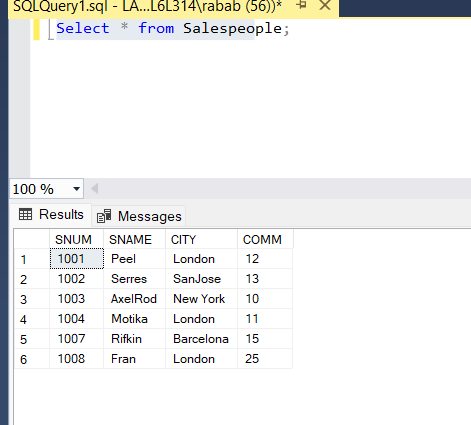
1) List all the columns of the Salespeople table.

**Query**- Select \* from Salespeople;

**Output-**

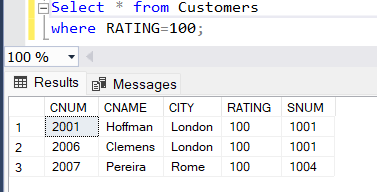


2) List all customers with a rating of 100.

**Query**- Select \* from Customers

where RATING=100;

**Output-**



3) Find the largest order taken by each salesperson on each date.

**Query**- Select count(o.SNUM) as Orderquantity,s.SNAME from Orders o

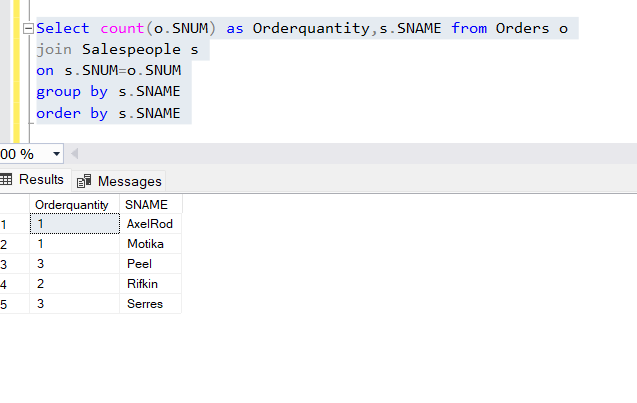
join Salespeople s

on s.SNUM=o.SNUM

group by s.SNAME

order by s.SNAME;

**Output-**

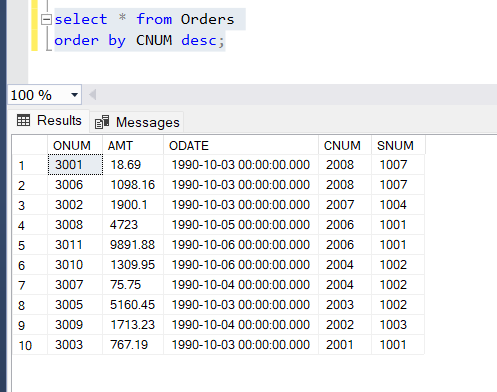


4) Arrange the Order table by descending customer number.

**Query**- select \* from Orders

order by CNUM desc;

**Output-**



5) Find which salespeople currently have orders in the order table

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

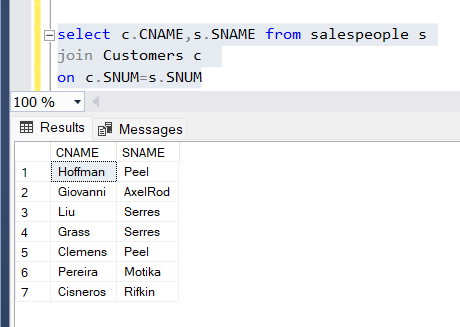
**Query**-

select c.CNAME,s.SNAME from salespeople s

join Customers c

on c.SNUM=s.SNUM

**Output-**



7) Find the names and numbers of all salespeople who have more than one customer.

**Query**-

select s.SNAME,s.SNUM,count(S.SNUM) AS Count from salespeople s

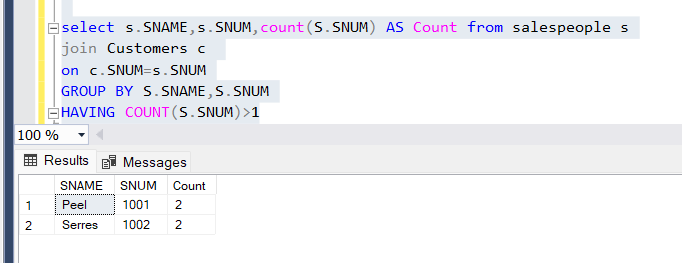
join Customers c

on c.SNUM=s.SNUM

GROUP BY S.SNAME,S.SNUM

HAVING COUNT(S.SNUM)>1

**Output-**



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

**Query**-

Select s.SNUM,s.SNAME,count(a.onum) as OrderQuantity from Orders a

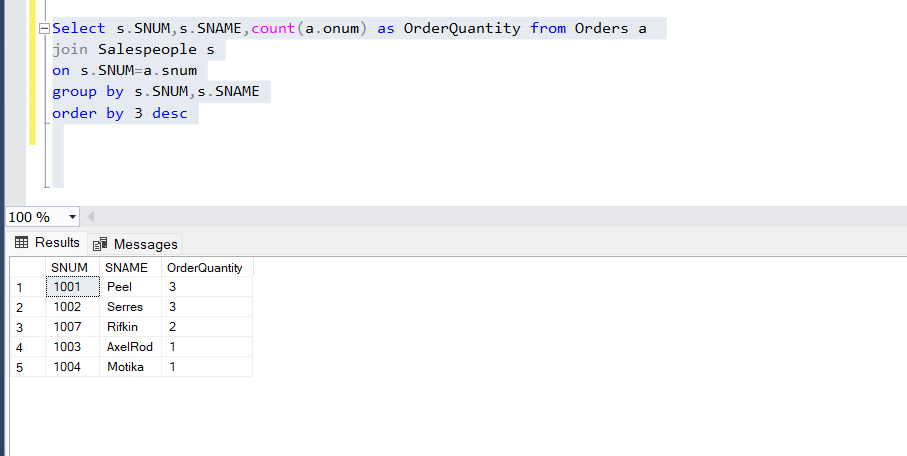
join Salespeople s

on s.SNUM=a.snum

group by s.SNUM,s.SNAME

order by 3 desc

**Output-**



9) List the customer table if and only if one or more of the customers in the Customer table are located in SanJose.

10) Match salespeople to customers according to what city they live in.

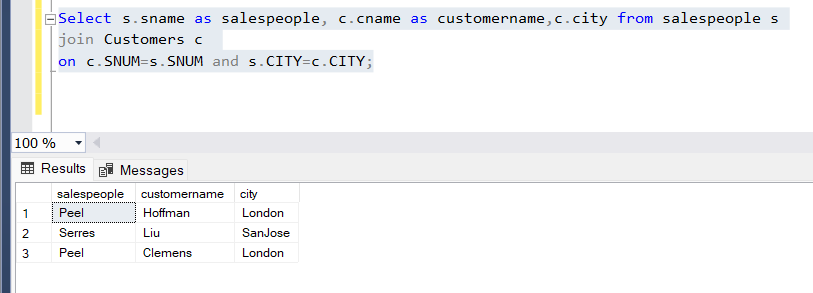
**Query**-

Select s.sname as salespeople, c.cname as customername,c.city from salespeople s

join Customers c

on c.SNUM=s.SNUM and s.CITY=c.CITY;

**Output-**



11) Find all the customers in SanJose who have a rating above 200.

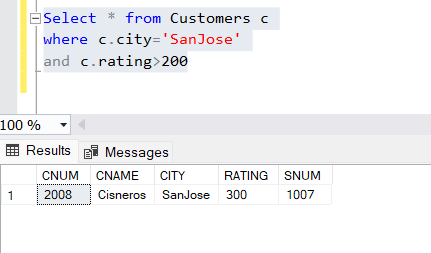
**Query**-

Select \* from Customers c

where c.city='SanJose'

and c.rating>200;

**Output-**



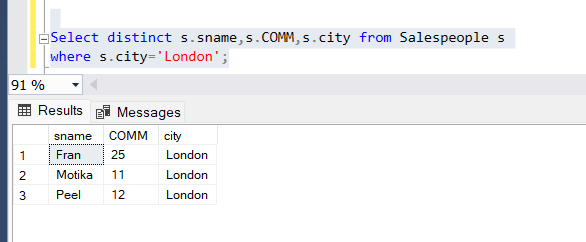
12) List the names and commissions of all salespeople in London.

**Query**-

Select distinct s.sname,s.COMM,s.city from Salespeople s

where s.city='London';

**Output-**



13) List all the orders of Salesperson Motika from the orders table.

**Query**-

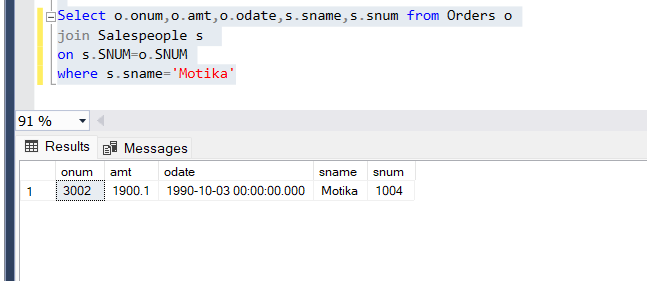
Select o.onum,o.amt,o.odate,s.sname,s.snum from Orders o

join Salespeople s

on s.SNUM=o.SNUM

where s.sname='Motika';

**Output-**



14) Find all customers who booked orders on October 3.

**Query**-

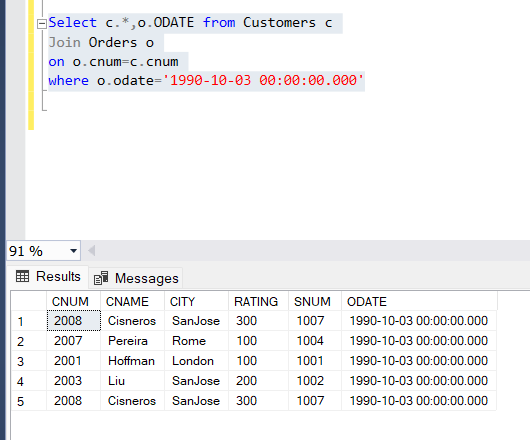
Select c.\*,o.ODATE from Customers c

Join Orders o

on o.cnum=c.cnum

where o.odate='1990-10-03 00:00:00.000'

**Output-**



15) 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 above

the maximum Amount.

