BHUMESH BODALIA

FYMCA **UCID:-2020510009**

PRACTICAL NO:-07

Questions on Functions

1) **Write a PL/SQL block to find factorial of a number without recursion using function.**

```
create or replace function fact(n in number)
return number IS
ans number := 1;

begin

for i in 1..n loop
ans := ans*i;
end loop;

return ans;
end;

declare
a number;
b number;

begin
a := 4;
b := fact(a);
dbms_output.put_line(a||'! = '||b);
end;
```

```
Statement processed.
4! = 24
```

2) **Write a PL/SQL block to find factorial of a number using recursive function.**

```
create or replace function fact(n number)
```

```
return number IS
ans number;

begin

if n = 0 then
ans := 1;
else
ans := n * fact(n-1);
end if;

return ans;
end;
/
declare
a number;
b number;

begin
a := 9;
b := fact(a);
dbms_output.put_line(a||'! = '||b);
end;
```

```
Function created.

Statement processed.
9! = 362880
```

3) Write a PL/SQL block to demonstarte Function Overloading.

```
declare
a number;
b number;
c number;
ans number;

        function overload(n1 number, n2 number, n3 number)
        return number IS
        ans number;

        begin
        ans := n1+n2+n3;
        return ans;
        end overload;

        function overload(n1 number, n2 number)
        return number IS
        ans number;

        begin
        ans := n1+n2;
        return ans;
        end overload;

begin
a := 65;
b := 34;
c := 31;
ans := overload(c,c,c);
dbms_output.put_line(ans);

ans := overload(a,b);
dbms_output.put_line(ans);
end;
```

```
Statement processed.
93
99
```

4) Write a PL/SQL block to display name of depositor having highest bank amount using function.

```
declare
function highest
return varchar as
  name varchar(20);
  cursor c IS select * from deposit_35;
```

```
   r c%rowtype;
   max deposit_35.amount%type:=0;
Begin
max:=0;
   for r IN c loop
max:=r.amount;
name:=r.cname;
   end loop;
   return name;
End highest;

begin
dbms_output.put_line('Highest Depositer is '||highest());
end;
```

```
Statement processed.
Highest Depositer is naren
```

5) Write a PL/SQL block to display number of depositors using function.

```
declare
count1 number;
function count_depositer
return varchar as
   name varchar(20);
   cursor c IS select * from deposit_35;
   r c%rowtype;

begin
   count1:=0;
   open c;
   fetch c into r;
   while(c%found) loop
      count1:=count1+1;
      fetch c into r;
   end loop;
   return count1;
end count_depositer;
```

```
begin
dbms_output.put_line('Total Number of depositers '||count_depositer());
end;
```

```
Statement processed.
Total Number of depositers 9
```

6) Write a PL/SQL block to display number of depositors whose name starts with A using function.

```
declare
count1 number;
function count_depositer
return varchar as
   name varchar(20);
   cursor c IS select * from deposit_35 where cname like 'a%';
   r c%rowtype;

begin
   count1:=0;
   open c;
   fetch c into r;
   while(c%found) loop
      count1:=count1+1;
      fetch c into r;
   end loop;
   return count1;
end count_depositer;

begin
dbms_output.put_line('Total Number of depositers '||count_depositer());
end;
```

```
Statement processed.
Total Number of depositers 1
```

7) **Write a PL/SQL block to display branch name with fifth highest amount in deposit using function.**

```
declare
function fifth_high_amt
return varchar as
   name varchar(20);
   cursor c is select bname bm from deposit_35 order by amount desc;
   r c%rowtype;
begin
   for r in c loop
      if(c%rowcount=5) Then
name := r.bm;
      end if;
   end loop;
   return name;
end fifth_high_amt;

begin
dbms_output.put_line('branch with fifth highest amount:
'||fifth_high_amt);
end;
```

```
Statement processed.
branch with fifth highest amount: m.g.road
```

Questions on Procedures

1) **Write a PL/SQL block to display depositor name and date whose opening account is after 12/3/1998 using procedure.**

```
create or replace procedure opening_date
is
cursor c is select cname, adate from deposit_35 where adate> '12-MAR-98';
is_record_found boolean := false;
not_found exception;
begin
   for i in c loop
is_record_found := true;
```

```
dbms_output.put_line(i.cname||' '||i.adate);
   end loop;

   if not is_record_found Then
      raise not_found;
   end if;

exception
   when not_found then
dbms_output.put_line('No Depositers were joined after date 12/3/1998.');
end opening_date;
/
declare
begin
opening_date();
end;
```

```
Procedure created.
Statement processed.
No Depositers were joined after date 12/3/1998.
```

