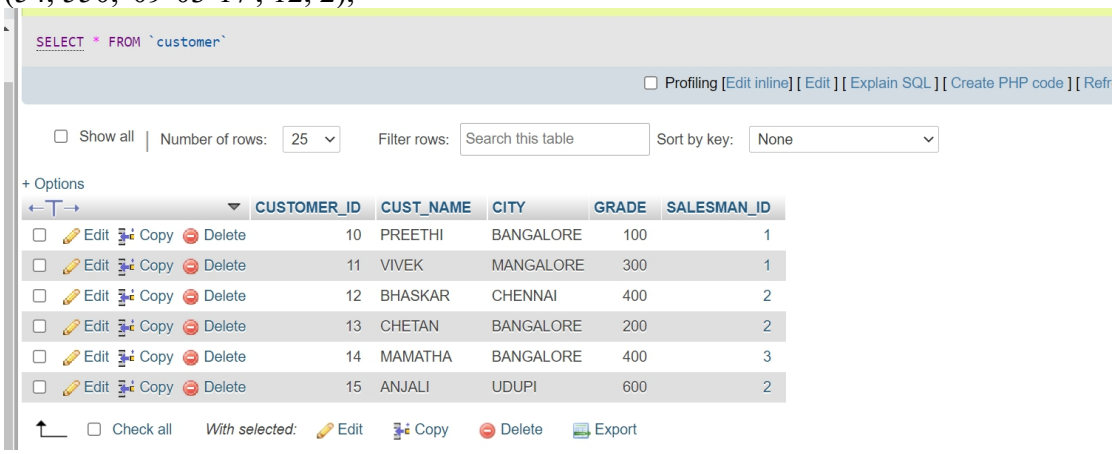
Program 6: Order Database Consider the following schema for Order Database:
SALESMAN (Salesman_id,Name, City, Commission) CUSTOMER
(Customer_id,Cust_Name, City, Grade, Salesman_id) ORDERS
(Ord_No,Purchase_Amt, Ord_Date, Customer_id, Salesman_id)

```
CREATE DATABASE order_database;

CREATE TABLE SALESMAN
(SALESMAN_ID int,
NAME VARCHAR (20),
CITY VARCHAR (20),
COMMISSION VARCHAR (20),
PRIMARY KEY (SALESMAN_ID));
CREATE TABLE CUSTOMER
(CUSTOMER_ID INT,
CUST_NAME VARCHAR (20),
CITY VARCHAR (20),
GRADE INT (3),
SALESMAN_ID int,
PRIMARY KEY (CUSTOMER_ID),
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID)
ON DELETE SET NULL ON UPDATE CASCADE);
CREATE TABLE ORDERS
(ORD_NO INT,
PURCHASE_AMT INT,
ORD_DATE DATE,
CUSTOMER_ID INT,
SALESMAN_ID INT,
PRIMARY KEY (ORD_NO),
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER (CUSTOMER_ID)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID)
ON DELETE CASCADE ON UPDATE CASCADE);


INSERT INTO SALESMAN VALUES (1, 'JASON','BANGALORE','25 %'),
(2, 'SHYAM','BANGALORE','20 %'),
(3, 'KUMARA','MYSORE','15 %'),
(4, 'SMITHA','DELHI','30 %'),
(5, 'HARISH','HYDERABAD','15%');
INSERT INTO CUSTOMER VALUES (10, 'PREETHI','BANGALORE', 100, 1),
(11, 'VIVEK','MANGALORE', 300, 1),
(12, 'BHASKAR','CHENNAI', 400, 2),
(13, 'CHETAN','BANGALORE', 200, 2),
(14, 'MAMATHA','BANGALORE', 400, 3),
(15,'ANJALI','UDUPI',600,2);
INSERT INTO ORDERS VALUES (50, 5000, '2017-04-04', 10, 1),
(51, 450, '20-01-17', 10, 2),
(52,1000,'24-02-17',13,2),
(53,3500,'13-04-17',14,3),
```

