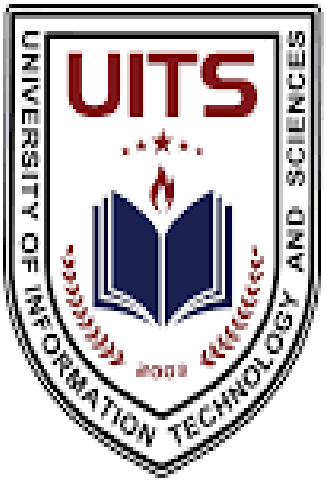


UITS

UNIVERSITY OF INFORMATION TECHNOLOGY AND SCIENCES

DBMS LAB REPORT

Course Code:CSE0612216S

Submitted To

Ms. Tania Akter Setu

Assistant Professor

Submitted By

Helal uddin Patwary

ID: 0432410005101086

Section: 3B2

Batch:55

Computer Science and Engineering

DBMS LAB Assessment Query with Results

Created Table Query:

```
CREATE TABLE salesman (
salesman_id INTEGER PRIMARY KEY,
name TEXT NOT NULL,
city TEXT NOT NULL,
commission decimal(3,2)
);
INSERT INTO salesman VALUES (5001, 'James Hoog', 'New York', 0.15);
INSERT INTO salesman VALUES (5002, 'Nail Knite', 'Paris', 0.13);
INSERT INTO salesman VALUES (5005, 'Pit Alex', 'London', 0.11);
INSERT INTO salesman VALUES (5006, 'Mc Lyon', 'Paris', 0.14);
INSERT INTO salesman VALUES (5007, 'Paul Adam', 'Rome', 0.13);
INSERT INTO salesman VALUES (5003, 'Lauson Hen', 'San Jose', 0.12);
```

```
CREATE TABLE customer (
customer_id INTEGER PRIMARY KEY,
cust_name TEXT NOT NULL,
city TEXT NOT NULL,
grade INTEGER,
salesman_id INTEGER
);
INSERT INTO customer VALUES (3002, 'Nick Rimando', 'New York', 100, 5001);
INSERT INTO customer VALUES (3007, 'Brad Davis', 'New York', 200, 5001);
INSERT INTO customer VALUES (3005, 'Graham Zusi', 'California', 200, 5002);
INSERT INTO customer VALUES (3008, 'Julian Green', 'London', 300, 5002);
INSERT INTO customer VALUES (3004, 'Fabian Johnson', 'Paris', 300, 5006);
INSERT INTO customer VALUES (3009, 'Geoff Cameron', 'Berlin', 100, 5003);
INSERT INTO customer VALUES (3003, 'Jozy Altidor', 'Moscow', 200, 5007);
INSERT INTO customer VALUES (3001, 'Brad Guzan', 'London', NULL, 5005);
```

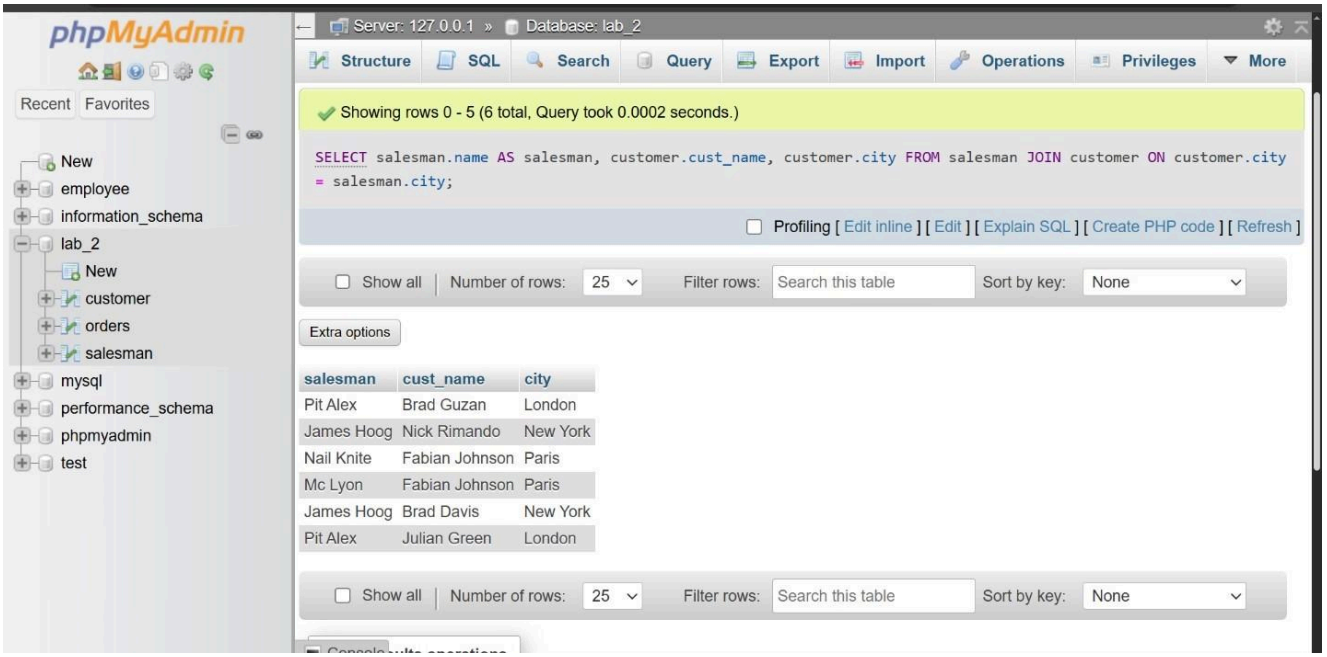
```
CREATE TABLE orders (
ord_no INTEGER PRIMARY KEY,
purch_amt INTEGER,
ord_date date,
customer_id INTEGER,

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

Questions Query with Results:

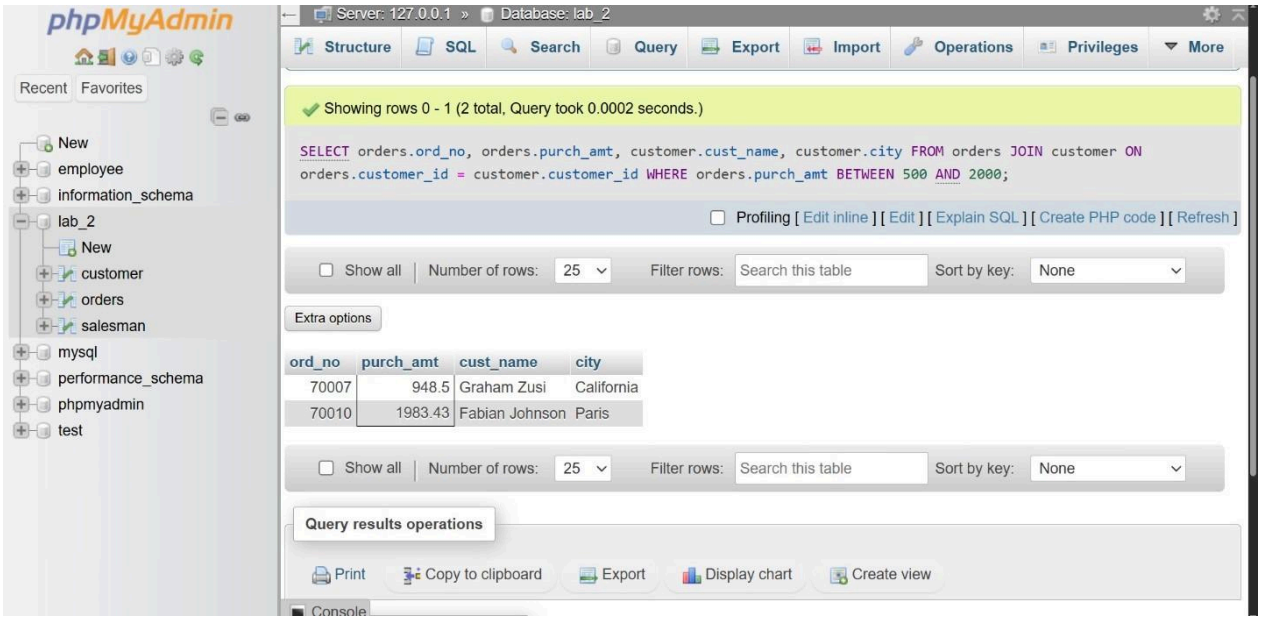
Q1. From the following tables write a SQL query to find the salesperson and customer who belong to the same city. Return Salesman, cust_name and city.

