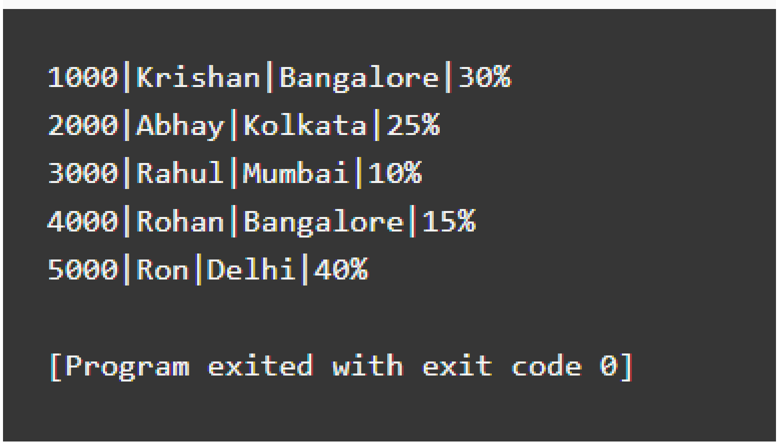
Lab 6 :-

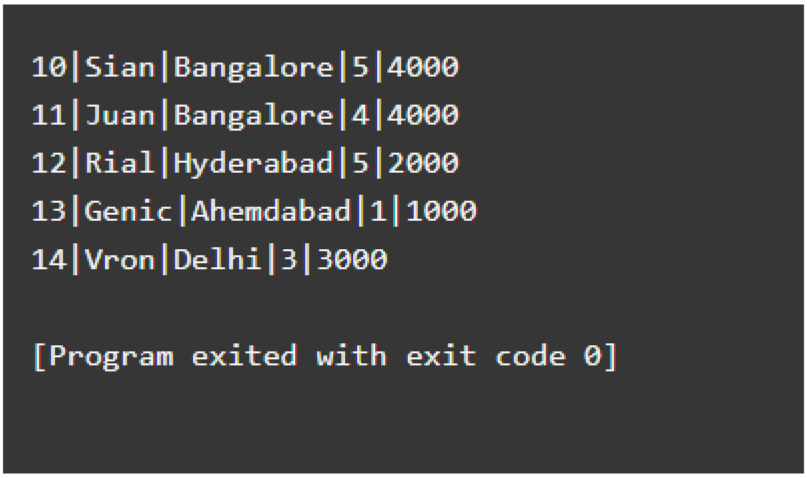
```
CREATE TABLE salesman
(salesman_id INT,
name VARCHAR(20),
city VARCHAR(20),
commission VARCHAR(20),
PRIMARY KEY (salesman_id));

CREATE TABLE customer
(customer_id INT,
cust_name VARCHAR(20),
city VARCHAR(20),
grade INT,
salesman_id INT,
PRIMARY KEY (customer_id),
FOREIGN KEY (salesman_id) REFERENCES salesman (salesman_id)
ON UPDATE CASCADE ON DELETE 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 UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)
ON UPDATE CASCADE ON DELETE CASCADE);


INSERT INTO SALESMAN VALUES(1000,'Krishan','Bangalore','30%'),
(2000,'Abhay','Kolkata','25%'),
(3000,'Rahul','Mumbai','10%'),
(4000,'Rohan','Bangalore','15%'),
(5000,'Ron','Delhi','40%');


INSERT INTO CUSTOMER VALUES(10,'Sian','Bangalore',5,4000),
(11,'Juan','Bangalore',4,4000),
(12,'Rial','Hyderabad',5,2000),
(13,'Genic','Ahemdabad',1,1000),
(14,'Vron','Delhi',3,3000);


INSERT INTO ORDERS VALUES(101,10000,'2021-06-12',10,4000),
(102,15000,'2018-05-07',10,2000),
(103,8000,'2020-11-01',14,3000),
(104,800,'2015-10-25',13,1000),
(105,100,'2020-11-01',11,2000);
```

SELECT * FROM salesman;