(54, 550, '09-03-17', 12, 2);



```
SELECT * FROM `customer`
```

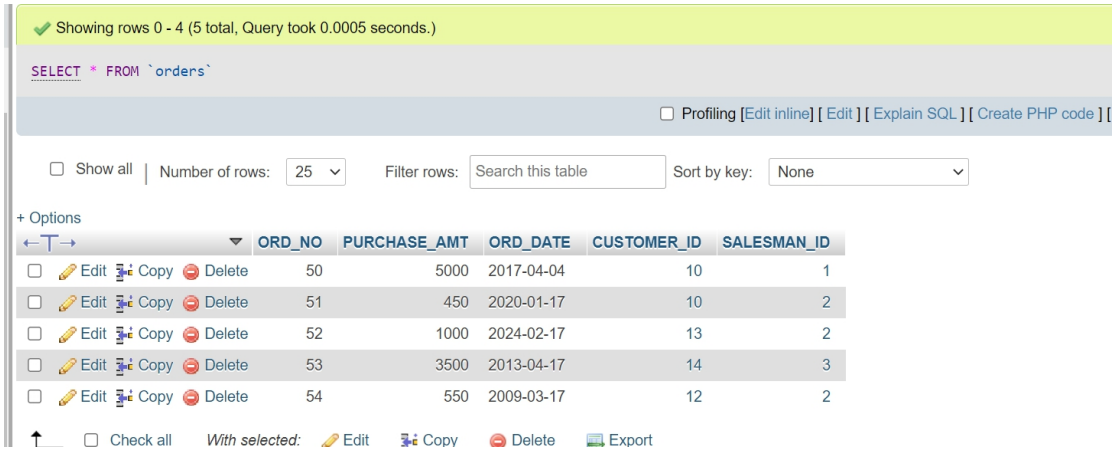
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refr

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
Edit Copy Delete	10	PREETHI	BANGALORE	100	1
Edit Copy Delete	11	VIVEK	MANGALORE	300	1
Edit Copy Delete	12	BHASKAR	CHENNAI	400	2
Edit Copy Delete	13	CHETAN	BANGALORE	200	2
Edit Copy Delete	14	MAMATHA	BANGALORE	400	3
Edit Copy Delete	15	ANJALI	UDUPI	600	2

Check all With selected: Edit Copy Delete Export



Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

```
SELECT * FROM `orders`
```

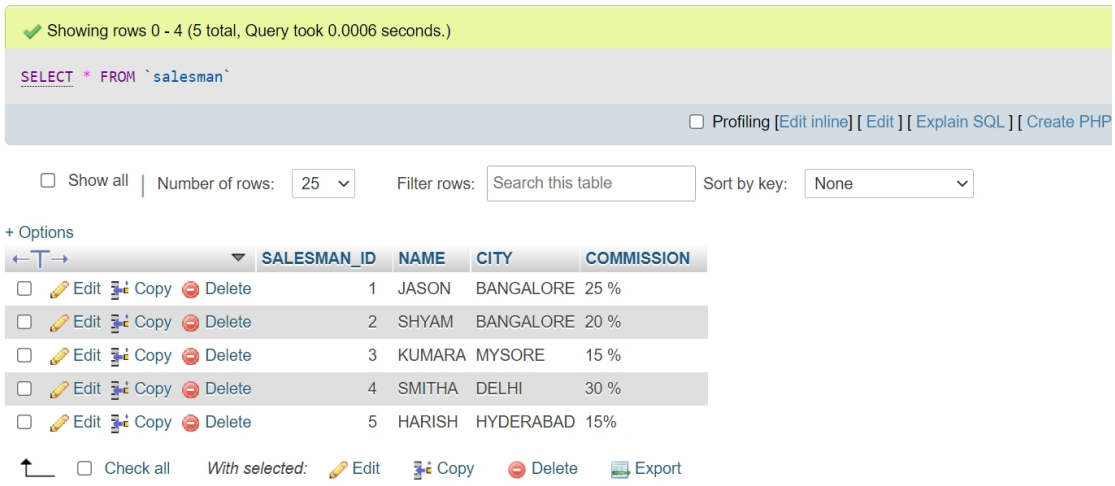
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
Edit Copy Delete	50	5000	2017-04-04	10	1
Edit Copy Delete	51	450	2020-01-17	10	2
Edit Copy Delete	52	1000	2024-02-17	13	2
Edit Copy Delete	53	3500	2013-04-17	14	3
Edit Copy Delete	54	550	2009-03-17	12	2