```
SELECT s.name AS Salesman, c.cust_name, c.city
FROM salesman s
JOIN customer c ON s.salesman_id = c.salesman_id
WHERE s.city = c.city;
```



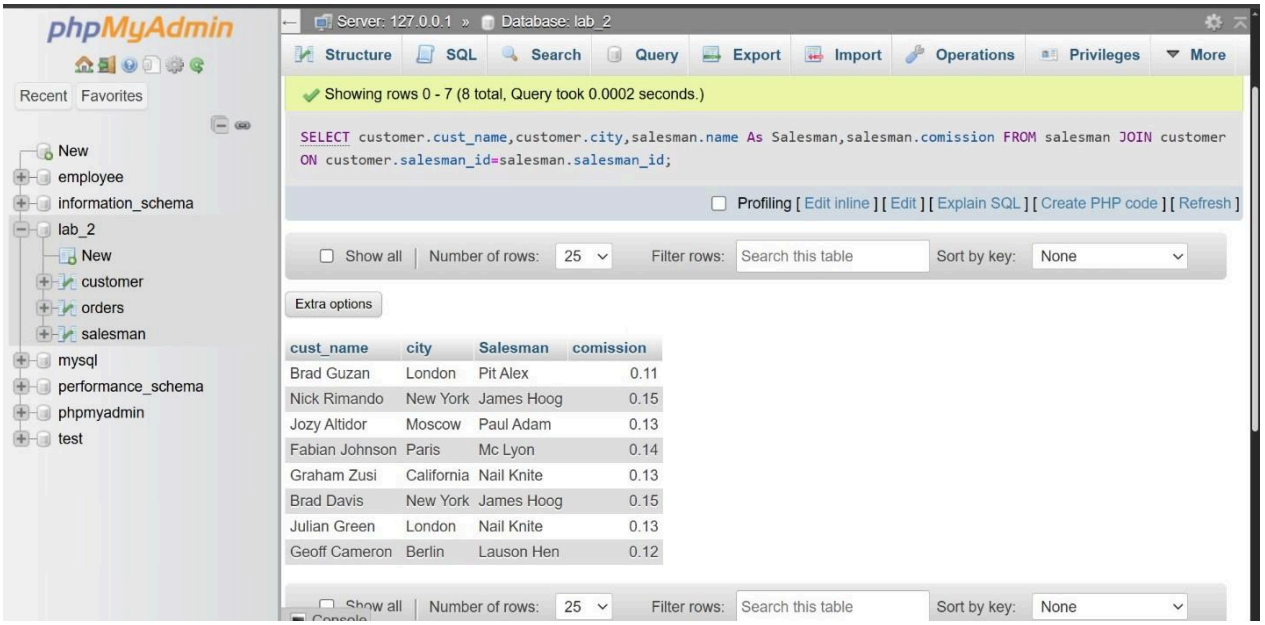
Q2. From the following tables write a SQL query to find those orders where order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

