**DBMS LAB**

**ASSIGNMENT-10**

**SNEHA SINGH**

**20198023**

**IT 5 B2**

**Q. Create the following tables:**

Create table Emp\_Detail (employee\_id number(4), first\_name varchar2(10), last\_name varchar2(4), Salary number(30), dept\_id number(5) );

insert into Emp\_Detail values(1,'John', 'aaa1',10000,11);

insert into Emp\_Detail values(2,'Dennis', 'b1',8000,12);

insert into Emp\_Detail values(3,'Albert', 'c1',11000,13);

insert into Emp\_Detail values(4,'Charles', 'd1',7000,14);

insert into Emp\_Detail values(5,'Richard','e1',17000,15);

Create table Department (Dept\_name varchar2(10), Manager\_id number(4), dept\_id number(5));

insert into Department values('LMNT','111',1000);

insert into Department values('SRTC', '222',100);

insert into Department values('STORE', '333',1100);

insert into Department values('TEST', '444',700);

insert into Department values('ENQIRY','555',1700);

**1.Implicit Cursor:**

**a) Write a program in PL/SQL, using a cursor display all information of employee having emp id 5 from emp\_detail table.**

DECLARE

employee\_id Emp\_Detail.employee\_id%type;

first\_name Emp\_Detail.first\_name%type;

last\_name Emp\_Detail.last\_name%type;

salary Emp\_Detail.salary%type;

dept\_id Emp\_Detail.dept\_id%type;

dept\_name Department.dept\_name%type;

manager\_id Department.manager\_id%type;

cursor c\_emp is select employee\_id, first\_name, last\_name, salary, dept\_id from Emp\_Detail where

employee\_id=5;

BEGIN

open c\_emp;

loop

fetch c\_emp into employee\_id, first\_name, last\_name, salary, dept\_id;

exit when c\_emp%notfound;

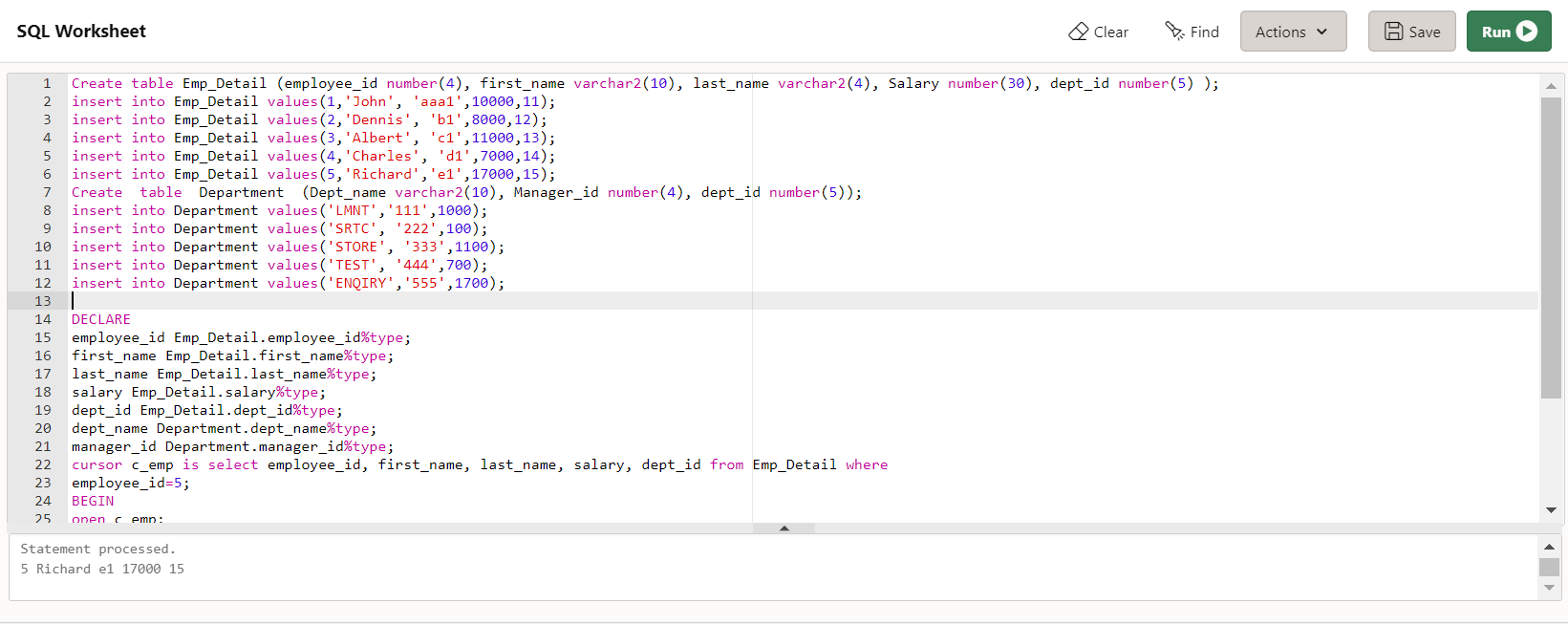
dbms\_output.put\_line(employee\_id||' '||first\_name||' '||last\_name||' '||salary||' '||dept\_id);

