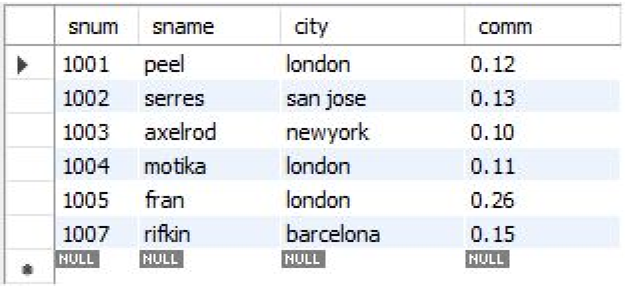
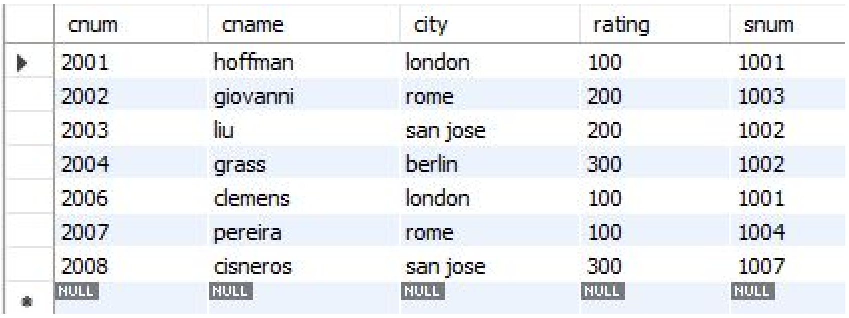
Salespeople



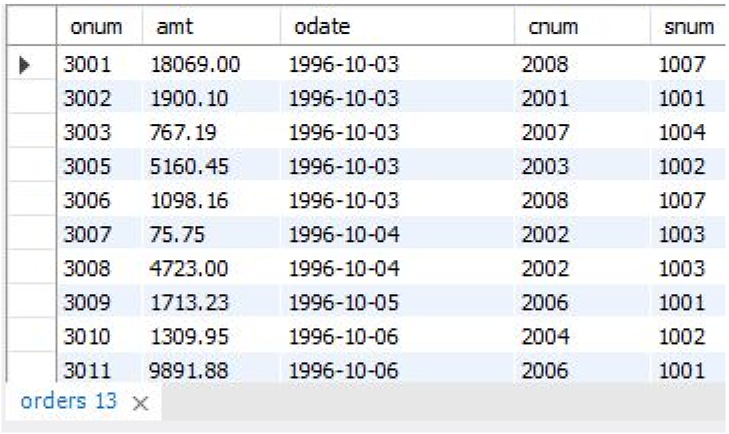
snum	sname	city	comm
1001	peel	london	0.12
1002	serres	san jose	0.13
1003	axelrod	newyork	0.10
1004	motika	london	0.11
1005	fran	london	0.26
1007	rifkin	barcelona	0.15
NULL	NULL	NULL	NULL

customer



cnum	cname	city	rating	snum
2001	hoffman	london	100	1001
2002	giovanni	rome	200	1003
2003	liu	san jose	200	1002
2004	grass	berlin	300	1002
2006	clemens	london	100	1001
2007	pereira	rome	100	1004
2008	cisneros	san jose	300	1007
NULL	NULL	NULL	NULL	NULL

Order



onum	amt	odate	cnum	snum
3001	18069.00	1996-10-03	2008	1007
3002	1900.10	1996-10-03	2001	1001
3003	767.19	1996-10-03	2007	1004
3005	5160.45	1996-10-03	2003	1002
3006	1098.16	1996-10-03	2008	1007
3007	75.75	1996-10-04	2002	1003
3008	4723.00	1996-10-04	2002	1003
3009	1713.23	1996-10-05	2006	1001
3010	1309.95	1996-10-06	2004	1002
3011	9891.88	1996-10-06	2006	1001

1.List all the columns of the Salespeople table.

---desc salespeople;

2. List all customers with a rating of 100.

---SELECT * FROM customers where rating=100;

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

---SELECT * FROM customers where city="null";

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

---select* from (select month(odate),amt,snum from orders) as timest order by snum;

5. Arrange the Orders table by descending customer number.

---SELECT * FROM vita_acts_ass1.orders order by cnum desc;

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

```
---select distinct snum from orders;
```

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

```
---select cname,sname from customers joins salespeople using(snum);
```

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

```
---select snum,sname from vita_acts_ass1.salespeople

where snum in (select snum  from customers group by snum having count(snum)>1);
```

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

```
---select snum,count(snum)  as num_of_order from orders group by snum order by num_of_order 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.