```
SELECT o.ord_no, o.purch_amt, c.cust_name, c.city
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
WHERE o.purch_amt BETWEEN 500 AND 2000;
```



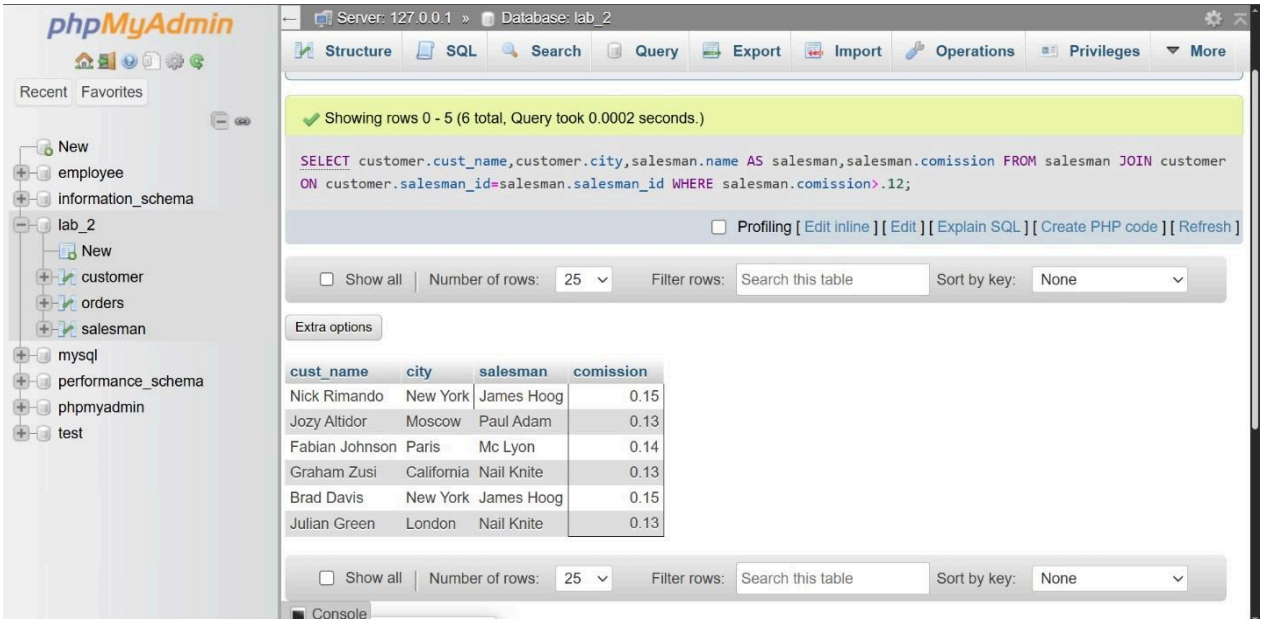
Q3. From the following tables write a SQL query to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission.

```
SELECT c.cust_name, c.city, s.name AS Salesman, s.commission
FROM customer c
JOIN salesman s ON c.salesman_id = s.salesman_id;
```



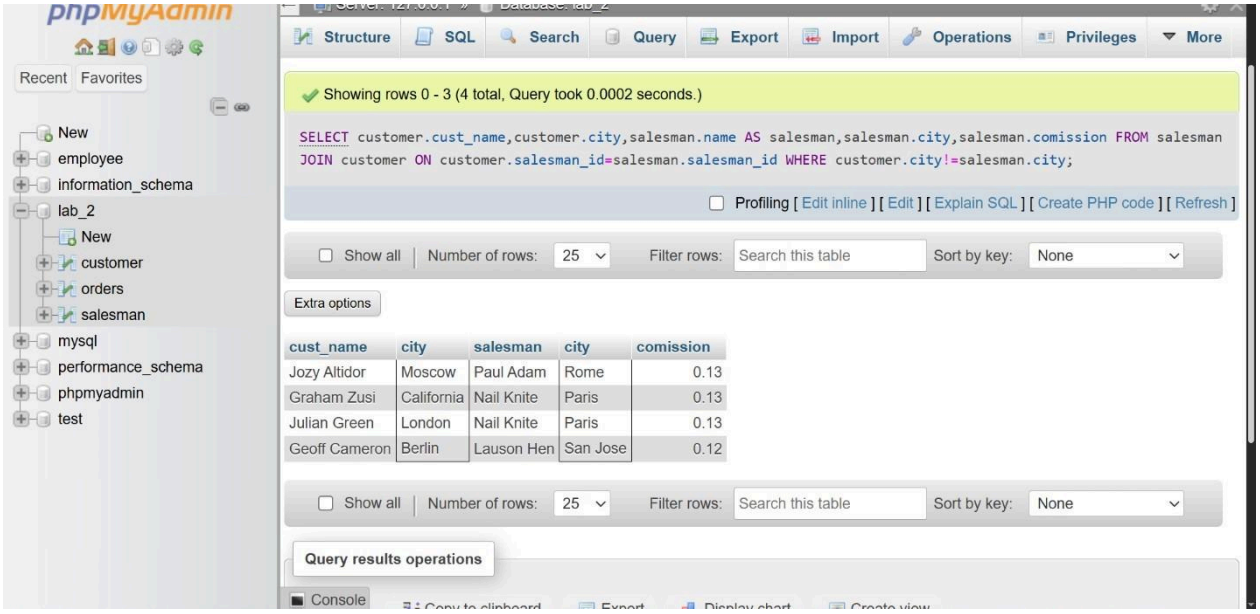
Q4. From the following tables write a SQL query to find those salespersons who received a commission from the company more than 12%. Return Customer Name, customer city, Salesman,commission

