SQL Queries:

1. List all the columns of the Salespeople table.

```
      Desc Salespeople;
```

2. List all customers with a rating of 100.

```
mqsql>select * from customer where rating=100;
```

3. Find all records in the Customer table with NULL values in the city column.

```
      mysql>select * from customer where city is null ;
```

4. Find the largest order taken by each salesperson on each date.

```
      mysql>select a.snum,b.odate,max(b.amt) from customer as a, orders
as b where a.cnum=b.cnum  group by a.snum,b.odate;
```

5. Arrange the Orders table by descending customer number.

```
      mysql>SELECT * FROM orders order by cnum desc;
```

6. Find which salespeople currently have orders in the Orders table.

```
      mysql> select a.snum,a.cnum,b.onum from customer as a,orders as b
where a.cnum=b.cnum;
```

7. List names of all customers matched with the salespeople serving them.

```
mysql> select c.cnum,c.cname,c.city,c.rating,s.snum from customer as c,
salepeople as s  where c.snum=s.snum;
```

8. Find the names and numbers of all salespeople who had more than one customer.

```
mysql> select count(c.cnum),s.sname ,s.snum from customer as c,salepeople
as s where c.snum=s.snum group by c.snum;
```

9. Count the orders of each of the salespeople and output the results in descending order.

```
      mysql> select c.snum,count(o.onum) from orders as o,customer as c
where c.cnum=o.cnum group by c.snum order by o.onum desc ;
```

10. List the Customer table if and only if one or more of the customers in the Customer table are
located in San Jose.

```
mysql> select * from customer where city='San jose';
```

11. Match salespeople to customers according to what city they lived in.

```
mysql> select c.cnum,c.cname,c.city,s.snum,s.sname,s.city from salepeople
as s,customer as c where c.city=s.city and c.snum=s.snum;
```

12. Find the largest order taken by each salesperson.

```
mysql> select a.snum,b.onum,max(b.amt)from customer as a,orders as b
where a.cnum=b.cnum group by a.snum;
```

13. Find customers in San Jose who have a rating above 200.

```
      mysql> select cnum,cname,city,rating from customer where city='San
jose' and rating>200;
```

14. List the names and commissions of all salespeople in London.

```
      mysql> select snum,sname,city,comm  from salepeople where
city='London';
```

15. List all the orders of salesperson Motika from the Orders table.

```
      mysql> select * from orders  where cnum=(select c.cnum from
customer as c,salepeople as s where c.snum=s.snum and s.sname='Motika');
```

16. Find all customers with orders on October 3.

```
mysql> select c.cnum,c.cname,o.onum,o.odate from customer as c, orders as
o where c.cnum=o.cnum and odate = '1996-03-10';
```

17. Give the sums of the amounts from the Orders table, grouped by date, eliminating all those dates where the SUM was not at least 2000.00 above the MAX amount.

```
      mysql> select sum(amt),onum,odate from orders group by odate having
sum(amt)>2000.00 ;
```

18. Select all orders that had amounts that were greater than at least one of the orders from October 6.

```
      mysql> select *from orders where amt>(select min(amt)from orders
group by odate having odate='1996-06-10');
```

19. Write a query that uses the EXISTS operator to extract all salespeople who have customers with a rating of 300.

```
mysql> select s.snum,s.sname,s.city,s.comm from salepeople as s, customer
as c where s.snum=c.snum and c.rating=300;

mysql> select * from salepeople where exists (select cnum from customer
where salepeople.snum=customer.snum and customer.rating=300);
```

20. Find all pairs of customers having the same rating.

```
      mysql> select n.cname,m.cname,m.rating from customer m,customer n
where m.rating=n.rating and n.cname!=m.cname order by rating;
+----------+----------+--------+
| cname    | cname    | rating |
+----------+----------+--------+
| Pereira  | Clemens  |    100 |
| Hoffman  | Pereira  |    100 |
| Clemens  | Hoffman  |    100 |
| Clemens  | Pereira  |    100 |
| Pereira  | Hoffman  |    100 |
| Hoffman  | Clemens  |    100 |
| Giovanni | Liu      |    200 |
| Liu      | Giovanni |    200 |
| Cinseros | Grass    |    300 |
| Grass    | Cinseros |    300 |
+----------+----------+--------+
```

