**JALA ASSIGNMENT**

**SQL Questions and Queries**

**TABLE SALESPEOPLE**

SNUM SNAME CITY COMM

1001 Peel London .12

1002 Serres San Jose .13

1004 Motika London .11

1007 Rafkin Barcelona .15

1003 Axelrod New york .1

**TABLE CUST**

CNUM CNAME CITY RATING SNUM

2001 Hoffman London 100 1001

2002 Giovanne Rome 200 1003

2003 Liu San Jose 300 1002

2004 Grass Brelin 100 1002

2006 Clemens London 300 1007

2007 Pereira Rome 100 1004

**ORDERS**

**ONUM AMT ODATE CNUM SNUM**

3001 18.69 03-OCT-94 2008 1007

3003 767.19 03-OCT-94 2001 1001

3002 1900.10 03-OCT-94 2007 1004

3005 5160.45 03-OCT-94 2003 1002

3006 1098.16 04-OCT-94 2008 1007

3009 1713.23 04-OCT-94 2002 1003

3007 75.75 05-OCT-94 2004 1002

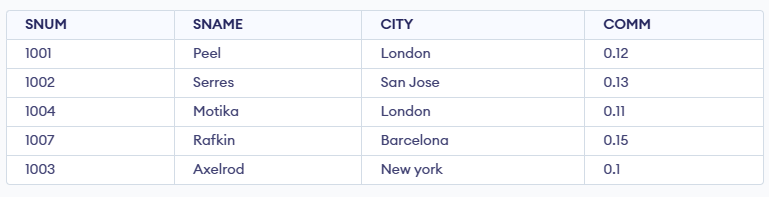
3008 4723.00 05-OCT-94 2006 1001

3010 1309.95 06-OCT-94 2004 1002

3011 9891.88 06-OCT-94 2006 1001

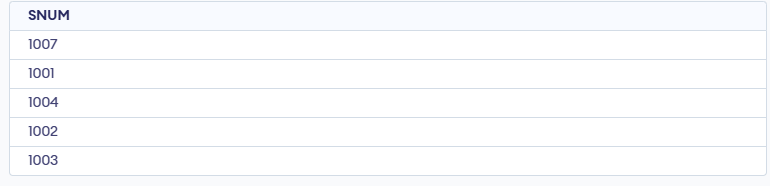
**1, Display snum,sname,city and comm of all salespeople.**

Select snum, sname, city, comm from salespeople;



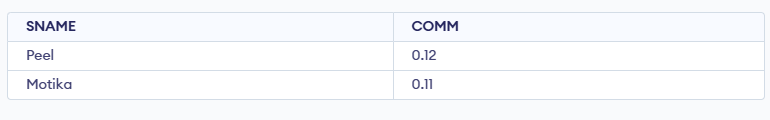
**2.Display all snum without duplicates from all orders**.

Select distinct snum from orders;



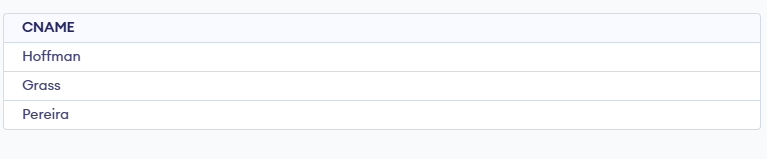
**3.Display names and commissions of all salespeople in london.**

Select sname,comm from salespeople where city = ‘London’;



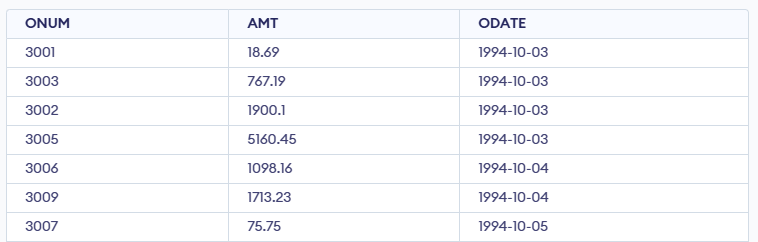
**4All customers with rating of 100.**

Select cname from cust where rating = 100;



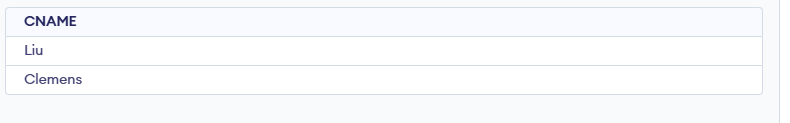
**5.Produce orderno, amount and date form all rows in the order table.**

Select ordno, amt, odate from orders;



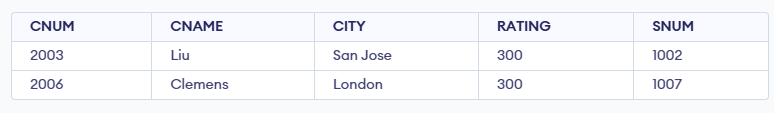
**6.All customers in San Jose, who have rating more than 200.**

Select cname from cust where rating > 200;



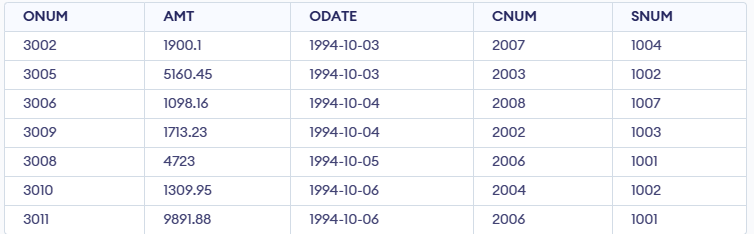
**7.All customers who were either located in San Jose or had a rating above 200.**

Select cname from cust where city = ‘San Jose’ or rating > 200;



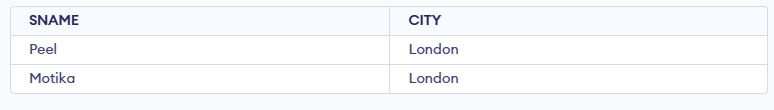
**8.All orders for more than $1000.**

Select \* from orders where amt > 1000;



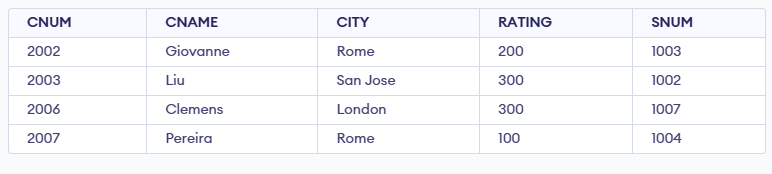
**9.Names and citires of all salespeople in london with commission above 0.10.**

Select sname, city from salepeople where comm > 0.10 and city = ‘London’;



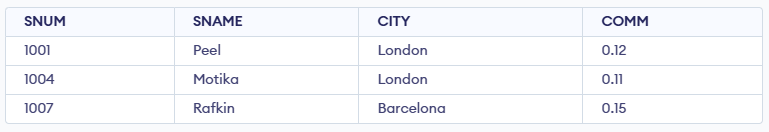
**10.All customers excluding those with rating <= 100 unless they are located in Rome**.

