LAB ASSIGNMENT 11

NAME SHISHU **REG** 2020CA089

1. 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.

```
MariaDB [assign11]> DELIMITER $$
MariaDB [assign11]> CREATE PROCEDURE getMax(IN f int,IN s int,OUT mx int)
    -> begin
    -> if f>s then
    -> set mx=f;
    -> else
    -> set mx=s;
    -> end if;
    -> END$$
Query OK, 0 rows affected (0.029 sec)
```

```
MariaDB [assign11]> call getMax(1,5,@mx);
Query OK, 0 rows affected (0.004 sec)

MariaDB [assign11]> select @mx as maximum_number;
+----------------+
| maximum_number |
+----------------+
|              5 |
+----------------+
1 row in set (0.000 sec)
```

2. Create and call a standalone function where this function returns the total number of customers from the Customers table. Use the following code to create 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);

```
MariaDB [assign11]> Create table customers (ID int(10), name varchar(10), age int(10), address varchar(10), Salary Float
 );
Query OK, 0 rows affected (0.021 sec)

MariaDB [assign11]> 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,'Al
bert',29,'California',45000.00);
Query OK, 1 row affected (0.052 sec)

Query OK, 1 row affected (0.002 sec)

Query OK, 1 row affected (0.004 sec)

Query OK, 1 row affected (0.002 sec)

MariaDB [assign11]> desc customers
    -> ;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| ID      | int(10)     | YES  |     | NULL    |       |
| name    | varchar(10) | YES  |     | NULL    |       |
| age     | int(10)     | YES  |     | NULL    |       |
| address | varchar(10) | YES  |     | NULL    |       |
| Salary  | float       | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
5 rows in set (0.016 sec)

MariaDB [assign11]> select * from customers;
+------+--------+------+------------+--------+
| ID   | name   | age  | address    | Salary |
+------+--------+------+------------+--------+
|  101 | John   |   32 | Lodan      |  20000 |
|    3 | Clerk  |   25 | Paris      |  15000 |
|    4 | Mark   |   22 | New York   |  85000 |
|    5 | Albert |   29 | California |  45000 |
+------+--------+------+------------+--------+
4 rows in set (0.000 sec)

MariaDB [assign11]>
```

Code:

```
CREATE OR REPLACE FUNCTION totalCustomers
RETURN number IS
  total number(2) := 0;
BEGIN
  SELECT count(*) into total
  FROM customers;

  RETURN total;
END;
```

It will create function, now code for calling it:

```
DECLARE
  c number(2);
BEGIN
  c := totalCustomers();
  dbms_output.put_line('Total no. of Customers: ' || c);
END;
/
```

```
Total no. of Customers: 4

Statement processed.
```

---

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

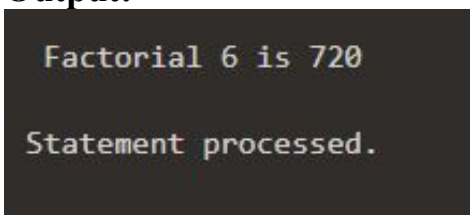
Code:

```
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;
/
```

Output:

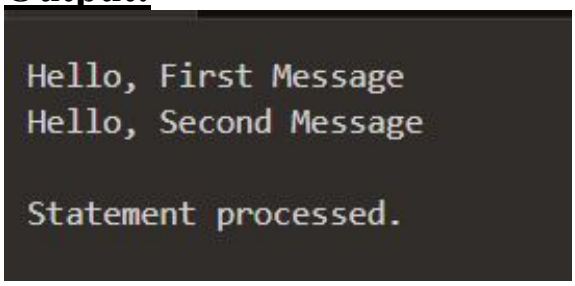


4. Create two procedures within a procedure and call them. Both should print the messages which distinguish them from each other.

Code:

```
CREATE OR REPLACE PROCEDURE messages IS
   PROCEDURE message1 (mesg VARCHAR2) IS
   BEGIN
      dbms_output.put_line(mesg);
   END;
   PROCEDURE message2 (mesg VARCHAR2) IS
   BEGIN
      dbms_output.put_line(mesg);
   END;
BEGIN
   message1('Hello, First Message');
   message2('Hello, Second Message');
END;
```

Output:


Hello, First Message
Hello, Second Message
Statement processed.