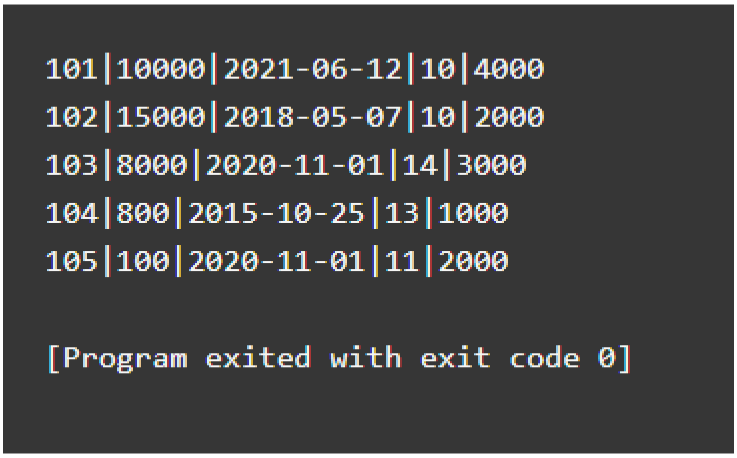
```
1000|Krishan|Bangalore|30%
2000|Abhay|Kolkata|25%
3000|Rahul|Mumbai|10%
4000|Rohan|Bangalore|15%
5000|Ron|Delhi|40%

[Program exited with exit code 0]
```

SELECT * FROM customer;



```
10|Sian|Bangalore|5|4000
11|Juan|Bangalore|4|4000
12|Rial|Hyderabad|5|2000
13|Genic|Ahemdabad|1|1000
14|Vron|Delhi|3|3000

[Program exited with exit code 0]
```

SELECT * FROM orders;

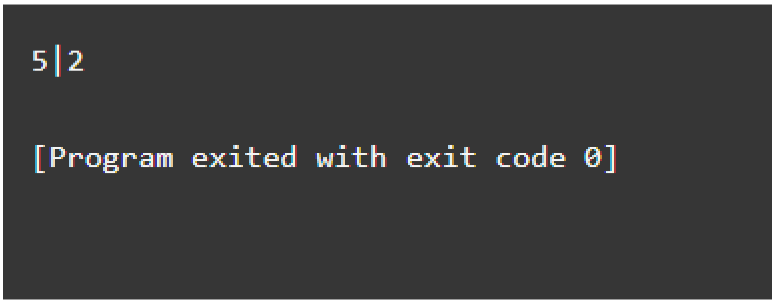


```
101|10000|2021-06-12|10|4000
102|15000|2018-05-07|10|2000
103|8000|2020-11-01|14|3000
104|800|2015-10-25|13|1000
105|100|2020-11-01|11|2000

[Program exited with exit code 0]
```

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

SELECT grade, COUNT (DISTINCT customer_id) FROM customer GROUP BY grade

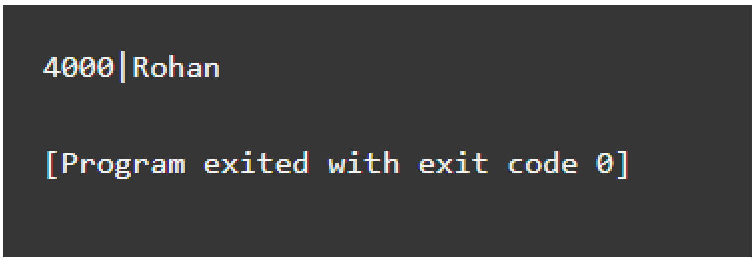
HAVING grade > (SELECT AVG(grade) FROM customer WHERE city = 'Bangalore');



```
5|2

[Program exited with exit code 0]
```

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

SELECT s.salesman_id, s.name FROM salesman s, customer c WHERE c.salesman_id=s.salesman_id GROUP BY s.salesman_id HAVING COUNT(s.salesman_id)>1;