**Query**-

select sum(o.AMT) as Amount,o.ODATE from Orders o

group by o.ODATE

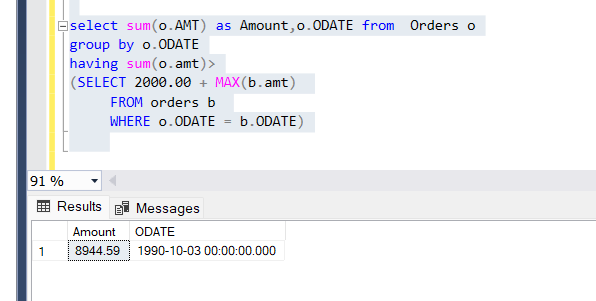
having sum(o.amt)>

(SELECT 2000.00 + MAX(b.amt)

FROM orders b

WHERE o.ODATE = b.ODATE);

**Output-**

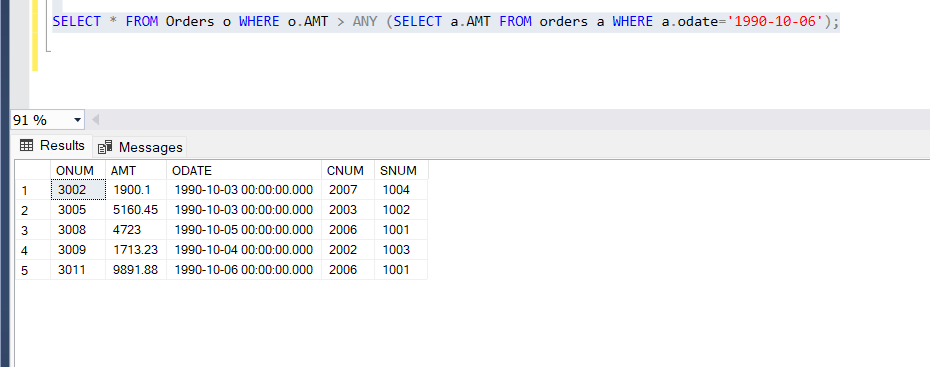


16) Select all orders that had amounts that were greater than at least one of the orders from October 6.

**Query**-

SELECT \* FROM Orders o WHERE o.AMT > ANY (SELECT a.AMT FROM orders a WHERE a.odate='1990-10-06');

**Output-**

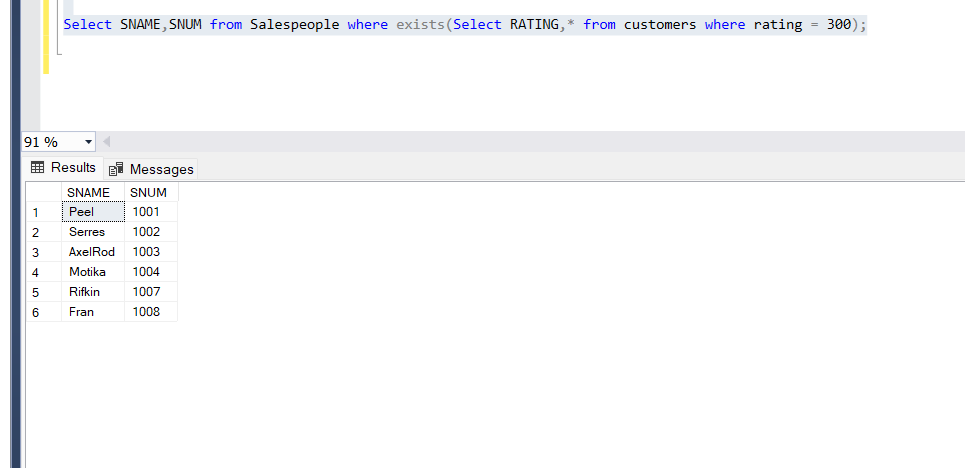
.

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

**Query**-

Select SNAME,SNUM from Salespeople where exists(Select RATING,\* from customers where rating = 300);

**Output-**



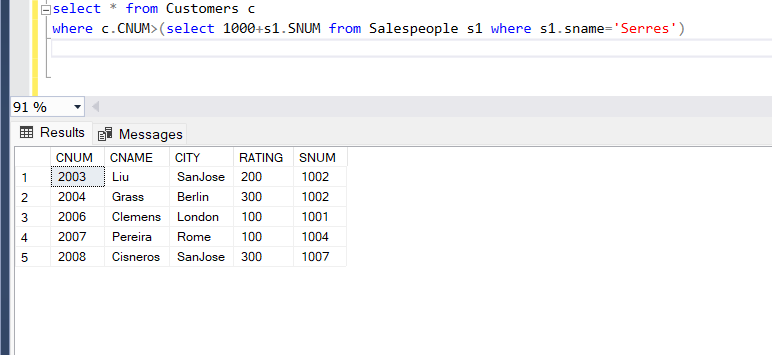
18) Find all customers whose cnum is 1000 above the snum of Serres.

**Query**-

select \* from Customers c

where c.CNUM>(select 1000+s1.SNUM from Salespeople s1 where s1.sname='Serres');

**Output-**

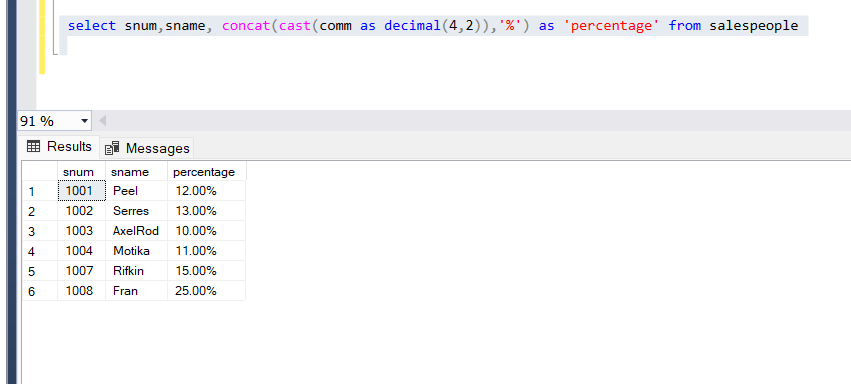


19) Give the salespeople’s commissions as percentages instead of decimal numbers.

**Query-**

select snum,sname, concat(cast(comm as decimal(4,2)),'%') as 'percentage' from salespeople

**Output-**



20) Find the largest order taken by each salesperson on each date, eliminating those Maximum orders, which are less than 3000.

**Query-**

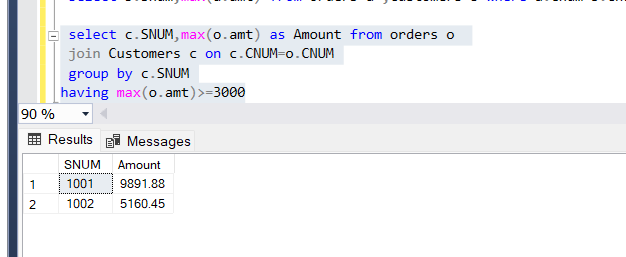
select c.SNUM,max(o.amt) as Amount from orders o

join Customers c on c.CNUM=o.CNUM

group by c.SNUM

having max(o.amt)>=3000

**Output-**



21) List all the largest orders for October 3, for each salesperson.

**Query-**

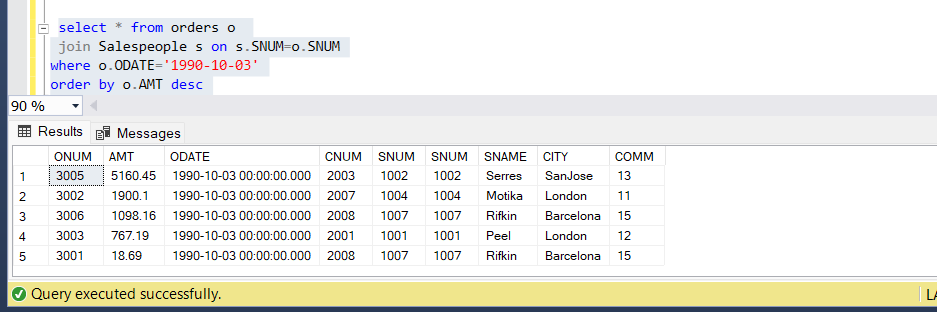
select \* from orders o

join Salespeople s on s.SNUM=o.SNUM

where o.ODATE='1990-10-03'

order by o.AMT desc;

**Output-**



22) Find all customers located in cities where Serres has customers.

**Query-**

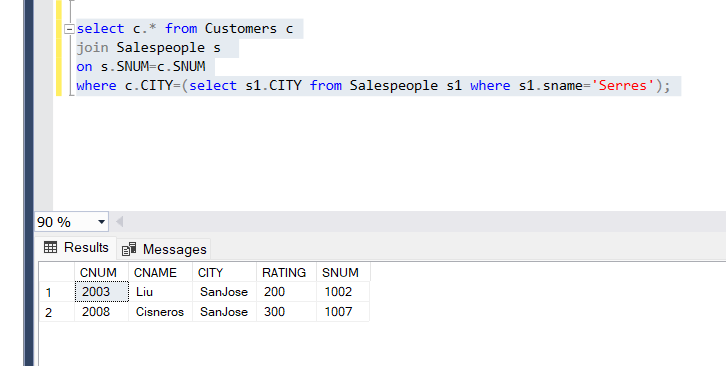
select c.\* from Customers c

join Salespeople s

on s.SNUM=c.SNUM

where c.CITY=(select s1.CITY from Salespeople s1 where s1.sname='Serres');

**Output-**



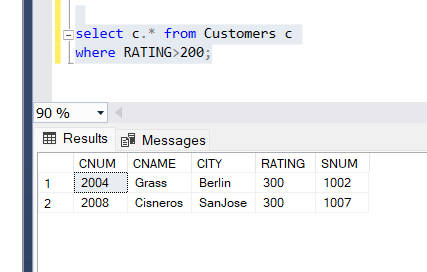
23) Select all customers with a rating above 200.

**Query-**

select c.\* from Customers c

where RATING>200;

**Output-**



24) Count the number of salespeople currently having orders in the orders table.