```
SELECT c.cust_name, c.city AS customer_city, s.name AS Salesman, s.commission
FROM customer c
JOIN salesman s ON c.salesman_id = s.salesman_id
WHERE s.commission > 0.12;
```



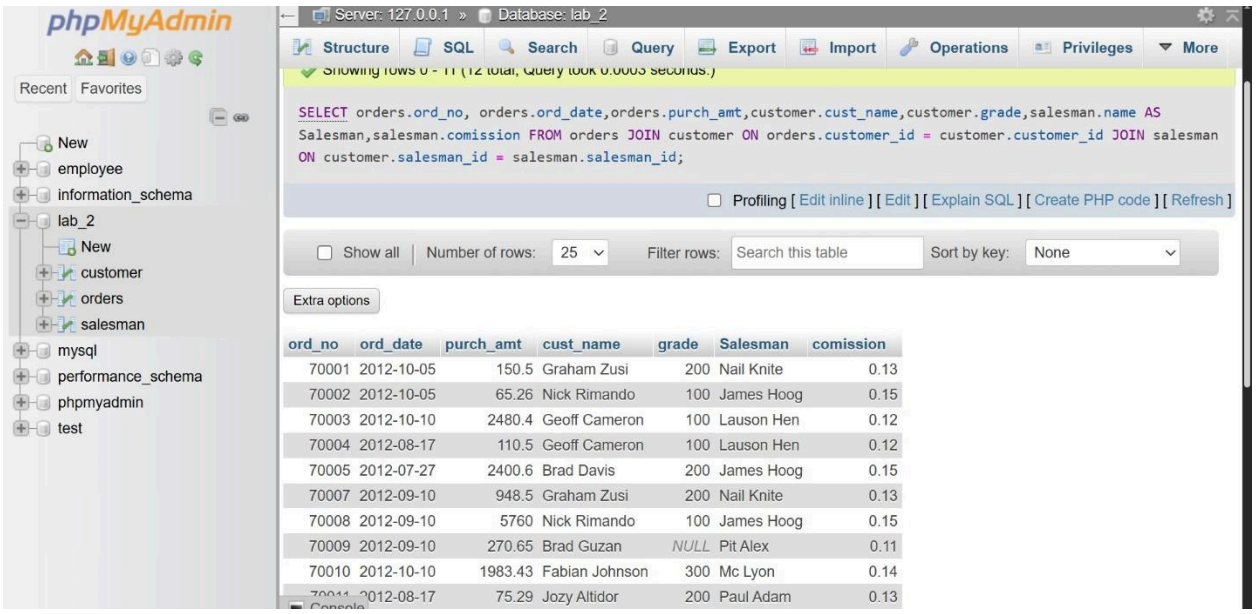
Q5. From the following tables write a SQL query to find those salespersons do not live in the same city where their customers live and received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, salesman city, commission.

SELECT c.cust_name, c.city AS customer_city, s.name AS Salesman, s.city AS salesman_city, s.commission
FROM customer c
JOIN salesman s ON c.salesman_id = s.salesman_id
WHERE s.city ! = c.city AND s.commission > 0.12;



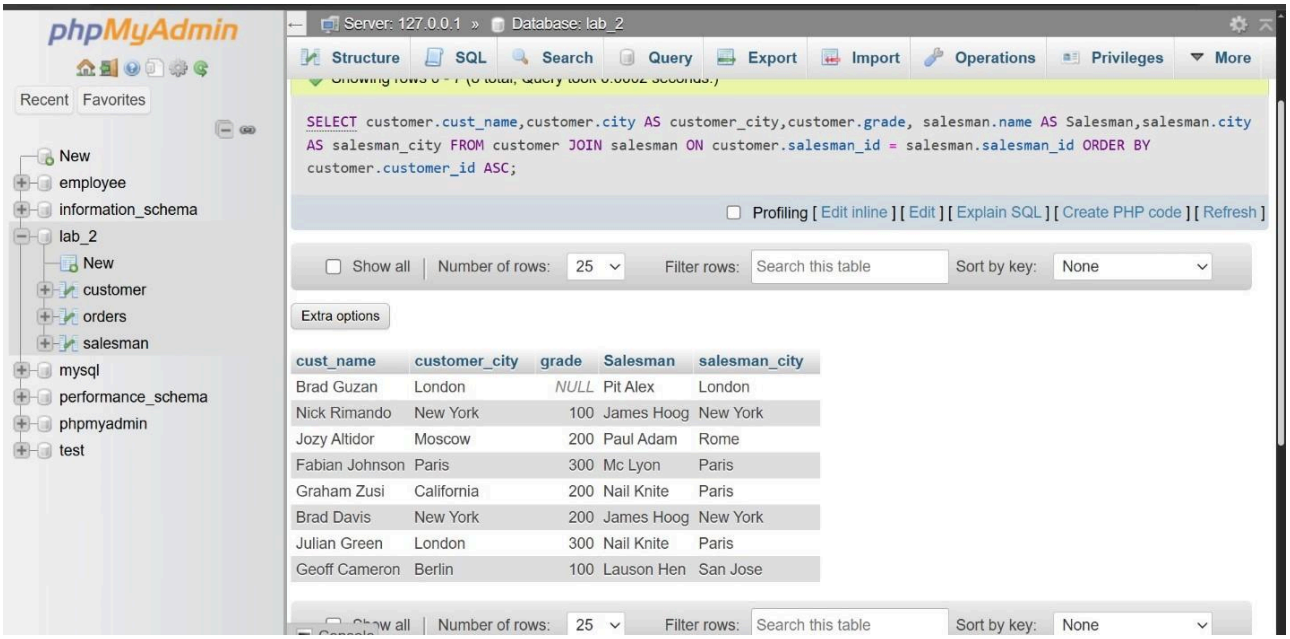
Q6. From the following tables write a SQL query to find the details of an order. Return ord_no,ord_date, purch_amt, Customer Name, grade, Salesman, commission.

```
SELECT o.order_no,o.ord_date, o.purchase_amt, c.cust_name,c.grade,s.name
 AS salesman, s.commission
FROM orders o
JOIN
   customer c ON o.customer_id = c.customer_id
JOIN
   salesman s ON o.salesman_id = s.salesman_id;
```



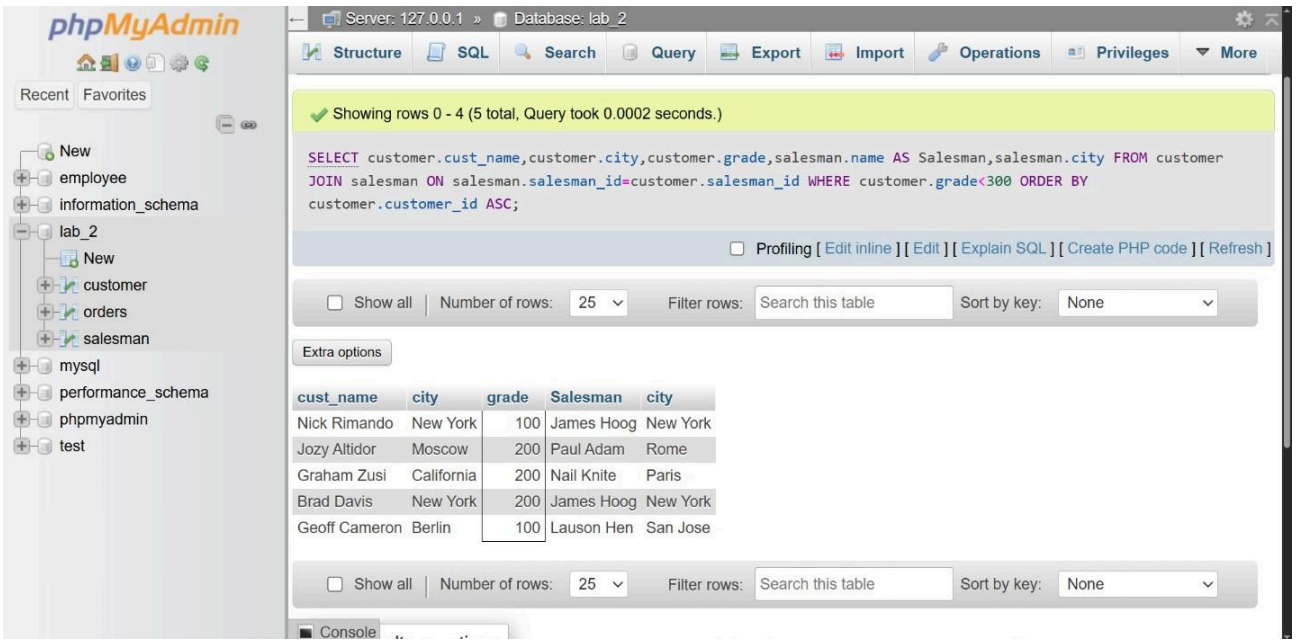
Q7. From the following tables write a SQL query to display the cust_name, customer city, grade,Salesman, salesman city. The result should be ordered by ascending on customer_id.

