Assignment 2 : Retrieve data using join with where clause

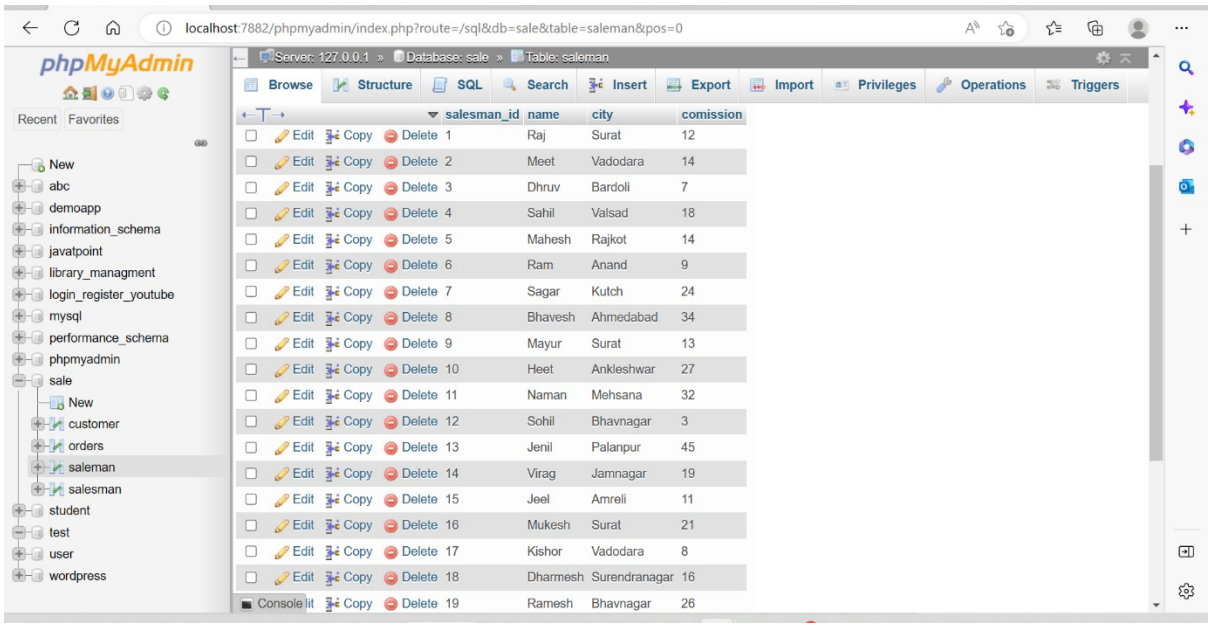
Sample table1: salesman

-salesman_id

-name

-city

-commission



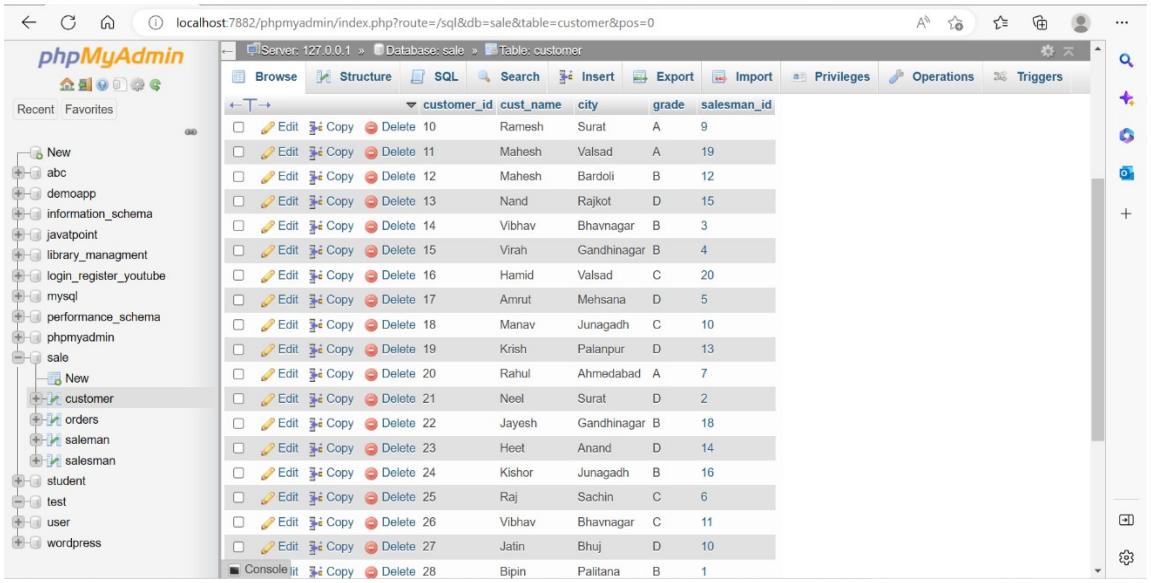
Sample table2: customer

-customer_id

-cust_name

-city -grade

-salesman_id



Sample table3: orders

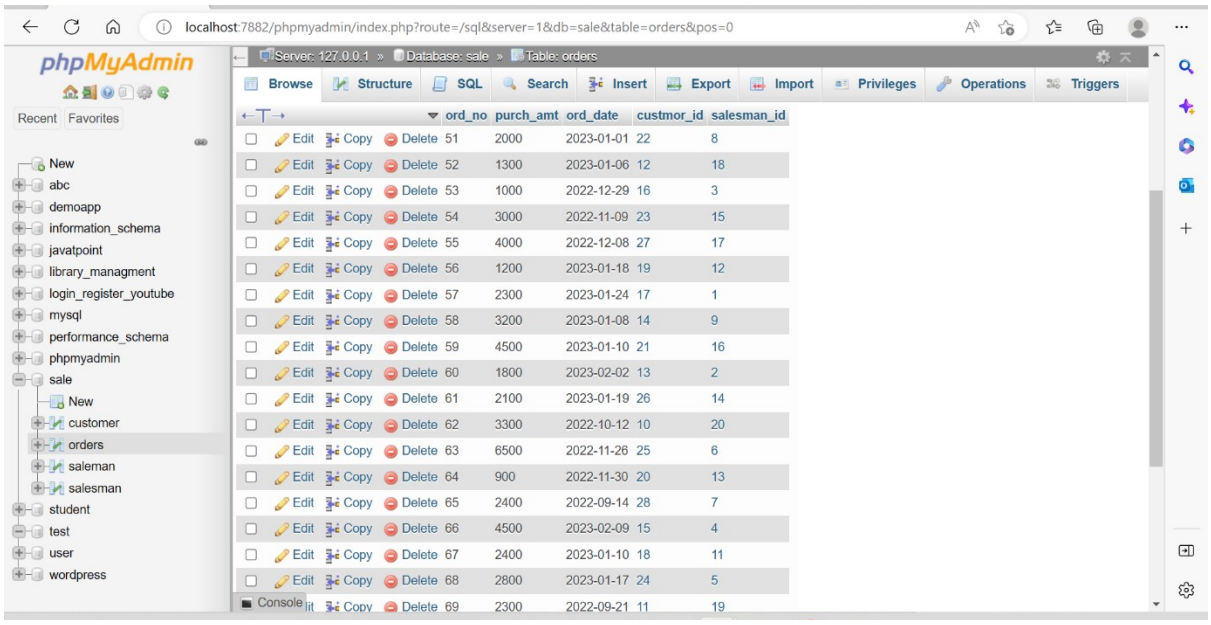
-ord_no

-purch_amt

-ord_date

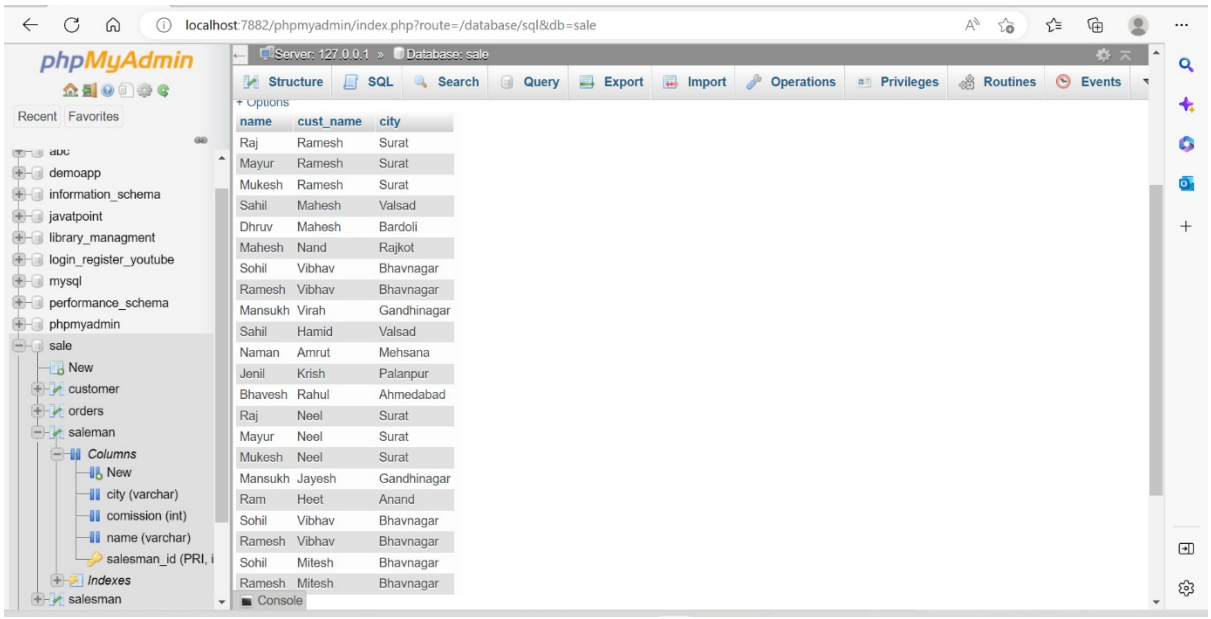
-customer_id

-salesman_id



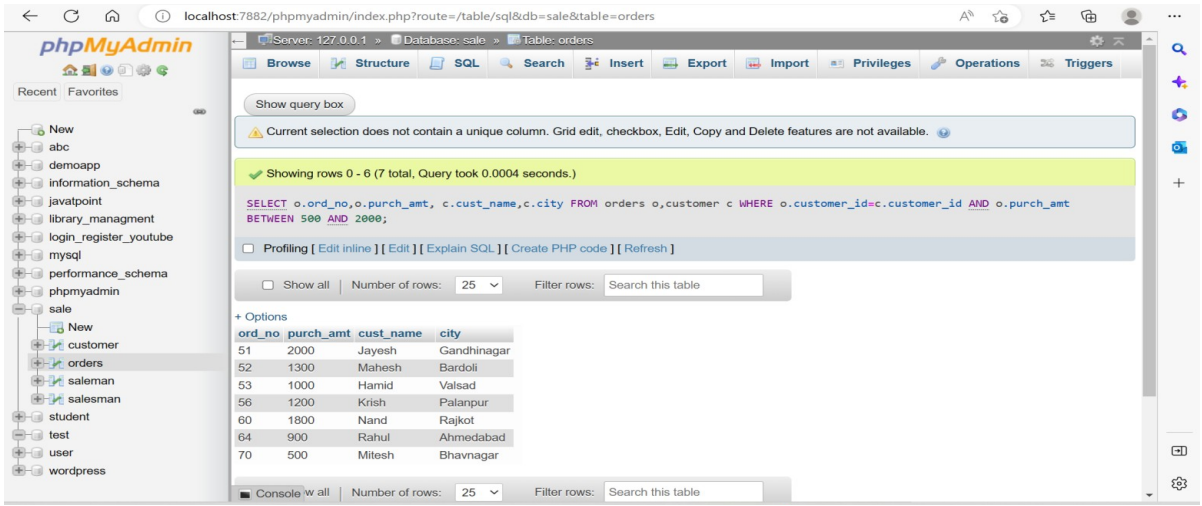
1. write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust_name and city