21. Find all customers whose CNUM is 1000 above the SNUM of Serres.

```
mysql> select * from customer where cnum=1000+(select snum from
salepeople where sname='Serres') ;
```

22. Give the salespeople’s commissions as percentages instead of decimal numbers.

```
mysql> select comm*100  from salepeople;
```

23. Find the largest order taken by each salesperson on each date, eliminating those MAX orders which are less than $3000.00 in value.

```
      mysql> select o.onum,o.odate,max(o.amt),c.snum,c.cnum from orders
as o,customer as c where o.cnum=c.cnum group by c.snum,o.odate having
max(o.amt)<3000.00;
```

24. List the largest orders for October 3, for each salesperson.

```
mysql> select o.onum,o.odate,max(o.amt),c.snum,c.cnum from orders as
o,customer as c where o.cnum=c.cnum group by c.snum having o.odate='1996-
03-10';
```

25. Find all customers located in cities where Serres (SNUM 1002) has customers.

```
mysql> select c.cnum,c.cname,s.snum,s.sname from customer as c,salepeople
as s where s.snum=c.snum and s.snum=1002 and sname='Serres';
```

26. Select all customers with a rating above 200.00.

```
mysql> select * from customer where rating>200.00 ;
```

27. Count the number of salespeople currently listing orders in the Orders table.

```
mysql> select c.snum,count(distinct(c.snum)),o.onum from customer as
c,orders as o where c.cnum=o.cnum group by c.snum ;

mysql> select count(distinct(c.snum)) from customer as c,orders as o
where c.cnum=o.cnum ;
```

28. Write a query that produces all customers serviced by salespeople with a commission above
12%. Output the customer’s name and the salesperson’s rate of commission.

```
mysql> select c.cnum,c.cname,c.city,c.rating,c.snum ,s.comm from customer
as c, salepeople as s where c.snum=s.snum and comm*100>12 ;
```

29. Find salespeople who have multiple customers.

```
      mysql> select s.snum ,s.sname,s.city,s.comm,count(c.snum),c.cnum
from customer as c, salepeople as s where c.snum=s.snum group by c.snum
having count(c.snum)>1;
```

30. Find salespeople with customers located in their city.

```
      mysql> select s.snum ,s.sname,s.city,c.city,c.cnum,c.cname from
customer as c, salepeople as s where c.snum=s.snum and c.city=s.city;
```

31. Find all salespeople whose name starts with ‘P’ and the fourth character is ‘l’.

```
mysql> select * from salepeople where sname like "P%__l";
```

32. Write a query that uses a subquery to obtain all orders for the customer named Cinseros.
Assume you do not know his customer number.

```
mysql> select * from orders where cnum=(select cnum from customer where
cname='Cinseros');
```

33. Find the largest orders for Serres and Rifkin.

```
mysql> select b.snum,c.sname,max(a.amt) from orders as a,customer as
b,salepeople as c where a.cnum=b.cnum and b.snum=c.snum and
(c.sname='Rifkin' or c.sname='Serres') group by b.snum;
```

34. Extract the Salespeople table in the following order : SNUM, SNAME, COMMISSION, CITY.

35. Select all customers whose names fall in between ‘A’ and ‘G’ alphabetical range.

```
mysql> select * from customer where substring(cname,1,1) in
("A","B","C","D","E","F","G");
```

36. Select all the possible combinations of customers that you can assign.

37. Select all orders that are greater than the average for October 4.

```
mysql> select * from orders where amt>(select avg(amt) from orders group
by odate having odate="1996-04-10");
```

38. Write a select command using a corelated subquery that selects the names and numbers of all
customers with ratings equal to the maximum for their city.

```
mysql> select cnum,cname,city,rating from customer c where rating=(Select
max(rating)from customer where city=c.city group by city);
```

39. Write a query that totals the orders for each day and places the results in descending order.

```
mysql> select onum,sum(amt) as total_order,odate from orders group by
odate order by total_order;
```

40. Write a select command that produces the rating followed by the name of each customer in
San Jose.

```
mysql> select rating,cname,cnum from customer where city="San jose";
```

