CSLR 51 : DBMS LAB-5

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

PROBLEM 1

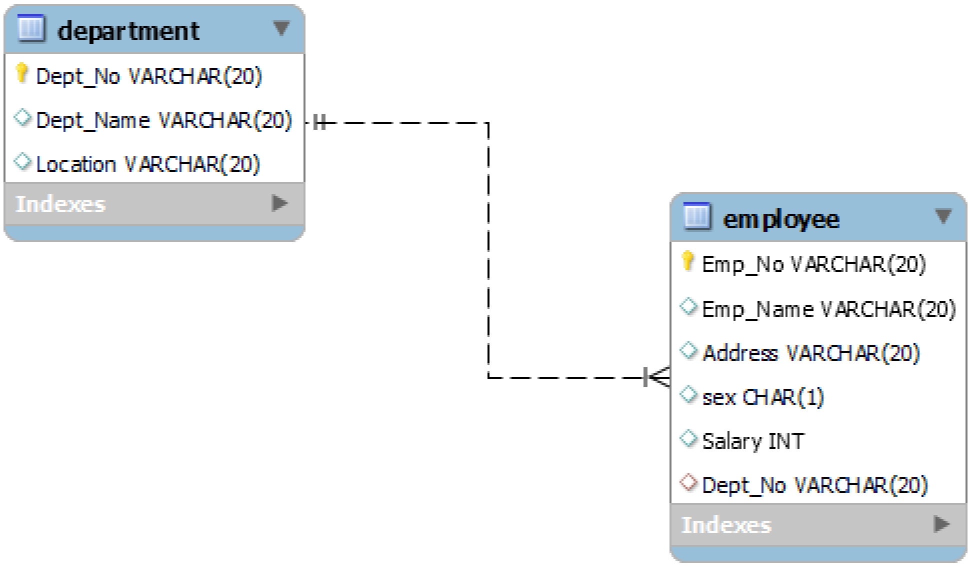
Question 1

Employee (Emp_No, Emp_Name, sex, Salary, Address, Dept_No)
Department (Dept_No, Dept_Name, Location)

Insert at least 10 employees and 4 departments so that all queries result at least 1 tuple

1. Write a procedure to display employee details given the Emp_No.
2. Write a procedure to delete an employee record given the Emp_Name.
3. Write a procedure to list all the employee names belonging to a particular department given the Dept_No.
4. Write a procedure to display the number of employees whose salary is greater than 30K.

ER Diagram :

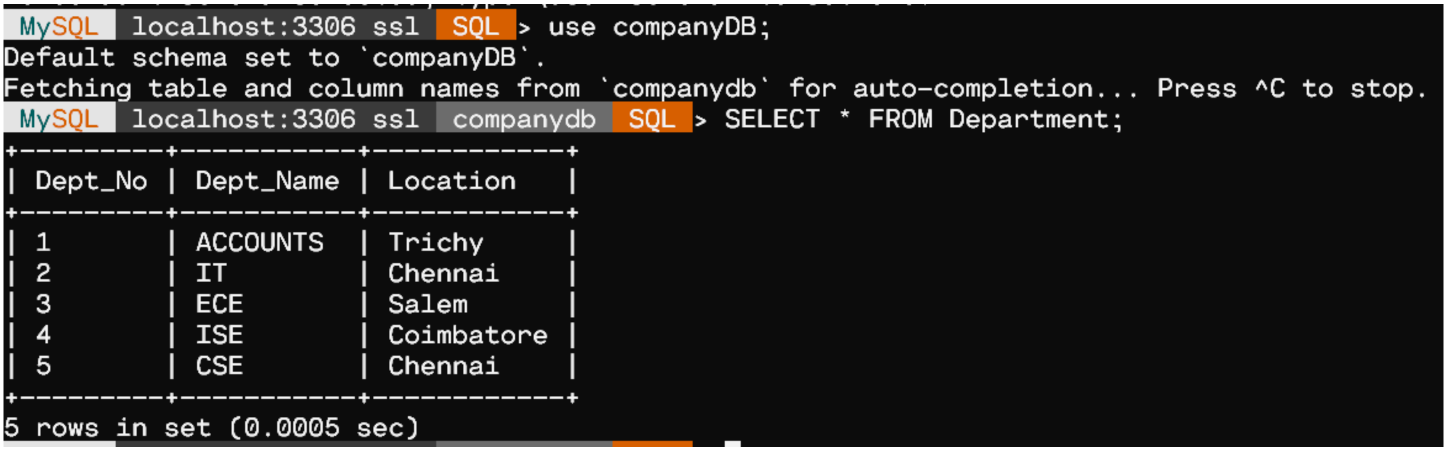


```
CREATE DATABASE CompanyDB;
USE CompanyDB;

CREATE TABLE Department  (
    Dept_No VARCHAR (20) PRIMARY KEY,
    Dept_Name VARCHAR (20),
    Location VARCHAR (20)
);
```

```
CREATE TABLE Employee (
    Emp_No VARCHAR (20) PRIMARY KEY,
    Emp_Name VARCHAR (20),
    Address VARCHAR (20),
    sex CHAR (1),
    Salary INTEGER,
    Dept_No VARCHAR (20),
    foreign key (Dept_No) REFERENCES Department (Dept_No)
);

INSERT INTO Department
    VALUES
        (1,'ACCOUNTS','Trichy'),
        (2,'IT','Chennai'),
        (3,'ECE','Salem'),
        (4,'ISE','Coimbatore'),
        (5,'CSE','Chennai');

INSERT INTO Employee
 VALUES
       ('ECE01','PREETI','BANGALORE','F', 40000,3),
       ('CSE01','JAMES','BANGALORE','M', 50000,5),
       ('CSE02','HEARN','BANGALORE','M', 70000,5),
       ('CSE03','EDWARD','MYSORE','M', 50000,5),
       ('CSE04','PAVAN','MANGALORE','M', 15000,5),
       ('CSE05','GIRISH','MYSORE','M', 20000,5),
       ('CSE06','NEHA','BANGALORE','F', 80000,5),
       ('ACC01','AHANA','MANGALORE','F', 35000,1),
       ('ACC02','SANTHOSH','MANGALORE','M', 30000,1),
       ('ISE01','VEENA','MYSORE','M', 10000,4),
       ('IT01','NAGESH','BANGALORE','M', 50000,2);
```