```
SELECT saleman.name,customer.cust_name, customer.city FROM saleman,customer
WHERE saleman.city=customer.city;
```



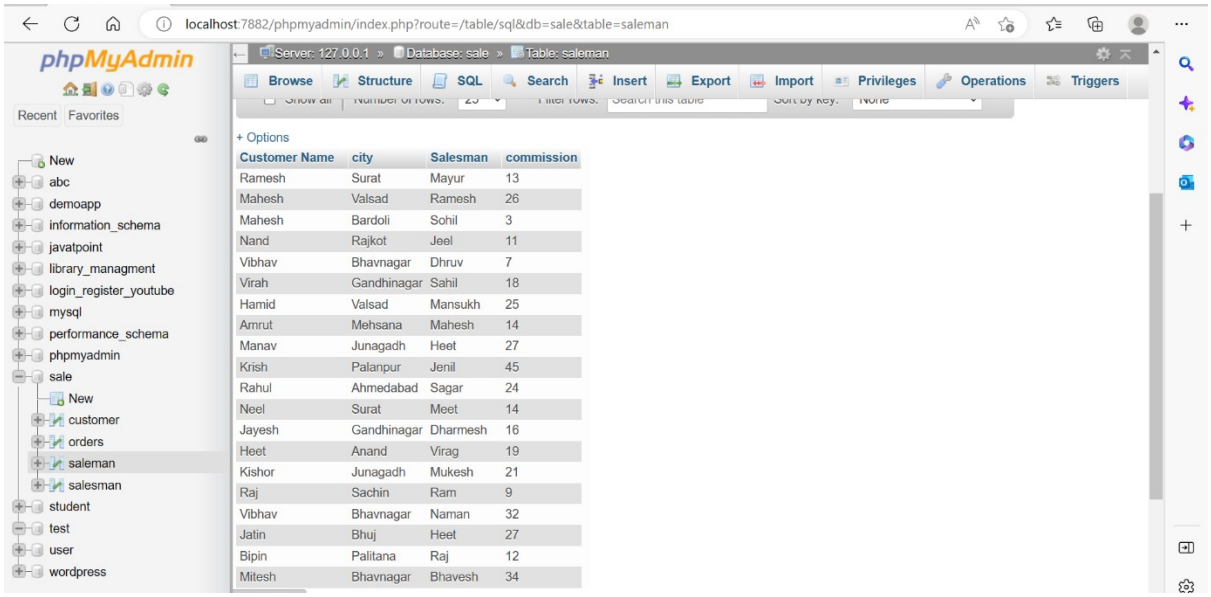
2. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city

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



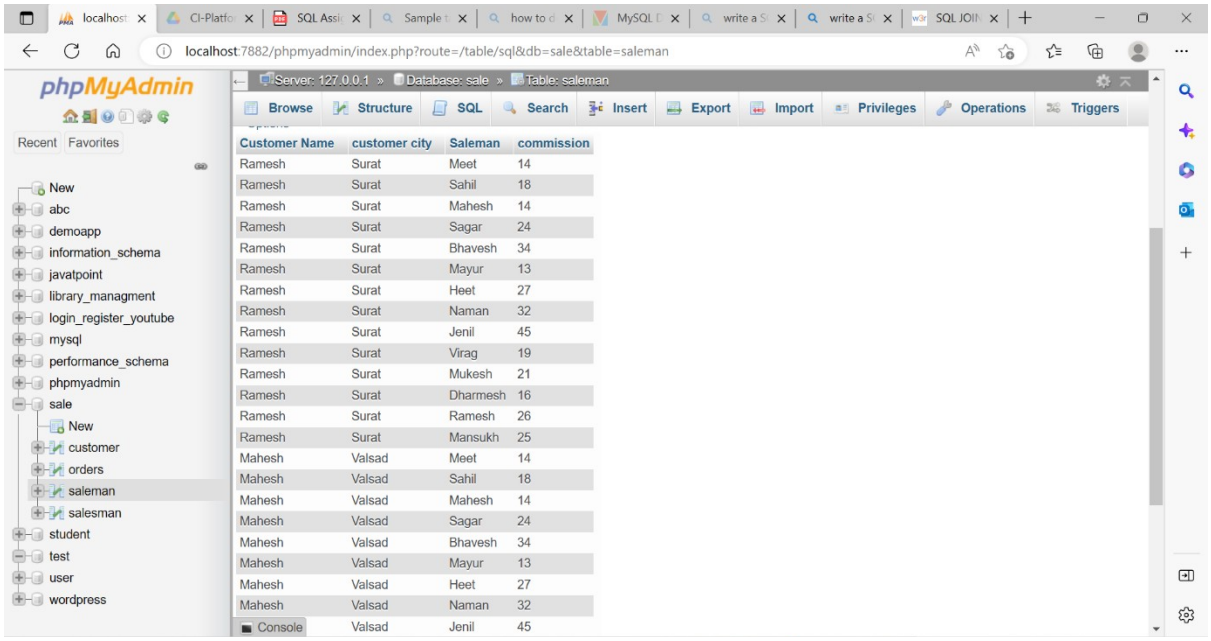
3. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission

```
SELECT c.cust_name AS "Customer Name", c.city, s.name AS "Salesman", s.commission FROM customer c INNER JOIN saleman s ON c.salesman_id=s.salesman_id;
```