41. Find all orders with amounts smaller than any amount for a customer in San Jose.

```
mysql> select * from orders where amt<(select max(o.amt) from customer as
c,orders as o where o.cnum=c.cnum group by c.city having c.city="San
jose");

mysql> select * from orders where amt<(select min(o.amt) from customer as
c,orders as o where o.cnum=c.cnum group by c.city having c.city="San
jose");
```

42. Find all orders with above average amounts for their customers.

```
mysql> select amt,cnum from orders o where amt>=(select avg(amt)from
orders where cnum=o.cnum group by cnum) ;
```

43. Write a query that selects the highest rating in each city.

```
mysql> select max(rating),city from customer group by city;
```

44. Write a query that calculates the amount of the salesperson’s commission on each order by a
customer with a rating above 100.00.

```
mysql> select o.onum,c.cnum,c.rating,s.snum,s.comm from orders as
o,customer as c,salepeople as s where o.cnum=c.cnum and s.snum=c.snum and
(c.rating>100.00);
```

45. Count the customers with ratings above San Jose’s average.

```
     mysql> select count(cnum)from customer where rating>(select
avg(rating) from customer group by city having city="San jose");
```

46. Write a query that produces all pairs of salespeople with themselves as well as duplicate rows
with the order reversed.

```
     mysql> select a.sname,b.sname from salepeople as a,salepeople as b
where a.sname!=b.sname;
```

47. Find all salespeople that are located in either Barcelona or London.

```
mysql> select * from salepeople where city="Bercelona" or city="London";
```

48. Find all salespeople with only one customer.

```
select *  from salepeople as s where snum=(select snum from customer
where snum=s.snum group by snum having count(snum)=1);
```

49. Write a query that joins the Customer table to itself to find all pairs of customers served by a single salesperson.

```
mysql> select a.cname,b.cname,a.snum,b.snum from customer as a,customer
as b where a.snum=b.snum and a.cname!=b.cname;
```

50. Write a query that will give you all orders for more than $1000.00

```
mysql> select * from orders where amt>1000.00;
```

51. Write a query that lists each order number followed by the name of the customer who made that order.

```
mysql> select o.onum,c.cname from orders as o,customer as c where
c.cnum=o.cnum;
```

52. Write 2 queries that select all salespeople (by name and number) who have customers in their cities who they do not service, one using a join and one a corelated subquery. Which solution is more elegant?

```
Select * from salepeople a,customer as b where a.snum!=b.snum and
a.city=b.city;
```

```
select * from salepeople a where city in(select city from customer where
b.snum!=a.snum and city=b.city);
```

53. Write a query that selects all customers whose ratings are equal to or greater than ANY (in the SQL sense) of Serres'?

```
mysql> select * from customer where rating>=(select min(c.rating) from
customer as c inner join salepeople as s on s.snum=c.snum and
s.sname="Serres"group by c.snum);
```

54. Write 2 queries that will produce all orders taken on October 3 or October 4.

```
mysql> select onum,amt,odate,cnum from orders where odate='1996-03-10' or
odate='1996-04-10';
```

```
mysql> select * from orders o where odate=(select odate from orders where
odate=o.odate group by odate having odate="1996-03-10" or odate= "1996-
04-10");
```

55. Write a query that produces all pairs of orders by a given customer. Name that customer and eliminate duplicates.

```
mysql> select m.onum,n.onum,m.cnum,n.cnum ,c.cname from orders as
m,orders as n,customer as c  where m.cnum=n.cnum and m.cnum=c.cnum  and
m.onum!=n.onum;
```

56. Find only those customers whose ratings are higher than every customer in Rome.

```
mysql> select * from customer where rating>(select max(rating) from
customer group by city having city="Rome");
```

57. Write a query on the Customers table whose output will exclude all customers with a rating <= 100.00, unless they are located in Rome.

```
mysql> select * from customer where rating>=100 and city!="Rome";
```

58. Find all rows from the Customers table for which the salesperson number is 1001.

```
mysql> select * from customer where snum=1001;
```

59. Find the total amount in Orders for each salesperson for whom this total is greater than the amount of the largest order in the table.

```
mysql> select c.snum,sum(o.amt) as total from orders as o,customer as c
where c.cnum=o.cnum group by c.snum having  total>(select max(amt) from
orders);
```

