=t.salary;
        end if;
        if (x=t.salary) then

dbms_output.put_line(t.pname);
        end if;
        if (x>t.salary) then
            exit;
        end if;
    end loop;
    close s;
end;
```

9) Procedure & function to calculate sum of two numbers

Procedure:

```
create or replace procedure addition(a in number,b in number) as
c number;
```

```
begin
    c:=a+b;
    dbms_output.put_line('sum is'||c);
end;
```

Execution:

```
sql> get procd9.sql;
Procedure created
sql> exec addition(4,7);
sum is 11
```

Creating Function:

```
create or replace function addition1(a in number,b in number) return number
as
c number;
begin
    c:=a+b;
    return (c);
```

```
end;
```

Calling Function:

```
declare
    x number:=&x;
    y number:=&y;
    z number;
begin
    z:=addition1(x,y);
    dbms_output.put_line('sum is'||z);
end;
```

10) Procedure & function to get details for given programmer

Procedure:

```
create or replace procedure
details(name in
programmer.pname%type)
```

```
as
p1 programmer.prof1%type;
p2 programmer.prof2%type;
pay programmer.salary%type;
begin
    select prof1,prof2,salary into p1,p2,pay
from programmer    where pname=name;

dbms_output.put_line('name:'||name);
    dbms_output.put_line('lang1:'||p1);
    dbms_output.put_line('lang2:'||p2);
    dbms_output.put_line('salary:'||pay);
end;
/
sql> exec details(‘vijaya’);
```

Creating Function:

```
create or replace function detail(name in
programmer.pname%type) return
varchar2 as
p1 programmer.prof1%type;
p2 programmer.prof2%type;
pay programmer.salary%type;
x varchar2(30);
begin
    select prof1,prof2,salary into
p1,p2,pay
from programmer
    where pname=name;
    x:=name||' '||p1||' '||p2||' '||pay;
    return (x);
end;
/
O /P: Function created
```

Calling function:

```
declare
      name programmer.pname%type:='& name';
      z varchar2(30);
begin
      z:=detail(name);
      dbms_output.put_line('details are:'||z);
end;
```

11) Function to calculate salary of highest paid male programmer

Creating function:

```
create or replace function payment(gender1 in programmer.gender%type) return varchar2
as
```

```
    i number;
    x varchar2(20):=' ';
cursor s is select pname,salary from
programmer where gender=gender1 and
salary=(select max(salary) from
programmer where gender=gender1);
    a  s%rowtype;
begin
    i:=0;
    open s;
    loop
        fetch s into a;
        exit when s%notfound;
        i:=i+1;
        x:=x||'('||i||')'||' '||a.pname||'
'||a.salary;
    end loop;
    close s;
    return(x);
end;
```

Calling function:

```
Declare
gender1
programmer.gender%type:='& gender1’;
    result varchar2(20);
begin
    result:=payment(gender1);
    dbms_output.put_line('highest paid programmer salary is'||result);
end;
```

12) write a package with one function of product of two numbers. Procedure generates mathematical table

Package specification:

```
create or replace package pack as
```

```
function product(a in number,b in number) return number;
procedure mathtable(a in number);
end pack;
/
```

o/p: Package created

Package body:

```
create or replace package body pack is
    function product(a in number,b in number) return number as
        res number;
    begin
        res:=a*b;
        return(res);
end product;
    procedure mathtable(a in number) as
        i number;
    begin
        for i in 1..10
        loop
            dbms_output.put_line(a||' * '||i||'  = '||a*i);
        end loop;
    end  mathtable;
end pack;
/
```

o/p: package body created

<u>Main program</u>:

```
declare
    n1 number:=& n1;
    n2 number:=& n2;
    p1 number;
    t number:=& t;
begin
```

```
    p1:=pack.product(n1,n2);
    dbms_output.put_line('product of two numbers  :'||p1);
    pack.mathtable(t);
end;
```

13) Write a package with one function and one procedure, function to find the costliest package details . Procedure to find the details of programmer & packages developed

Package specification:

```
create or replace package abc as
    function costly(cost in software.scost%type) return varchar2;
    procedure detailsp(name in software.pname%type);
end;
/
```

o/p: package created

Package body:

```
create or replace package body abc is
function costly(cost in software.scost%type) return varchar2
as
    x varchar2(30):=' ';
    i number:=0;
    cursor s is select title from software where scost=cost;
    a s%rowtype;
begin
    open s;
    loop
        fetch s into a;
        exit when s%notfound;
        i:=i+1;
        x:=x||'   '||i||' -> '||a.title;
    end loop;
    close s;
    return(x);
end costly;
procedure detailsp(name in software.pname%type) as
    cursor s is select title,dev_d from software where pname=name;
    a s%rowtype;
begin
    open s;
    loop
        fetch s into a;
        exit when s%notfound;

dbms_output.put_line('Name          :'||name);
```

```
        dbms_output.put_line('Title
      :'||a.title);
        dbms_output.put_line('Platform
      :'||a.dev_d);
    end loop;
    close s;
end detailsp;
end abc;
/
```

o/p: package body created

Main program:

```
declare
    cost software.scost%type;
    tit varchar2(20);
    name
software.pname%type:='& name';
begin
```

```
    select max(scost) into cost from software;
    tit:=abc.costly(cost);
    dbms_output.put_line('title of costliest package is '||tit);
    abc.detailsp(name);
end;
```