```
---select * from customers where city="san jose";
```

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

```
---select cname,snum,city from customers where city in (select city from salespeople);
```

12. Find the largest order taken by each salesperson.

```
---select snum,count(onum),max(amt) from orders group by snum;
```

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

```
---select cnum,cname,city,rating from customers where city="san jose" and rating>200;
```

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

```
---select sname,comm from salespeople where city="london";
```

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

```
---select onum,snum from orders where snum in (select snum from salespeople where sname="motika");
```

16. Find all customers with orders on October 3.

```
---select distinct cnum from orders where day(odate)=3;
```

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.

```
---select day(odate) as odt,sum(amt) as maxsum from orders group by odt having maxsum>max(amt)+2000;
```

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

```
---select onum from orders where amt>(select min(amt) from orders where day(odate)=6);
```

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

with a rating of 300.

```
---select snum,sname from salespeople where exists (select rating from customers where salespeople.snum=customers.snum and rating=300);
```

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

```
---select cname,rating from customers

 where rating in(select distinct rating from customers group by rating  having count(rating)>1)

 order by rating;
```

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

```
---select cnum from customers where  cnum>(select (snum+1000) from salespeople where sname="serres");
```

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

```
---select sname,format(comm*100,'P2') as percent from salespeople;
```

23.Find the largest order taken by each salesperson on each date, eliminating those MAX orders

which are less than $3000.00 in value.

```
---select snum,odate,max(amt)as b from orders group by snum,odate having b>3000;
```

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

```
---select snum,max(amt),onum from orders where day(odate)=3 group by snum;
```

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

```
---select cnum,cname from customers

 where  snum in (select snum from salespeople where sname="serres")

 and city in(select city from salespeople where sname="serres")  ;
```

26. Select all customers with a rating above 200.00.

```
--- select cnum,cname,city,rating from customers where rating>200;
```

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

```
---select count(distinct snum) from orders;
```

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.

```
---select cname,comm from salespeople join customers using(snum) where format(comm*100,'P')>12;
```

29. Find salespeople who have multiple customers.

```
---select snum,count(snum) from customers group by snum having count(snum)>1;
```

30. Find salespeople with customers located in their city.

```
---select cnum,cname,city from customers where city in(select city from salespeople);
```

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

```
---select* from salespeople where sname like "p__l";
```

32. Write a query that uses a subquery to obtain all orders for the customer named Cisneros.

Assume you do not know his customer number.

```
---select onum from orders where cnum in (select cnum from customers where cname="cisneros");
```

33. Find the largest orders for Serres and Rifkin

```
--select sname,max(amt) from orders join salespeople using(snum) group by snum having snum=(select snum from salespeople where sname="serres") or snum=(select snum from salespeople where sname="rifkin");
```

```
--- select snum,max(amt) from orders group by snum

 having snum in (select snum from salespeople where sname="serres" or sname="rifkin");
```

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

```
---select snum,sname,comm,city from salespeople;
```

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

```
---select cname from customers where cname  between "a" and "g";
```

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

```
---select * from customers join (select distinct snum from customers) as dt using(snum);
```

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

```
---select onum,amt from orders where amt>(select avg(amt) from orders where day(odate)=4);
```

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.

```
---select cnum,cname,max(rating) from customers group by city;
```

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

```
---select odate,count(odate) from orders group by odate order by count(odate) desc;
```

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

San Jose.

```
---select cnum,cname,city,max(rating) from customers group by city;
```

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

```
---select onum,amt from orders

 where cnum in (select cnum from customers where city="san jose");
```

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

```
---select onum,amt from orders where amt>(select avg(amt) from orders);
```

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

```
---select max(rating),city from customers 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.

```
---select rating,onum,ac,snum,cnum from(

 select amt*comm as ac,onum,snum,cnum from (select amt,cnum,snum,onum,comm

 from orders join salespeople using(snum)) as t order by snum) as dt

join customers using(cnum,snum) where rating>100 order by snum;
```

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

```
---select count(cnum) from customers where rating>(select avg(rating) from customers where city="san jose");
```

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

```
---select a.sname,b.sname from salespeople a,salespeople b order by b.sname desc;
```