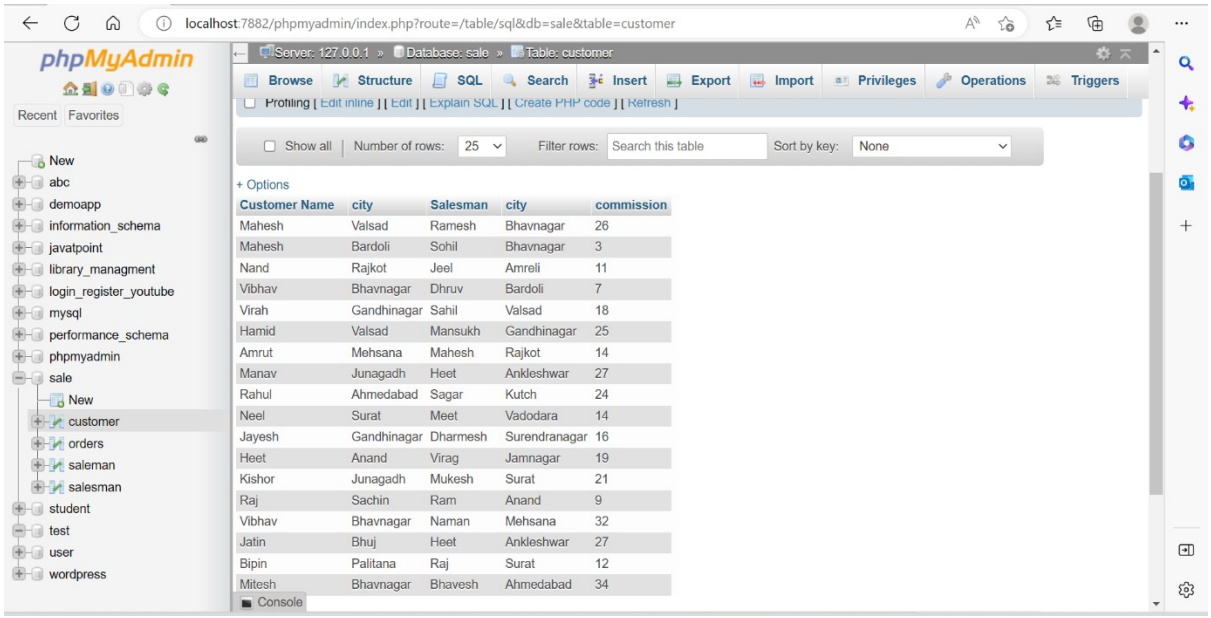
4. write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission.

```
select c.cust_name as 'Customer Name', c.city as 'customer city', s.name as 'Saleman', s.commission FROM customer c,saleman s WHERE s.commission > 12;
```



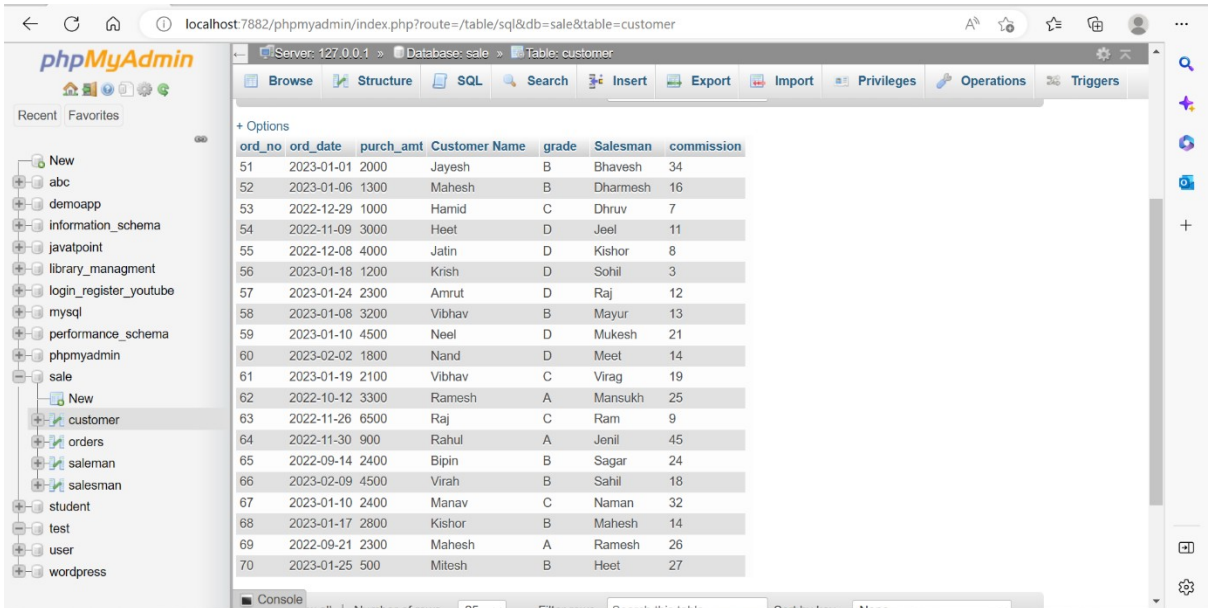
5. write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission

```
SELECT c.cust_name AS "Customer Name", c.city, s.name AS "Salesman", s.city,s.commission FROM customer c INNER JOIN saleman s ON c.salesman_id=s.salesman_id WHERE s.commission>.12 AND c.city<>s.city;
```



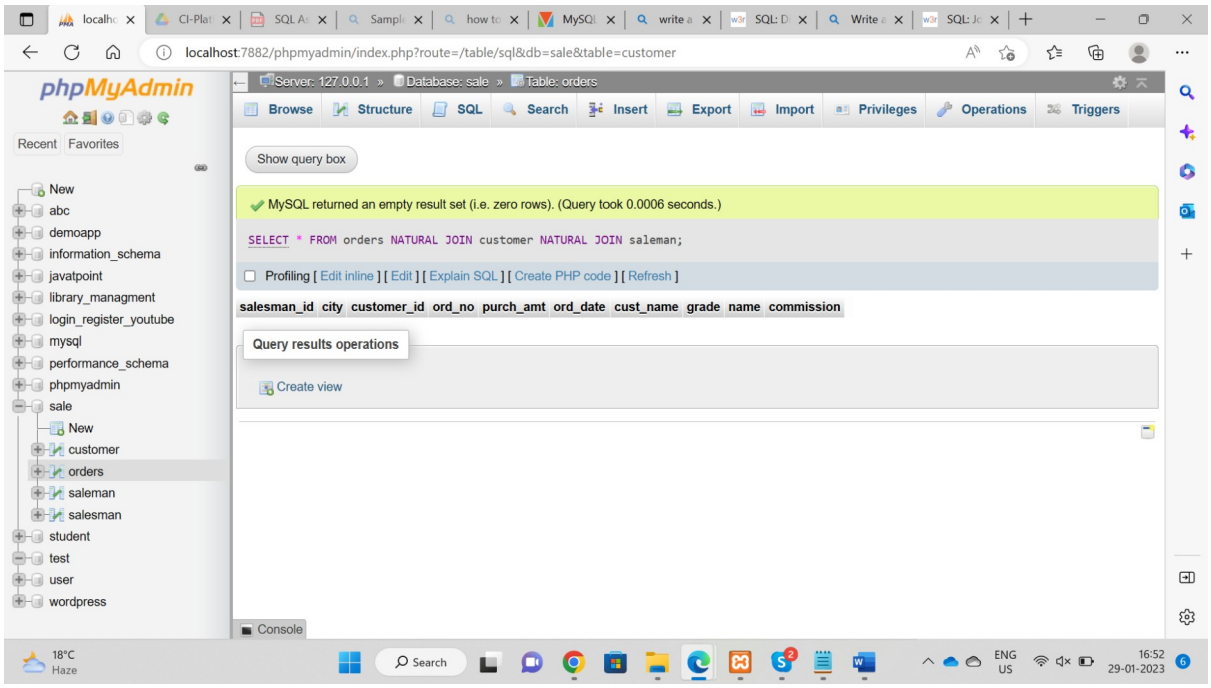
6. write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission

```
SELECT o.ord_no,o.ord_date,o.purch_amt, c.cust_name AS "Customer Name", c.grade, s.name AS "Salesman", s.commission FROM orders o INNER JOIN customer c ON o.customer_id=c.customer_id INNER JOIN saleman s ON o.salesman_id=s.salesman_id;
```