end loop;

close c\_emp;

end;

/

****

**b) Write a program in PL/SQL, using an implicit cursor with for loop, display the first name, dept id and dept name of employees having salary more than 5000.**

BEGIN

FOR item IN(SELECT dept\_name,d.dept\_id,last\_name,salary

FROM department d JOIN emp\_detail e

ON e.dept\_id = d.dept\_id

WHERE salary > 5000)

LOOP

DBMS\_OUTPUT.PUT\_LINE(item.last\_name||' '||item.dept\_name

||' '||item.dept\_id||' '||' '||item.salary);

END LOOP;

END;

/

****

**2.Explicit Cursor:**

declare

employee\_id Emp\_Detail.employee\_id%type;

first\_name Emp\_Detail.first\_name%type;

last\_name Emp\_Detail.last\_name%type;

salary Emp\_Detail.salary%type;

dept\_id Emp\_Detail.dept\_id%type;

dept\_name Department.dept\_name%type;

manager\_id Department.manager\_id%type;

cursor c\_emp is select employee\_id, first\_name, last\_name, salary, dept\_id from Emp\_Detail;

begin

open c\_emp;

loop

fetch c\_emp into employee\_id, first\_name, last\_name, salary, dept\_id;

exit when c\_emp%notfound;

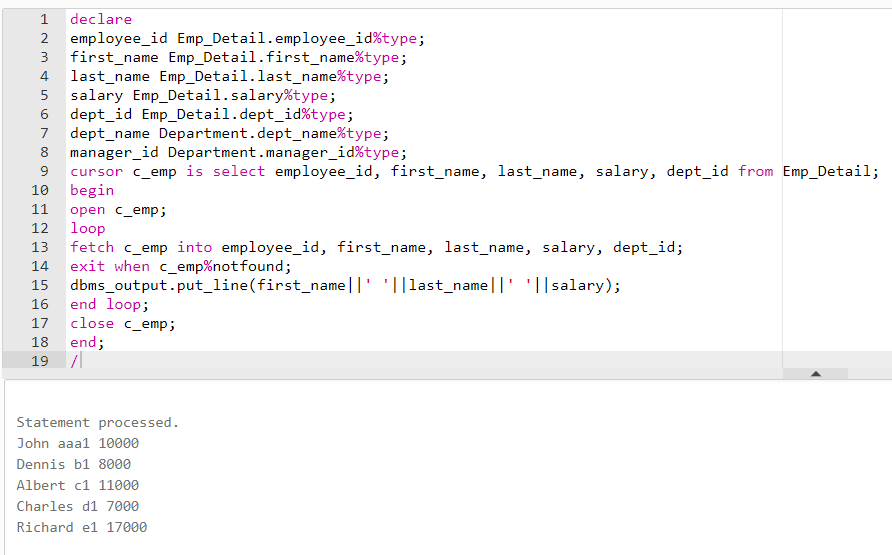
dbms\_output.put\_line(first\_name||' '||last\_name||' '||salary);

end loop;

close c\_emp;

end;