**Query-**

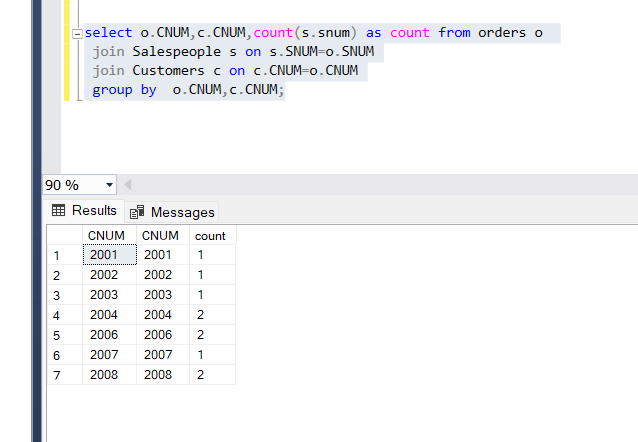
select o.CNUM,c.CNUM,count(s.snum) as count from orders o

join Salespeople s on s.SNUM=o.SNUM

join Customers c on c.CNUM=o.CNUM

group by o.CNUM,c.CNUM;

**Output-**



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

**Query-**

Select c.CNAME as Customers,s.SNAME as Salesman,

s.COMM as Commision

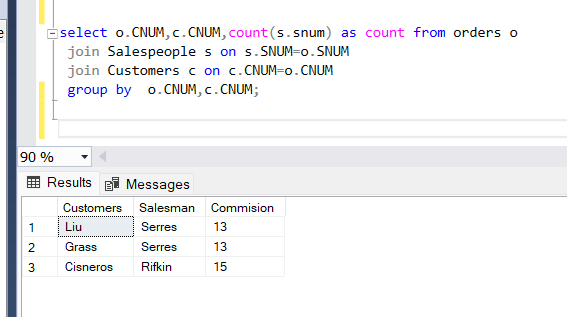
from Customers c

join Salespeople s

on s.SNUM=c.SNUM

where s.COMM>12;

**Output-**



26) Find salespeople who have multiple customers.

**Query-**

SELECT \*

FROM Salespeople s

WHERE s.SNUM IN (

SELECT DISTINCT c.SNUM

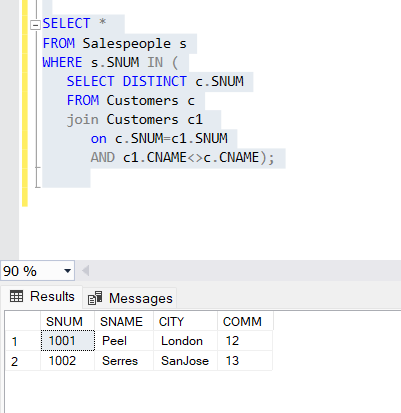
FROM Customers c

join Customers c1

on c.SNUM=c1.SNUM

AND c1.CNAME<>c.CNAME);

**Output-**

****

27) Find salespeople with customers located in their own cities.

**Query-**

Select s.SNUM,s.SNAME as Salespeoplename,c.CNAME,s.CITY,c.CITY

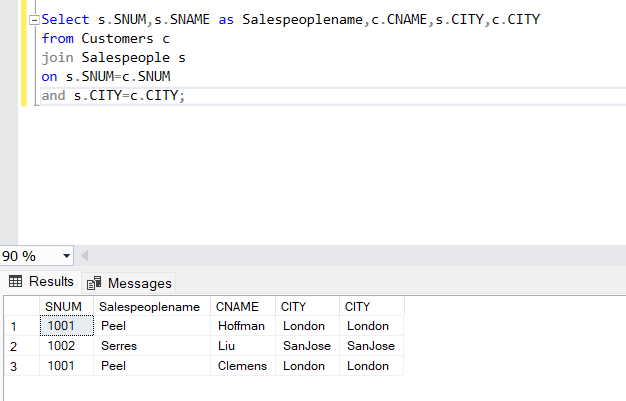
from Customers c

join Salespeople s

on s.SNUM=c.SNUM

and s.CITY=c.CITY;

**Output-**



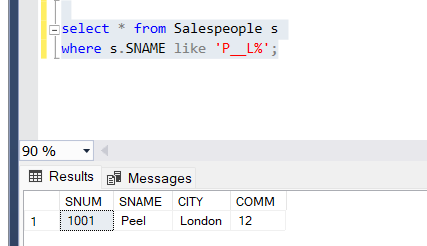
28) Find all salespeople whose name starts with ‘P’ and fourth character is ‘I’.

**Query-**

select \* from Salespeople s

where s.SNAME like 'P\_\_L%';

**Output-**



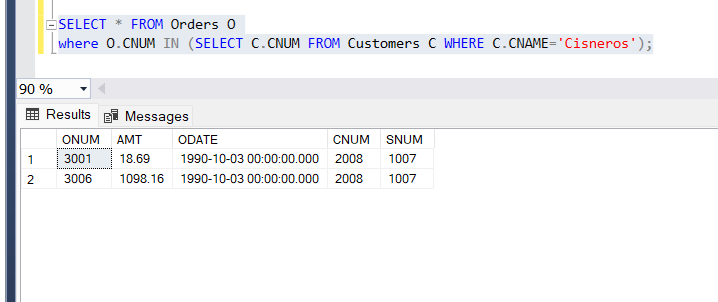
29) Write a query that uses a subquery to obtain all orders for the customer named ‘Cisneros’. Assume you do not know his customer number.

**Query-**

SELECT \* FROM Orders O

where O.CNUM IN (SELECT C.CNUM FROM Customers C WHERE C.CNAME='Cisneros');

**Output-**

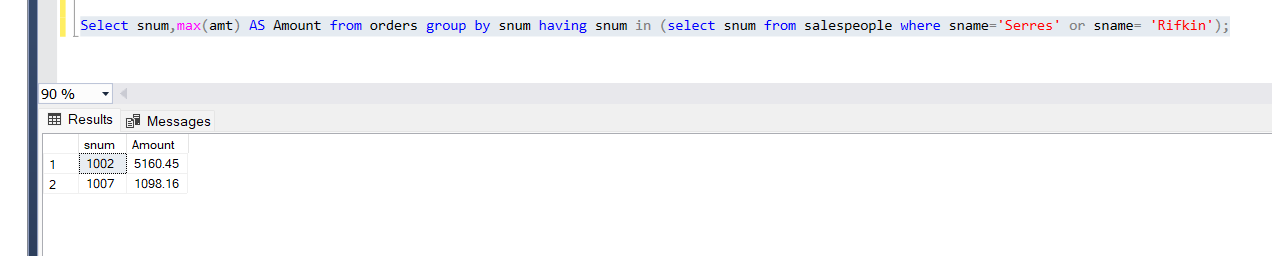
****

30) Find the largest orders for Serres and Rifkin.

**Query-**

Select snum,max(amt) AS Amount from orders group by snum having snum in (select snum from salespeople where sname='Serres' or sname= 'Rifkin');

**Output-**

****

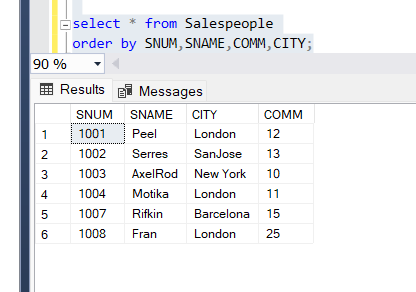
31) Sort the salespeople table in the following order: snum, sname, commission, city.

**Query-**

select \* from Salespeople

order by SNUM,SNAME,COMM,CITY;

**Output-**

****

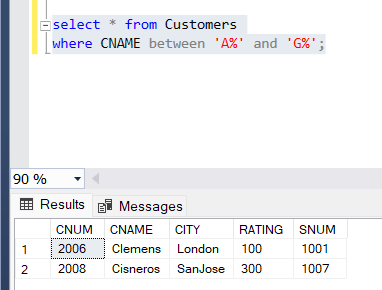
32) Select all customers whose names fall in between ‘A’ and ‘G’ alphabetical range.

**Query-**

select \* from Customers

where CNAME between 'A%' and 'G%';

**Output-**



33) Select all the possible combinations of customers you can assign.

**Query-**

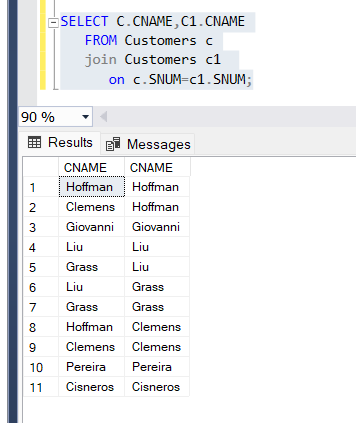
SELECT C.CNAME,C1.CNAME

FROM Customers c

join Customers c1

on c.SNUM=c1.SNUM;

**Output-**

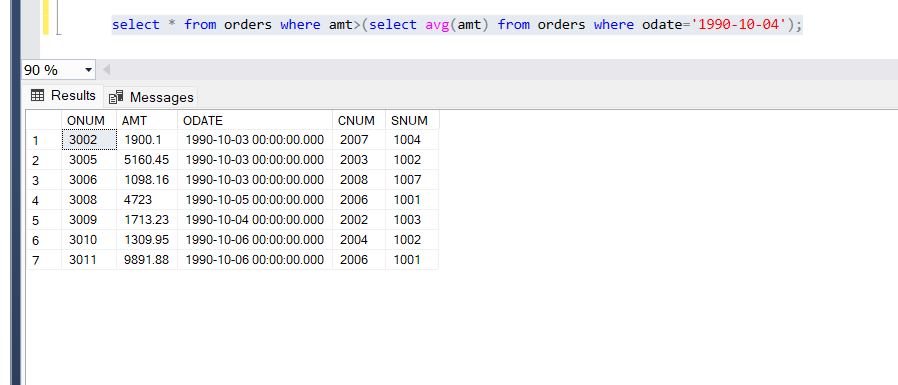
****

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

**Query-**

select \* from orders where amt>(select avg(amt) from orders where odate='1990-10-04');