SELECT \* FROM CUST WHERE RATING > 100 OR CITY = 'Rome';

****

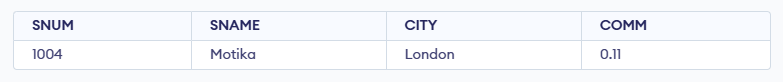
**11.All salespeople either in Barcelona or in london.**

Select sname, cityfrom salespeoplewhere city in (‘Barcelona’,’London’);

****

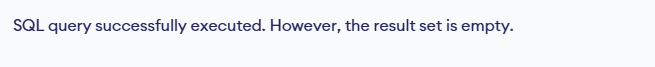
**12.All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded)**

Select sname, commfrom salespeoplewhere comm > 0.10 and comm < 0.12;

****

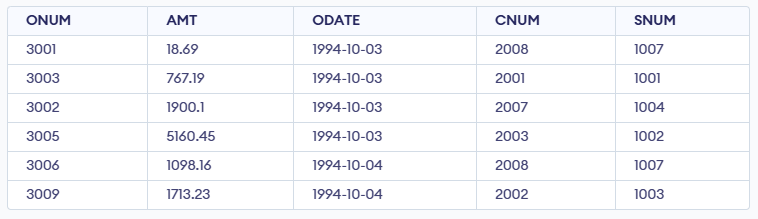
**13.All customers with NULL values in city column.**

Select cname from cust where city is null;



**14.All orders taken on Oct 3Rd and Oct 4th 1994.**

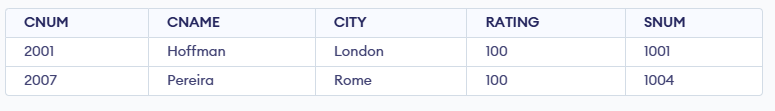
Select \*from orders where odate in (‘03-OCT-94’,’04-OCT-94’);



**15.All customers serviced by peel or Motika.**

Select cname from cust, orders where orders.cnum = cust.cnum and orders.snum in ( select snum

from salespeople where sname in 'Peel','Motika'));



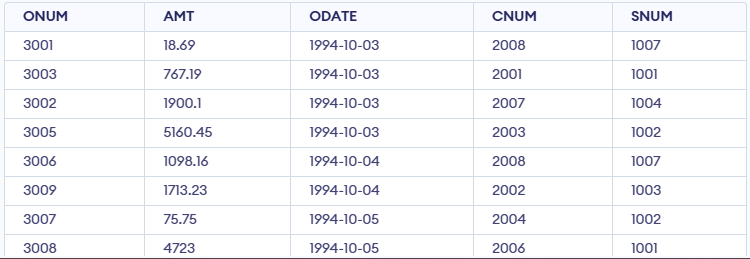
**16.All customers whose names begin with a letter from A to B.**

SELECT \* FROM CUST WHERE CNAME BETWEEN 'A' AND 'B';



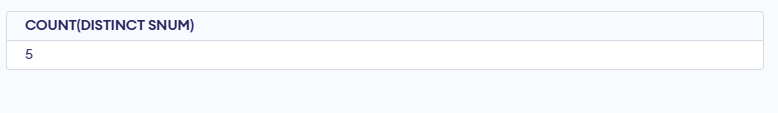
**17.All orders except those with 0 or NULL value in amt field.**

SELECT \* FROM ORDERS WHERE AMT IS NOT NULL AND AMT <> 0;



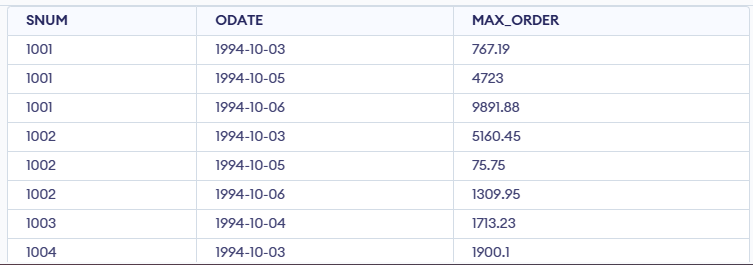
**18.Count the number of salespeople currently listing orders in the order table.**

Select count(distinct snum) from orders;



19.**Largest order taken by each salesperson, datewise.**

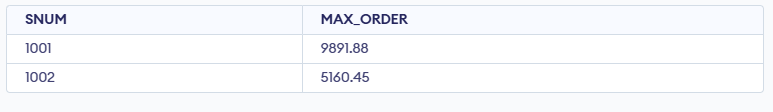
SELECT SNUM, ODATE, MAX(AMT) AS MAX\_ORDER FROM ORDERS GROUP BY SNUM, ODATE;



20.**Largest order taken by each salesperson with order value more than $3000.**

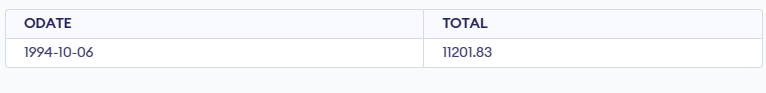
SELECT SNUM, MAX(AMT) AS MAX\_ORDER FROM ORDERS

WHERE AMT > 3000 GROUP BY SNUM;



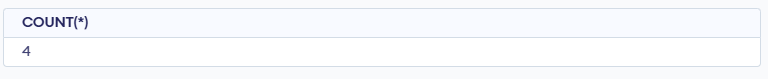
**21.Which day had the hightest total amount ordered.**

Select odate, amt, snum, cnum from orders where amt = (select max(amt) from orders)



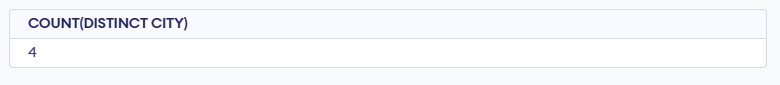
**22.Count all orders for Oct 3rd.**

Select count(\*) from orders where odate = ‘03-OCT-94’;



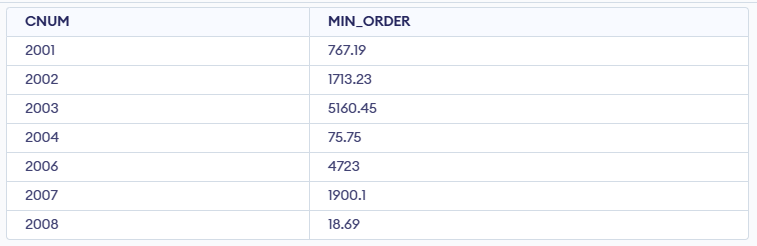
**23.Count the number of different non NULL city values in customers table.**

Select count(distinct city) from cust;

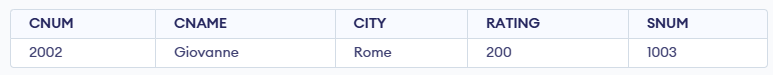


**24.Select each customer’s smallest order.**

Select cnum, min(amt) from orders group by cnum;

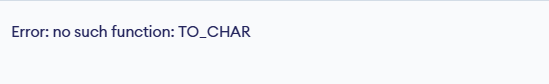


**25.First customer in alphabetical order whose name begins with G.**

SELECT \* FROM CUST WHERE CNAME LIKE 'G%' ORDER BY CNAME ASC LIMIT 1;