```
SELECT
   c.cust_name,
   c.city AS customer_city,
   c.grade,
   s.name AS salesman,
   s.city AS salesman_city
FROM
   customer c
JOIN
   salesman s ON c.salesman_id = s.salesman_id
ORDER BY
   c.customer_id ASC;
```



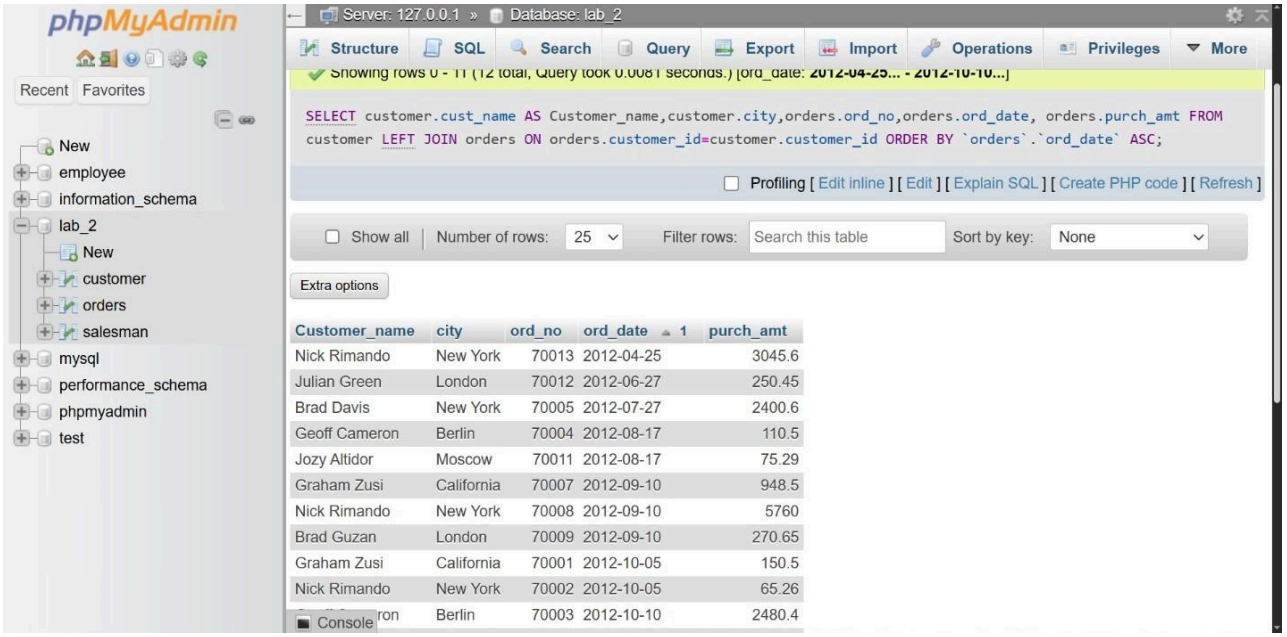
Q8. From the following tables write a SQL query to find those customers whose grade less than 300.Return cust_name, customer city, grade, Salesman, saleman city. The result should be ordered by ascending customer_id