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

```
---select snum,sname,city from  salespeople where city="barcelona" or city="london";
```

48. Find all salespeople with only one customer.

```
---select snum,count(cnum) as custcount from customers group by snum having custcount=1;
```

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

```
---select cnum,count(cnum) as sellcount from customers group by snum having sellcount=1;
```

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

```
---select onum,amt from orders where amt>1000;
```

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

```
---select onum,cname from orders join customers using(cnum) order by onum;
```

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 snum from salespeople where city in(

select city from salespeople join customers using(city,snum))

and snum not in (select snum from salespeople join customers using(city,snum));

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

---select * from customers where rating>= any

(select rating from salespeople join customers using(snum) where sname="serres");

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

---select onum,day(odate) from orders where day(odate)=3 or day(odate)=4;

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

---select cnum,amt from (select distinct cnum from orders) as c join

(select amt from orders) as a order by cnum;

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

---select cnum,rating from customers where rating>(select max(rating) from customers where 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.

---select * from customers where   city="rome" or not rating<=100;

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

---select * from customers 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.

---select snum,sum(amt) from orders group by snum having sum(amt)>(select max(amt) from orders);

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

---select * from orders where amt=0 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.

---select sname,rating,cname from salespeople join customers   where rating<200;

62. List all Salespeople’s names and the Commission they have earned.(comm can be calculated how many oders placed by customers with salespeople)

```
---select snum,sname,comm*sa from salespeople

 join (select snum,sum(amt) as sa from orders group by snum) as t using(snum);
```

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.(hoffmans rating 100)

```
---select cname,city,rating from customers where

rating=(select rating from customers where cnum=(select cnum from customers where cname="hoffman"));
```

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

```
--select sname,cname from salespeople join customers using(snum) where
salespeople.sname<customers.cname;(simplest)

--select sname,snum from salespeople where sname<any( select cname from customers where
salespeople.snum=customers.snum);

--select sname,snum from salespeople where exists( select cname from customers

where salespeople.snum=customers.snum and salespeople.sname<customers.cname);
```

65. Write a query that produces the names and ratings of all customers of all who have above
average orders.(avg amount is around 2699)

```
--select cname,rating,amt from customers join orders using(cnum) where amt>(select avg(amt) from orders);

---select cname,cnum,amt from customers join orders using(cnum) where amt>(

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

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

```
---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.

```
--- select onum,cnum,amt from orders;
```

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

```
---select count(rating) from customers;
```

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

after the order number.

```
---select sname,cname,onum from salespeople join customers using(snum) join orders using(snum,cnum);
```

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

```
---select snum,comm from salespeople where snum in(select snum from customers where city="london");
```

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

```
---select sname from salespeople where snum = any( select snum from customers

 where salespeople.city != customers.city and  salespeople.snum = customers.snum);
```

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

```
---select sname from salespeople where exists (select cname,sname from customers join salespeople

 where salespeople.city=customers.city and salespeople.snum<>customers.snum);
```

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

```
---select * from customers

 where snum in(select snum from salespeople where sname="peel" or 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.)

```
--- select odate,count(distinct snum) from orders group by odate;
```

75. Find all orders attributed to salespeople in London.

```
---select onum,amt,snum from orders

 where snum in ( select snum from salespeople where city="london");
```

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

```
---select * from orders where cnum=any

 (select cnum from customers join salespeople using(snum) where salespeople.city<>customers.city);
```

77. Find all salespeople who have customers with more than one current order.(cnum 2002,2006,20008 has more than 1 order in ordertable)

```
---select snum,count(cnum)as cc from orders group by snum having cc>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.

```
---select cnum,snum from customers where snum in

 (select snum from customers group by snum having count(snum)>1 and snum in(select snum from orders));
```

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

---select * from customers 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*).

---select city,max(rating) from customers 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).

---select distinct snum from orders;

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.