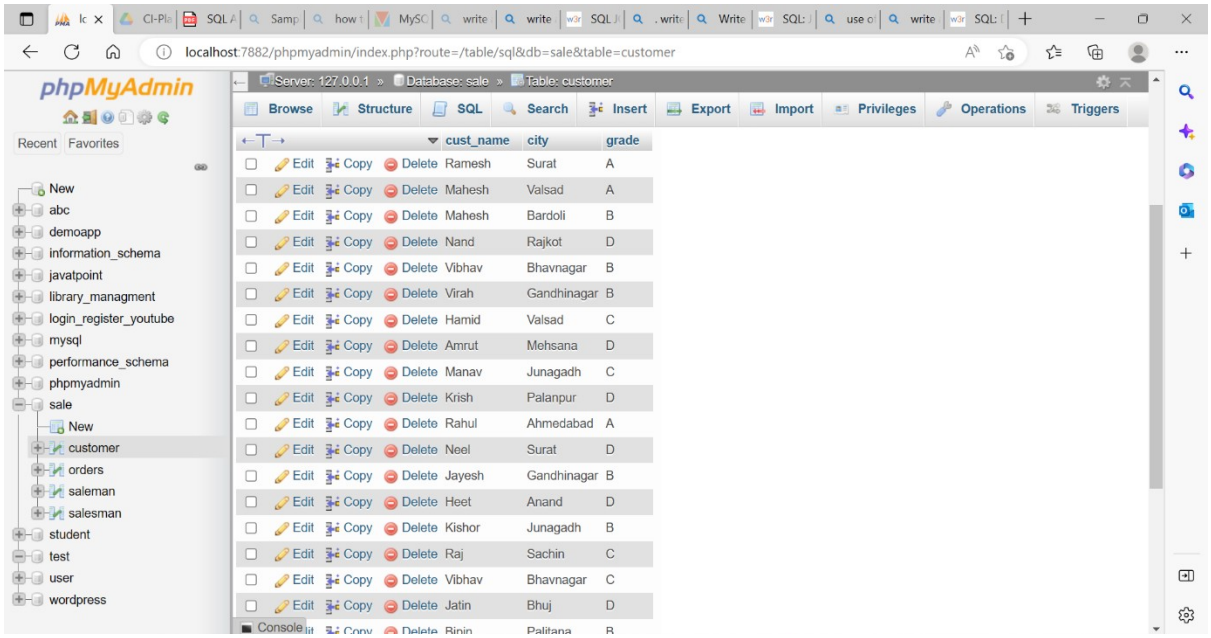
7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.

```
SELECT * FROM orders NATURAL JOIN customer NATURAL JOIN saleman;
```



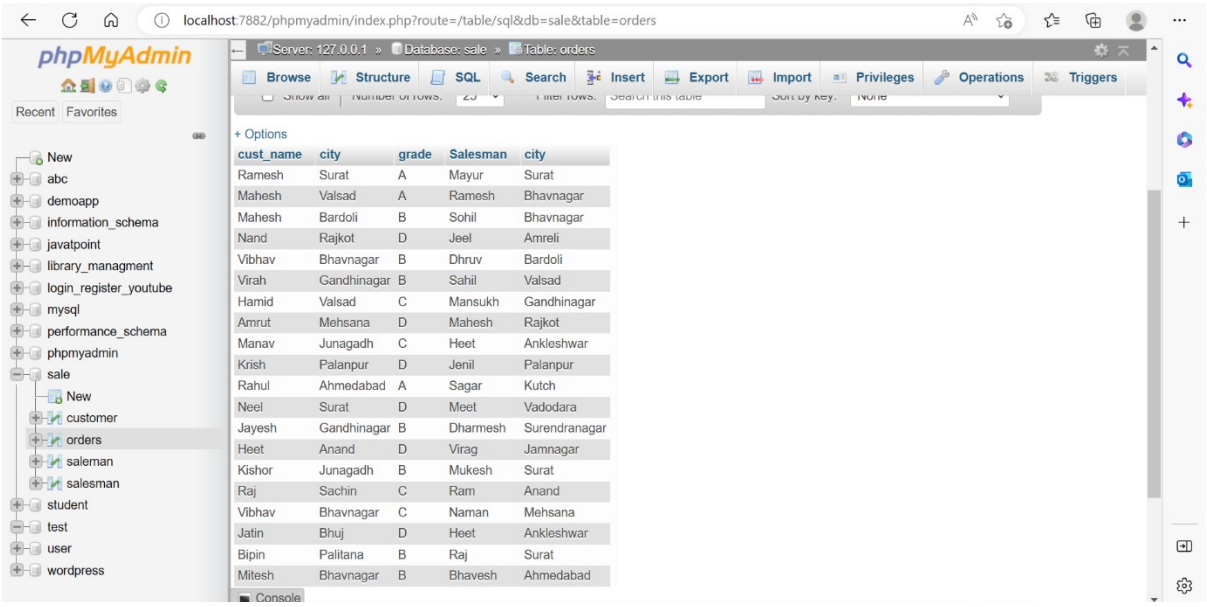
8. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id.

```
SELECT cust_name,city,grade FROM customer ORDER BY customer_id;
```



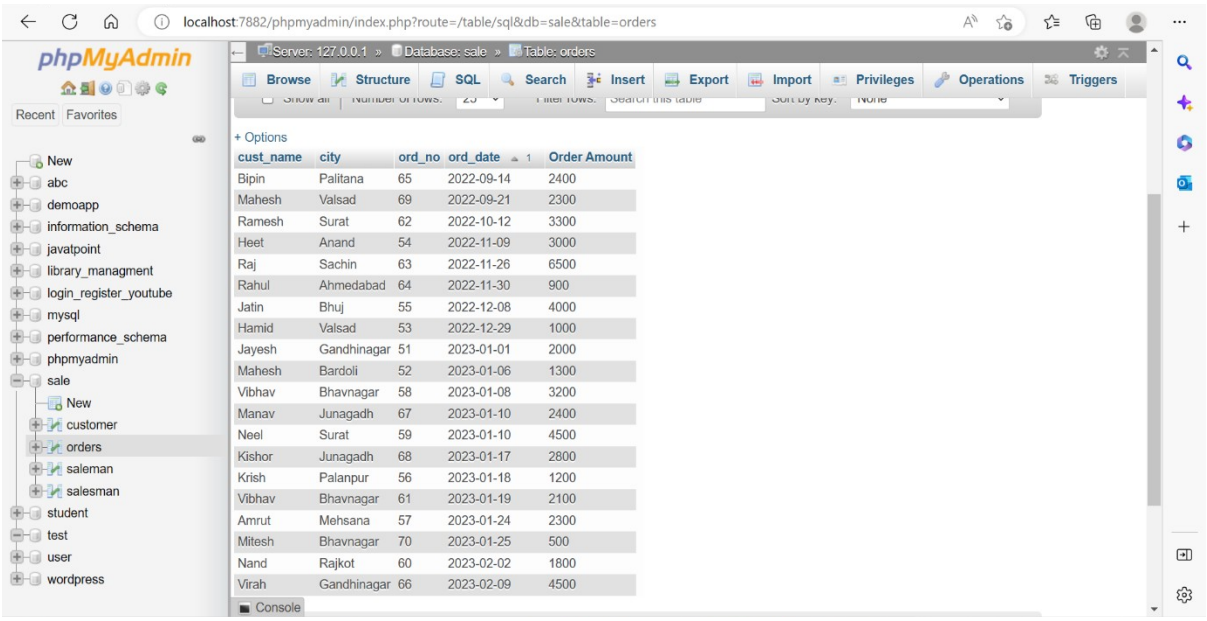
9. write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id.

```
SELECT c.cust_name,c.city,c.grade, s.name AS "Salesman", s.city FROM custom
er c LEFT OUTER JOIN saleman s ON c.salesman_id=s.salesman_id ORDER BY c.cu
stomer_id;
```