```
SELECT
    c.cust_name,
    c.city AS customer_city,
    c.grade,
    s.name AS salesman,
    s.city AS salesman_city
FROM
    customer c
JOIN
    salesman s ON c.salesman_id = s.salesman_id
WHERE
    c.grade < 300
ORDER BY
    c.customer_id ASC;
```



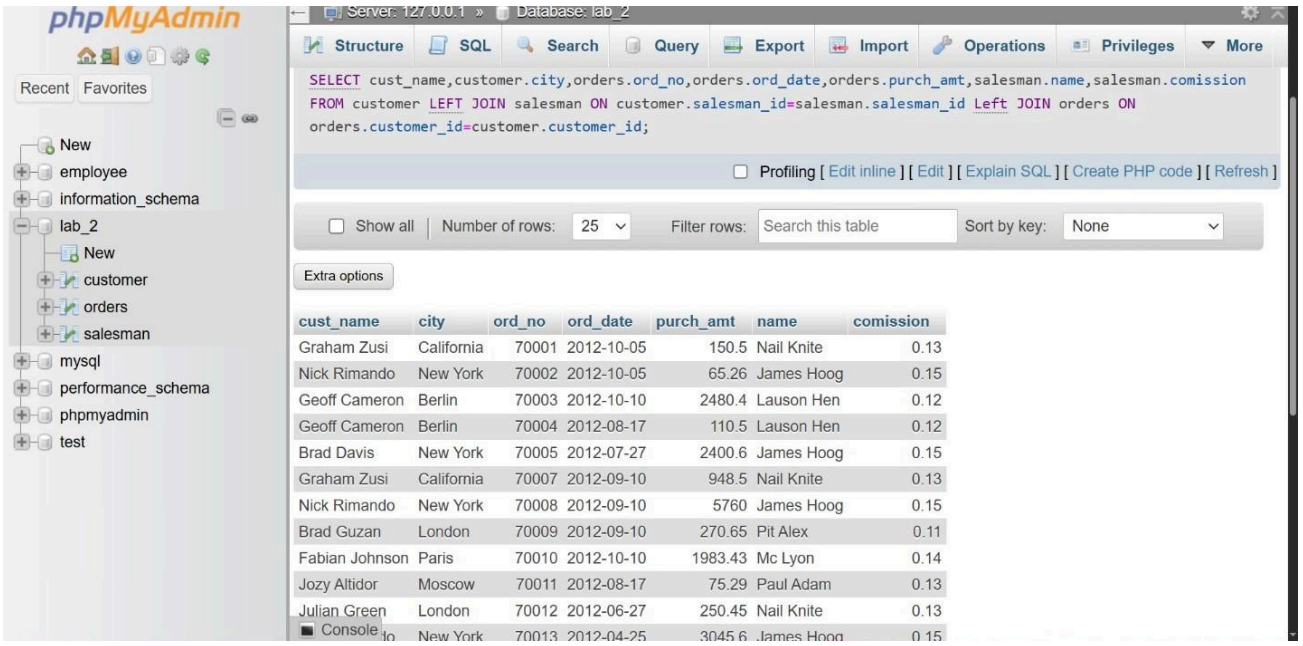
Q9. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    customer c
LEFT JOIN
    orders o ON c.customer_id = o.customer_id
ORDER BY
    o.ord_date ASC;
```



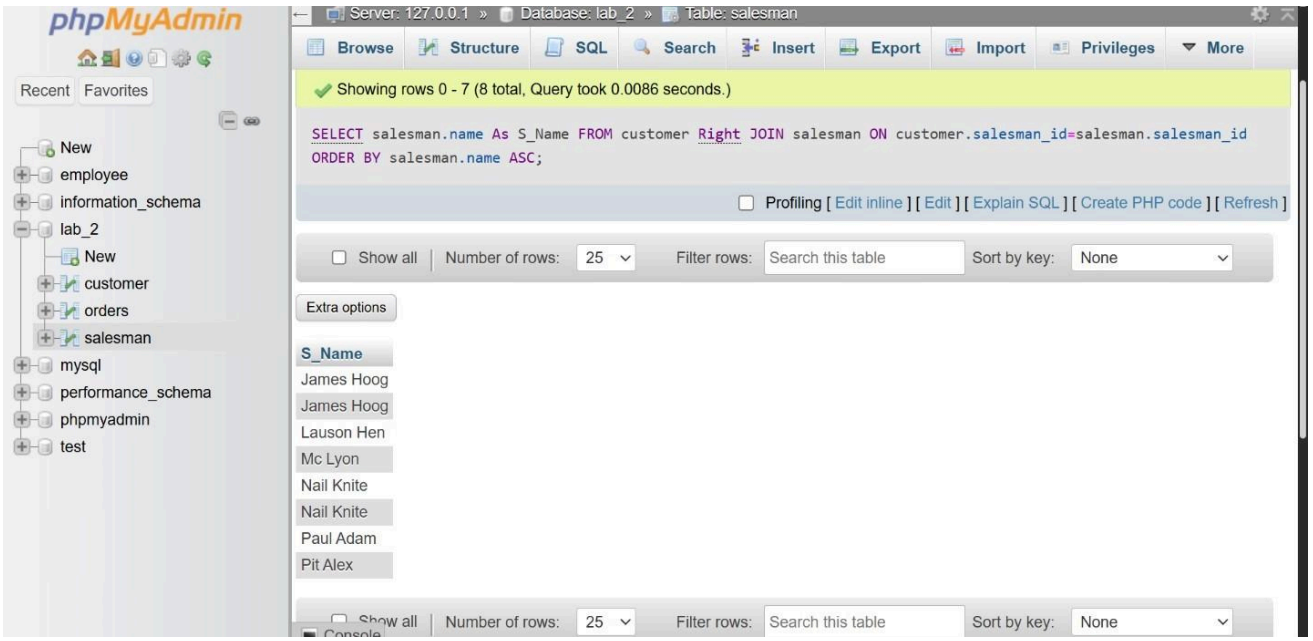
Q10. Write a SQL statement to make a report with customer name, city, order number, order date, order amount salesman name and commission to find that either any of the existing customers have placed no order or placed one or more orders by their salesman or by own.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt,
    s.name AS salesman,
    s.commission
FROM
    customer c
LEFT JOIN
    orders o ON c.customer_id = o.customer_id
LEFT JOIN
    salesman s ON c.salesman_id = s.salesman_id
ORDER BY
    o.ord_date ASC;
```



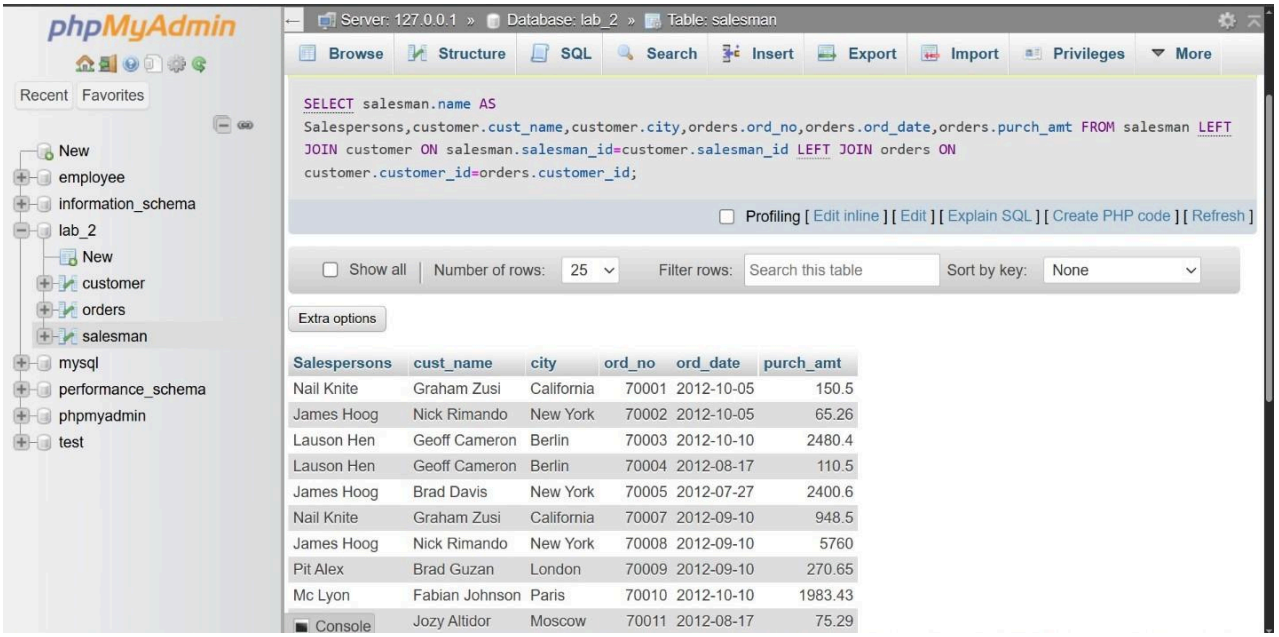
Q11. Write a SQL statement to make a list in ascending order for the salesmen who works either for one or more customer or not yet join under any of the customers.