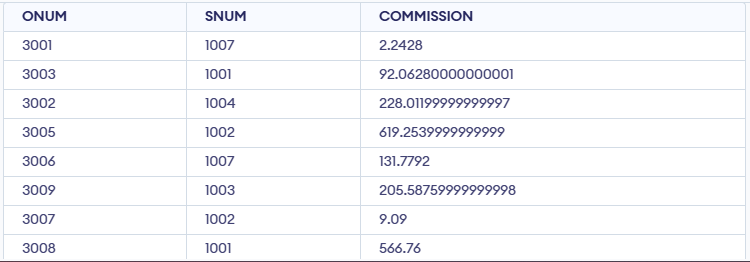
**26.Get the output like “ For dd/mm/yy there are \_\_\_ orders.**

Select 'For ' || to\_char(odate,'dd/mm/yy') || ' there are '||count(\*) || ' Orders'from orders group by odate;

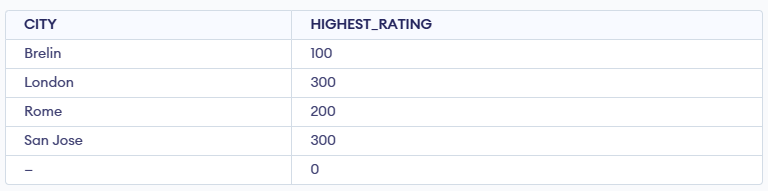


**27.Assume that each salesperson has a 12% commission. Produce order no., salesperson no., and amount of salesperson’s commission for that order.**

Select onum, snum, amt, amt \* 0.12 from orders order by snum;

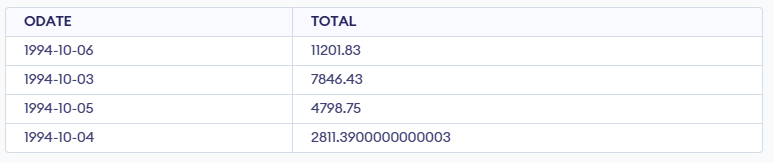


**28.Find highest rating in each city. Put the output in this form. For the city (city), the highest rating is : (rating).**

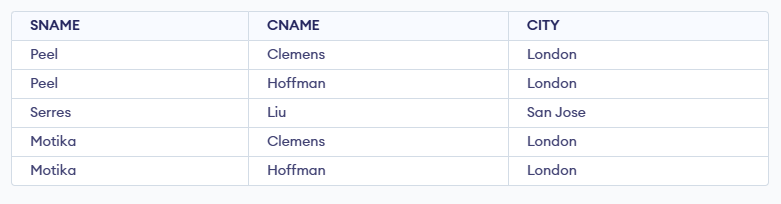
SELECT CITY, MAX(RATING) AS HIGHEST\_RATING FROM CUST GROUP BY CITY;

**29.Display the totals of orders for each day and place the results in descending order.**

SELECT ODATE, SUM(AMT) AS TOTAL FROM ORDERS GROUP BY ODATE ORDER BY TOTAL DESC;

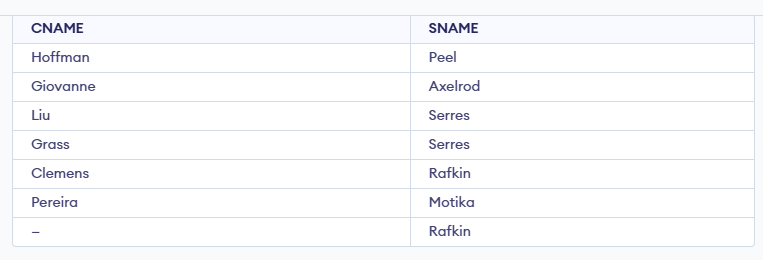


**30.All combinations of salespeople and customers who shared a city. (ie same city).**

SELECT S.SNAME, C.CNAME, S.CITY FROM SALESPEOPLE S JOIN CUST C ON S.CITY = C.CITY;

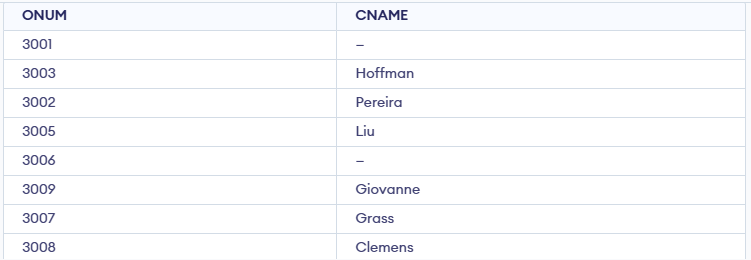
**31. Name of all customers matched with the salespeople serving them.**

SELECT C.CNAME, S.SNAME FROM CUST C JOIN SALESPEOPLE S ON C.SNUM = S.SNUM;



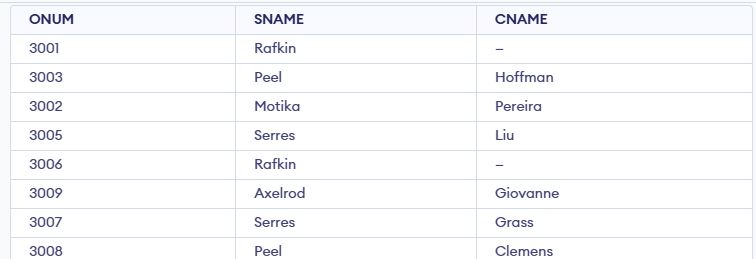
**32. List each order number followed by the name of the customer who made the order.**

SELECT O.ONUM, C.CNAME FROM ORDERS O JOIN CUST C ON O.CNUM = C.CNUM;



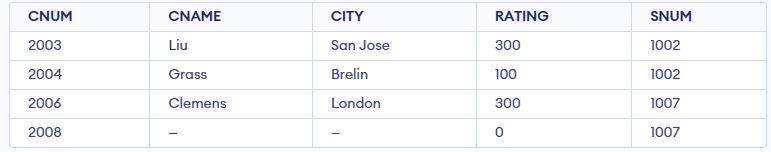
**33. Names of salesperson and customer for each order after the order number.**

SELECT O.ONUM, S.SNAME, C.CNAME FROM ORDERS O JOIN SALESPEOPLE S ON O.SNUM = S.SNUM JOIN CUST C ON O.CNUM = C.CNUM;



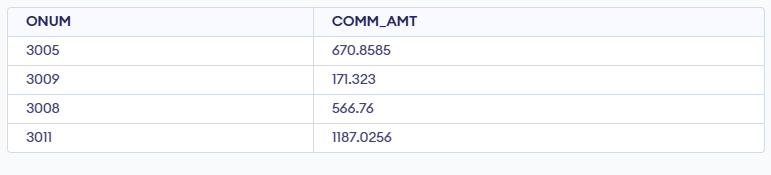
**34. Produce all customer serviced by salespeople with a commission above 12%.**

SELECT C.\* FROM CUST C JOIN SALESPEOPLE S ON C.SNUM = S.SNUM WHERE S.COMM > 0.12;



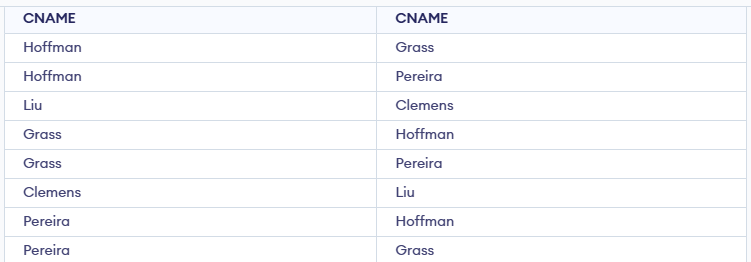
**35. Calculate the amount of the salesperson’s commission on each order with a rating above 100.**