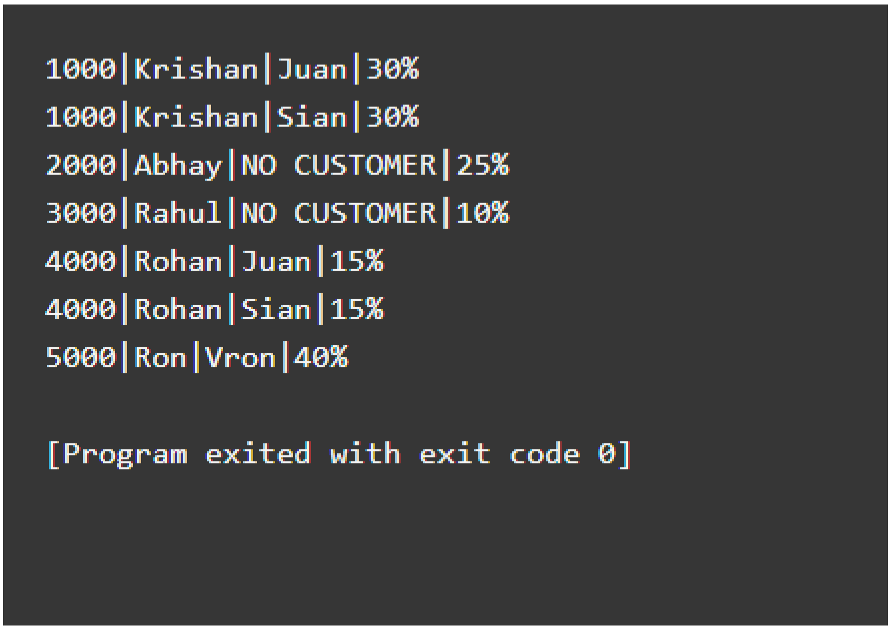
```
4000|Rohan

[Program exited with exit code 0]
```

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 salesman_id, name, 'NO CUSTOMER', commission FROM salesman WHERE city NOT IN (SELECT city FROM customer);



```
1000|Krishan|Juan|30%
1000|Krishan|Sian|30%
2000|Abhay|NO CUSTOMER|25%
3000|Rahul|NO CUSTOMER|10%
4000|Rohan|Juan|15%
4000|Rohan|Sian|15%
5000|Ron|Vron|40%

[Program exited with exit code 0]
```

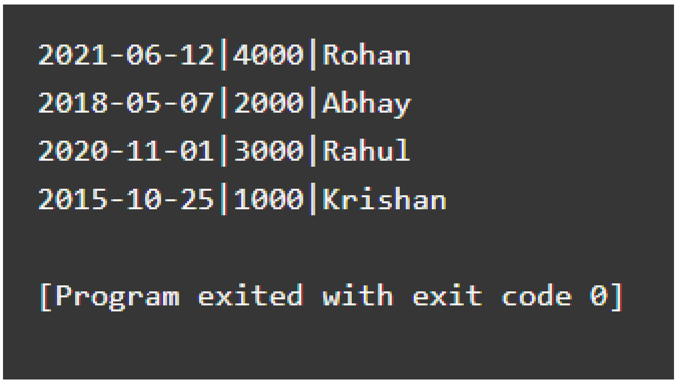
4. Create a view that finds the salesman who has the customer with the highest order of a day.

CREATE VIEW high_order 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 o1 WHERE o1.ord_date = o.ord_date);

SELECT * FROM high_order;

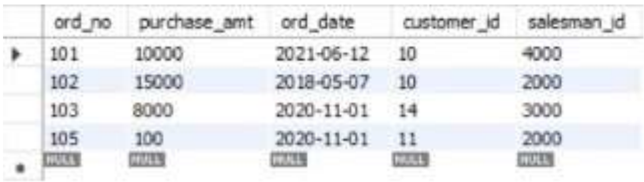


```
2021-06-12|4000|Rohan
2018-05-07|2000|Abhay
2020-11-01|3000|Rahul
2015-10-25|1000|Krishan

[Program exited with exit code 0]
```

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

DELETE from salesman WHERE salesman_id=1000;

SELECT * FROM orders;



ord_no	purchase_amt	ord_date	customer_id	salesman_id
101	10000	2021-06-12	10	4000
102	15000	2018-05-07	10	2000
103	8000	2020-11-01	14	3000
105	100	2020-11-01	11	2000
NULL	NULL	NULL	NULL	NULL