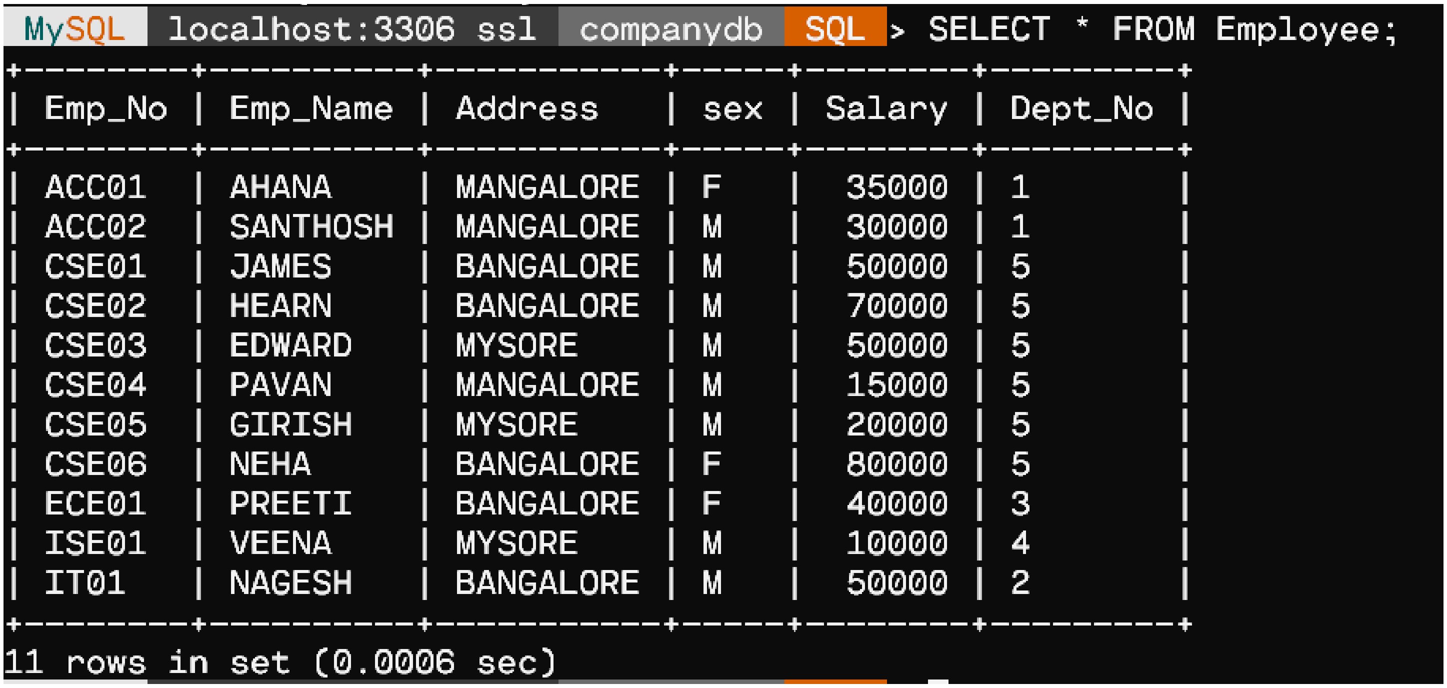
```
/*------------Queries------------*/
```

```
SELECT * FROM Department;
```



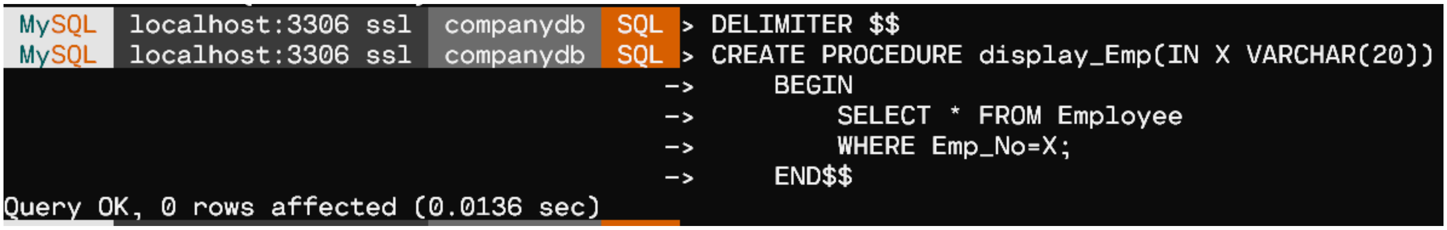
```
 MySQL  localhost:3306 ssl  SQL > use companyDB;
Default schema set to `companyDB`.
Fetching table and column names from `companydb` for auto-completion... Press ^C to stop.
 MySQL  localhost:3306 ssl  companydb  SQL > SELECT * FROM Department;
+---------+-----------+------------+
| Dept_No | Dept_Name | Location   |
+---------+-----------+------------+
| 1       | ACCOUNTS  | Trichy     |
| 2       | IT        | Chennai    |
| 3       | ECE       | Salem      |
| 4       | ISE       | Coimbatore |
| 5       | CSE       | Chennai    |
+---------+-----------+------------+
5 rows in set (0.0005 sec)
```

```
SELECT * FROM Employee;
```



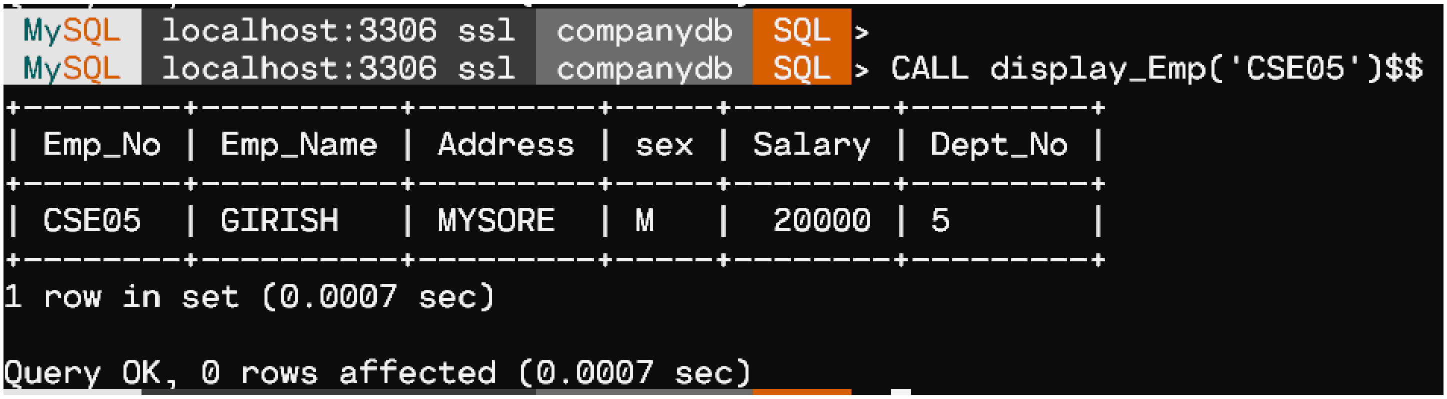
```
 MySQL  localhost:3306 ssl  companydb  SQL > SELECT * FROM Employee;
+--------+----------+-----------+-----+--------+---------+
| Emp_No | Emp_Name | Address   | sex | Salary | Dept_No |
+--------+----------+-----------+-----+--------+---------+
| ACC01  | AHANA    | MANGALORE | F   |  35000 | 1       |
| ACC02  | SANTHOSH | MANGALORE | M   |  30000 | 1       |
| CSE01  | JAMES    | BANGALORE | M   |  50000 | 5       |
| CSE02  | HEARN    | BANGALORE | M   |  70000 | 5       |
| CSE03  | EDWARD   | MYSORE    | M   |  50000 | 5       |
| CSE04  | PAVAN    | MANGALORE | M   |  15000 | 5       |
| CSE05  | GIRISH   | MYSORE    | M   |  20000 | 5       |
| CSE06  | NEHA     | BANGALORE | F   |  80000 | 5       |
| ECE01  | PREETI   | BANGALORE | F   |  40000 | 3       |
| ISE01  | VEENA    | MYSORE    | M   |  10000 | 4       |
| IT01   | NAGESH   | BANGALORE | M   |  50000 | 2       |
+--------+----------+-----------+-----+--------+---------+
11 rows in set (0.0006 sec)
```

```
/*1.Write a procedure to display employee details given the Emp_No.*/

DELIMITER $$
CREATE PROCEDURE display_Emp(IN X VARCHAR(20))
    BEGIN
        SELECT * FROM Employee
        WHERE Emp_No=X;
    END$$
```