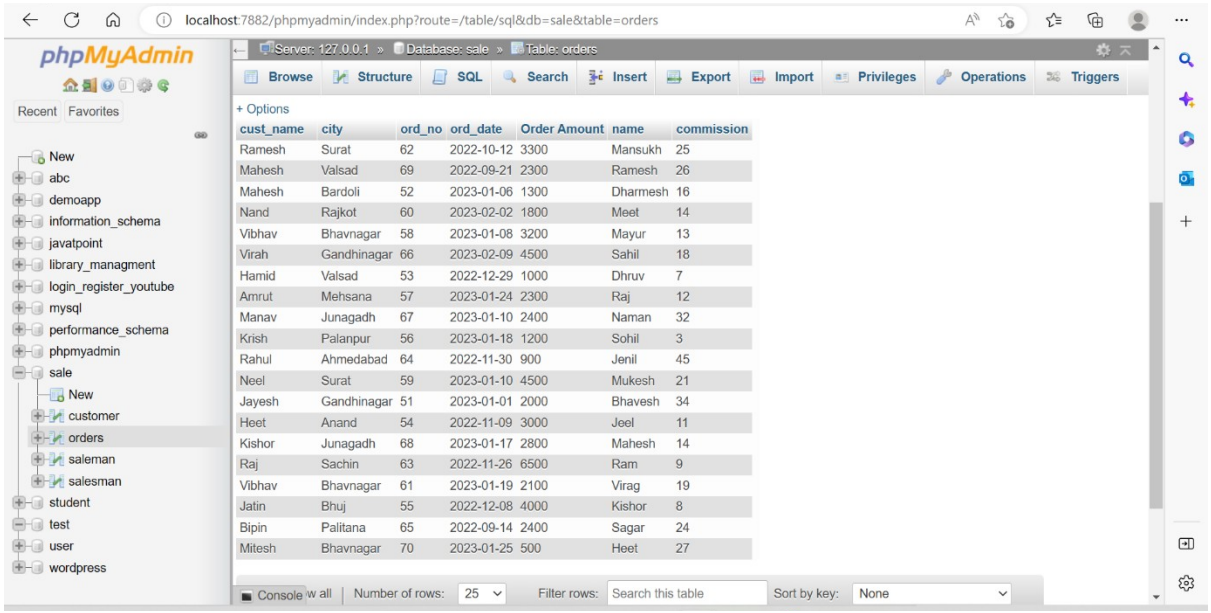
10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not

```
SELECT c.cust_name,c.city, o.ord_no, o.ord_date,o.purch_amt AS "Order
Amount" FROM customer c LEFT OUTER JOIN orders o ON c.customer_id=o.custome
r_id order by o.ord_date;
```



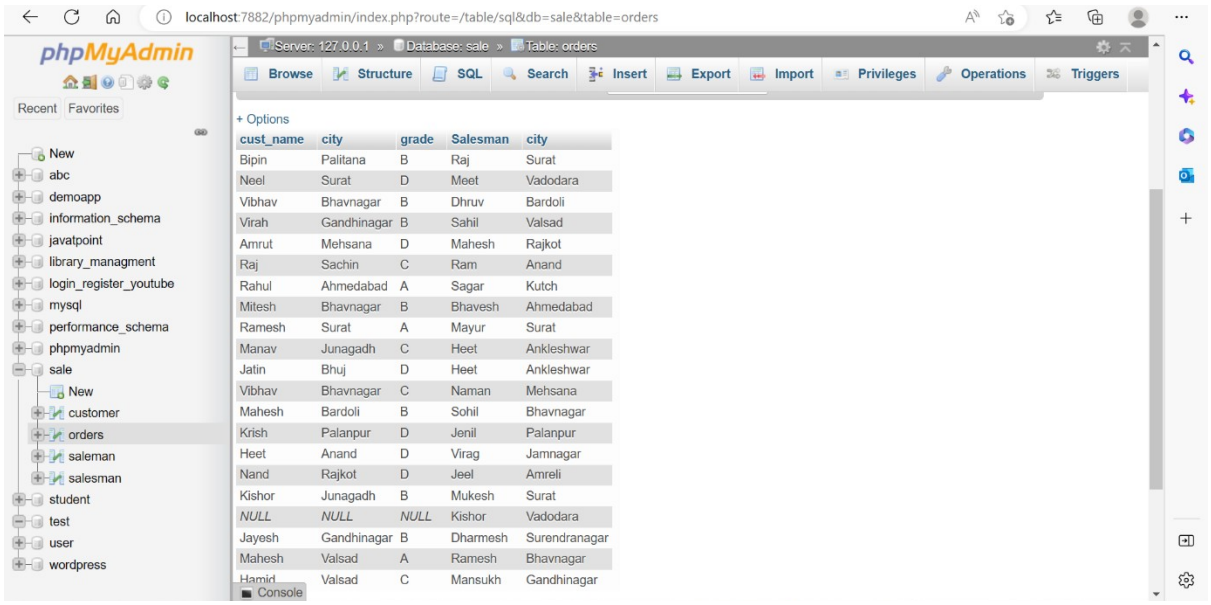
11. Write a SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves

```
SELECT c.cust_name,c.city, o.ord_no, o.ord_date,o.purch_amt AS "Order Amount", s.name,s.commission FROM customer c LEFT OUTER JOIN orders o ON c.customer_id=o.customer_id LEFT OUTER JOIN saleman s ON s.salesman_id=o.salesman_id;
```



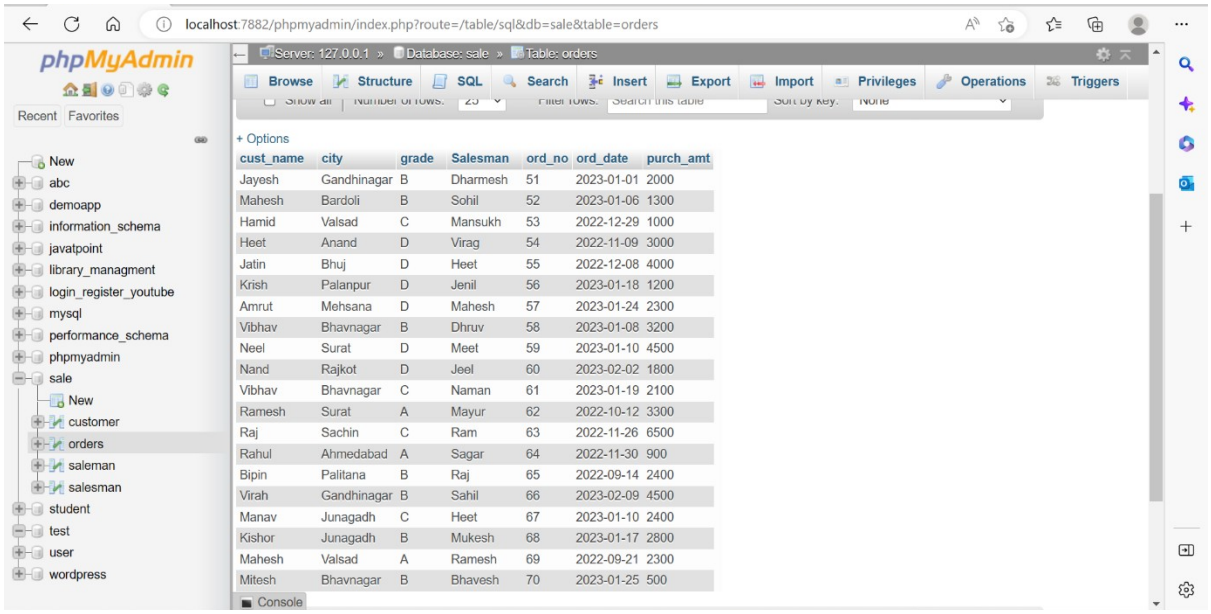
12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers

```
SELECT c.cust_name,c.city,c.grade, s.name AS "Salesman", s.city FROM customer c RIGHT OUTER JOIN saleman s ON s.salesman_id=c.salesman_id ORDER BY s.salesman_id;
```