SELECT O.ONUM, O.AMT \* S.COMM AS COMM\_AMT FROM ORDERS O JOIN CUST C ON O.CNUM = C.CNUM JOIN SALESPEOPLE S ON O.SNUM = S.SNUM WHERE C.RATING > 100;



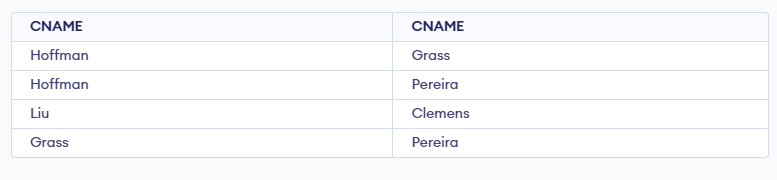
3**6. Find all pairs of customers having the same rating.**

SELECT C1.CNAME, C2.CNAME FROM CUST C1, CUST C2 WHERE C1.RATING = C2.RATING AND C1.CNUM <> C2.CNUM;



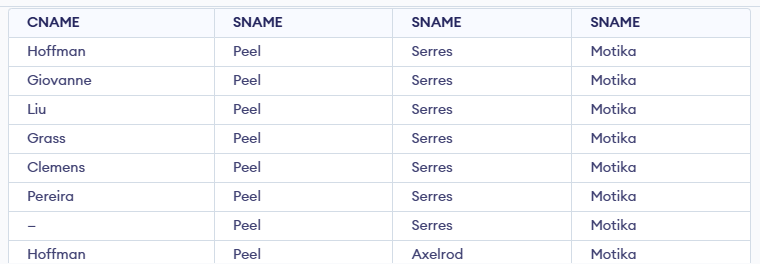
**37. Find all pairs of customers having the same rating, each pair coming once only.**

SELECT C1.CNAME, C2.CNAME FROM CUST C1, CUST C2 WHERE C1.RATING = C2.RATING AND C1.CNUM < C2.CNUM;



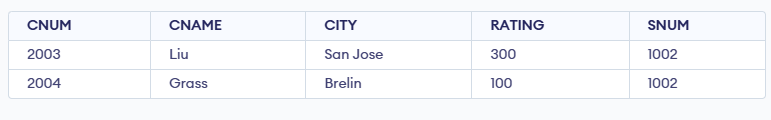
**38. Policy is to assign three salesperson to each customers. Display all such combinations.**

SELECT C.CNAME, S1.SNAME, S2.SNAME, S3.SNAME FROM CUST C, SALESPEOPLE S1, SALESPEOPLE S2, SALESPEOPLE S3 WHERE S1.SNUM < S2.SNUM AND S2.SNUM < S3.SNUM;



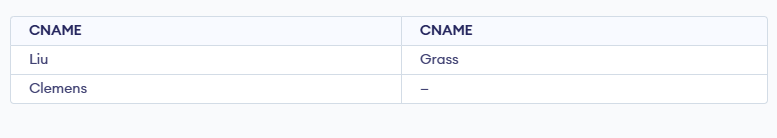
**39. Display all customers located in cities where salesman Serres has customer.**

SELECT DISTINCT C1.\* FROM CUST C1 WHERE C1.CITY IN (SELECT C2.CITY FROM CUST C2 JOIN SALESPEOPLE S ON C2.SNUM = S.SNUM WHERE S.SNAME = 'Serres');



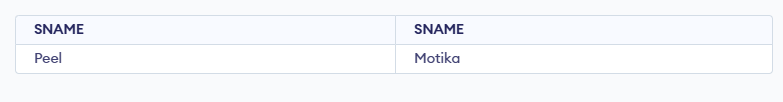
**40. Find all pairs of customers served by single salesperson.**

SELECT C1.CNAME, C2.CNAME FROM CUST C1, CUST C2 WHERE C1.SNUM = C2.SNUM AND C1.CNUM < C2.CNUM;



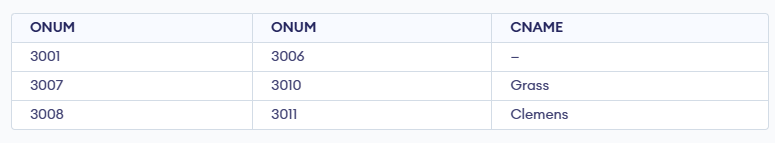
**41. Produce all pairs of salespeople which are living in the same city. Exclude combinations of salespeople with themselves as well as duplicates with the order reversed.**

SELECT S1.SNAME, S2.SNAME FROM SALESPEOPLE S1, SALESPEOPLE S2 WHERE S1.CITY = S2.CITY AND S1.SNUM < S2.SNUM;



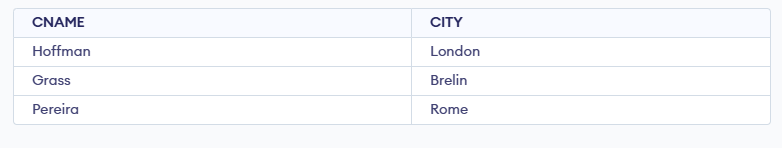
**42. Produce all pairs of orders by given customer, names that customers and eliminates duplicates.**

SELECT O1.ONUM, O2.ONUM, C.CNAME FROM ORDERS O1, ORDERS O2, CUST C WHERE O1.CNUM = O2.CNUM AND O1.CNUM = C.CNUM AND O1.ONUM < O2.ONUM;



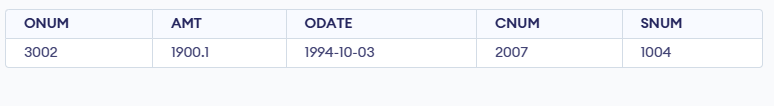
**43. Produce names and cities of all customers with the same rating as Hoffman.**

SELECT CNAME, CITY FROM CUST WHERE RATING = (SELECT RATING FROM CUST WHERE CNAME = 'Hoffman');



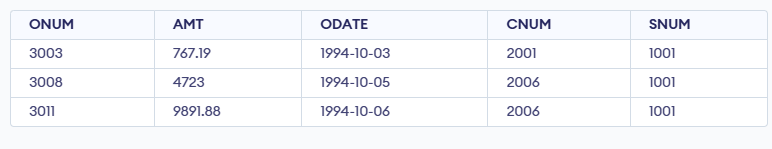
**44. Extract all the orders of Motika.**

SELECT \* FROM ORDERS WHERE SNUM = (SELECT SNUM FROM SALESPEOPLE WHERE SNAME = 'Motika');



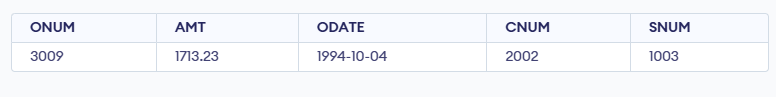
**45. All orders credited to the same salesperson who services Hoffman.**

SELECT \* FROM ORDERS WHERE SNUM = (SELECT SNUM FROM CUST WHERE CNAME = 'Hoffman');