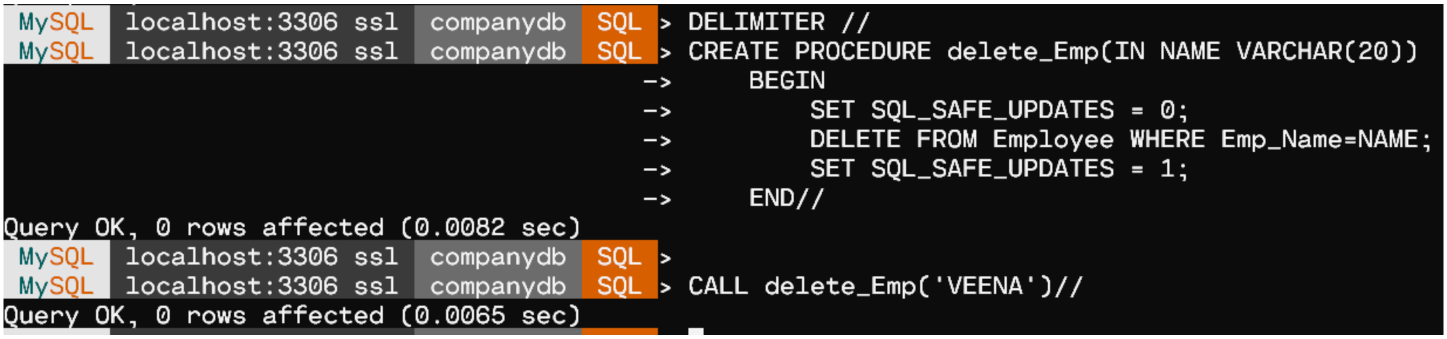
```
MySQL  localhost:3306 ssl  companydb  SQL > DELIMITER $$
MySQL  localhost:3306 ssl  companydb  SQL > CREATE PROCEDURE display_Emp(IN X VARCHAR(20))
                                         ->     BEGIN
                                         ->         SELECT * FROM Employee
                                         ->         WHERE Emp_No=X;
                                         ->     END$$
Query OK, 0 rows affected (0.0136 sec)
```

```
CALL display_Emp('CSE05')$$
```



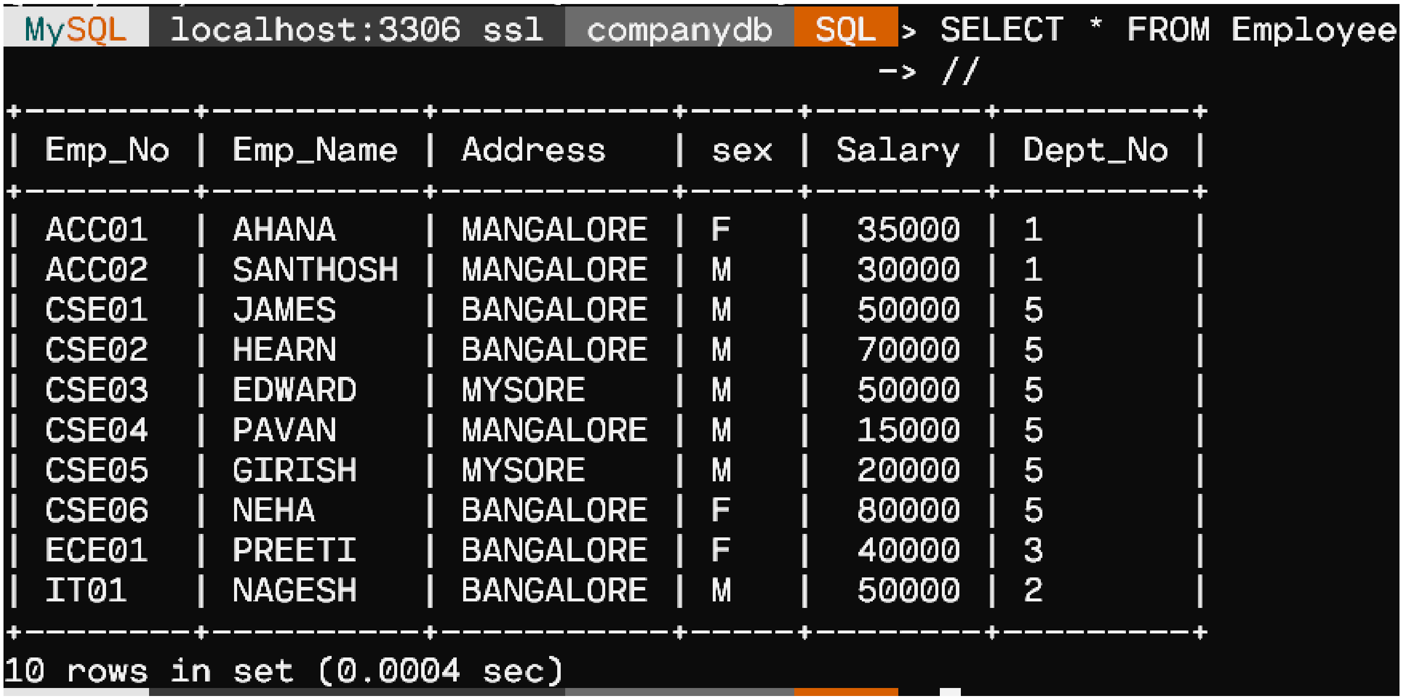
```
MySQL  localhost:3306 ssl  companydb  SQL >
MySQL  localhost:3306 ssl  companydb  SQL > CALL display_Emp('CSE05')$$
+--------+----------+---------+-----+--------+---------+
| Emp_No | Emp_Name | Address | sex | Salary | Dept_No |
+--------+----------+---------+-----+--------+---------+
| CSE05  | GIRISH   | MYSORE  | M   |  20000 | 5       |
+--------+----------+---------+-----+--------+---------+
1 row in set (0.0007 sec)

Query OK, 0 rows affected (0.0007 sec)
```

```
/*2.Write a procedure to delete an employee record given the Emp_Name
.*/

DELIMITER //
CREATE PROCEDURE delete_Emp(IN NAME VARCHAR(20))
    BEGIN
        SET SQL_SAFE_UPDATES = 0;
        DELETE FROM Employee WHERE Emp_Name=NAME;
        SET SQL_SAFE_UPDATES = 1;
    END//

CALL delete_Emp('VEENA')//
```