**Output-**

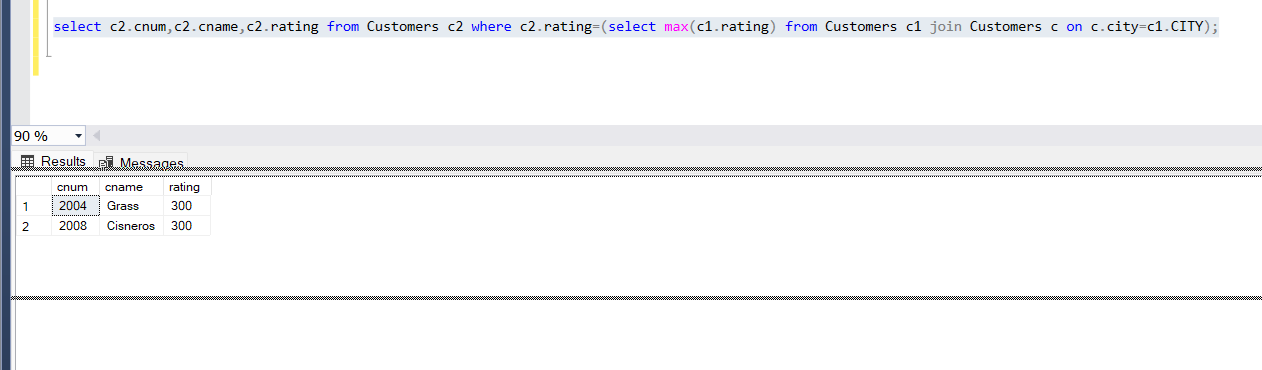
****

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

**Query-**

select c2.cnum,c2.cname,c2.rating from Customers c2 where c2.rating=(select max(c1.rating) from Customers c1 join Customers c on c.city=c1.CITY);

**Output-**

****

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

**Query-**

SELECT o.ODATE

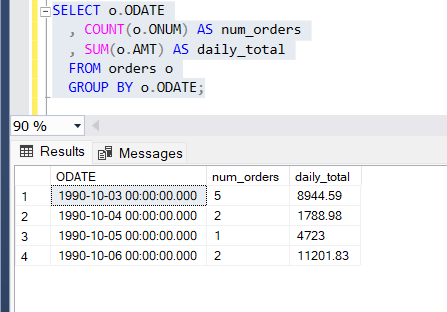
, COUNT(o.ONUM) AS num\_orders

, SUM(o.AMT) AS daily\_total

FROM orders o

GROUP BY o.ODATE;

**Output-**

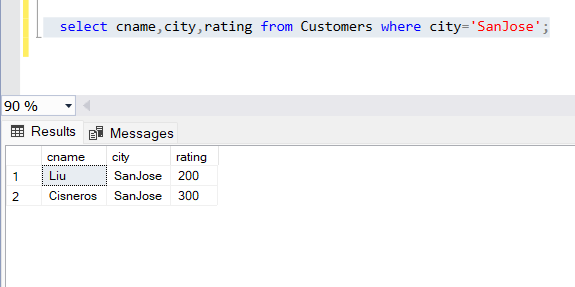


37) Write a select command that produces the rating followed by the name of each customer in SanJose.

**Query-**

select cname,city,rating from Customers where city='SanJose';

**Output-**

****

38) Find all orders with amounts smaller than any amount for a customer in SanJose.

**Query-**

SELECT \*

FROM orders

WHERE AMT < ANY

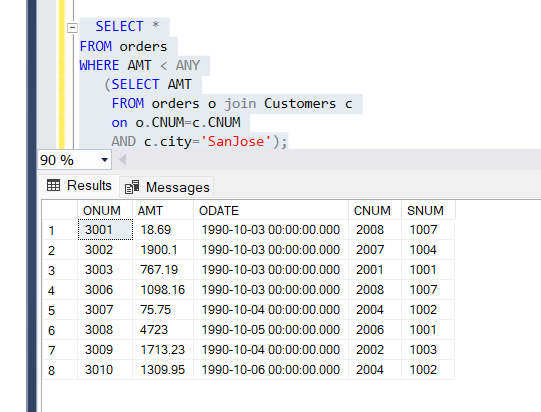
(SELECT AMT

FROM orders o join Customers c

on o.CNUM=c.CNUM

AND c.city='SanJose');

**Output-**



39) Find all orders with above average amounts for their customers.

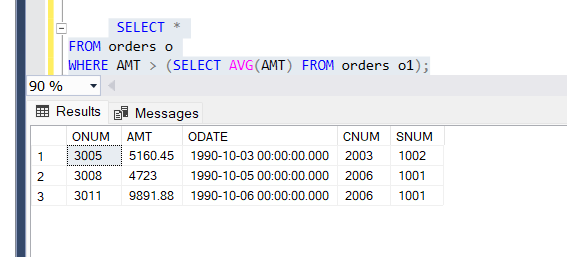
**Query-**

SELECT \*

FROM orders o

WHERE AMT > (SELECT AVG(AMT) FROM orders o1);

**Output-**

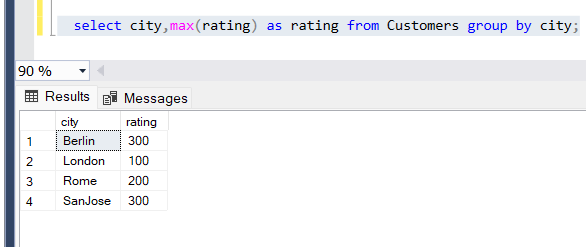


40) Write a query that selects the highest rating in each city.

**Query-**

select city,max(rating) as rating from Customers group by city;

**Output-**



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

**Query-**

select o.ONUM as order\_no,c.CNAME as customer\_name,s.COMM as Commission,o.AMT\*s.COMM as Total

from Salespeople s

join orders o

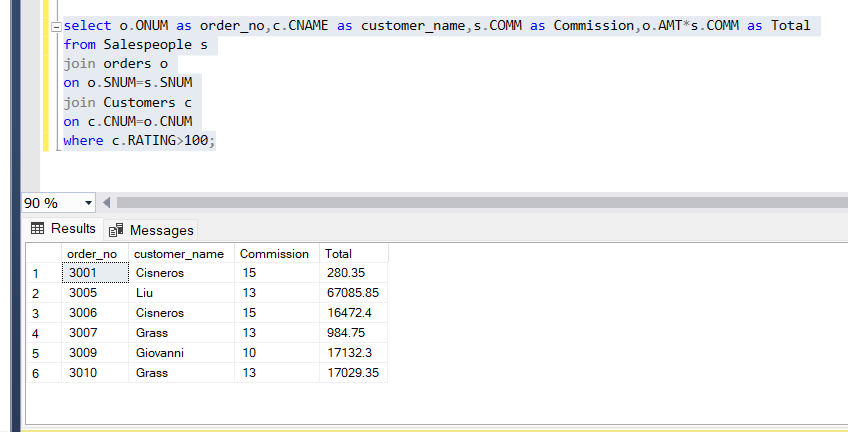
on o.SNUM=s.SNUM

join Customers c

on c.CNUM=o.CNUM

where c.RATING>100;

**Output-**

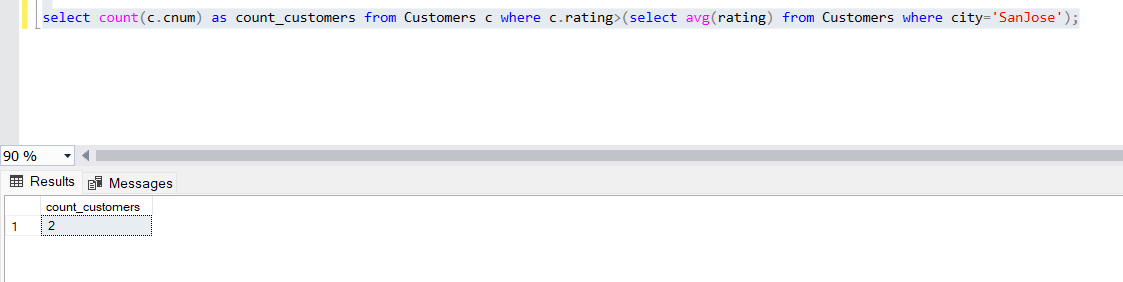


42) Count the customers with ratings above SanJose’s average.

**Query-**

select count(c.cnum) as count\_customers from Customers c where c.rating>(select avg(rating) from Customers where city='SanJose');

**Output-**

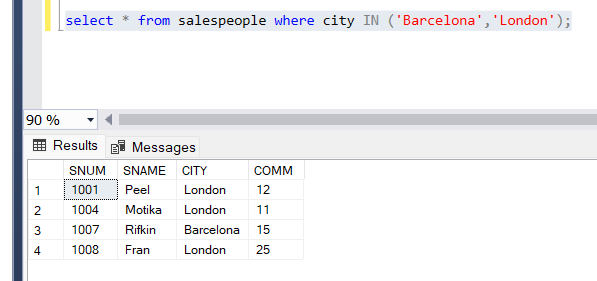


43) Find all salespeople that are located in either Barcelona or London.

**Query-**

select \* from salespeople where city IN ('Barcelona','London');

**Output-**



44) Find all salespeople with only one customer

**Query-**

SELECT \*

FROM Salespeople

WHERE snum IN (

SELECT DISTINCT a.snum

FROM Customers a

WHERE NOT EXISTS (

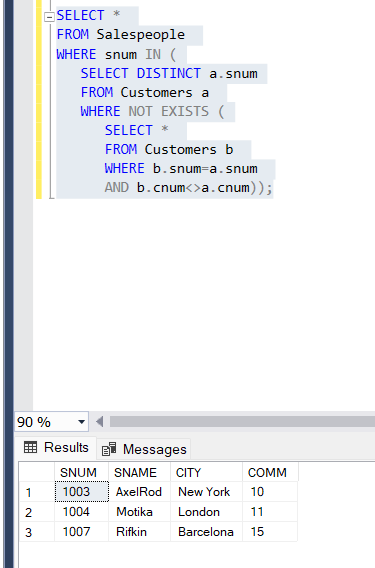
SELECT \*

FROM Customers b

WHERE b.snum=a.snum

AND b.cnum<>a.cnum));

**Output-**



45) Write a query that joins the Customer table to itself to find all pairs or customers served by a single salesperson.

**Query-**

select a.cname,b.cname,c.sname from Customers a

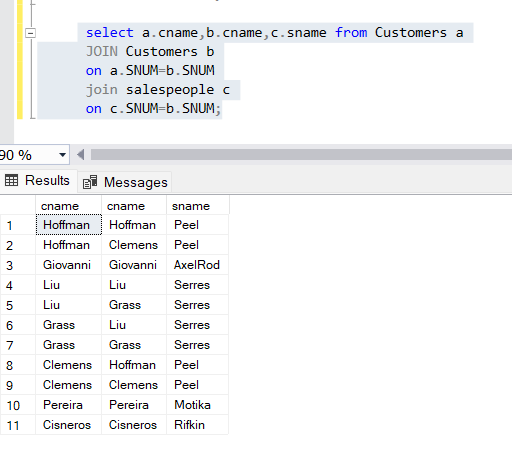
JOIN Customers b

on a.SNUM=b.SNUM

join salespeople c

on c.SNUM=b.SNUM;

**Output-**



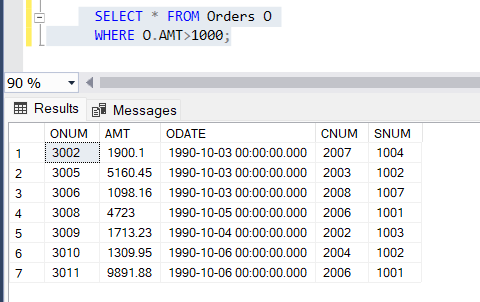
46) Write a query that will give you all orders for more than $1000.00.