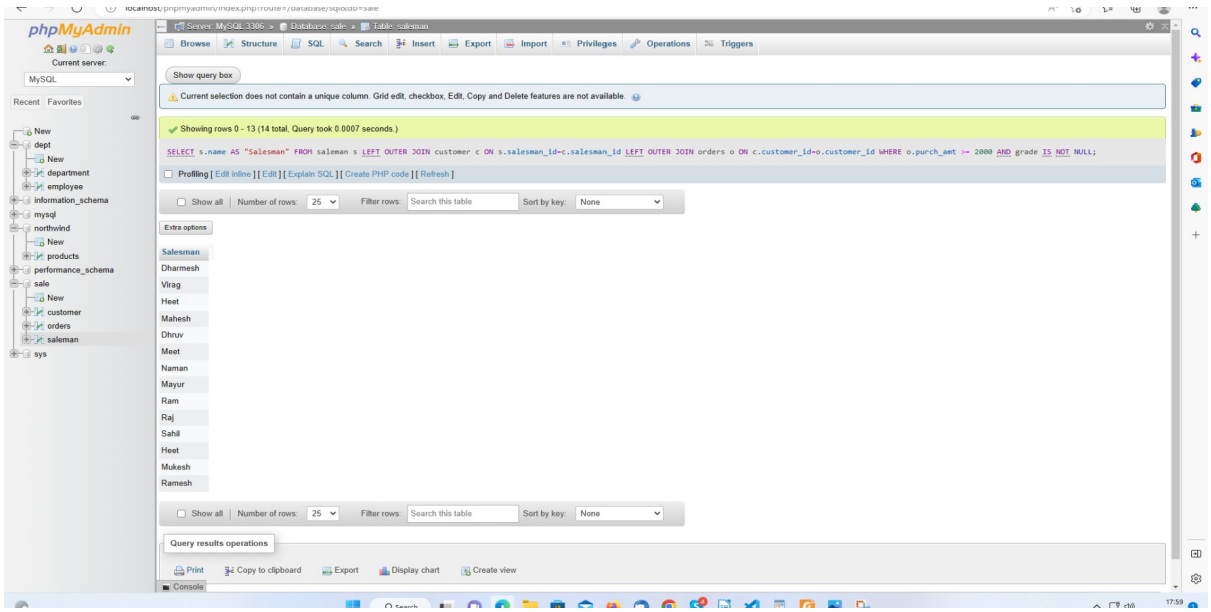
13. write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

```
SELECT c.cust_name,c.city,c.grade, s.name AS "Salesman", o.ord_no, o.ord_date, o.purch_amt FROM customer c RIGHT OUTER JOIN saleman s ON s.salesman_id=c.salesman_id RIGHT OUTER JOIN orders o ON o.customer_id=c.customer_id;
```



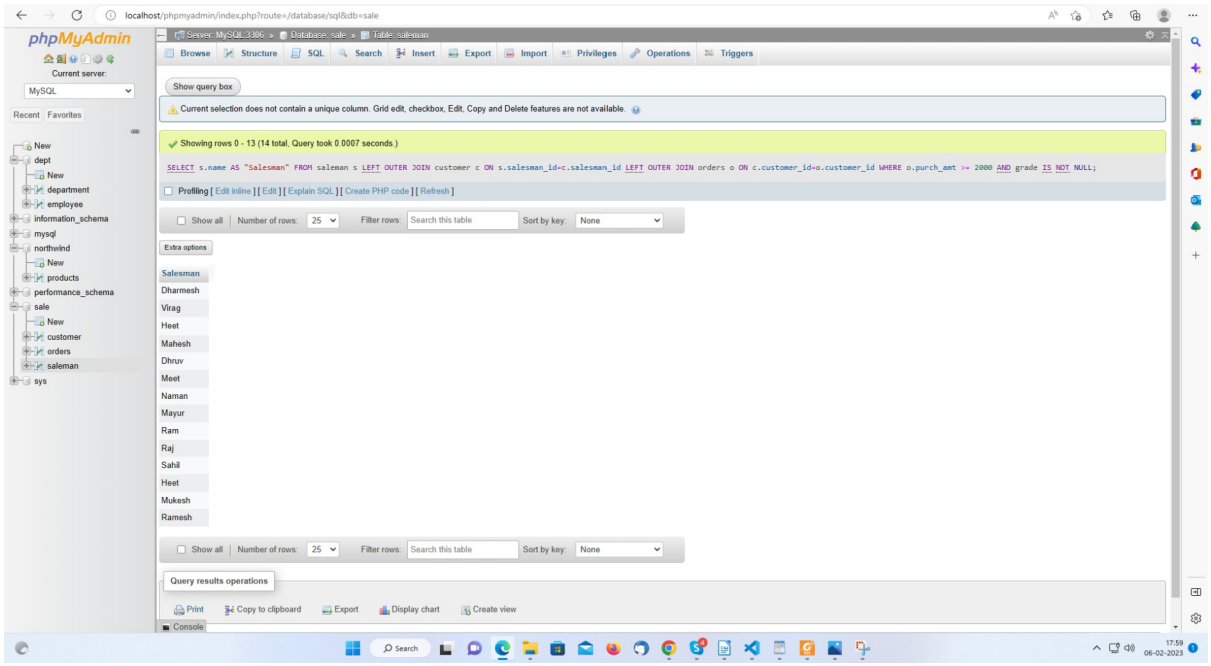
14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customers. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

```
SELECT s.name AS "Salesman" FROM saleman s LEFT OUTER JOIN customer c ON s.salesman_id=c.salesman_id LEFT OUTER JOIN orders o ON c.customer_id=o.customer_id WHERE o.purch_amt >= 2000 AND grade IS NOT NULL;
```



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

```
SELECT s.name AS "Salesman" FROM saleman s LEFT OUTER JOIN customer c ON s
.salesman_id=c.salesman_id LEFT OUTER JOIN orders o ON c.customer_id=o.cus
tomer_id WHERE o.purch_amt >= 2000 AND grade IS NOT NULL;
```



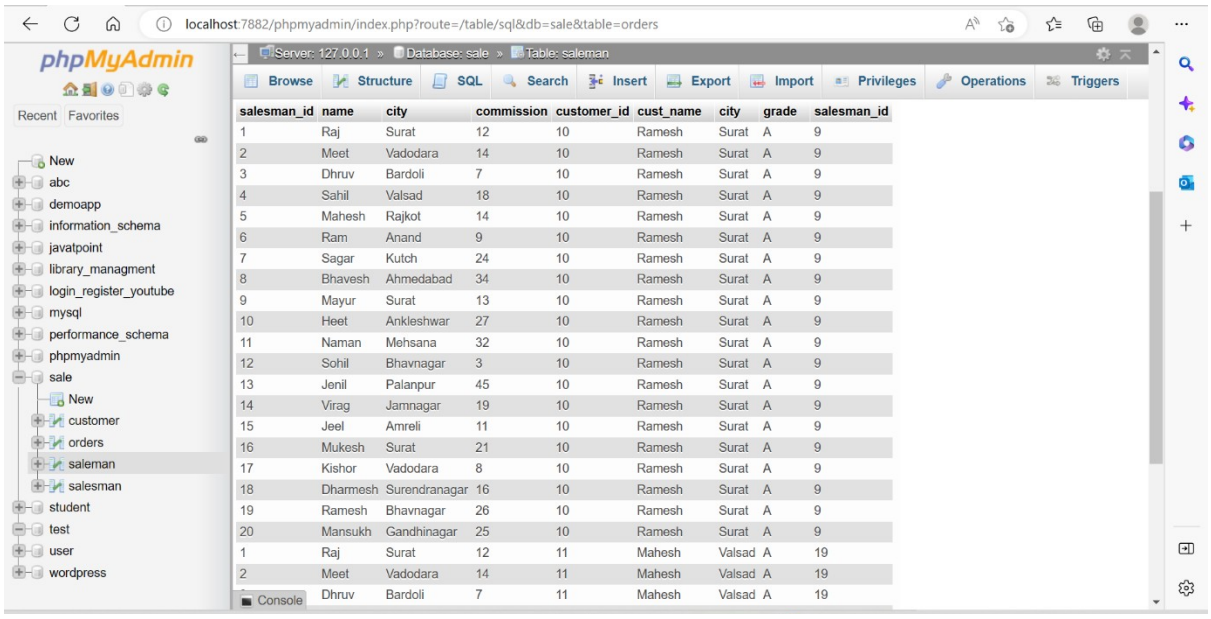
16. Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