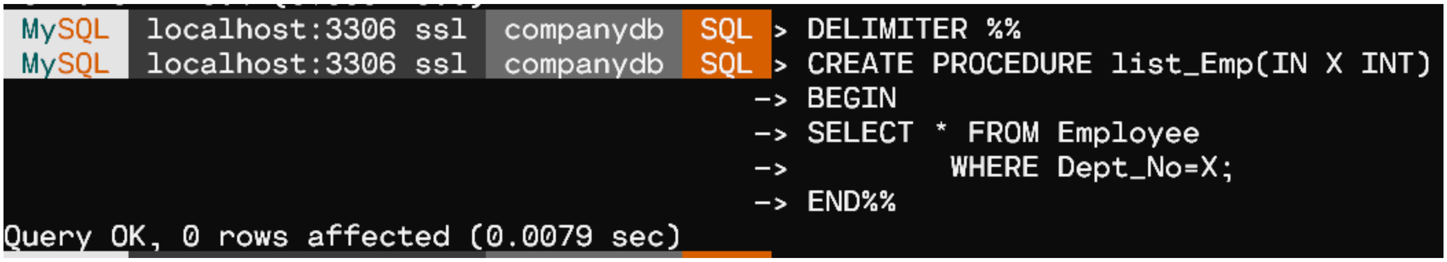
```
MySQL  localhost:3306 ssl  companydb  SQL > DELIMITER //
MySQL  localhost:3306 ssl  companydb  SQL > CREATE PROCEDURE delete_Emp(IN NAME VARCHAR(20))
                                        ->     BEGIN
                                        ->         SET SQL_SAFE_UPDATES = 0;
                                        ->         DELETE FROM Employee WHERE Emp_Name=NAME;
                                        ->         SET SQL_SAFE_UPDATES = 1;
                                        ->     END//
Query OK, 0 rows affected (0.0082 sec)
MySQL  localhost:3306 ssl  companydb  SQL >
MySQL  localhost:3306 ssl  companydb  SQL > CALL delete_Emp('VEENA')//
Query OK, 0 rows affected (0.0065 sec)
```

```
SELECT * FROM Employee//
```



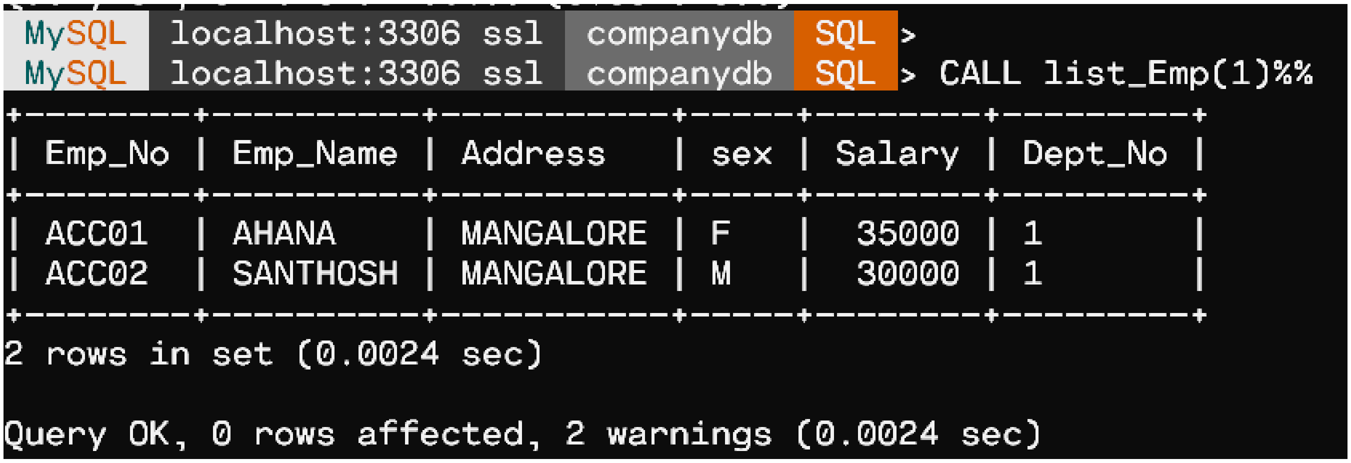
```
MySQL  localhost:3306 ssl  companydb  SQL > SELECT * FROM Employee
                                        -> //
+--------+----------+-----------+-----+--------+---------+
| Emp_No | Emp_Name | Address   | sex | Salary | Dept_No |
+--------+----------+-----------+-----+--------+---------+
| ACC01  | AHANA    | MANGALORE | F   |  35000 | 1       |
| ACC02  | SANTHOSH | MANGALORE | M   |  30000 | 1       |
| CSE01  | JAMES    | BANGALORE | M   |  50000 | 5       |
| CSE02  | HEARN    | BANGALORE | M   |  70000 | 5       |
| CSE03  | EDWARD   | MYSORE    | M   |  50000 | 5       |
| CSE04  | PAVAN    | MANGALORE | M   |  15000 | 5       |
| CSE05  | GIRISH   | MYSORE    | M   |  20000 | 5       |
| CSE06  | NEHA     | BANGALORE | F   |  80000 | 5       |
| ECE01  | PREETI   | BANGALORE | F   |  40000 | 3       |
| IT01   | NAGESH   | BANGALORE | M   |  50000 | 2       |
+--------+----------+-----------+-----+--------+---------+
10 rows in set (0.0004 sec)
```

```
/*3. Write a procedure to list all the employee names belonging to a
particular department given the Dept_No.*/

DELIMITER %%
CREATE PROCEDURE list_Emp(IN X INT)
    BEGIN
        SELECT * FROM Employee
        WHERE Dept_No=X;
    END%%
```