60. Write a query that selects all orders save those with zeroes or NULLs in the amount field.

```
mysql> select * from orders where amt=00.00 or amt is null;
```

61. Produce all combinations of salespeople and customer names such that the former precedes the latter alphabetically, and the latter has a rating of less than 200.

```
mysql> select s.snum,s.sname,c.cname ,c.rating from customer as
c,salepeople as s where c.snum=s.snum  and c.rating<200 order by sname;
```

62. List all Salespeople’s names and the Commission they have earned.

```
mysql> select sname,comm from salepeople;
```

63. Write a query that produces the names and cities of all customers with the same rating as Hoffman. Write the query using Hoffman’s CNUM rather than his rating, so that it would still be usable if his rating changed.

```
mysql> select * from customer where rating=(select rating from customer
where cnum=(select cnum from customer where cname="Hoffman"));
```

```
select * from customer where rating=(select rating from customer where
cname=”Hoffman”) and cname!=”Hoffman”;
```

64. Find all salespeople for whom there are customers that follow them in alphabetical order.

```
mysql> select s.snum,s.sname,c.cname from salepeople as s,customer as c
where s.snum=c.snum  and substring(s.sname,1,1)< substring(c.cname,1,1);
```

65. Write a query that produces the names and ratings of all customers of all who have above

average orders.

```
mysql> select c.cname,c.rating,o.amt from customer as c,orders as o where
c.cnum=o.cnum and o.amt>(select avg(amt) from orders);
```

66. Find the SUM of all purchases from the Orders table.

```
mysql> select sum(amt)from orders;
```

67. Write a SELECT command that produces the order number, amount and date for all rows in
the order table.

```
mysql> select onum,amt,odate from orders;
```

68. Count the number of nonNULL rating fields in the Customers table (including repeats).

```
mysql> select count(*) from customer where rating=0 or rating  is not
null;
```

69. Write a query that gives the names of both the salesperson and the customer for each order
after the order number.

```
mysql> select o.onum,s.sname,c.cname,s.snum,c.cnum from orders as
o,salepeople as s,customer as c where s.snum=c.snum and c.cnum=o.cnum;
```

70. List the commissions of all salespeople servicing customers in London.

```
mysql> select s.sname,c.city from salepeople as s,customer as c where
s.snum=c.snum and c.city="London" ;
```

71. Write a query using ANY or ALL that will find all salespeople who have no customers located in
their city.

```
mysql> select all(s.sname),c.city from salepeople as s,customer as c
where s.snum=c.snum and c.city=s.city ;
```

72. Write a query using the EXISTS operator that selects all salespeople with customers located in
their cities who are not assigned to them.

```
mysql> select s.sname,c.cname,c.city from salepeople as s,customer as c
where exists(select s.sname,c.cname from salepeople as s,customer as c)
and s.snum!=c.snum and c.city=s.city ;
```

73. Write a query that selects all customers serviced by Peel or Motika. (Hint : The SNUM field
relates the two tables to one another.)

```
mysql> select s.sname,c.cnum,c.cname,c.city,c.rating from salepeople as
s,customer as c where s.snum=c.snum and (s.sname="Peel" or
s.sname="Motika") ;
```

74. Count the number of salespeople registering orders for each day. (If a salesperson has more
than one order on a given day, he or she should be counted only once.)

```
mysql> select count(distinct(c.snum)),c.snum,o.odate from customer as
c,orders as o where c.cnum=o.cnum  group  by c.snum, o.odate order by
o.odate;
```

75. Find all orders attributed to salespeople in London.

```
mysql> select s.snum,s.city,o.onum,o.odate from customer as c,orders as
o,salepeople as s where c.cnum=o.cnum and s.snum=c.snum and
s.city="London";
```

76. Find all orders by customers not located in the same cities as their salespeople.

```
mysql> select s.snum,s.city,o.onum,o.odate from customer as c,orders as
o,salepeople as s where c.cnum=o.cnum and s.snum=c.snum and
s.city!=c.city;
```

77. Find all salespeople who have customers with more than one current order.

```
mysql> select * from salepeople  s where snum=(select snum from customer
c where s.snum=snum and cnum=(select cnum from orders where c.cnum=cnum
group by cnum having count(cnum)>1));
```