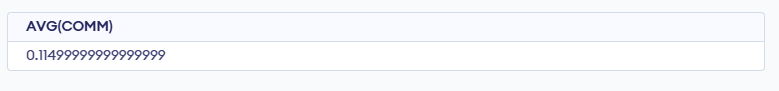
**46. All orders that are greater than the average for Oct 4.**

SELECT \* FROM ORDERS WHERE ODATE = '1994-10-04' AND AMT > (SELECT AVG(AMT) FROM ORDERS WHERE ODATE = '1994-10-04');



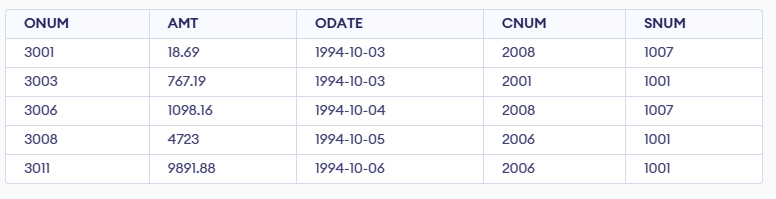
**47. Find average commission of salespeople in London**.

SELECT AVG(COMM) FROM SALESPEOPLE WHERE CITY = 'London';



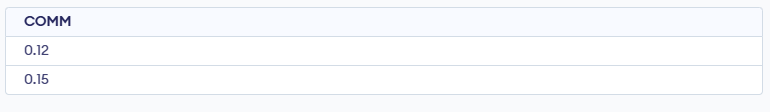
**48. Find all orders attributed to salespeople servicing customers in London.**

SELECT \* FROM ORDERS WHERE SNUM IN (SELECT SNUM FROM CUST WHERE CITY = 'London');



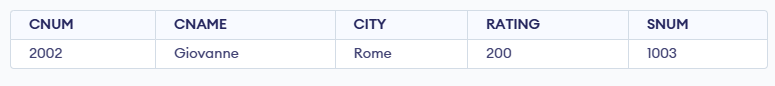
**49. Extract commissions of all salespeople servicing customers in London.**

SELECT DISTINCT S.COMM FROM SALESPEOPLE S JOIN CUST C ON S.SNUM = C.SNUM WHERE C.CITY = 'London';



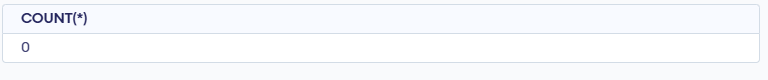
**50. Find all customers whose cnum is 1000 above the snum of serres.**

SELECT \* FROM CUST WHERE CNUM = (SELECT SNUM + 1000 FROM SALESPEOPLE WHERE SNAME = 'Serres');



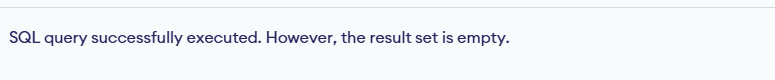
**51. Count the customers with rating above San Jose’s average.**

SELECT COUNT(\*) FROM CUST WHERE RATING > (SELECT AVG(RATING) FROM CUST WHERE CITY = 'San Jose');



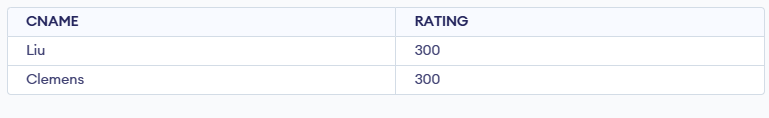
**52. Obtain all orders for the customer named Cisnerous. (Assume you don’t know his customer no. (cnum)).**

SELECT \* FROM ORDERS WHERE CNUM = (SELECT CNUM FROM CUST WHERE CNAME = 'Cisnerous');



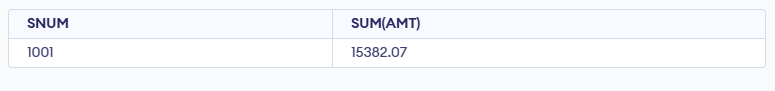
**53. Produce the names and rating of all customers who have above average orders.**

SELECT DISTINCT C.CNAME, C.RATING FROM CUST C JOIN ORDERS O ON C.CNUM = O.CNUM GROUP BY C.CNUM, C.CNAME, C.RATING HAVING AVG(O.AMT) > (SELECT AVG(AMT) FROM ORDERS);



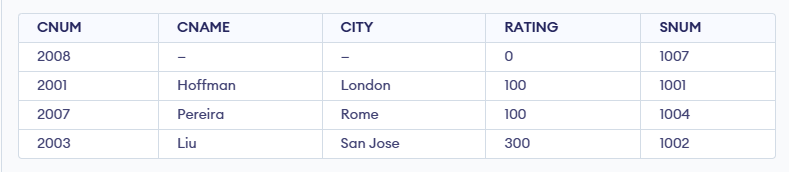
**54. Find total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the table.**

SELECT SNUM, SUM(AMT) FROM ORDERS GROUP BY SNUM HAVING SUM(AMT) > (SELECT MAX(AMT) FROM ORDERS);



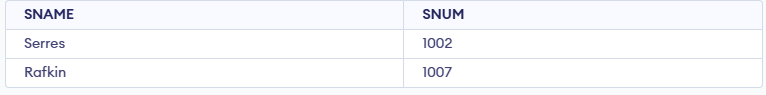
**55. Find all customers with order on 3rd Oct.**

SELECT DISTINCT C.\* FROM CUST C JOIN ORDERS O ON C.CNUM = O.CNUM WHERE O.ODATE = '1994-10-03';



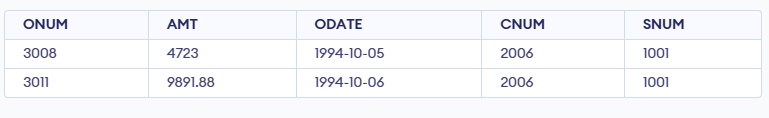
**56. Find names and numbers of all salesperson who have more than one customer.**

SELECT S.SNAME, S.SNUM FROM SALESPEOPLE S WHERE S.SNUM IN (SELECT SNUM FROM CUST GROUP BY SNUM HAVING COUNT(\*) > 1);



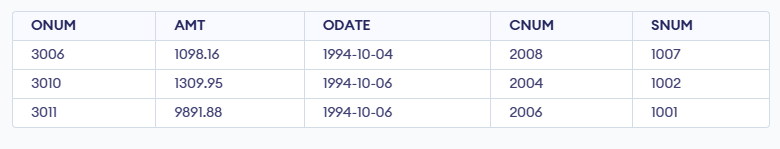
**57. Check if the correct salesperson was credited with each sale.**

SELECT \* FROM ORDERS O WHERE NOT EXISTS (SELECT 1 FROM CUST C WHERE C.CNUM = O.CNUM AND C.SNUM = O.SNUM);



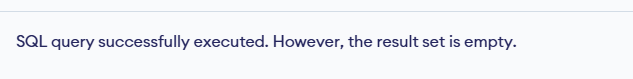
**58. Find all orders with above average amounts for their customers.**

SELECT O.\* FROM ORDERS O WHERE O.AMT > (SELECT AVG(O2.AMT) FROM ORDERS O2 WHERE O2.CNUM = O.CNUM);