**Query-**

SELECT \* FROM Orders O

WHERE O.AMT>1000;

**Output-**



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

**Query-**

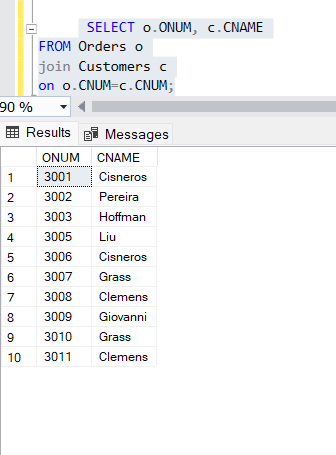
SELECT o.ONUM, c.CNAME

FROM Orders o

join Customers c

on o.CNUM=c.CNUM;

**Output-**



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

49) Write two queries that will produce all orders taken on October 3 or October 4.

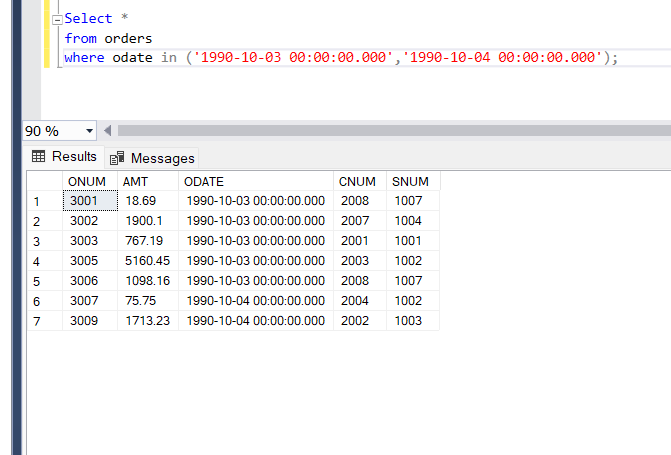
**Query-**

Select \*

from orders

where odate in ('1990-10-03 00:00:00.000','1990-10-04 00:00:00.000');

**Output-**



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

**Query-**

SELECT \*

FROM Customers

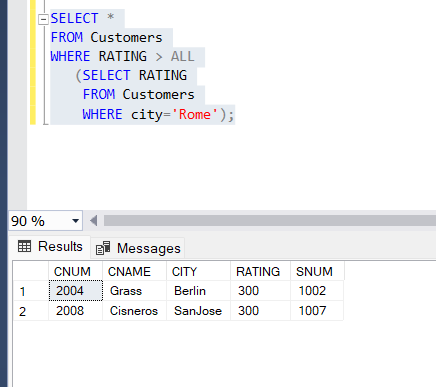
WHERE RATING > ALL

(SELECT RATING

FROM Customers

WHERE city='Rome');

**Output-**



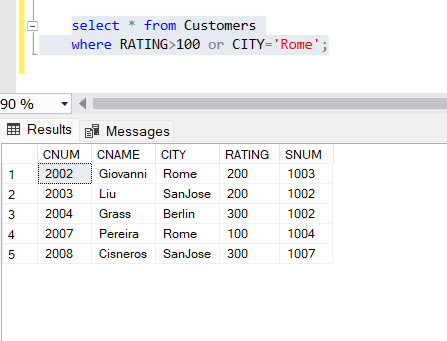
51) Write a query on the Customers table whose output will exclude all customers with a rating<= 100.00, unless they are located in Rome.

**Query-**

select \* from Customers

where RATING>100 or CITY='Rome';

**Output-**



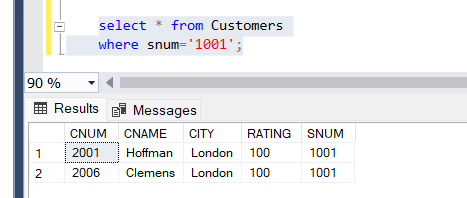
52) Find all rows from the customer’s table for which the salesperson number is 1001.

**Query-**

select \* from Customers

where snum='1001';

**Output-**

****

53) Find the total amount in orders for each salesperson where their total of amounts are greater than the amount of the largest order in the table.

**Query-**

Select snum,sum(amt)

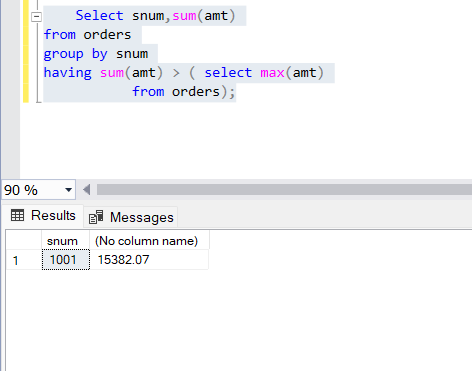
from orders

group by snum

having sum(amt) > ( select max(amt)

from orders);

**Output-**

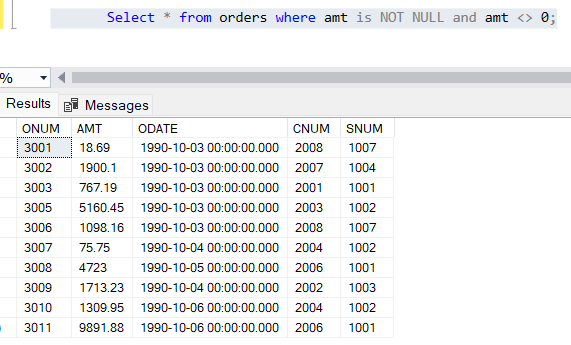
****

54) Write a query that selects all orders save those with zeroes or NULL in the amount file.

**Query-**

Select \* from orders where amt is NOT NULL and amt <> 0;

**Output-**

****

55) 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.

**Query-**

Select s.SNAME,c.CNAME,c.RATING from Salespeople s

join Customers c

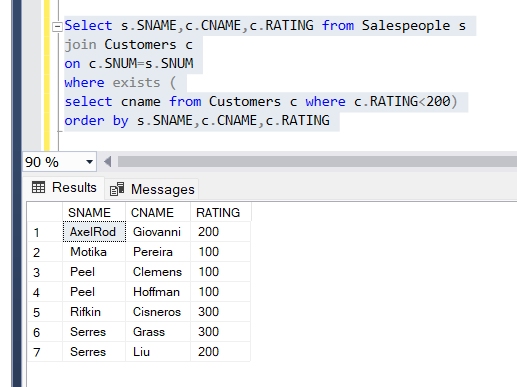
on c.SNUM=s.SNUM

where exists (

select cname from Customers c where c.RATING<200)

order by s.SNAME,c.CNAME,c.RATING;

**Output-**

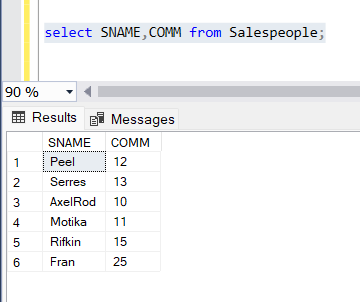


56) Find all salespeople name and commission.

**Query-**

select SNAME,COMM from Salespeople;

**Output-**

****

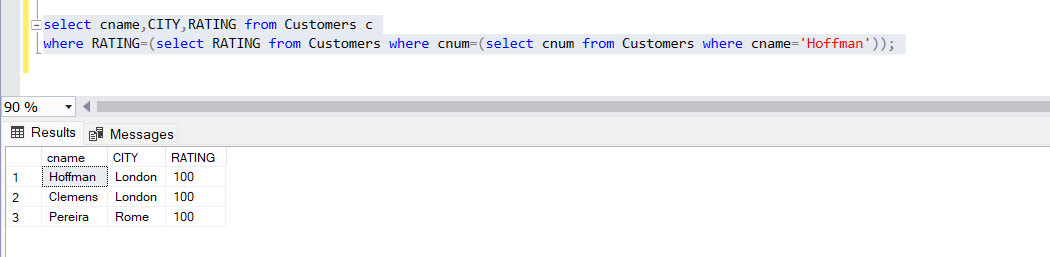
57) Write a query that produces the names and cities of all customers with the same rating as Hoffman. Write the query using Hoffman’s cnum rather than his rating, so that it would still be usable if his rating is changed.

**Query-**

select cname,CITY,RATING from Customers c

where RATING=(select RATING from Customers where cnum=(select cnum from Customers where cname='Hoffman'));

**Output-**

****

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

**Query-**

SELECT \*

FROM Salespeople s

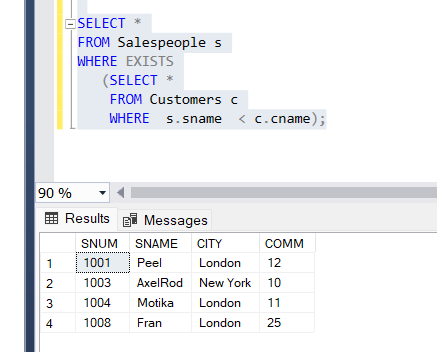
WHERE EXISTS

(SELECT \*

FROM Customers c

WHERE s.sname < c.cname);

**Output-**

****

59) Write a query that produces the names and ratings of all customers who have average orders.

**Query-**

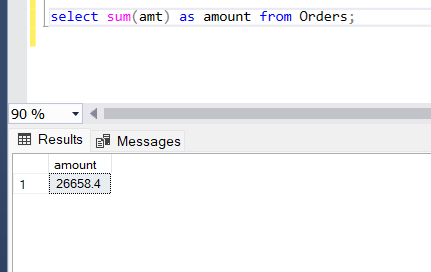
**Output-**

60) Find the SUM of all Amounts from the orders table.

**Query-**

select sum(amt) as amount from Orders;

**Output-**

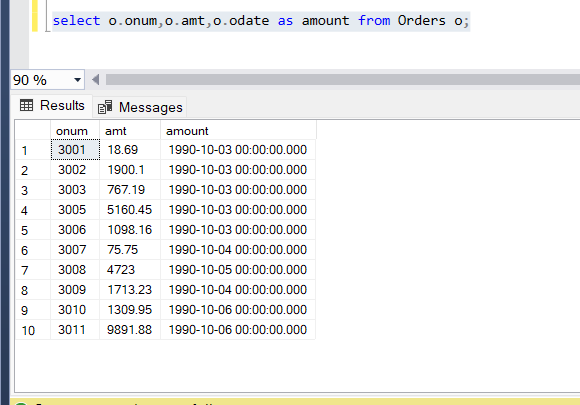
****

61) Write a SELECT command that produces the order number, amount, and the date from rows in the order table.