Check all With selected: Edit Copy Delete Export



Showing rows 0 - 4 (5 total, Query took 0.0006 seconds.)

```
SELECT * FROM `salesman`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

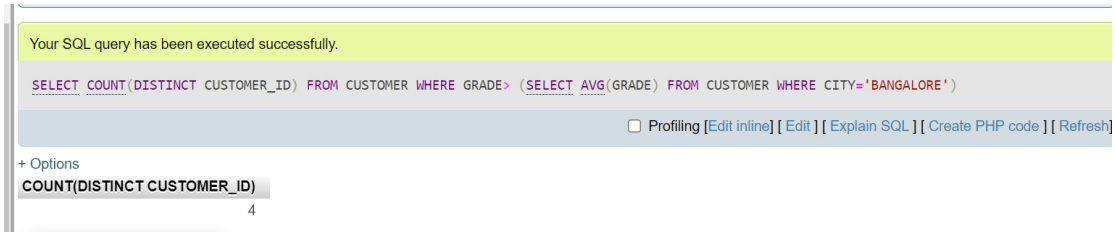
	SALESMAN_ID	NAME	CITY	COMMISSION
Edit Copy Delete	1	JASON	BANGALORE	25 %
Edit Copy Delete	2	SHYAM	BANGALORE	20 %
Edit Copy Delete	3	KUMARA	MYSORE	15 %
Edit Copy Delete	4	SMITHA	DELHI	30 %
Edit Copy Delete	5	HARISH	HYDERABAD	15%

Check all With selected: Edit Copy Delete Export

Write SQL queries to

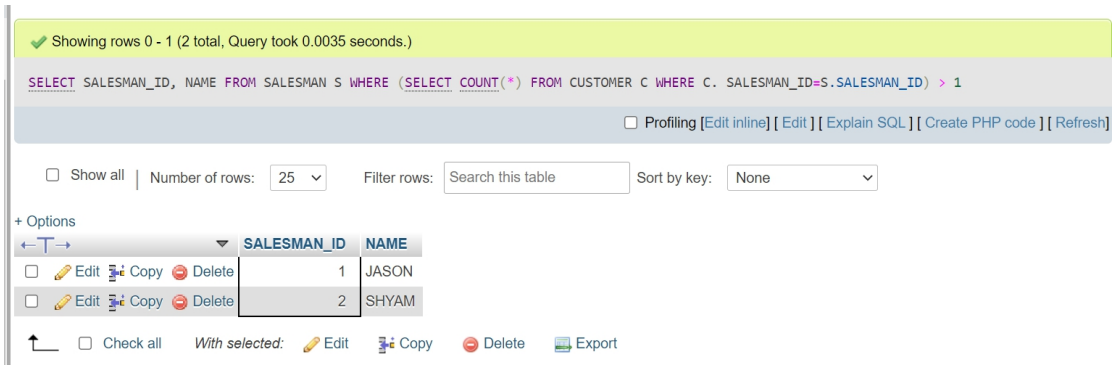
1. Count the customers with grades above Bangalore's average.

SELECT COUNT(DISTINCT CUSTOMER_ID) FROM CUSTOMER WHERE GRADE> (SELECT AVG(GRADE) FROM CUSTOMER WHERE CITY='BANGALORE');



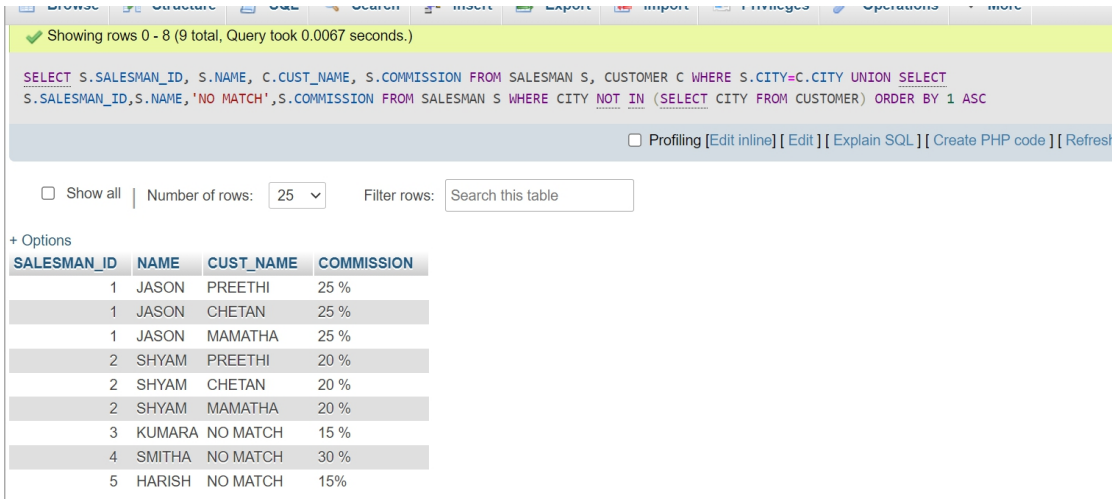
2. Find the name and numbers of all salesmen who had more than one customer.