/



**a) Write a program in PL/SQL using an explicit cursor, retrieve the records from the emp\_detail table and display the details of employees whose salary is more than 8000.**

declare

employee\_id Emp\_Detail.employee\_id%type;

first\_name Emp\_Detail.first\_name%type;

last\_name Emp\_Detail.last\_name%type;

salary Emp\_Detail.salary%type;

dept\_id Emp\_Detail.dept\_id%type;

dept\_name Department.dept\_name%type;

manager\_id Department.manager\_id%type;

cursor c\_emp is select employee\_id, first\_name, last\_name, salary,dept\_id from Emp\_Detail where

salary>8000;

begin

open c\_emp;

loop

fetch c\_emp into employee\_id, first\_name, last\_name, salary, dept\_id;

exit when c\_emp%notfound;

dbms\_output.put\_line(employee\_id||' '||first\_name||' '||last\_name||' '||salary||' '||dept\_id); end loop;

close c\_emp;

end;

/



**b) Write a PL/SQL block, using a cursor attribute to update the table & increase salary of each customer by 2000.**

declare

employee\_id Emp\_Detail.employee\_id%type;

first\_name Emp\_Detail.first\_name%type;

last\_name Emp\_Detail.last\_name%type;

salary Emp\_Detail.salary%type;

dept\_id Emp\_Detail.dept\_id%type;

dept\_name Department.dept\_name%type;

manager\_id Department.manager\_id%type;

cursor c\_emp is select employee\_id, first\_name, last\_name, salary, dept\_id from Emp\_Detail for

update;

begin

open c\_emp;

loop

fetch c\_emp into employee\_id, first\_name, last\_name, salary, dept\_id;

exit when c\_emp%notfound;

update Emp\_Detail set salary = salary + 2000 where current of c\_emp;

dbms\_output.put\_line(employee\_id||' '||first\_name||' '||last\_name||' '||salary||' '||dept\_id);

end loop;

close c\_emp;

end;

/



**3.Procedure & Function:**

**a) Create a procedure to find the minimum of two values, which are passed to it using IN mode and returns their minimum value using OUT parameters.**

DECLARE

a number;

b number;

c number;

PROCEDURE findMin(x IN number, y IN number, z OUT number) IS

BEGIN

IF x < y THEN

z:= x;

ELSE

z:= y;

END IF;

END;

BEGIN

a:= 23;

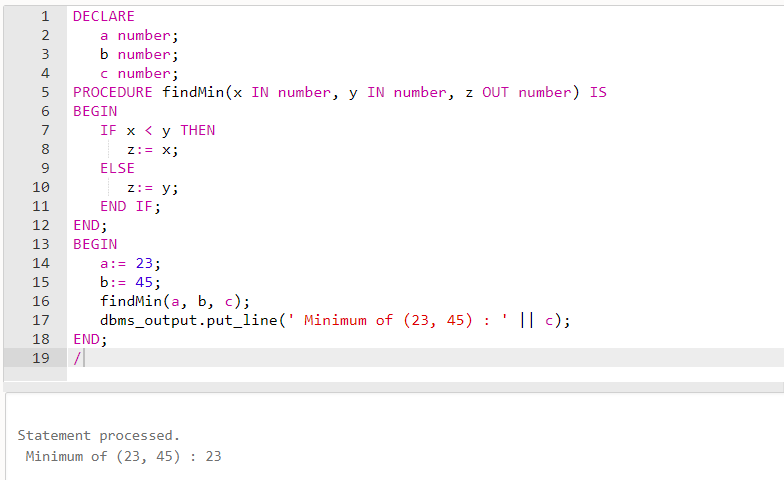
b:= 45;

findMin(a, b, c);

dbms\_output.put\_line(' Minimum of (23, 45) : ' || c);

END;

/

****

**b) Create and call a standalone function where this function returns the total number of customers from the Customers table.**

create table customers (ID number(10), name varchar2(10), age number(10), address varchar2(10), Salary Float);

insert into customers values(101,'John',32,'Lodan',20000.00);

insert into customers values(3,'Clerk',25,'Paris',15000.00);

insert into customers values(4,'Mark',22,'New York',85000.00);

insert into customers values(5,'Albert',29,'California',45000.00);

-- Declare function total\_customers

create or replace function total\_customers return number is

total number(2) := 0;

begin

select count(\*) into total from customers;

return total;

end;