---select rating,cnum,cname from customers order by rating;

83. Find the average commission for salespeople in London.

---select avg(comm) from salespeople where city="london";

84. Find all orders credited to the same salesperson who services Hoffman (CNUM 2001).and(snum=1001)

--- select onum,amt from orders where snum in (select snum from customers where cnum=2001);

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

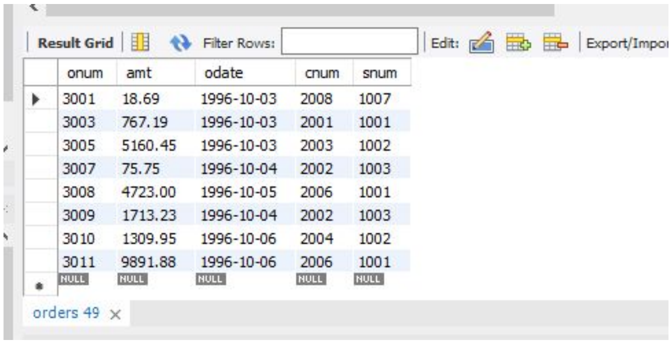
---select * from salespeople where comm between 0.10 and 0.12;

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

---select * from salespeople where city="london" and comm>0.1;

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));
```



	onum	amt	odate	cnum	snum
▶	3001	18.69	1996-10-03	2008	1007
	3003	767.19	1996-10-03	2001	1001
	3005	5160.45	1996-10-03	2003	1002
	3007	75.75	1996-10-04	2002	1003
	3008	4723.00	1996-10-05	2006	1001
	3009	1713.23	1996-10-04	2002	1003
	3010	1309.95	1996-10-06	2004	1002
	3011	9891.88	1996-10-06	2006	1001
●	NULL	NULL	NULL	NULL	NULL

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

---select cnum,onum,min(amt) from orders group by cnum;

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

---select cname from customers where cname like "g%" order by cname;

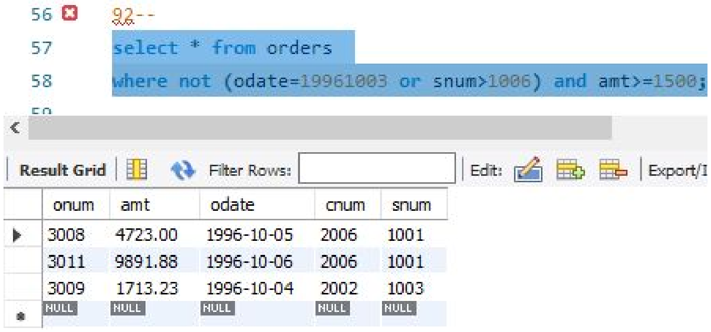
90. Write a query that counts the number of different nonNULL city values in the Customers table.

---select count(distinct city) from customers;

91. Find the average amount from the Orders table.

---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);



```
92--
select * from orders
where not (odate=19961003 or snum>1006) and amt>=1500;
```

onum	amt	odate	cnum	snum
3008	4723.00	1996-10-05	2006	1001
3011	9891.88	1996-10-06	2006	1001
3009	1713.23	1996-10-04	2002	1003
NULL	NULL	NULL	NULL	NULL

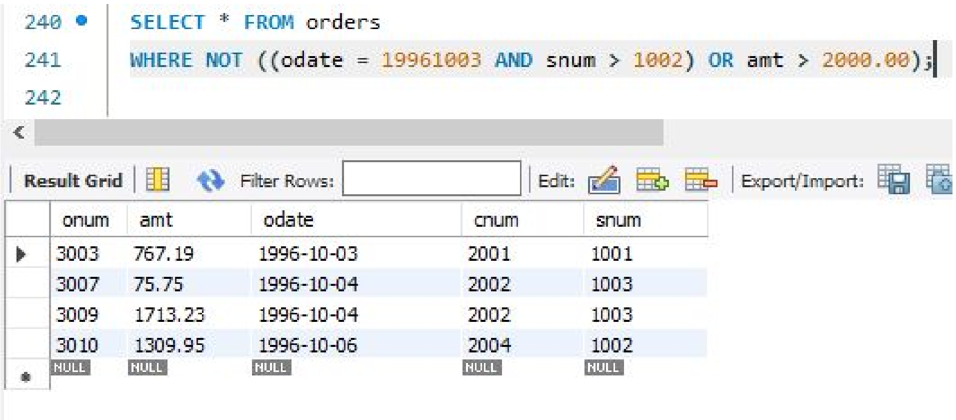
93. Find all customers who are not located in San Jose and whose rating is above 200.

---select * from customers 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);

---select * from salespeople;

95. Evaluate the following query :
SELECT * FROM orders
WHERE NOT ((odate = 10/03/96 AND snum > 1002) OR amt > 2000.00);