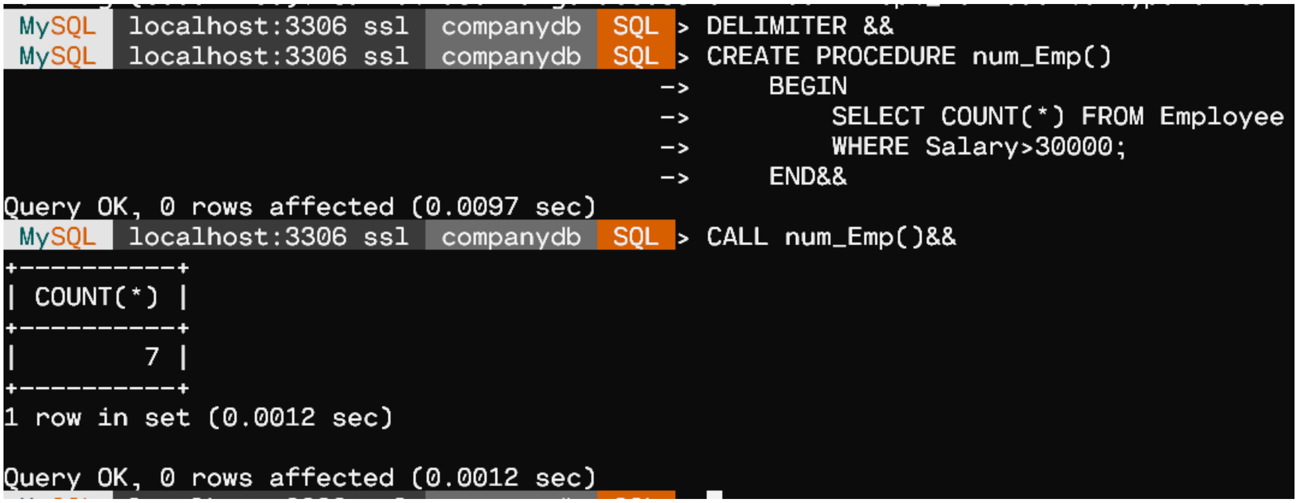
```
MySQL  localhost:3306 ssl  companydb  SQL > DELIMITER %%
MySQL  localhost:3306 ssl  companydb  SQL > CREATE PROCEDURE list_Emp(IN X INT)
                                         -> BEGIN
                                         -> SELECT * FROM Employee
                                         ->        WHERE Dept_No=X;
                                         -> END%%
Query OK, 0 rows affected (0.0079 sec)
```

```
CALL list_Emp(1)%%
```



```
MySQL  localhost:3306 ssl  companydb  SQL >
MySQL  localhost:3306 ssl  companydb  SQL > CALL list_Emp(1)%%
+--------+----------+-----------+-----+--------+---------+
| Emp_No | Emp_Name | Address   | sex | Salary | Dept_No |
+--------+----------+-----------+-----+--------+---------+
| ACC01  | AHANA    | MANGALORE | F   |  35000 | 1       |
| ACC02  | SANTHOSH | MANGALORE | M   |  30000 | 1       |
+--------+----------+-----------+-----+--------+---------+
2 rows in set (0.0024 sec)

Query OK, 0 rows affected, 2 warnings (0.0024 sec)
```

```
/*4. Write a procedure to display the number of employees whose salar
y is greater than 30K.*/
DELIMITER &&
CREATE PROCEDURE num_Emp()
    BEGIN
        SELECT COUNT(*) FROM Employee
        WHERE Salary>30000;
    END&&
CALL num_Emp()&&
```



```
MySQL  localhost:3306 ssl  companydb  SQL > DELIMITER &&
MySQL  localhost:3306 ssl  companydb  SQL > CREATE PROCEDURE num_Emp()
                                         ->     BEGIN
                                         ->         SELECT COUNT(*) FROM Employee
                                         ->         WHERE Salary>30000;
                                         ->     END&&
Query OK, 0 rows affected (0.0097 sec)
MySQL  localhost:3306 ssl  companydb  SQL > CALL num_Emp()&&
+----------+
| COUNT(*) |
+----------+
|        7 |
+----------+
1 row in set (0.0012 sec)

Query OK, 0 rows affected (0.0012 sec)
```

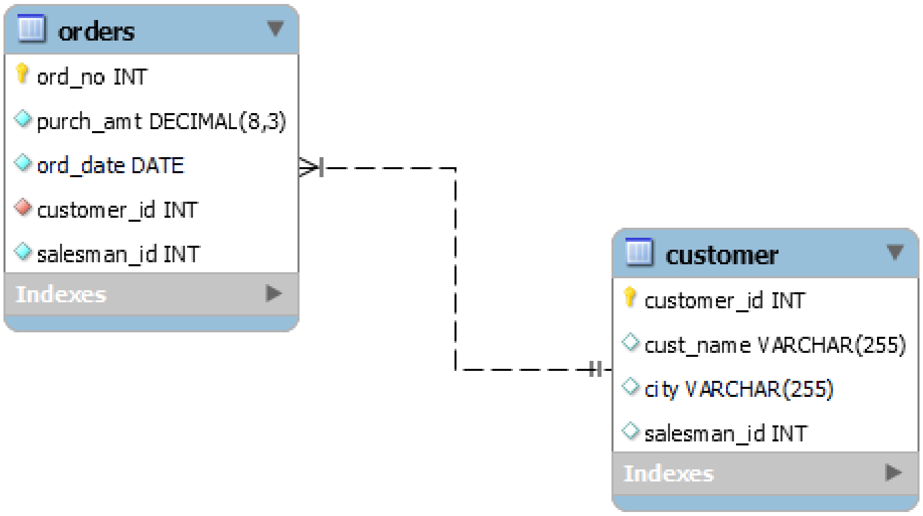
PROBLEM 2

Question 2

Orders (ord_no, purch_amt, ord_date, customer_id, salesman_id)
Customer (customer_id, cust_name, city, salesman_id)

Insert at least 8 tuples in both relation so that all queries result at least 1 tuple
1. Use SQL functions to find the highest purchase amount ordered by each customer with their ID and highest purchase amount.
2. Use SQL functions to find the number of customers in each city.
3. Write a user defined function to display customer details given the customer_id.
4. Write a user defined function to decrease the purch_amt of an order by 5% given the ord_no.

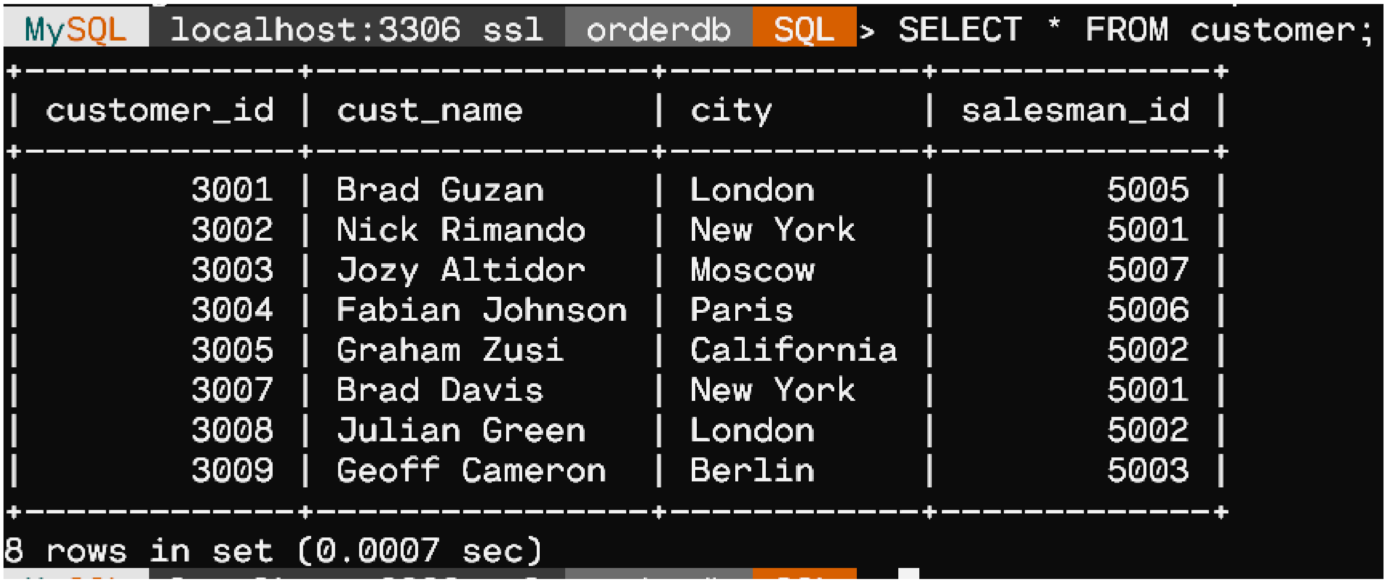
ER Diagram :



```
CREATE DATABASE orderDB;
USE orderDB;

CREATE TABLE customer(
    customer_id int NOT NULL,
    cust_name varchar(255),
    city varchar(255),
    salesman_id int,
    UNIQUE (customer_id),
    PRIMARY KEY (customer_id)
);
```