```
SELECT
   s.salesman_id,
   s.name,
   s.city,
   s.commission
FROM
   salesman s
LEFT JOIN
   customer c ON s.salesman_id = c.salesman_id
ORDER BY
   s.salesman_id ASC;
```



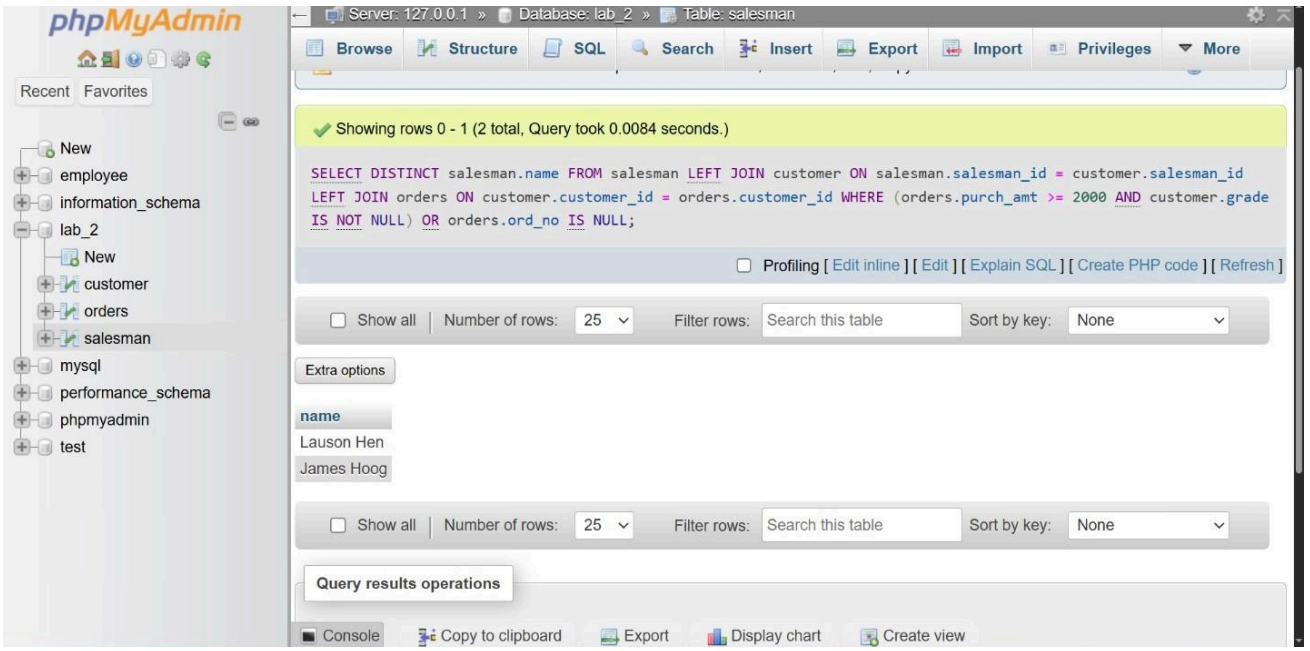
Q12. Write a SQL statement to make a list in ascending order for the salesmen who works either for one or more customer or not yet join under any of the customers.

```
SELECT
    s.name AS salesman,
    c.cust_name,
    c.city,
    c.grade,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    salesman s
LEFT JOIN
    customer c ON s.salesman_id = c.salesman_id
LEFT JOIN
    orders o ON c.customer_id = o.customer_id;
```



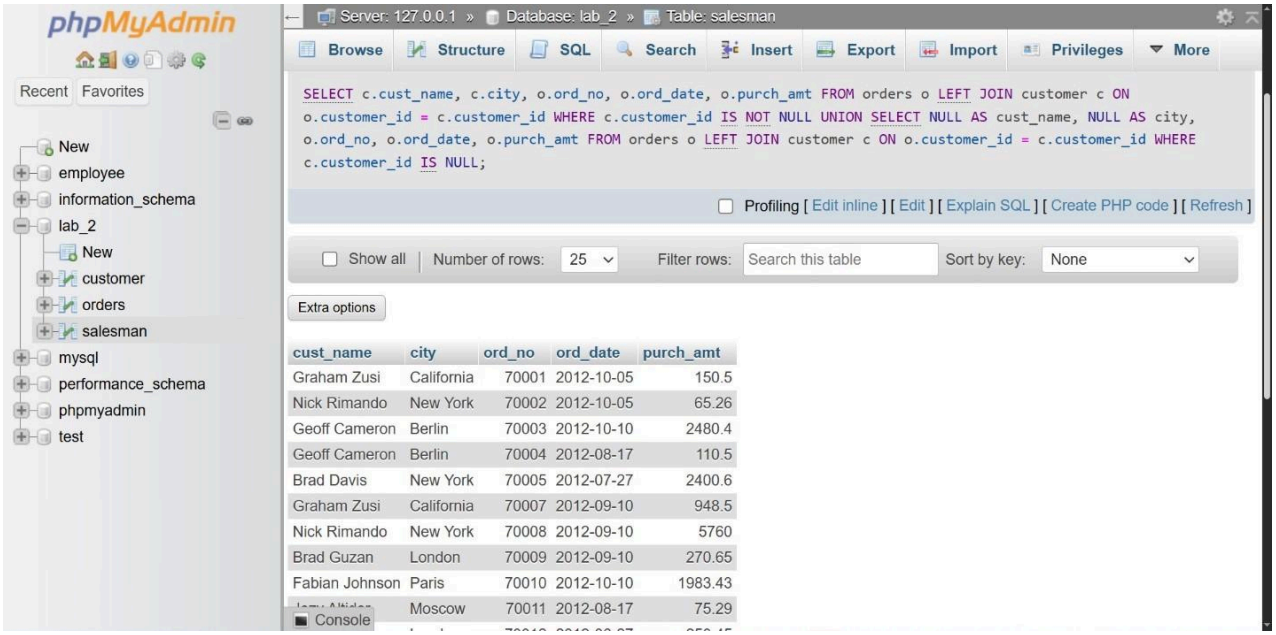
Q13. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