78. Write a query that extracts from the Customers table every customer assigned to a
salesperson who currently has at least one other customer (besides the customer being
selected) with orders in the Orders table.

```
mysql> select * from customer where snum in (select b.snum from orders as
a,customer as b where a.cnum=b.cnum  group by b.snum having
count(distinct(a.cnum)>1));
```

79. Write a query that selects all customers whose names begin with ‘C’.

```
mysql> select * from customer  where cname like "C%";
```

80. Write a query on the Customers table that will find the highest rating in each city. Put the output
in this form : for the city (city) the highest rating is : (rating).

```
mysql> select city as city,max(rating)as highest_rating from customer
group by city;
concate:

select concat(“for the city”,city,”highest rating is ”,max(rating)) as
result from customer group by city;
```

81. Write a query that will produce the SNUM values of all salespeople with orders currently in the
Orders table (without any repeats).

```
mysql> select c.snum from orders as o,customer as c where c.cnum=o.cnum
group by c.snum;
```

82. Write a query that lists customers in descending order of rating. Output the rating field first,
followed by the customer’s names and numbers.

```
mysql> select rating, cname from customer order by rating desc;
```

83. Find the average commission for salespeople in London.

```
mysql> select avg (o.amt*s.comm),c.snum,s.sname,s.city from orders
o,customer as c,salepeople as s  where s.snum=c.snum and c.cnum=o.cnum
group by c.snum having s.city="London";
```

84. Find all orders credited to the same salesperson who services Hoffman (CNUM 2001).

```
mysql> select o.onum,c.cnum,c.cname,c.snum from customer as c,orders as o
where c.cnum=o.cnum and c.snum=(select snum from customer where
cname="Hoffman");
```

85. Find all salespeople whose commission is in between 0.10 and 0.12 (both inclusive).

```
mysql> select * from salepeople where comm<= .12 and comm>=.10;
or
mysql> select * from salepeople where comm between .10 and .12 ;
```

86. Write a query that will give you the names and cities of all salespeople in London with a
commission above 0.10.

```
mysql> select sname,city,comm from salepeople where comm>0.10;
```

87. What will be the output from the following query?

```
SELECT * FROM ORDERS where (amt < 1000 OR NOT (odate = 10/03/1996 AND
cnum >
2003));
```

retrive the order table where amount is than 1000 or order is not placed on 10/03/1996 and not the customer id > 2003.

88. Write a query that selects each customer’s smallest order.

```
mysql> select min(amt),cnum from orders group by cnum;
```

89. Write a query that selects the first customer in alphabetical order whose name begins with G.

```
mysql> select *  from customer where cname like "G%" order by cname;
```

90. Write a query that counts the number of different nonNULL city values in the Customers table.

```
mysql> select count(city),city from customer where city is not null group
by city  ;
```

91. Find the average amount from the Orders table.

```
mysql> select avg(amt) from orders;
```

92. What would be the output from the following query?

```
SELECT * FROM ORDERS
WHERE NOT (odate = 10/03/96 OR snum > 1006) AND amt >=
1500);
```

```
correct>-mysql> SELECT onum,odate,amt,snum FROM orders,customer WHERE
orders.cnum=customer.cnum and  not (odate ="1996-03-10" and customer.snum
> 1006) AND amt >=1500;
```

place the orders where order date is not 10/03/96 or salesperson number is not greater than 1006 and order will be greater than 1500.

93. Find all customers who are not located in San Jose and whose rating is above 200.

```
mysql> select * from customer  where city!="San jose"and rating>200 ;
```

94. Give a simpler way to write this query :

```
SELECT snum, sname city, comm FROM salespeople
WHERE (comm > + 0.12 OR comm < 0.14);
+------+---------+------+
| snum | city    | comm |
+------+---------+------+
| 1001 | Peel    | 0.12 |
| 1002 | Serres  | 0.13 |
| 1003 | AxelRod | 0.10 |
| 1004 | Motika  | 0.11 |
| 1005 | Fran    | 0.26 |
| 1007 | Rifkin  | 0.15 |
+------+---------+------+
```

give those salespeople details where comm is greater than .12 or less than .14;

95. Evaluate the following query :

```
SELECT * FROM orders
WHERE NOT ((odate = 10/03/96 AND snum > 1002) OR amt > 2000.00);
```

retrive the order table when order is not placed on 10/03/1996 and not the salespeople id > 1002 or amount is greater than 2000.00.