**Query-**

select o.onum,o.amt,o.odate as amount from Orders o;

**Output-**

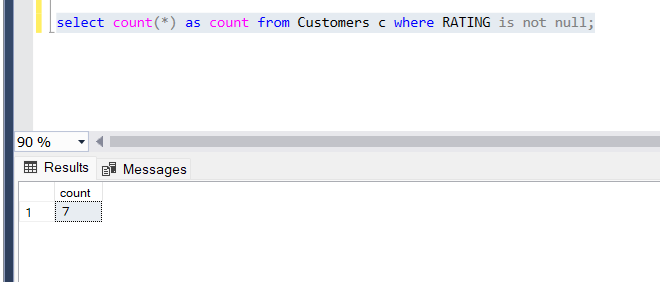
****

62) Count the number of non NULL rating fields in the Customers table (including repeats).

**Query-**

select count(\*) as count from Customers c where RATING is not null;

**Output-**

****

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

**Query-**

Select o.onum,s.SNAME,c.CNAME from Salespeople s

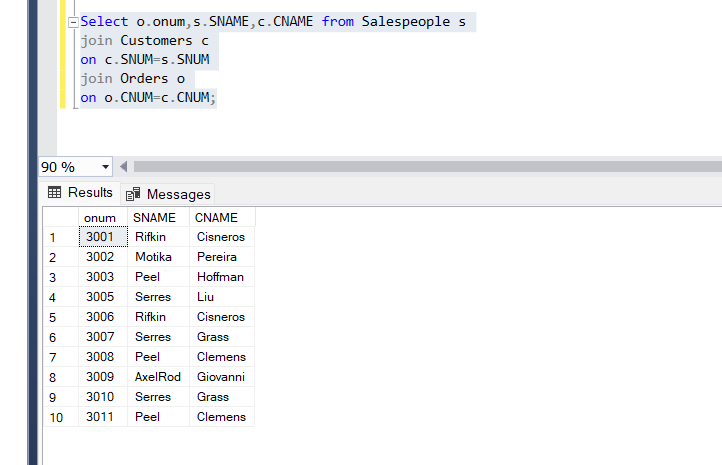
join Customers c

on c.SNUM=s.SNUM

join Orders o

on o.CNUM=c.CNUM;

**Output-**

****

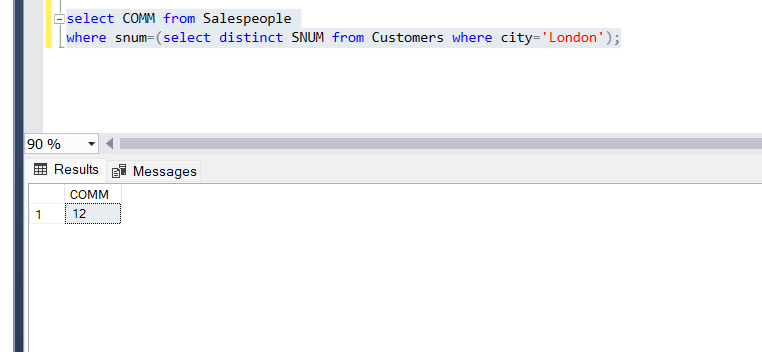
64) List the commissions of all salespeople servicing customers in London.

**Query-**

select COMM from Salespeople

where snum=(select distinct SNUM from Customers where city='London');

**Output-**

****

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

**Query-**

Select sname

from salespeople s1

where s1.snum in ( select s.snum

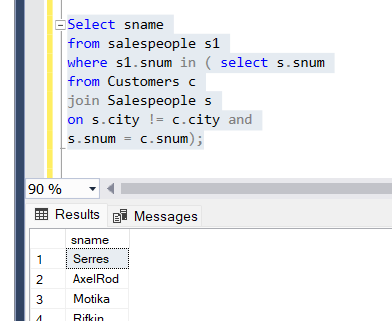
from Customers c

join Salespeople s

on s.city != c.city and

s.snum = c.snum);

**Output-**

****

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

**Query-**

**Output-**

67) Write a query that selects all customers serviced by Peel or Motika. (Hint: The snum field relates the 2 tables to one another.)

**Query-**

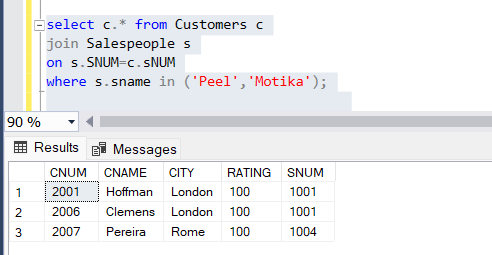
select c.\* from Customers c

join Salespeople s

on s.SNUM=c.sNUM

where s.sname in ('Peel','Motika');

**Output-**

****

68) 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

**Query-**

Select count(distinct s.snum) as count\_sales,o.ODATE

from Salespeople s

join Customers c

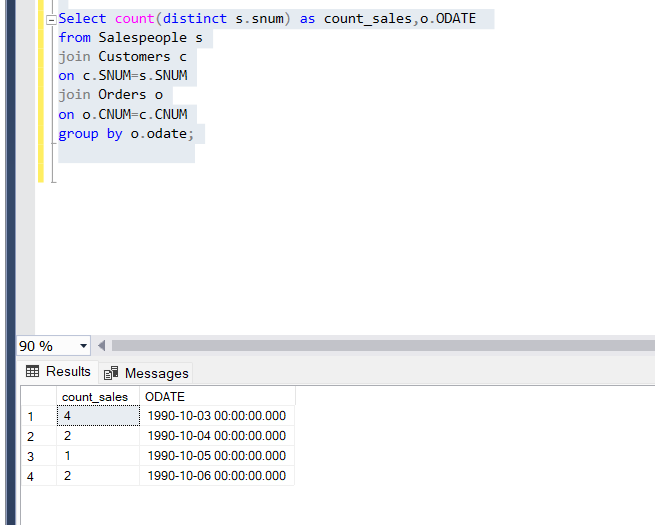
on c.SNUM=s.SNUM

join Orders o

on o.CNUM=c.CNUM

group by o.odate;

**Output-**

****

69) Find all orders attributed to salespeople who live in London.

**Query-**

Select o.\*

from Salespeople s

join Customers c

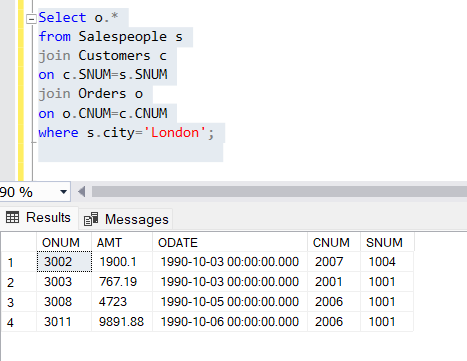
on c.SNUM=s.SNUM

join Orders o

on o.CNUM=c.CNUM

where s.city='London';

**Output-**



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

**Query-**

Select o.\*,c.CNAME,s.city,c.city

from Salespeople s

join Customers c

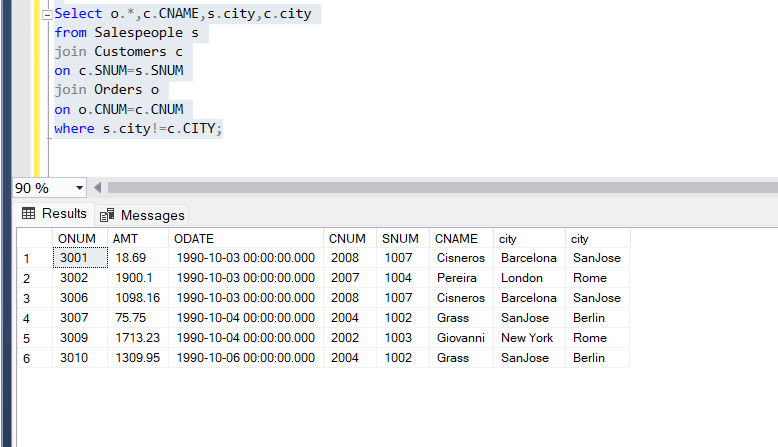
on c.SNUM=s.SNUM

join Orders o

on o.CNUM=c.CNUM

where s.city!=c.CITY;

**Output-**

****

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

**Query-**

Select s.sname,c.cname,count(o.onum) as count\_order

from Salespeople s

join Customers c

on c.SNUM=s.SNUM

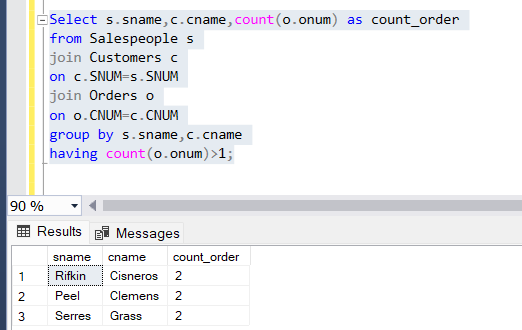
join Orders o

on o.CNUM=c.CNUM

group by s.sname,c.cname

having count(o.onum)>1;

**Output-**



72) Write a query that extracts from the customer’s table every customer assigned to a salesperson, who is currently having at least one another customer(besides the customer being selected) with orders in the Orders Table.