```
SELECT DISTINCT
   s.name AS salesman
FROM
   salesman s
LEFT JOIN
   customer c ON s.salesman_id = c.salesman_id
LEFT JOIN
   orders o ON c.customer_id = o.customer_id
WHERE
   (o.purchase_amt >= 2000 AND c.grade IS NOT NULL)
   OR c.customer_id IS NULL;
```



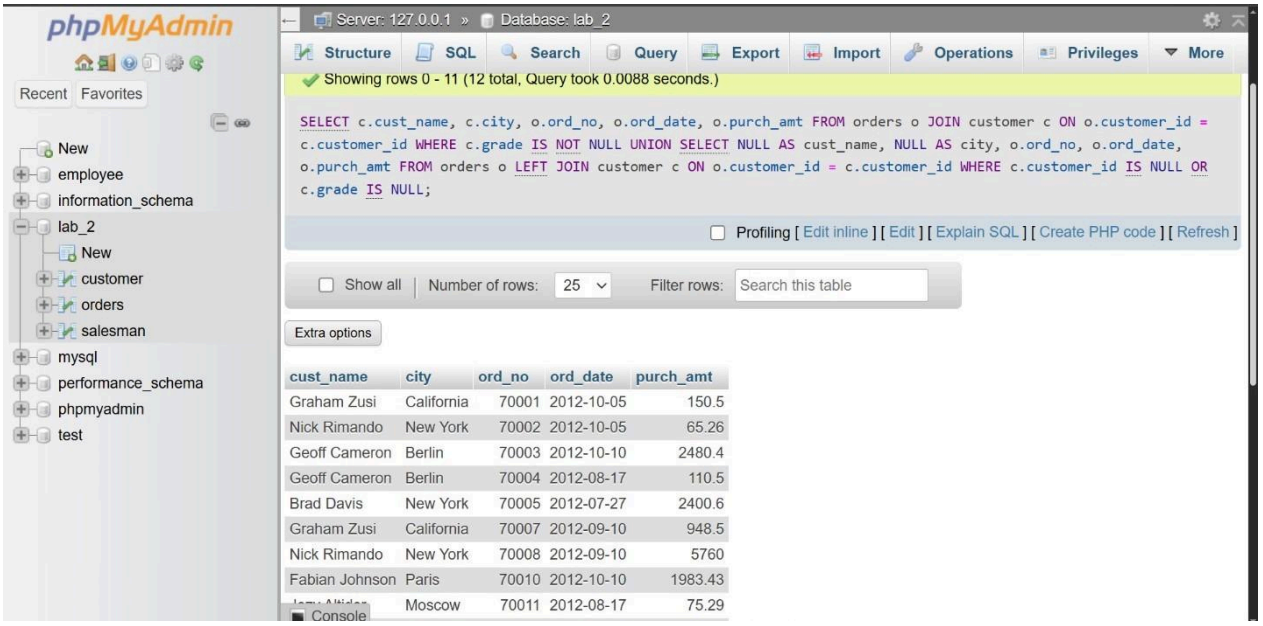
Q14. Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for those customers from the existing list who placed one or more orders or which order(s) have been placed by the customer who is not on the list.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    orders o
LEFT JOIN
    customer c ON o.customer_id = c.customer_id;
```



Q15. Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who is neither in the list nor have a grade.

```
SELECT
    c.cust_name,
    c.city,
    o.order_no,
    o.ord_date,
    o.purchase_amt
FROM
    orders o
LEFT JOIN
    customer c ON o.customer_id = c.customer_id
WHERE
    (c.grade IS NOT NULL)
    OR (c.customer_id IS NULL OR c.grade IS NULL);
```


phpMyAdmin
Recent
Favorites
New
employee
information_schema
lab_2
New
customer
orders
salesman
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: lab_2
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
More
Showing rows 0 - 11 (12 total, Query took 0.0088 seconds.)
SELECT c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt FROM orders o JOIN customer c ON o.customer_id = c.customer_id WHERE c.grade IS NOT NULL UNION SELECT NULL AS cust_name, NULL AS city, o.ord_no, o.ord_date, o.purch_amt FROM orders o LEFT JOIN customer c ON o.customer_id = c.customer_id WHERE c.customer_id IS NULL OR c.grade IS NULL;
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Extra options
cust_name city ord_no ord_date purch_amt
Graham Zusi California 70001 2012-10-05 150.5
Nick Rimando New York 70002 2012-10-05 65.26
Geoff Cameron Berlin 70003 2012-10-10 2480.4
Geoff Cameron Berlin 70004 2012-08-17 110.5
Brad Davis New York 70005 2012-07-27 2400.6
Graham Zusi California 70007 2012-09-10 948.5
Nick Rimando New York 70008 2012-09-10 5760
Fabian Johnson Paris 70010 2012-10-10 1983.43
Moscow 70011 2012-08-17 75.29
Console