/

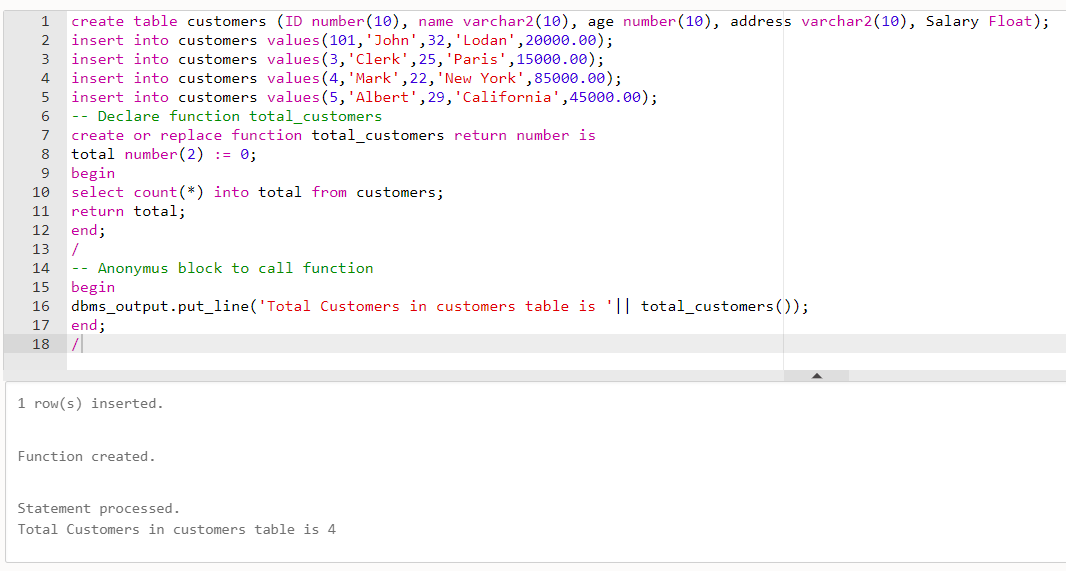
-- Anonymus block to call function

begin

dbms\_output.put\_line('Total Customers in customers table is '|| total\_customers());

end;

/

****

**c) Create a function to calculate the factorial of a given number. The function should call itself recursively.**

DECLARE

num number;

factorial number;

FUNCTION fact(x number)

RETURN number

IS

f number;

BEGIN

IF x=0 THEN

f := 1;

ELSE

f := x \* fact(x-1);

END IF;

RETURN f;

END;

BEGIN

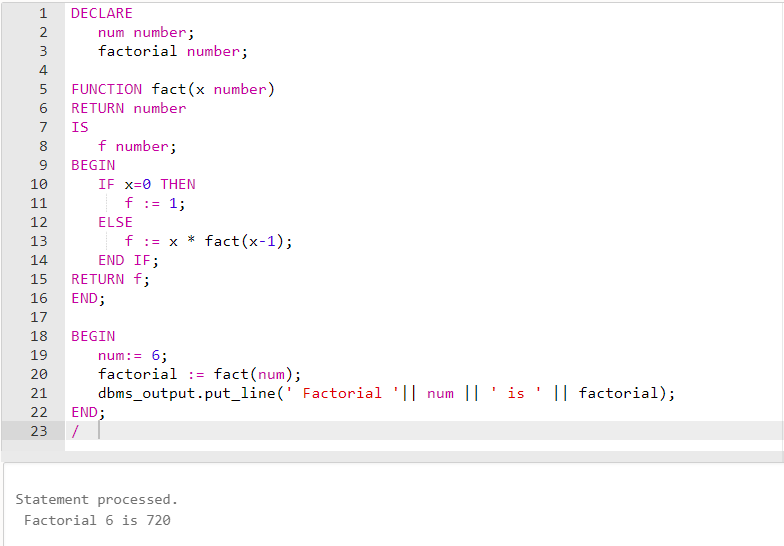
num:= 6;

factorial := fact(num);

dbms\_output.put\_line(' Factorial '|| num || ' is ' || factorial);

END;

/

****

**d) Create two procedures within a procedure and call them. Both should print the messages which distinguish them from each other.**

-- Create outside\_procedure and two inside\_procedure

create or replace

procedure outside\_procedure

is

procedure inside\_procedure1

is

begin

dbms\_output.put\_line('inside\_procedure1');

end;

procedure inside\_procedure2

is

begin

dbms\_output.put\_line('inside\_procedure2');

end;

begin

dbms\_output.put\_line('outside\_procedure');

inside\_procedure1;

inside\_procedure2;

end;

-- Call outside\_procedure using exec

exec outside\_procedure;

**-- Output : outside\_procedure**

**-- inside\_procedure1**

**-- inside\_procedure2**