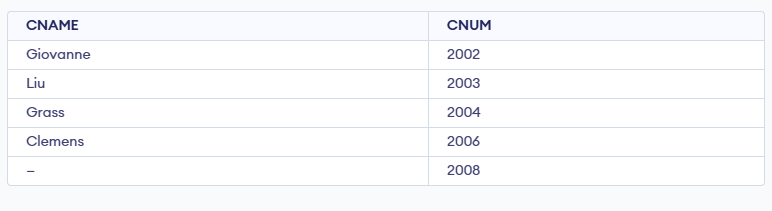
**59. Find the sums of the amounts from order table grouped by date, eliminating all those dates where the sum was not at least 2000 above the maximum amount.**

SELECT ODATE, SUM(AMT) FROM ORDERS GROUP BY ODATE HAVING SUM(AMT) >= (SELECT MAX(AMT) + 2000 FROM ORDERS);



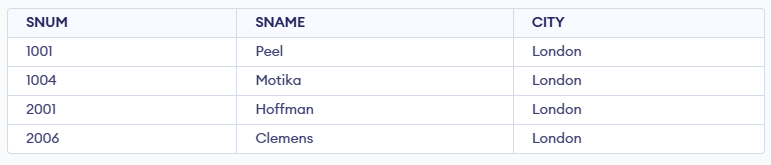
**60. Find names and numbers of all customers with ratings equal to the maximum for their city.**

SELECT C1.CNAME, C1.CNUM FROM CUST C1 WHERE RATING = (SELECT MAX(C2.RATING) FROM CUST C2 WHERE C1.CITY = C2.CITY);



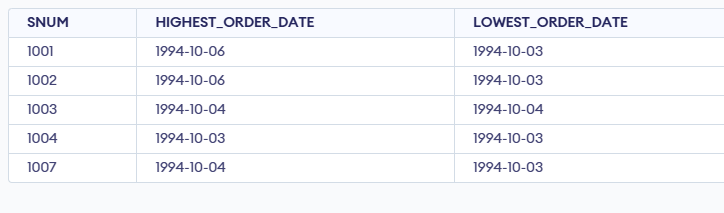
**61.Find all salespeople who have customers in their cities who they don’t service. ( Both way using Join and Correlated subquery.)**

SELECT SNUM, SNAME, CITY FROM SALESPEOPLE WHERE CITY = 'London' UNION SELECT CNUM, CNAME, CITY FROM CUST WHERE CITY = 'London';



**62.Extract cnum,cname and city from customer table if and only if one or more of the customers in the table are located in San Jose.**

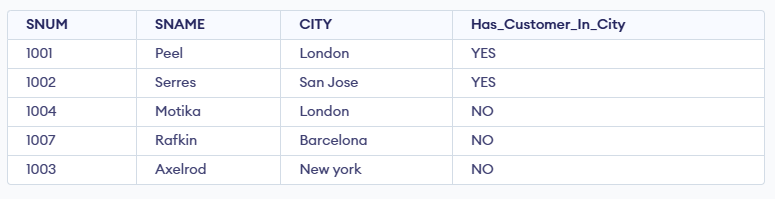
SELECT SNUM, MAX(ODATE) AS HIGHEST\_ORDER\_DATE, MIN(ODATE) AS LOWEST\_ORDER\_DATE FROM ORDERS GROUP BY SNUM;



**63.Find salespeople no. who have multiple customers.**

**SQL:** SELECT S.SNUM, S.SNAME, S.CITY, CASE WHEN EXISTS (SELECT 1 FROM CUST C WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM) THEN 'YES' ELSE 'NO' END AS Has\_Customer\_In\_City

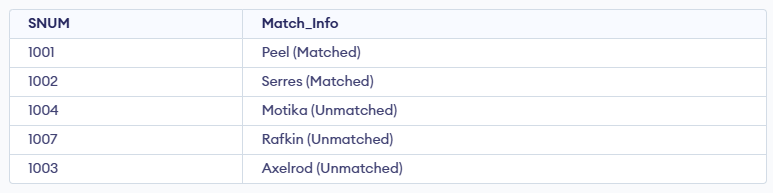
FROM SALESPEOPLE S**;**

****

**64.Find salespeople number, name and city who have multiple customers.**

**SQL**: SELECT S.SNUM, S.SNAME || ' (' || CASE WHEN EXISTS (SELECT 1 FROM CUST C WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM) THEN 'Matched' ELSE 'Unmatched' END || ')' AS Match\_Info

FROM SALESPEOPLE S;

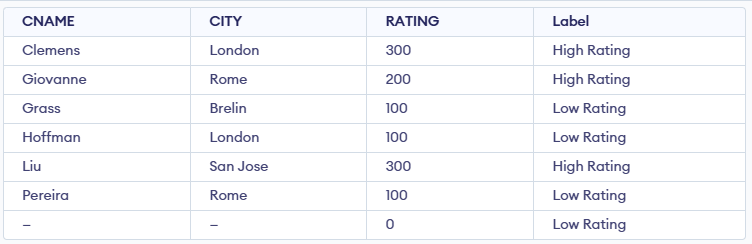


**65.Find salespeople who serve only one customer.**

**SQL:** SELECT CNAME, CITY, RATING, 'High Rating' AS Label FROM CUST WHERE RATING >= 200

UNION

SELECT CNAME, CITY, RATING, 'Low Rating' AS Label FROM CUST WHERE RATING < 200;

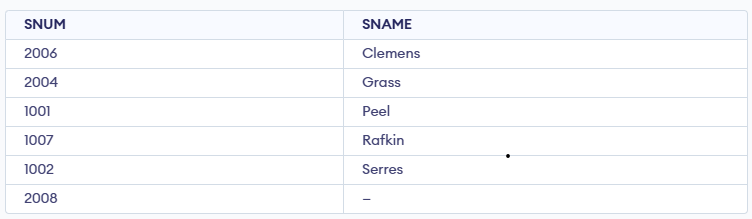


**66.Extract rows of all salespeople with more than one current order.**

**SQL**: SELECT S.SNUM, S.SNAME FROM SALESPEOPLE S WHERE S.SNUM IN (SELECT SNUM FROM ORDERS GROUP BY SNUM HAVING COUNT(\*) > 1)

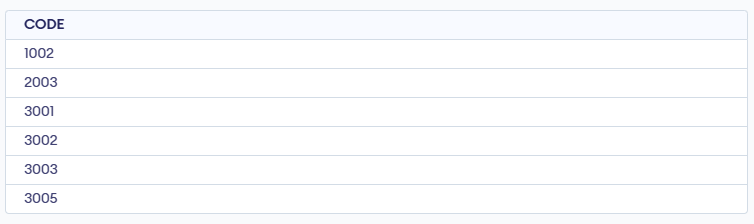
UNION

SELECT C.CNUM, C.CNAME FROM CUST C WHERE C.CNUM IN (SELECT CNUM FROM ORDERS GROUP BY CNUM HAVING COUNT(\*) > 1)ORDER BY 2;



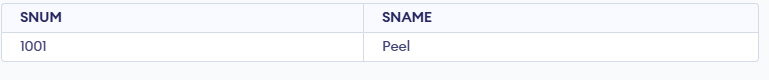
**67.Find all salespeople who have customers with a rating of 300. (use EXISTS)**