SELECT SALESMAN_ID, NAME FROM SALESMAN S WHERE (SELECT COUNT(*) FROM CUSTOMER C WHERE C. SALESMAN_ID=S.SALESMAN_ID) > 1;



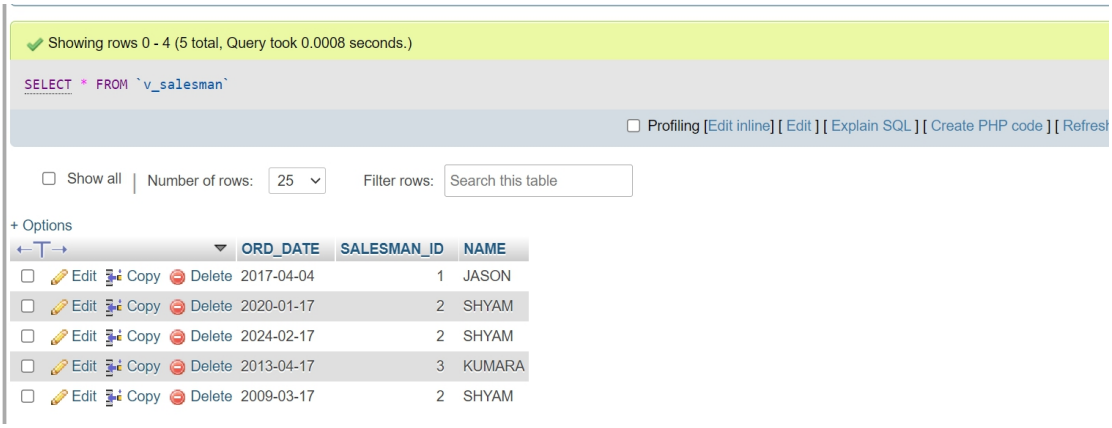
3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

SELECT S.SALESMAN_ID, S.NAME, C.CUST_NAME, S.COMMISSION FROM SALESMAN S, CUSTOMER C WHERE S.CITY=C.CITY UNION SELECT S.SALESMAN_ID,S.NAME,'NO MATCH',S.COMMISSION FROM SALESMAN S WHERE CITY NOT IN (SELECT CITY FROM CUSTOMER) ORDER BY 1 ASC;



4. Create a view that finds the salesman who has the customer with the highest order of a day.

CREATE VIEW V_SALESMAN AS SELECT O.ORD_DATE, S.SALESMAN_ID, S.NAME FROM SALESMAN S,ORDERS O WHERE S.SALESMAN_ID = O.SALESMAN_ID AND O.PURCHASE_AMT= (SELECT MAX(PURCHASE_AMT) FROM ORDERS C WHERE C.ORD_DATE=O.ORD_DATE);



5. Demonstrate the DELETE operation by removing salesman with id 1. All his orders must also be deleted.

DELETE FROM SALESMAN WHERE SALESMAN_ID=1;