**Query-**

Select

o.cnum,o.snum,max(c.cname) as cust\_name

from Customers c

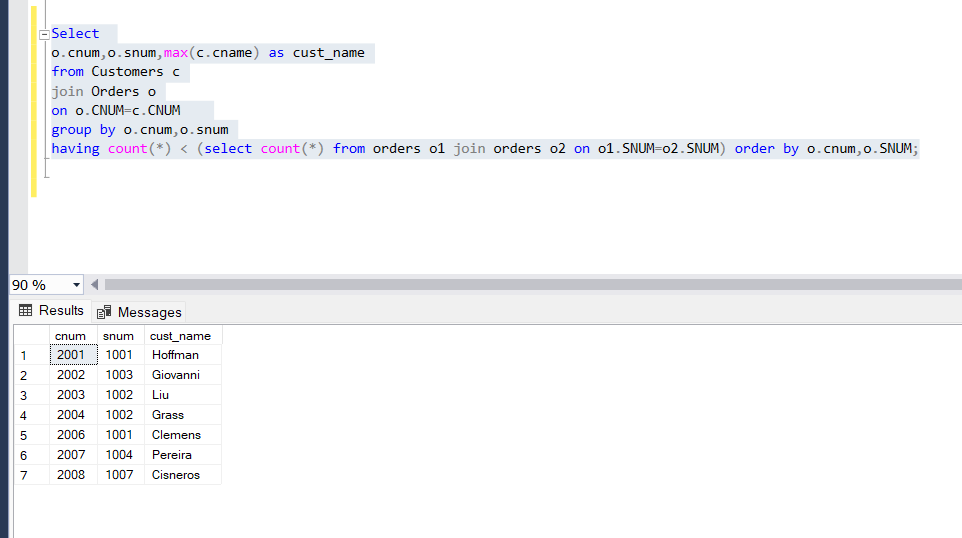
join Orders o

on o.CNUM=c.CNUM

group by o.cnum,o.snum

having count(\*) < (select count(\*) from orders o1 join orders o2 on o1.SNUM=o2.SNUM) order by o.cnum,o.SNUM;

**Output-**



73) Write a query on the customer’s 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).

**Query-**

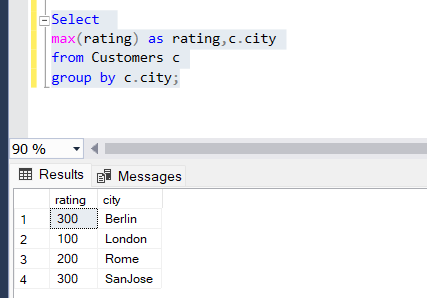
Select

max(rating) as rating,c.city

from Customers c

group by c.city;

**Output-**

****

74) Write a query that will produce the snum values of all salespeople with orders, having amt greater than 1000 in the Orders Table(without

repeats).

**Query-**

select distinct s.snum

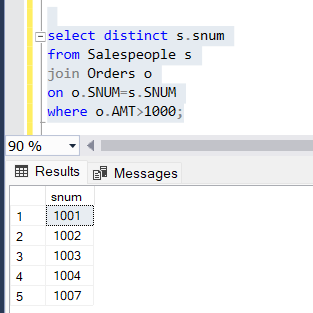
from Salespeople s

join Orders o

on o.SNUM=s.SNUM

where o.AMT>1000;

**Output-**

****

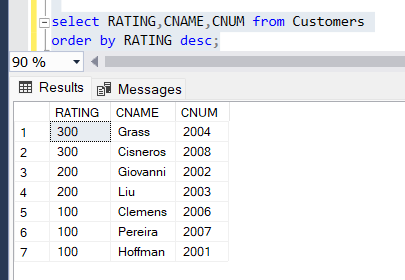
75) Write a query that lists customers in a descending order of rating. Output the rating field first, followed by the customer’s names and numbers.

**Query-**

select RATING,CNAME,CNUM from Customers

order by RATING desc;

**Output-**

****

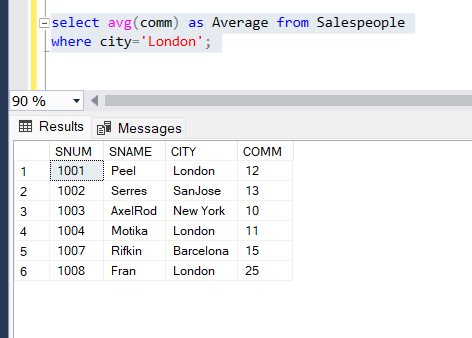
76) Find the average commission for salespeople in London.

**Query-**

select avg(comm) as Average from Salespeople

where city='London';

**Output-**



77) Find all orders credited to the same salesperson who services Hoffman.(cnum 2001).

**Query-**

select o.onum,s.sname,c.cname, o.amt

from Salespeople s

join Orders o

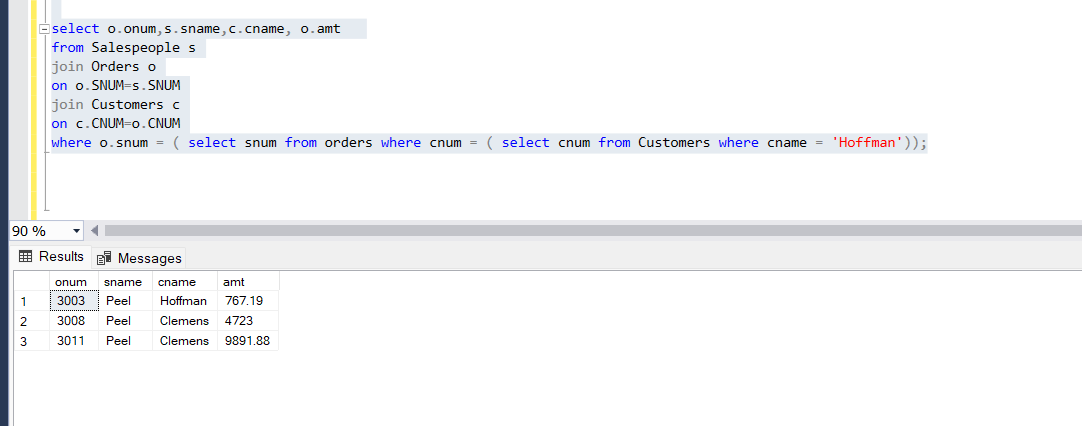
on o.SNUM=s.SNUM

join Customers c

on c.CNUM=o.CNUM

where o.snum = ( select snum from orders where cnum = ( select cnum from Customers where cname = 'Hoffman'));

**Output-**

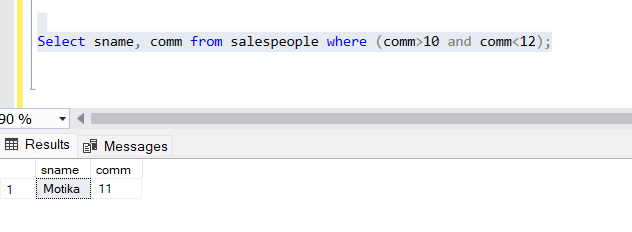
****

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

**Query-**

Select sname, comm from salespeople where (comm>10 and comm<12);

**Output-**



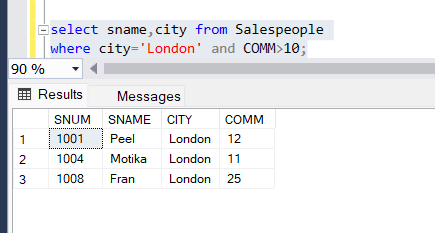
79) Write a query that will give you the names and cities of all salespeople in London with a commission above 0.10.

**Query-**

select sname,city from Salespeople

where city='London' and COMM>10;

**Output-**

****

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

**Query-**

select c.\*,o.\*

from Customers c

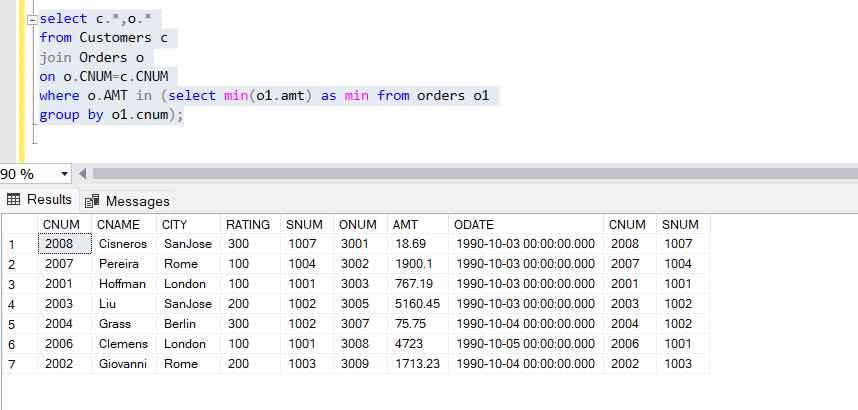
join Orders o

on o.CNUM=c.CNUM

where o.AMT in (select min(o1.amt) as min from orders o1

group by o1.cnum);

**Output-**



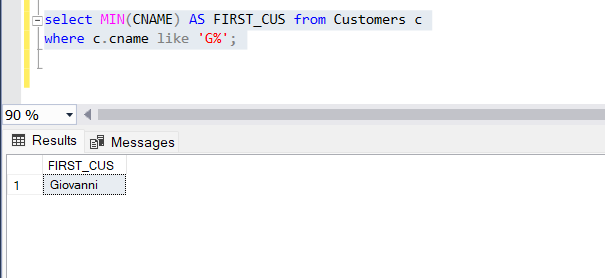
81) Write a query that selects the first customer in alphabetical order whose name begins with ‘G’.

**Query-**

select MIN(CNAME) AS FIRST\_CUS from Customers c

where c.cname like 'G%';

**Output-**

****

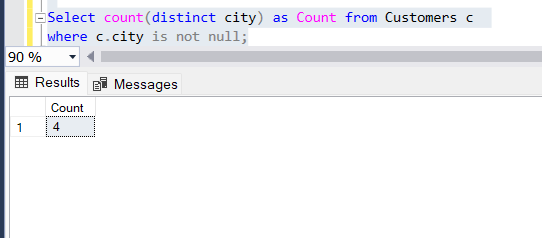
82) Write a query that counts the number of different non NULL city values in the customers table.

**Query-**

Select count(distinct city) as Count from Customers c

where c.city is not null;

**Output-**

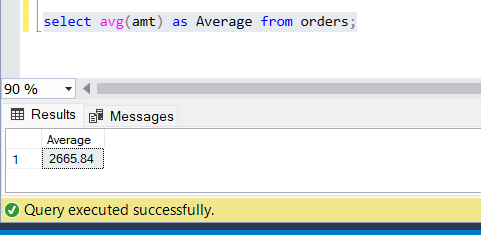
****

83) Find the average amount from the Orders Table.

**Query-**

select avg(amt) as Average from orders;

**Output-**

****

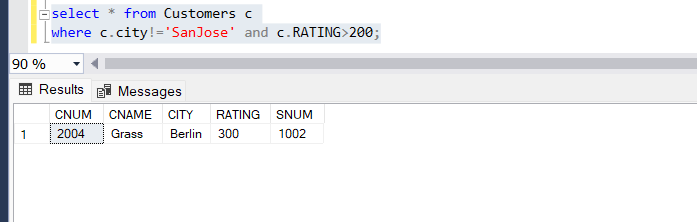
84) Find all customers who are not located in SanJose and whose rating is above 200.