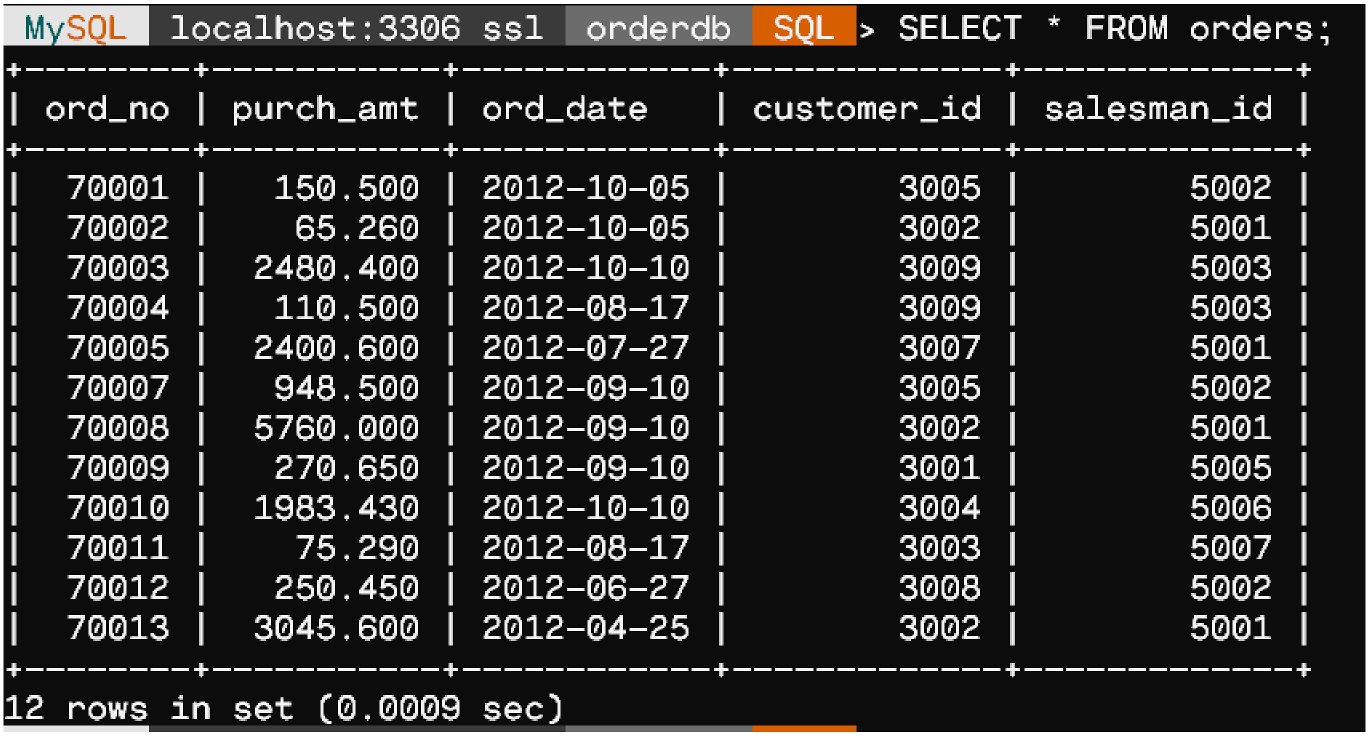
```
INSERT INTO customer
 VALUES
    (3002 , 'Nick Rimando' , 'New York' , 5001),
    (3007 , 'Brad Davis' , 'New York'  , 5001),
    (3005 , 'Graham Zusi' , 'California' , 5002),
    (3008 , 'Julian Green' , 'London' , 5002),
    (3004 , 'Fabian Johnson' , 'Paris' , 5006),
    (3009 , 'Geoff Cameron' , 'Berlin' , 5003),
    (3003 , 'Jozy Altidor' , 'Moscow' , 5007),
    (3001 , 'Brad Guzan' , 'London' , 5005);

CREATE TABLE orders(
    ord_no INT NOT NULL,
    purch_amt DECIMAL(8,3) NOT NULL,
    ord_date DATE NOT NULL,
    customer_id INT NOT NULL,
    salesman_id INT NOT NULL,
    PRIMARY KEY (ord_no),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
    );

INSERT INTO orders
 VALUES
    ('70001', '150.5', '2012-10-05', '3005', '5002'),
    ('70009', '270.65', '2012-09-10', '3001', '5005'),
    ('70002', '65.26', '2012-10-05', '3002', '5001'),
    ('70004', '110.5', '2012-08-17', '3009', '5003'),
    ('70007', '948.5', '2012-09-10', '3005', '5002'),
    ('70005', '2400.6', '2012-07-27', '3007', '5001'),
    ('70008', '5760', '2012-09-10', '3002', '5001'),
    ('70010', '1983.43', '2012-10-10', '3004', '5006'),
    ('70003', '2480.4', '2012-10-10', '3009', '5003'),
    ('70012', '250.45', '2012-06-27', '3008', '5002'),
    ('70011', '75.29', '2012-08-17', '3003', '5007'),
    ('70013', '3045.6', '2012-04-25', '3002', '5001');
```

```
        /*------------- Queries---------------- */

SET GLOBAL log_bin_trust_function_creators = 1;

SELECT * FROM customer;
```