96. Which salespersons attend to customers not in the city they have been assigned to?

```
mysql> select s.snum,s.sname,s.city,c.cnum,c.cname,c.city from customer
as c,salepeople as s where s.snum=c.snum and s.city!=c.city;
```

97. Which salespeople get commission greater than 0.11 are serving customers rated less than
250?

```
mysql> select s.snum,s.sname,s.comm,c.cnum,c.cname,c.rating from customer
as c,salepeople as s where s.snum=c.snum and  s.comm>0.11 and
c.rating<250;
```

98. Which salespeople have been assigned to the same city but get different commission
percentages?

```
mysql> select a.snum,a.sname,b.sname,a.city,b.city,a.comm,b.comm from
salepeople as a,  salepeople as b where a.city=b.city and a.comm!=b.comm
and a.sname!=b.sname;
```

99. Which salesperson has earned the most by way of commission?

```
mysql> select sum(o.amt*s.comm) as commision,c.snum,s.sname,s.city from
orders o,customer as c,salepeople as s  where s.snum=c.snum and
c.cnum=o.cnum group by c.snum order by commision desc limit 1;
```

100.Does the customer who has placed the maximum number of orders have the maximum rating?

```
mysql> select count(o.cnum) as value,c.cnum,c.rating from orders as o,
customer as c where c.cnum=o.cnum and rating=(select max(rating) from
customer) group by o.cnum having value=(select count(*) from orders group
by cnum order by count(*) desc limit 1) ;
```

101.Has the customer who has spent the largest amount of money been given the highest rating?

```
mysql> select a.cnum as cus_a,b.cnum as cus_b from customer as a,customer
b  where a.cnum=(select cnum from orders where amt=(select max(amt) from
orders)) and b.cnum=(select cnum from customer c  where b.cnum=c.cnum and
rating=(select max(rating) from customer))and a.cnum=b.cnum;
```

102.List all customers in descending order of customer rating.

```
mysql> select * from customer order by rating desc;
```

103.On which days has Hoffman placed orders?

```
mysql> select c.cname,c.cnum ,o.odate from customer as c,orders as o
where o.cnum=c.cnum and c.cname="Hoffman";
```

104.Do all salespeople have different commissions?

```
mysql> select snum,sname,count(comm) from salepeople group by comm ;
```

yes.

105.Which salespeople have no orders between 10/03/1996 and 10/05/1996?

```
mysql> select c.snum,s.sname,count(o.onum),o.odate from salepeople as
s,customer as c,orders as o where s.snum=c.snum and c.cnum=o.cnum and
o.odate not  between "1996-03-10" and "1996-05-10" group by c.snum;
```

106.How many salespersons have succeeded in getting orders?

```
mysql> select c.snum,s.sname,count(o.onum) from salepeople as s,customer
as c,orders as o where s.snum=c.snum and c.cnum=o.cnum  group by c.snum;
```

107.How many customers have placed orders?

```
mysql> select count(distinct(cnum)) from orders ;
```

108.On which date has each salesperson booked an order of maximum value?

```
mysql> Select  a.odate,a.amt, m.sid,m.name from orders as a,(select
s.snum as sid,s.sname as name,max(o.amt)  as value from salepeople as
s,customer as c,orders as o where s.snum=c.snum and c.cnum=o.cnum  group
by c.snum)as m where m.value =a.amt ;
```

109.Who is the most successful salesperson?

```
mysql> select cu.snum,s.sname,count(o.onum),max(o.amt),o.odate from
salepeople as s,customer as cu,orders as o where s.snum=cu.snum and
cu.cnum=o.cnum group by cu.snum order by count(o.onum) desc limit 1;
```

110.Who is the worst customer with respect to the company?

```
mysql> select cu.cnum,cu.cname,count(o.onum) from customer as cu,orders
as o where  cu.cnum=o.cnum group by o.cnum order by count(o.cnum) limit
4;
```