```
SELECT * FROM orders
WHERE NOT ((odate = 19961003 AND snum > 1002) OR amt > 2000.00);
```

onum	amt	odate	cnum	snum
3003	767.19	1996-10-03	2001	1001
3007	75.75	1996-10-04	2002	1003
3009	1713.23	1996-10-04	2002	1003
3010	1309.95	1996-10-06	2004	1002
NULL	NULL	NULL	NULL	NULL

96. Which salespersons attend to customers not in the city they have been assigned to?
---select snum,sname from salespeople where snum not in(select snum from customers join salespeople using(snum,city));

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

```
---select snum,rating from salespeople join customers using(snum) where comm>0.11 and rating<250;
```

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

```
---select snum,city,comm from salespeople where city in
 (select city from salespeople group by city having count(distinct comm)>1);
```

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

```
---select snum,sname,max(comm*sa) from salespeople
 join (select snum,sum(amt) as sa from orders group by snum) as t using(snum);
```

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

```
---select cnum,tt,rating from customers join (select cnum,count(onum)as tt from orders group by cnum ) as tf

using(cnum) order by tt desc; // NO
```

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

```
---select cnum,max(tt), rating from customers

join (select cnum,sum(amt)as tt from orders group by cnum) as dt using(cnum); //NO
```

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

```
---select * from customers order by rating desc;
```

103.On which days has Hoffman placed orders?

```
---select cnum,odate from orders where cnum in

(select cnum from customers where cname="hoffman");
```

104.Do all salespeople have different commissions?

```
---select count(distinct snum),count(distinct comm) from salespeople;  //YES
```

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

```
---select snum from orders where odate not between 19961003 and 19961005;
```

106.How many salespersons have succeeded in getting orders?

```
--select COUNT(distinct snum) from orders;
```

107.How many customers have placed orders?

```
--select count(distinct cnum) from orders;
```

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

```
---select snum,max(amt),odate from orders group by snum;
```

109.Who is the most successful salesperson?

```
---select snum,sum(amt) from orders group by snum order by sum(amt) desc limit 1 ;
```

110.Who is the worst customer with respect to the company?(Hoffman 2001)

```
---select cnum,sum(amt) from orders group by cnum order by sum(amt) limit 1;
```

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

```
--select cnum,sum(amt) from orders where cnum=any(

select cnum from customers join salespeople using(snum) where sname="peel" or sname="serres")

 group by cnum having sum(amt)<200;
```

112.Which customers have the same rating?

```
---select * from customers where rating in

(select rating from customers group by rating having count(rating)>1) order by rating;
```

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

```
---select * from orders where amt>(select avg(amt) from orders group by odate having odate=19961004);
```

114.Which customers have above average orders?

```
---select cnum,sum(amt) as sa from orders group by cnum having sa>(select avg(amt) from orders);
```

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

```
---select* from customers where rating>

 (select avg(rating) from customers group by  city having 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.

```
---select sum(amt),snum from orders group by snum having sum(amt)>(select max(amt) from orders);
```

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

```
--- select snum,sname from salespeople where snum in

 (select snum from customers group by snum having count(cnum)>1);
```

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

```
--- select snum,sname from salespeople where snum not in   (select snum from salespeople join customers
using(snum,city)) and   snum in (select distinct snum from customers);
```

119.Which customers’ rating should be lowered?;

```
---select a.cnum,b.cname,b.rating,sum(a.amt) as totalspend from orders a,customers b where a.cnum=b.cnum group by
cnum order by totalspend desc;
```

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

```
---select count(snum) from customers group by city having city="berlin";
```

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

```
---select b.snum,c.sname,sum(a.amt) as totalsalesbySP,sum(amt)*c.comm from orders a,customers b,salespeople c

 where a.cnum=b.cnum and b.snum=c.snum group by b.snum   order by totalsalesbySP desc;
```

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?

```
---select (select sum(amt) from customers join orders using (cnum) where (city="rome" or city="london")) as a,

(select sum(a.amt*c.comm) as totalsales from orders a,customers b,salespeople c where a.cnum=b.cnum and
b.snum=c.snum and (c.city='London' or c.city='New York') )as b;
```

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

```
---select sname,snum,onum,ma,odate from salespeople join

(select onum,max(amt) as ma,snum,odate from orders group by snum) as dt using(snum);
```

124.Which salesperson(s) should be fired?

```
---select snum,sname from salespeople where snum not in (select distinct snum from customers);
```

125.What is the total income for the company?

```
---select sum(amt)-ct from orders join

 (select sum(comm*amt) as ct from salespeople join orders using(snum)) as ss ;
```