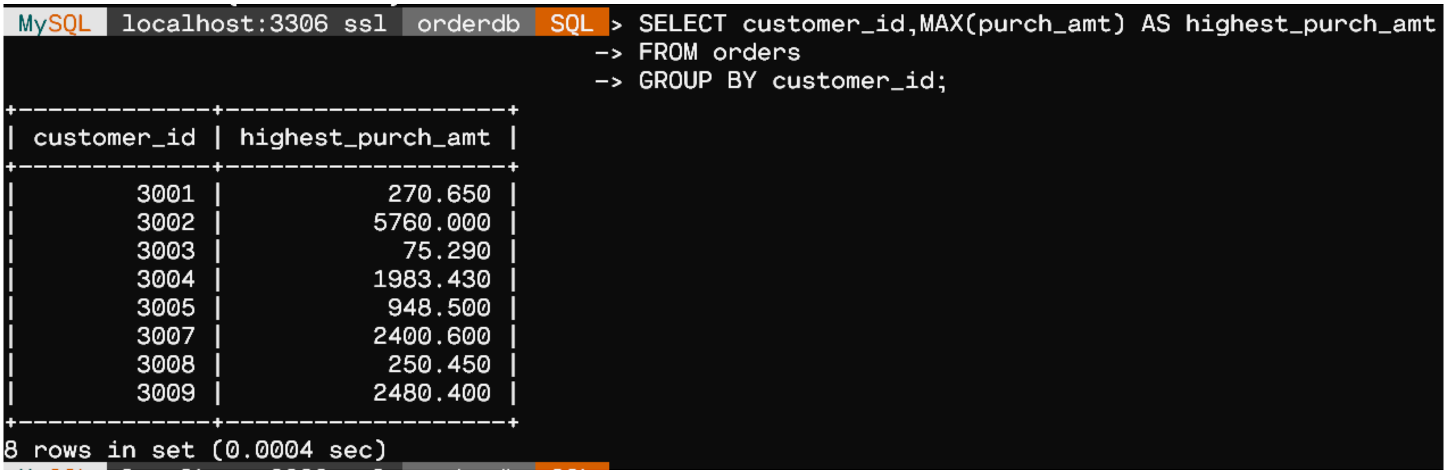
```
MySQL  localhost:3306 ssl  orderdb  SQL > SELECT * FROM customer;
+-------------+----------------+------------+-------------+
| customer_id | cust_name      | city       | salesman_id |
+-------------+----------------+------------+-------------+
|        3001 | Brad Guzan     | London     |        5005 |
|        3002 | Nick Rimando   | New York   |        5001 |
|        3003 | Jozy Altidor   | Moscow     |        5007 |
|        3004 | Fabian Johnson | Paris      |        5006 |
|        3005 | Graham Zusi    | California |        5002 |
|        3007 | Brad Davis     | New York   |        5001 |
|        3008 | Julian Green   | London     |        5002 |
|        3009 | Geoff Cameron  | Berlin     |        5003 |
+-------------+----------------+------------+-------------+
8 rows in set (0.0007 sec)
```

```
SELECT * FROM orders;
```



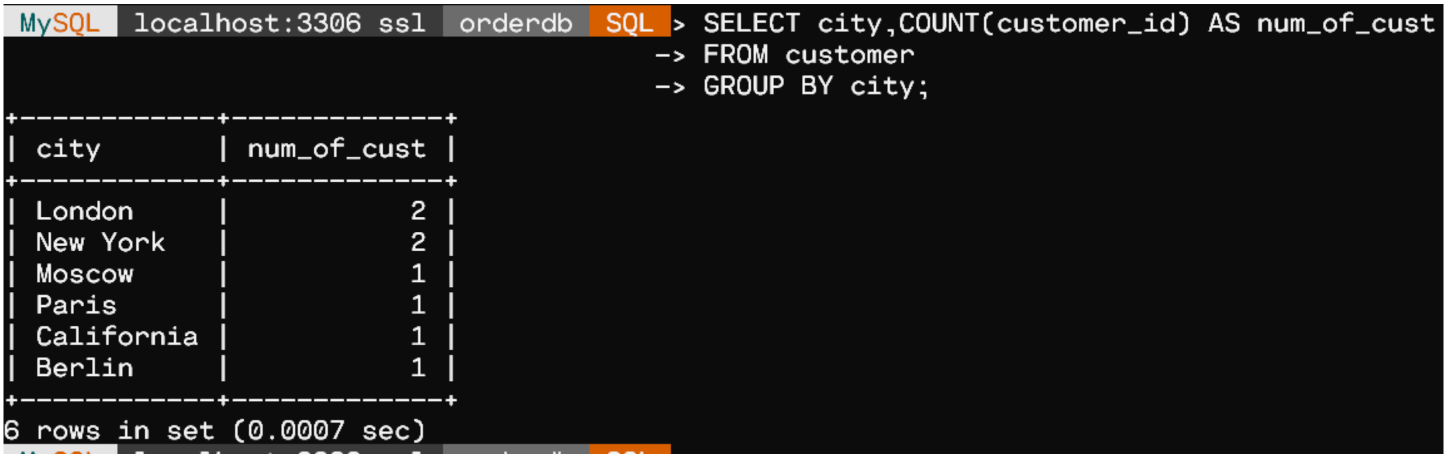
```
MySQL  localhost:3306 ssl  orderdb  SQL > SELECT * FROM orders;
+--------+-----------+------------+-------------+-------------+
| ord_no | purch_amt | ord_date   | customer_id | salesman_id |
+--------+-----------+------------+-------------+-------------+
|  70001 |   150.500 | 2012-10-05 |        3005 |        5002 |
|  70002 |    65.260 | 2012-10-05 |        3002 |        5001 |
|  70003 |  2480.400 | 2012-10-10 |        3009 |        5003 |
|  70004 |   110.500 | 2012-08-17 |        3009 |        5003 |
|  70005 |  2400.600 | 2012-07-27 |        3007 |        5001 |
|  70007 |   948.500 | 2012-09-10 |        3005 |        5002 |
|  70008 |  5760.000 | 2012-09-10 |        3002 |        5001 |
|  70009 |   270.650 | 2012-09-10 |        3001 |        5005 |
|  70010 |  1983.430 | 2012-10-10 |        3004 |        5006 |
|  70011 |    75.290 | 2012-08-17 |        3003 |        5007 |
|  70012 |   250.450 | 2012-06-27 |        3008 |        5002 |
|  70013 |  3045.600 | 2012-04-25 |        3002 |        5001 |
+--------+-----------+------------+-------------+-------------+
12 rows in set (0.0009 sec)
```

```
/*  1. Use SQL functions to find the highest purchase amount ordered
by
       each customer with their ID and highest purchase amount */

SELECT customer_id,MAX(purch_amt) AS highest_purch_amt
FROM orders
GROUP BY customer_id;
```