111.Are all customers not having placed orders greater than 200 totally been serviced by
salespersons Peel or Serres?

```
mysql> select cu.cnum,cu.cname,count(o.onum),sum(o.amt),s.sname from
customer as cu,orders as o,salepeople as s where  cu.cnum=o.cnum  and
cu.snum=s.snum and (s.sname="peel" or s.sname="Serres") group by o.cnum
having sum(o.amt)<200;
Empty set (0.00 sec)
```

112.Which customers have the same rating?

```
mysql> select c.cname,b.cname from customer as c,customer as b where
c.cnum!=b.cnum  and c.rating=b.rating group by c.rating;
```

113.Find all orders greater than the average for October 4th.

```
mysql> select * from orders where amt>(select avg(amt) from orders where
odate="1996-04-10");
```

114.Which customers have above average orders?

```
mysql> select o.cnum from orders as o where  o.amt>(select avg(amt) from
orders where odate="1996-04-10");
```

115.List all customers with ratings above San Jose’s average.

```
mysql> select * from customer where rating>(select avg(rating) from
customer where city="san jose");
```

116.Select the total amount in orders for each salesperson for whom the total is greater than the
amount of the largest order in the table.

```
mysql>  select
cu.snum,s.sname,count(o.onum),sum(o.amt),max(o.amt),o.odate from
salepeople as s,customer as cu,orders as o where s.snum=cu.snum and
cu.cnum=o.cnum group by cu.snum having sum(o.amt)>(select max(amt) from
orders);
```

117.Give names and numbers of all salespersons who have more than one customer.

```
mysql> select s.snum,s.sname ,count(c.snum) as value from customer as c ,
salepeople as s where  c.snum=s.snum group by c.snum having value>1;
```

118.Select all salespersons by name and number who have customers in their city whom they
don’t service.

```
mysql> select s.snum,s.sname,s.city,c.cnum,c.cname, c.city as value from
customer as c , salepeople as s where  c.snum!=s.snum and s.city=c.city;
```

119.Which customers’ rating should be lowered?

```
select c.cname,c.rating,count(o.onum) from customer as c,orders as o
where o.cnum=c.cnum group by o.cnum having count(o.onum)=(select
count(onum) from orders group by cnum order by count(onum) limit 1) and
c.rating=(select min(rating ) from customer);
```

120.Is there a case for assigning a salesperson to Berlin?

```
mysql> select c.sname,c.city,
b.cname,b.city,b.rating,sum(a.amt),round(sum(a.amt)*100/(select sum(amt)
from orders),2) as pc from orders as a,customer as b,salepeople as c
where  a.cnum=b.cnum and b.snum=c.snum and b.city="Berlin" group by
a.cnum;
```

121.Is there any evidence linking the performance of a salesperson to the commission that he or
she is being paid?

```
mysql> select s.sname,sum(o.amt),s.comm as commision from salepeople as
s,customer as c ,orders as  o where c.snum=s.snum and c.cnum=o.cnum group
by c.snum;
```

122.Does the total amount in orders by customer in Rome and London exceed the commission
paid to salespersons in London and New York by more than 5 times?

```
mysql> select sum(a.amt) as sales ,(SELECT (sum(o.amt)*s.comm)*5 from
orders as o, customer as c,salepeople as s  where c.cnum=o.cnum  and
c.snum=s.snum and (s.city="London" or s.city="New work") )as commision
from orders as a,customer as b where a.cnum=b.cnum and (b.city="London"
or b.city="Rome");
```

123.Which is the date, order number, amt and city for each salesperson (by name) for the
maximum order he has obtained?

```
mysql> select a.odate,a.onum, a.amt,m.sid,m.name from orders as a,(select
max(a.amt) as maxamt,b.snum as sid,c.sname as name from orders a,customer
b,salepeople as c where a.cnum=b.cnum  and b.snum=c.snum group by b.snum)
m where a.amt=maxamt order by m.sid;
```

124.Which salesperson(s) should be fired?

```
mysql> select s.snum,c.cnum from salepeople as s left outer join customer
as c on s.snum=c.snum where cnum is null;
```

125.What is the total income for the company?

```
Select sum(amt) from orders ;
```