SELECT c.cust_name AS 'customer name', c.city, o.ord_no as 'order no.', o.ord_date as 'order date', o.purch_amt AS 'purchase
amount' FROM customer c JOIN orders o ON c.customer_id=o.customer_id WHERE (c.grad e is not NULL and o.ord_no is null) or (o.ord_no in (SELECT oe.ord_no from orders oe W HERE oe.customer_id NOT IN (SELECT customer_id from customer)));

NO DATA

17. Write a SQL query to combine each row of the salesman table with each row of the customer table

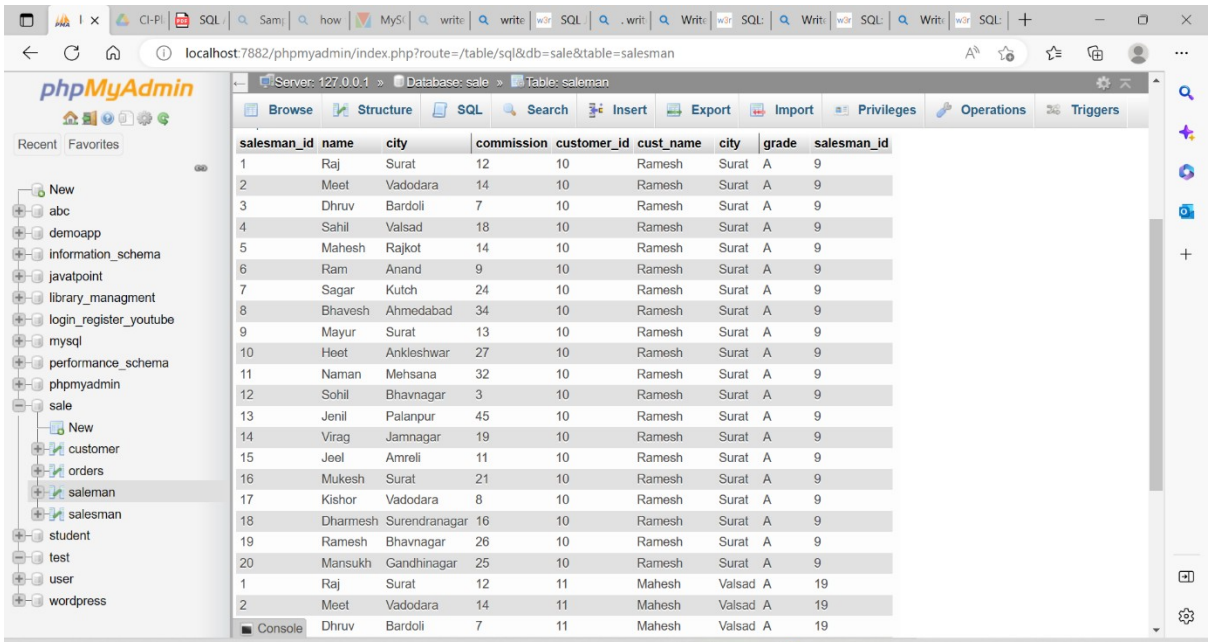
```
SELECT * FROM saleman s CROSS JOIN customer c;
```



salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	Raj	Surat	12	10	Ramesh	Surat	A	9
2	Meet	Vadodara	14	10	Ramesh	Surat	A	9
3	Dhruv	Bardoli	7	10	Ramesh	Surat	A	9
4	Sahil	Valsad	18	10	Ramesh	Surat	A	9
5	Mahesh	Rajkot	14	10	Ramesh	Surat	A	9
6	Ram	Anand	9	10	Ramesh	Surat	A	9
7	Sagar	Kutch	24	10	Ramesh	Surat	A	9
8	Bhavesh	Ahmedabad	34	10	Ramesh	Surat	A	9
9	Mayur	Surat	13	10	Ramesh	Surat	A	9
10	Heet	Ankleshwar	27	10	Ramesh	Surat	A	9
11	Naman	Mehsana	32	10	Ramesh	Surat	A	9
12	Sohil	Bhavnagar	3	10	Ramesh	Surat	A	9
13	Jenil	Palanpur	45	10	Ramesh	Surat	A	9
14	Virag	Jamnagar	19	10	Ramesh	Surat	A	9
15	Jeel	Amreli	11	10	Ramesh	Surat	A	9
16	Mukesh	Surat	21	10	Ramesh	Surat	A	9
17	Kishor	Vadodara	8	10	Ramesh	Surat	A	9
18	Dharmesh	Surendranagar	16	10	Ramesh	Surat	A	9
19	Ramesh	Bhavnagar	26	10	Ramesh	Surat	A	9
20	Mansukh	Gandhinagar	25	10	Ramesh	Surat	A	9
1	Raj	Surat	12	11	Mahesh	Valsad	A	19
2	Meet	Vadodara	14	11	Mahesh	Valsad	A	19
	Dhruv	Bardoli	7	11	Mahesh	Valsad	A	19

18. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city

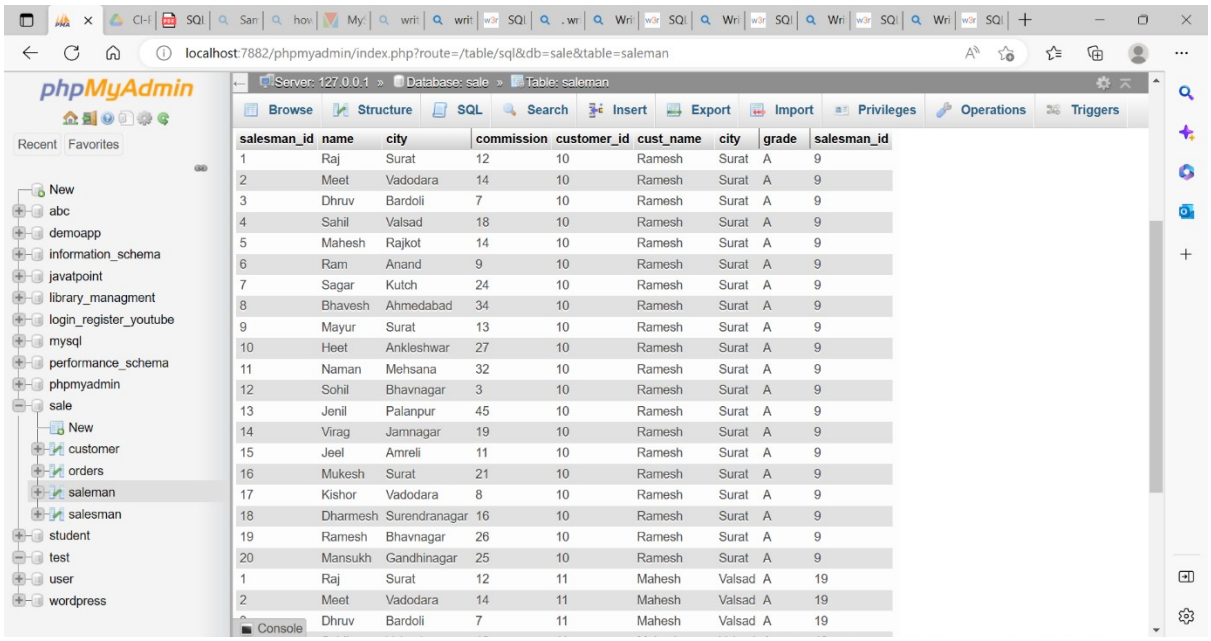
```
SELECT * FROM saleman s CROSS JOIN customer c WHERE s.city IS NOT NULL;
```



salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	Raj	Surat	12	10	Ramesh	Surat	A	9
2	Meet	Vadodara	14	10	Ramesh	Surat	A	9
3	Dhruv	Bardoli	7	10	Ramesh	Surat	A	9
4	Sahil	Valsad	18	10	Ramesh	Surat	A	9
5	Mahesh	Rajkot	14	10	Ramesh	Surat	A	9
6	Ram	Anand	9	10	Ramesh	Surat	A	9
7	Sagar	Kutch	24	10	Ramesh	Surat	A	9
8	Bhavesh	Ahmedabad	34	10	Ramesh	Surat	A	9
9	Mayur	Surat	13	10	Ramesh	Surat	A	9
10	Heet	Ankleshwar	27	10	Ramesh	Surat	A	9
11	Naman	Mehsana	32	10	Ramesh	Surat	A	9
12	Sohil	Bhavnagar	3	10	Ramesh	Surat	A	9
13	Jenil	Palanpur	45	10	Ramesh	Surat	A	9
14	Virag	Jamnagar	19	10	Ramesh	Surat	A	9
15	Jeel	Amreli	11	10	Ramesh	Surat	A	9
16	Mukesh	Surat	21	10	Ramesh	Surat	A	9
17	Kishor	Vadodara	8	10	Ramesh	Surat	A	9
18	Dharmesh	Surendranagar	16	10	Ramesh	Surat	A	9
19	Ramesh	Bhavnagar	26	10	Ramesh	Surat	A	9
20	Mansukh	Gandhinagar	25	10	Ramesh	Surat	A	9
1	Raj	Surat	12	11	Mahesh	Valsad	A	19
2	Meet	Vadodara	14	11	Mahesh	Valsad	A	19
	Dhruv	Bardoli	7	11	Mahesh	Valsad	A	19

19. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for every customer and vice versa for those salesmen who belong to a city and customers who require a grade

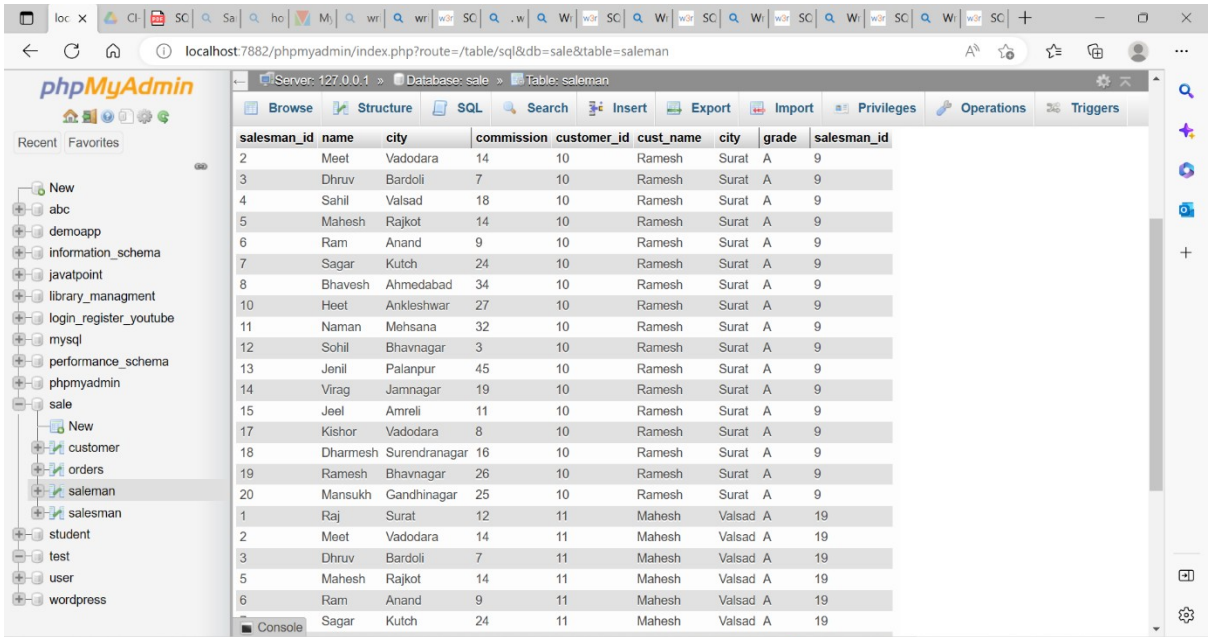
```
SELECT * FROM saleman s CROSS JOIN customer c WHERE s.city IS NOT NULL AND c.grade IS NOT NULL;
```



salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	Raj	Surat	12	10	Ramesh	Surat	A	9
2	Meet	Vadodara	14	10	Ramesh	Surat	A	9
3	Dhruv	Bardoli	7	10	Ramesh	Surat	A	9
4	Sahil	Valsad	18	10	Ramesh	Surat	A	9
5	Mahesh	Rajkot	14	10	Ramesh	Surat	A	9
6	Ram	Anand	9	10	Ramesh	Surat	A	9
7	Sagar	Kutch	24	10	Ramesh	Surat	A	9
8	Bhavesh	Ahmedabad	34	10	Ramesh	Surat	A	9
9	Mayur	Surat	13	10	Ramesh	Surat	A	9
10	Heet	Ankleshwar	27	10	Ramesh	Surat	A	9
11	Naman	Mehsana	32	10	Ramesh	Surat	A	9
12	Sohil	Bhavnagar	3	10	Ramesh	Surat	A	9
13	Jenil	Palanpur	45	10	Ramesh	Surat	A	9
14	Virag	Jamnagar	19	10	Ramesh	Surat	A	9
15	Jeel	Amreli	11	10	Ramesh	Surat	A	9
16	Mukesh	Surat	21	10	Ramesh	Surat	A	9
17	Kishor	Vadodara	8	10	Ramesh	Surat	A	9
18	Dharmesh	Surendranagar	16	10	Ramesh	Surat	A	9
19	Ramesh	Bhavnagar	26	10	Ramesh	Surat	A	9
20	Mansukh	Gandhinagar	25	10	Ramesh	Surat	A	9
1	Raj	Surat	12	11	Mahesh	Valsad	A	19
2	Meet	Vadodara	14	11	Mahesh	Valsad	A	19
	Dhruv	Bardoli	7	11	Mahesh	Valsad	A	19

20. Write a SQL statement to make a Cartesian product between salesman and customer i.e. each salesman will appear for all customers and vice versa for those salesmen who must belong to a city which is not the same as his customer and the customers should have their own grade

```
SELECT * FROM saleman s CROSS JOIN customer c WHERE s.city IS NOT NULL AND c.grade IS NOT NULL AND s.city<>c.city;
```



salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
2	Meet	Vadodara	14	10	Ramesh	Surat	A	9
3	Dhruv	Bardoli	7	10	Ramesh	Surat	A	9
4	Sahil	Valsad	18	10	Ramesh	Surat	A	9
5	Mahesh	Rajkot	14	10	Ramesh	Surat	A	9
6	Ram	Anand	9	10	Ramesh	Surat	A	9
7	Sagar	Kutch	24	10	Ramesh	Surat	A	9
8	Bhavesh	Ahmedabad	34	10	Ramesh	Surat	A	9
10	Heet	Ankleshwar	27	10	Ramesh	Surat	A	9
11	Naman	Mehsana	32	10	Ramesh	Surat	A	9
12	Sohil	Bhavnagar	3	10	Ramesh	Surat	A	9
13	Jenil	Palanpur	45	10	Ramesh	Surat	A	9
14	Virag	Jamnagar	19	10	Ramesh	Surat	A	9
15	Jeel	Amreli	11	10	Ramesh	Surat	A	9
17	Kishor	Vadodara	8	10	Ramesh	Surat	A	9
18	Dharmesh	Surendranagar	16	10	Ramesh	Surat	A	9
19	Ramesh	Bhavnagar	26	10	Ramesh	Surat	A	9
20	Mansukh	Gandhinagar	25	10	Ramesh	Surat	A	9
1	Raj	Surat	12	11	Mahesh	Valsad	A	19
2	Meet	Vadodara	14	11	Mahesh	Valsad	A	19
3	Dhruv	Bardoli	7	11	Mahesh	Valsad	A	19
5	Mahesh	Rajkot	14	11	Mahesh	Valsad	A	19
6	Ram	Anand	9	11	Mahesh	Valsad	A	19
7	Sagar	Kutch	24	11	Mahesh	Valsad	A	19