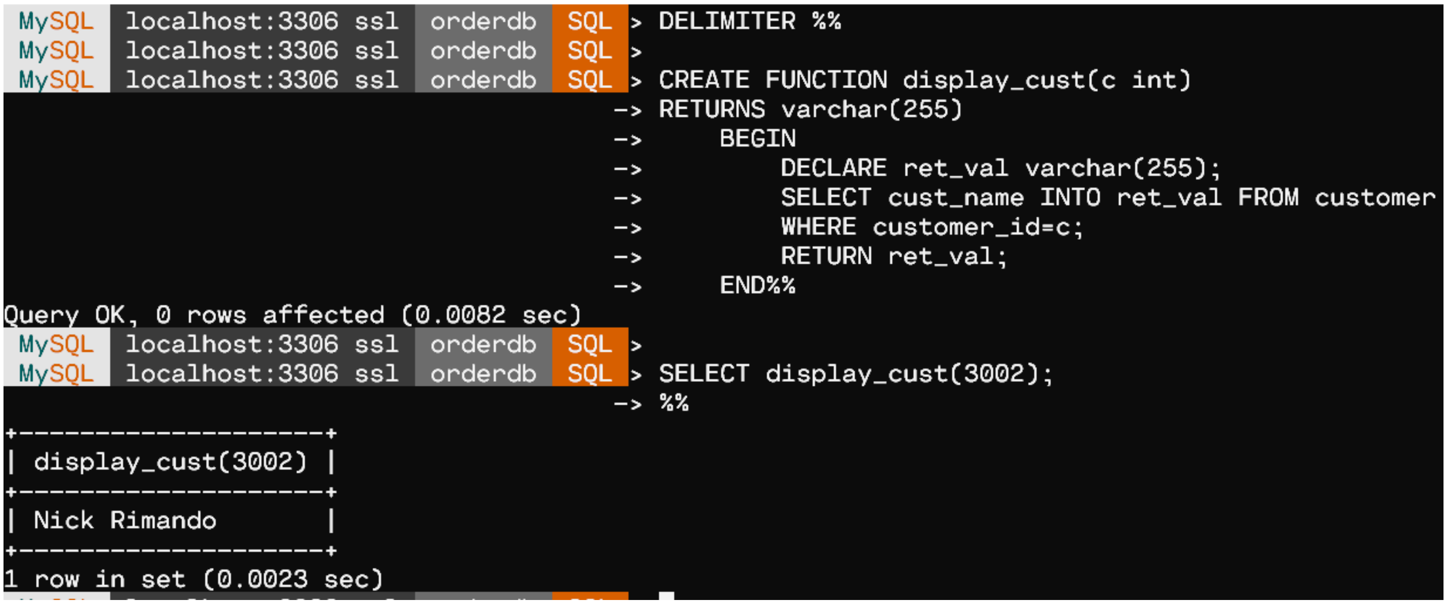
```
MySQL  localhost:3306 ssl  orderdb  SQL > SELECT customer_id,MAX(purch_amt) AS highest_purch_amt
                                       -> FROM orders
                                       -> GROUP BY customer_id;
+-------------+-------------------+
| customer_id | highest_purch_amt |
+-------------+-------------------+
|        3001 |           270.650 |
|        3002 |          5760.000 |
|        3003 |            75.290 |
|        3004 |          1983.430 |
|        3005 |           948.500 |
|        3007 |          2400.600 |
|        3008 |           250.450 |
|        3009 |          2480.400 |
+-------------+-------------------+
8 rows in set (0.0004 sec)
```

```
-  2. Use SQL functions to find the number of customers in each city.

SELECT city,COUNT(customer_id) AS num_of_cust
FROM customer
GROUP BY city;
```



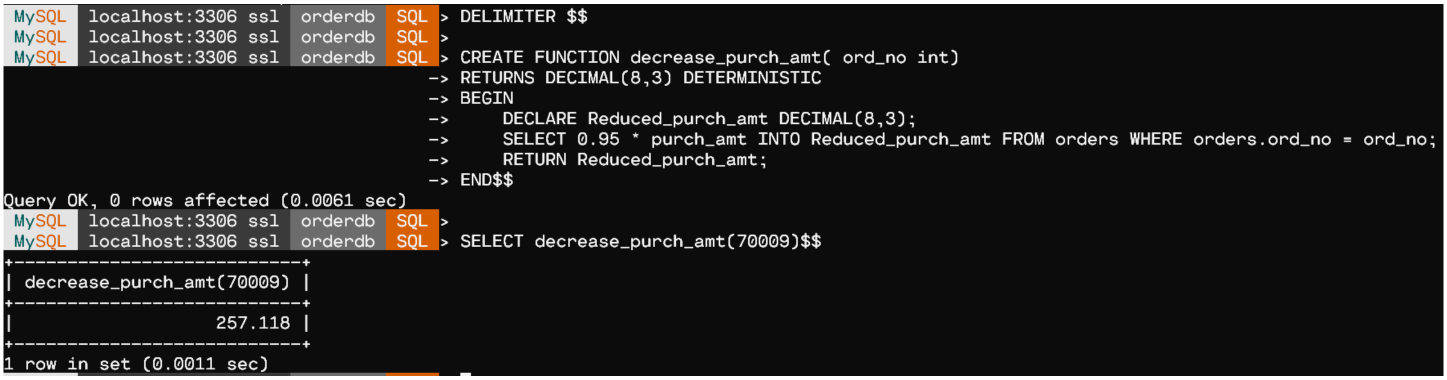
```
MySQL  localhost:3306 ssl  orderdb  SQL > SELECT city,COUNT(customer_id) AS num_of_cust
                                       -> FROM customer
                                       -> GROUP BY city;
+------------+-------------+
| city       | num_of_cust |
+------------+-------------+
| London     |           2 |
| New York   |           2 |
| Moscow     |           1 |
| Paris      |           1 |
| California |           1 |
| Berlin     |           1 |
+------------+-------------+
6 rows in set (0.0007 sec)
```

```
-  3. Write a user defined function to display customer details given
 the customer_id.

DELIMITER %%

CREATE FUNCTION display_cust(c int)
RETURNS varchar(255)
    BEGIN
        DECLARE ret_val varchar(255);
        SELECT cust_name INTO ret_val FROM customer
        WHERE customer_id=c;
        RETURN ret_val;
    END%%

SELECT display_cust(3002)%%
```



```
 MySQL  localhost:3306 ssl  orderdb  SQL > DELIMITER %%
 MySQL  localhost:3306 ssl  orderdb  SQL >
 MySQL  localhost:3306 ssl  orderdb  SQL > CREATE FUNCTION display_cust(c int)
                                        -> RETURNS varchar(255)
                                        ->     BEGIN
                                        ->         DECLARE ret_val varchar(255);
                                        ->         SELECT cust_name INTO ret_val FROM customer
                                        ->         WHERE customer_id=c;
                                        ->         RETURN ret_val;
                                        ->     END%%
Query OK, 0 rows affected (0.0082 sec)
 MySQL  localhost:3306 ssl  orderdb  SQL >
 MySQL  localhost:3306 ssl  orderdb  SQL > SELECT display_cust(3002);
                                        -> %%
+--------------------+
| display_cust(3002) |
+--------------------+
| Nick Rimando       |
+--------------------+
1 row in set (0.0023 sec)
```

```
-  4. Write a user defined function to decrease the purch_amt of an order by 5% given the ord_no.
DELIMITER $$

CREATE FUNCTION decrease_purch_amt( ord_no int)
RETURNS DECIMAL(8,3) DETERMINISTIC
BEGIN
    DECLARE Reduced_purch_amt DECIMAL(8,3);
    SELECT 0.95 * purch_amt INTO Reduced_purch_amt FROM orders WHERE
orders.ord_no = ord_no;
    RETURN Reduced_purch_amt;
END$$

SELECT decrease_purch_amt(70009)$$
```



```
 MySQL  localhost:3306 ssl  orderdb  SQL > DELIMITER $$
 MySQL  localhost:3306 ssl  orderdb  SQL >
 MySQL  localhost:3306 ssl  orderdb  SQL > CREATE FUNCTION decrease_purch_amt( ord_no int)
                                        -> RETURNS DECIMAL(8,3) DETERMINISTIC
                                        -> BEGIN
                                        ->     DECLARE Reduced_purch_amt DECIMAL(8,3);
                                        ->     SELECT 0.95 * purch_amt INTO Reduced_purch_amt FROM orders WHERE orders.ord_no = ord_no;
                                        ->     RETURN Reduced_purch_amt;
                                        -> END$$
Query OK, 0 rows affected (0.0061 sec)
 MySQL  localhost:3306 ssl  orderdb  SQL >
 MySQL  localhost:3306 ssl  orderdb  SQL > SELECT decrease_purch_amt(70009)$$
+---------------------------+
| decrease_purch_amt(70009) |
+---------------------------+
|                   257.118 |
+---------------------------+
1 row in set (0.0011 sec)
```