2) **Write a PL/SQL block to display deposito details of a specific name using procedure.**

```
declare
customer_name varchar(20);
procedure depo_details(customername in varchar2) is
cursor c is select * from deposit_35 where cname = customername;

begin

for i in c loop
dbms_output.put_line('Account Number: '||i.actno);
dbms_output.put_line('Name: '||i.cname);
dbms_output.put_line('Branch: '||i.bname);
dbms_output.put_line('Amount: '||i.amount);
dbms_output.put_line('Date: '||i.adate);
end loop;
end depo_details;
```

```
begin
customer_name:='sunil';
depo_details(customer_name);
end;
```

```
Statement processed.
Account Number: 107
Name: Bhumesh
Branch: andheri
Amount: 1001
Date: 05-SEP-95
```

3) **Write a PL/SQL block to update (replace) all the names of customers with first character capital and others in lower case using procedure.**

4) **Write a PL/SQL block to display amount in the format 99,999,99 using procedure.**

```
create or replace procedure change_format
is
cursor c is select amount from deposit_35;
begin
   for i in c loop
dbms_output.put_line(to_char(i.amount,'99,999,99'));
   end loop;
end change_format;
/
begin
change_format();
end;
```

```
Procedure created. Statement processed.
10,01
50,00
35,00
12,00
30,00
20,00
10,01
50,00
70,00
```

5) **Write a PL/SQL block to display average amount of depositors using procedure.**

```
create or replace procedure average_amt(average out number)
is
begin
select avg(amount) into average from deposit_35;
end average_amt;
/

declare
average number(8,2);
begin
average_amt(average);
dbms_output.put_line('Average amount '||average);
end;
```

```
Procedure created.
Statement processed.
Average amount 3189.11
```

Questions on PL/SQL Packages

1) **Create a package which consists of a function of addition of two numbers and a procedure for multiplication of two numbers.**

```
create or replace package pack1 as

procedure addition(a in number, b in number, c out number);
function multiply(a in number, b in number) return number;
end pack1;
/

create or replace package body pack1 as
procedure addition(a in number, b in number, c out number) as

begin
c := a+b;
```

```
end addition;

function multiply(a in number, b in number) return number as
begin
return(a*b);
end multiply;
end pack1;
/
declare
n1 number := 35;
n2 number := 14;
ans number;

begin
ans := pack1.multiply(n1,n2);
dbms_output.put_line(n1||' X '||n2||' = '|| ans);

pack1.addition(n1,n2,ans);
dbms_output.put_line(n1||' + '||n2||' = '|| ans);
end;
```

```
Package created.

Package Body created.

Statement processed.
35 X 14 = 490
35 + 14 = 49
```

2) **Create a package which consists of a function of factorial of a number and a procedure of factorial of a two number.**

3) **Create a package which consists of function to display name of depositor having highest bank amount and a procedure to display depositor name and date whose opening account is after 12/3/1998 using procedure**

4) **Create a package which consists of a procedure to display amount in the format 99,999,99 and a function to to display average amount of depositors.**

```
create or replace package pack3 as function avg_amt(a in number) return
varchar2;
PROCEDURE get(a in number);
end pack3;
/
Create or replace package body pack3 as function avg_amt(a in number)
return varchar2 is x number(10,2);
BEGIN
Select round(AVG(amount),2) into x FROM deposit_35;
return x;
END avg_amt;

PROCEDURE get(a in number) IS
cursor c1 is Select TO_CHAR(amount,'99,999,99') amt from deposit_35;
z c1%rowtype;
begin
open c1;
fetch c1 into z;
while(c1%found)
loop
dbms_output.put_line(z.amt);
fetch c1 into z;
end loop;
close c1;
END get;
end pack3;
/

declare
avg1 number :=0;
avg2 number :=0;
avg3 number :=0;
begin
avg3 := pack3.avg_amt(avg1); dbms_output.put_line('avg bank amount
:'||avg3);
pack3.get(avg2);
```

```
end;
```

```
Package created. Package Body created. Statement processed.
avg bank amount :3189.11

10,01
50,00
35,00
12,00
30,00
20,00
10,01
50,00
70,00
```