**SQL:** SELECT SNUM AS CODE FROM SALESPEOPLE WHERE CITY = 'San Jose' UNION SELECT CNUM FROM CUST WHERE CITY = 'San Jose'UNION SELECT ONUM FROM ORDERS WHERE ODATE = '1994-10-03';



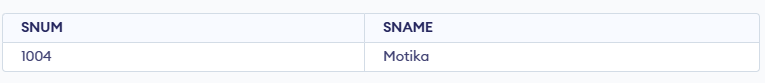
**68.Find all salespeople who have customers with a rating of 300. (use Join).**

**SQL**: SELECT DISTINCT S.SNUM, S.SNAME FROM SALESPEOPLE S, CUST C WHERE S.CITY = 'London' AND C.CITY = 'London' AND S.SNUM = C.SNUM;



**69. Select all salespeople with customers located in their cities who are not assigned to them. (use EXISTS).**

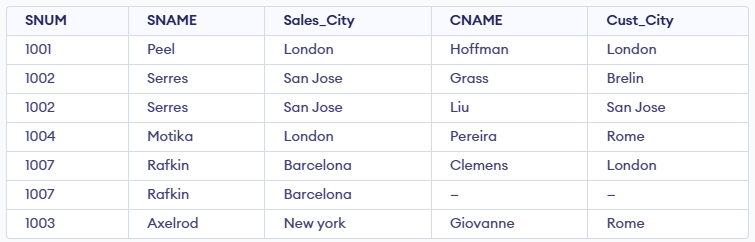
**SQL**: SELECT S.SNUM, S.SNAME FROM SALESPEOPLE S WHERE S.CITY = 'London' AND S.SNUM NOT IN (SELECT SNUM FROM CUST WHERE CITY = 'London');



**70.Extract from customers table every customer assigned the a salesperson who currently has at least one other customer ( besides the customer being selected) with orders in order table.**

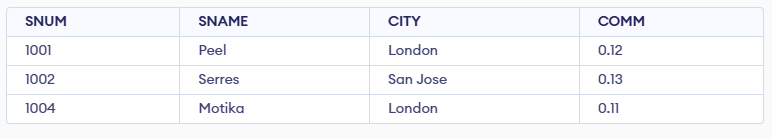
**SQL:** SELECT S.SNUM, S.SNAME, S.CITY AS Sales\_City, C.CNAME, C.CITY AS Cust\_City

FROM SALESPEOPLE S LEFT OUTER JOIN CUST C ON S.SNUM = C.SNUM;



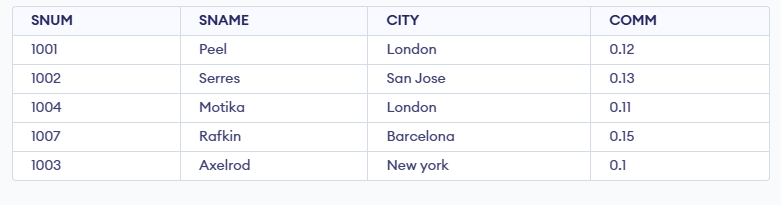
## 71. Find salespeople with customers located in their cities ( using both ANY and IN).

-- Using IN  
SELECT \* FROM SALESPEOPLE   
WHERE CITY IN (SELECT CITY FROM CUST);

  
  
-- Using ANY  
SELECT \* FROM SALESPEOPLE   
WHERE CITY = ANY (SELECT CITY FROM CUST);

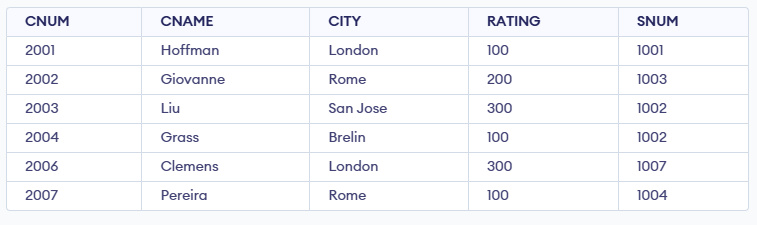
## 72. Find all salespeople for whom there are customers that follow them in alphabetical order. (Using ANY and EXISTS)

-- Using ANY  
SELECT \* FROM SALESPEOPLE S  
WHERE S.SNAME < ANY (SELECT CNAME FROM CUST);  
  
-- Using EXISTS  
SELECT \* FROM SALESPEOPLE S  
WHERE EXISTS (  
 SELECT \* FROM CUST C  
 WHERE C.CNAME > S.SNAME  
);



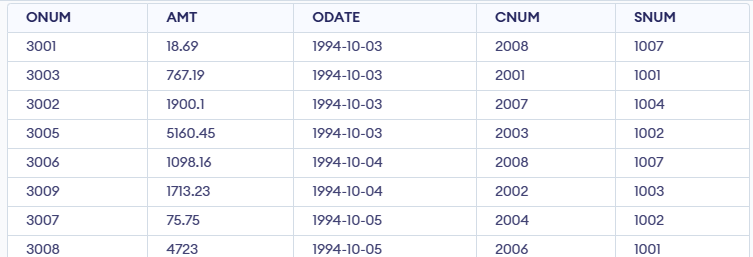
## 73. Select customers who have a greater rating than any customer in Rome.

SELECT \* FROM CUST  
WHERE RATING > EXISTS (SELECT RATING FROM CUST WHERE CITY = 'Rome');



## 74. Select all orders that had amounts that were greater than at least one of the orders from Oct 6th.

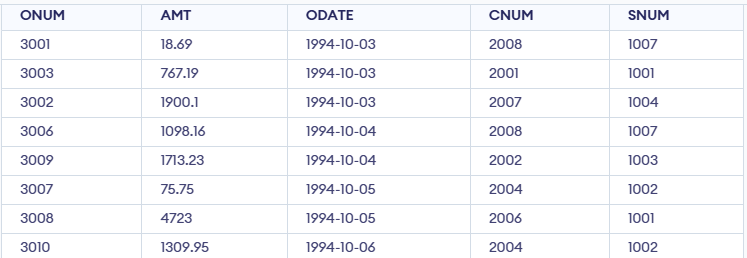
SELECT \* FROM ORDERS  
WHERE AMT > EXISTS (SELECT AMT FROM ORDERS WHERE ODATE = '1994-10-06');



## 75. Find all orders with amounts smaller than any amount for a customer in San Jose. (Both using ANY and without ANY)

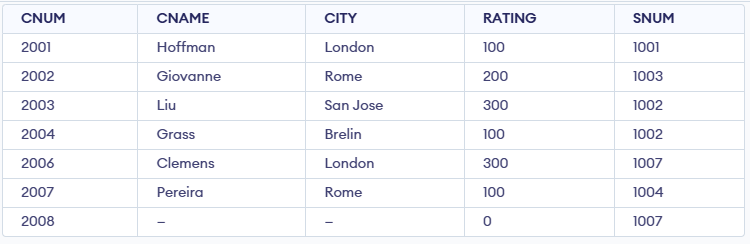
-- Using ANY  
SELECT \* FROM ORDERS  
WHERE AMT < ANY (  
 SELECT AMT FROM ORDERS O  
 JOIN CUST C ON O.CNUM = C.CNUM  
 WHERE C.CITY = 'San Jose'  
);

  
  
-- Without using ANY  
SELECT \* FROM ORDERS O1  
WHERE AMT < (  
 SELECT MIN(O2.AMT)  
 FROM ORDERS O2  
 JOIN CUST C ON O2.CNUM = C.CNUM  
 WHERE C.CITY = 'San Jose'  
);



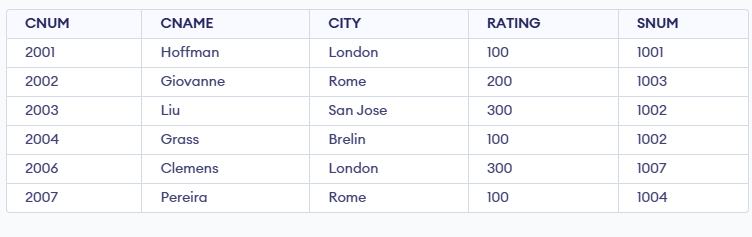
## 76. Select those customers whose ratings are higher than every customer in Paris. ( Using both ALL and NOT EXISTS).