**Query-**

select \* from Customers c

where c.city!='SanJose' and c.RATING>200;

**Output-**

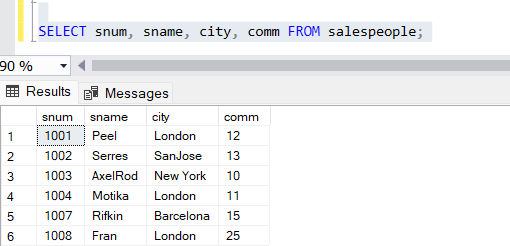
****

85) Give a simpler way to write this query.SELECT snum, sname, city, comm FROM salespeople WHERE (comm > + 0.12 OR comm < 0.14);

**Query-**

SELECT snum, sname, city, comm FROM salespeople;

**Output-**



86) Which salespersons attend to customers not in the city they have been assigned to?

**Query-**

select sname,s.city,c.city

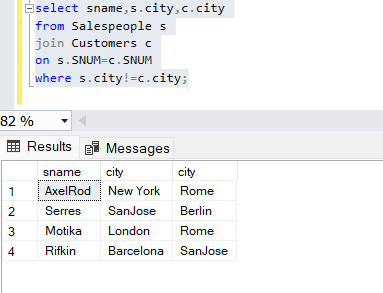
from Salespeople s

join Customers c

on s.SNUM=c.SNUM

where s.city!=c.city;

**Output-**



87) Which salespeople get commission greater than 0.11 are serving customers rated less than 250?

**Query-**

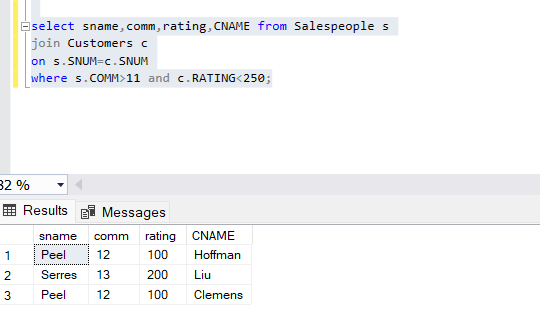
select sname,comm,rating,CNAME from Salespeople s

join Customers c

on s.SNUM=c.SNUM

where s.COMM>11 and c.RATING<250;

**Output-**



88) Which salespeople have been assigned to the same city but get different commission percentages?

**Query-**

select s.sname,s.city,c.city,s.comm

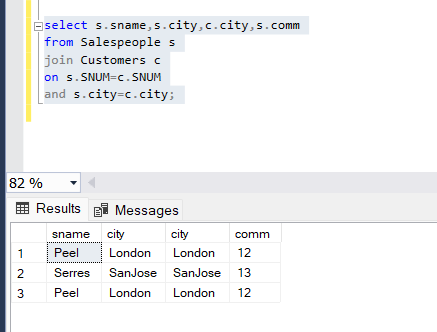
from Salespeople s

join Customers c

on s.SNUM=c.SNUM

and s.city=c.city;

**Output-**

****

89) Which salesperson has earned the maximum commission?

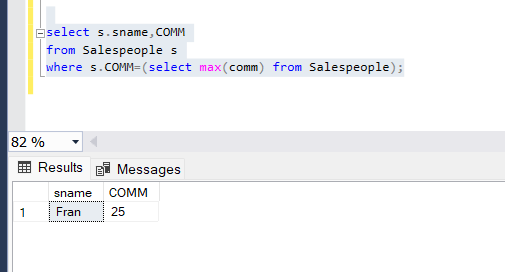
**Query-**

select s.sname,COMM

from Salespeople s

where s.COMM=(select max(comm) from Salespeople);

**Output-**

****

90) Does the customer who has placed the maximum number of orders have the maximum rating?

**Query-**

select count(o.onum) as count,cname from Customers c

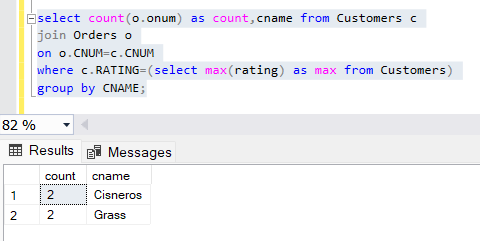
join Orders o

on o.CNUM=c.CNUM

where c.RATING=(select max(rating) as max from Customers)

group by CNAME;

**Output-**

****

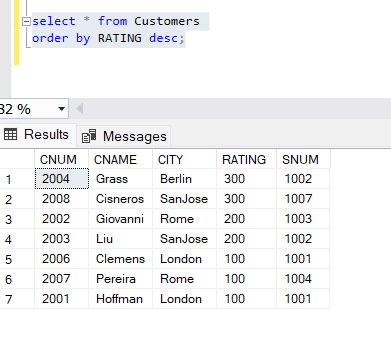
91) List all customers in descending order of customer rating.

**Query-**

select \* from Customers

order by RATING desc;

**Output-**



92) On which days has Hoffman placed orders?

**Query-**

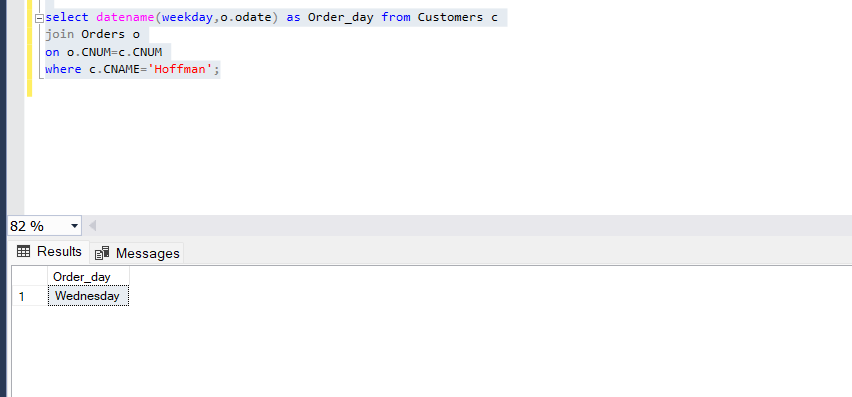
select datename(weekday,o.odate) as Order\_day from Customers c

join Orders o

on o.CNUM=c.CNUM

where c.CNAME='Hoffman';

**Output-**



93) Which salesmen have no orders between 10/03/1990 and 10/05/1990?

**Query-**

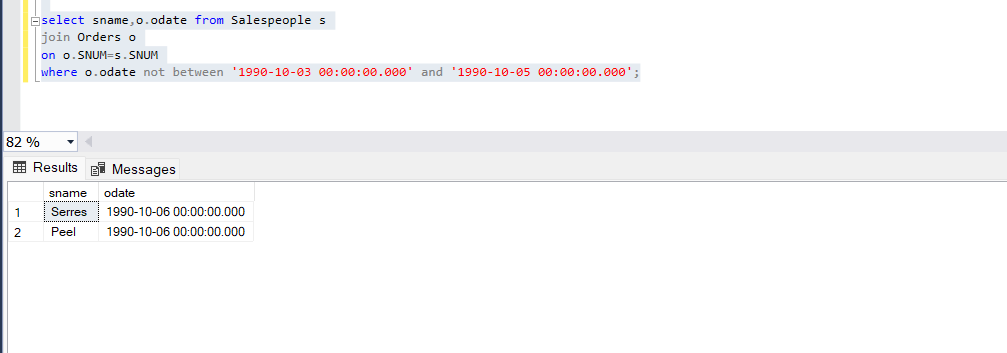
select sname,o.odate from Salespeople s

join Orders o

on o.SNUM=s.SNUM

where o.odate not between '1990-10-03 00:00:00.000' and '1990-10-05 00:00:00.000';

**Output-**



94) How many salespersons have succeeded in getting orders?

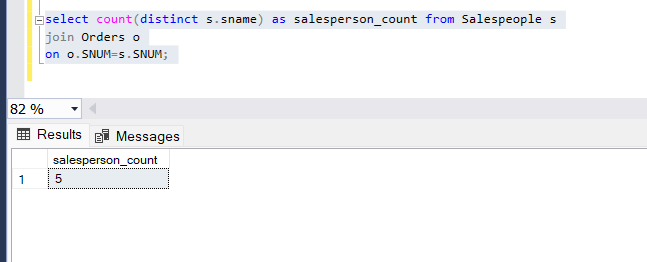
**Query-**

select sname,o.odate from Salespeople s

join Orders o

on o.SNUM=s.SNUM

**Output-**



95) How many customers have placed orders?

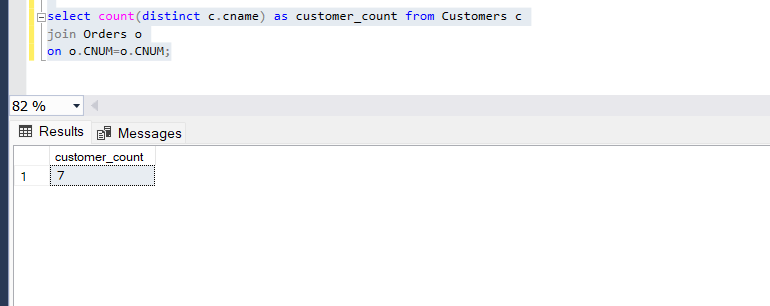
**Query-**

select count(distinct c.cname) as customer\_count from Customers c

join Orders o

on o.CNUM=o.CNUM;

**Output-**

****

96) On which date has each salesman booked an order of maximum value?

**Query-**

**Output-**

97) Who is the most successful salesperson?

**Query-**

**Output-**

98) Which customers have the same rating?

**Query-**

**Output-**

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

**Query-**

**Output-**

100) List all customers with ratings above Grass’s average.

**Query-**

**Output-**

101) Which customers have above average orders?

**Query-**

**Output-**

102) Select the total amount in orders for each salesperson for which the total is greater than the amount of the largest order in the table.

**Query-**

**Output-**

103) Give names and numbers of all salespersons that have more than one customer?

**Query-**

**Output-**

104) Select all salespeople by name and number who have customers in their city whom they don’t service.

**Query-**

**Output-**

105) Does the total amount in orders by customer in Rome and London, exceed the commission paid to salesperson in London, and New York by

more than 5 times?

**Query-**

**Output-**

106) Which are the date, order number, amt and city for each salesperson (by name) for themaximum order he has obtained?

**Query-**

**Output-**

107) Which salesperson is having lowest commission?

**Query-**

**Output-**