-- Using ALL  
SELECT \* FROM CUST  
WHERE RATING > ALL (  
 SELECT RATING FROM CUST WHERE CITY = 'Paris'  
);  
  
-- Using NOT EXISTS  
SELECT \* FROM CUST C1  
WHERE NOT EXISTS (  
 SELECT \* FROM CUST C2  
 WHERE C2.CITY = 'Paris' AND C1.RATING <= C2.RATING  
);



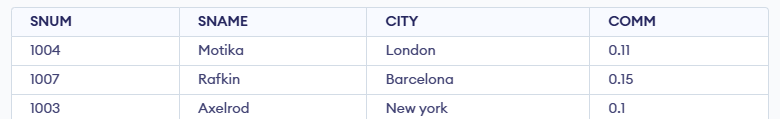
## 77. Select all customers whose ratings are equal to or greater than ANY of the Seeres.

SELECT \* FROM CUST  
WHERE RATING >= EXISTS (  
 SELECT RATING FROM CUST WHERE SNUM = (  
 SELECT SNUM FROM SALESPEOPLE WHERE SNAME = 'Serres'  
 )  
);



## 78. Find all salespeople who have no customers located in their city. ( Both using ANY and ALL)

-- Using ALL  
SELECT \* FROM SALESPEOPLE S  
WHERE CITY <> ALL (SELECT CITY FROM CUST WHERE S.SNUM = CUST.SNUM);  
  
-- Using NOT EXISTS  
SELECT \* FROM SALESPEOPLE S  
WHERE NOT EXISTS (  
 SELECT \* FROM CUST C  
 WHERE S.SNUM = C.SNUM AND S.CITY = C.CITY  
);



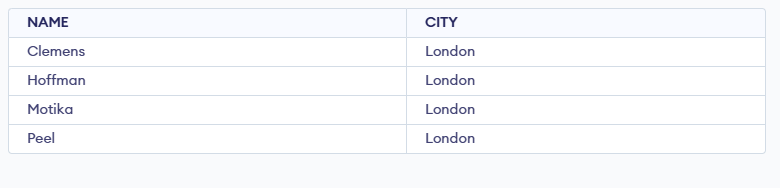
## 79. Find all orders for amounts greater than any for the customers in London.

SELECT \* FROM ORDERS  
WHERE AMT > ANY (  
 SELECT AMT FROM ORDERS O  
 JOIN CUST C ON O.CNUM = C.CNUM  
 WHERE C.CITY = 'London'  
);



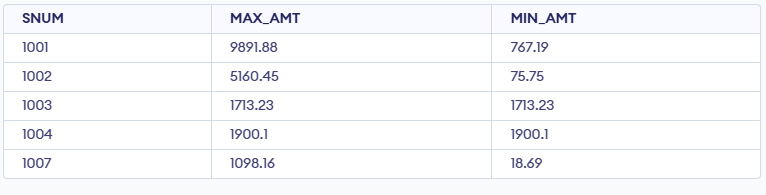
## 80. Find all salespeople and customers located in London.

SELECT SNAME AS NAME, CITY FROM SALESPEOPLE WHERE CITY = 'London'  
UNION  
SELECT CNAME AS NAME, CITY FROM CUST WHERE CITY = 'London';



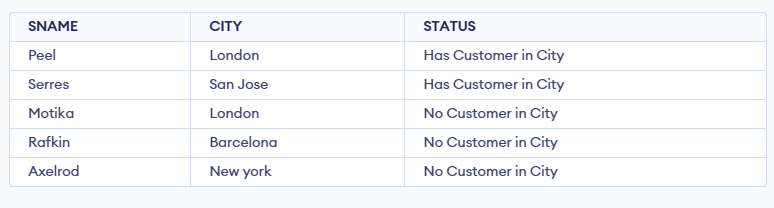
## 81. For every salesperson, dates on which highest and lowest orders were brought.

SELECT SNUM, MAX(AMT) AS MAX\_AMT, MIN(AMT) AS MIN\_AMT FROM ORDERS  
GROUP BY SNUM;



## 82. List all of the salespeople and indicate those who don’t have customers in their cities as well as those who do have.

SELECT S.SNAME, S.CITY,  
CASE  
 WHEN EXISTS (SELECT \* FROM CUST C WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM) THEN 'Has Customer in City'  
 ELSE 'No Customer in City'  
END AS STATUS  
FROM SALESPEOPLE S;

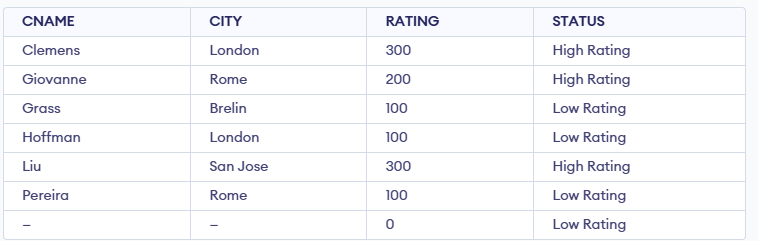


## 83. Append strings to the selected fields, indicating whether or not a given salesperson was matched to a customer in his city.

SELECT S.SNAME || ' works in ' || S.CITY || ' - ' ||  
CASE  
 WHEN EXISTS (SELECT \* FROM CUST C WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM) THEN 'Matched'  
 ELSE 'Not Matched'  
END AS INFO  
FROM SALESPEOPLE S;

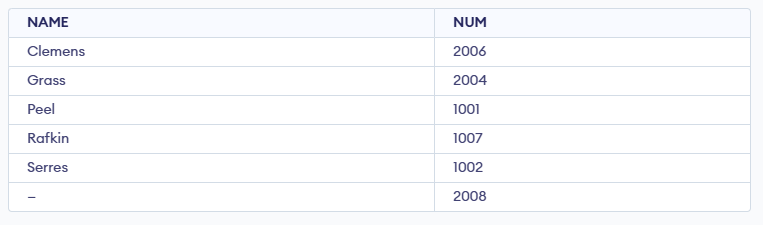


## 84. Create a union of two queries that shows the names, cities and ratings of all customers. Those with a rating of 200 or greater will also have the words ‘High Rating’, while the others will have the words ‘Low Rating’.

SELECT CNAME, CITY, RATING, 'High Rating' AS STATUS FROM CUST WHERE RATING >= 200  
UNION SELECT CNAME, CITY, RATING, 'Low Rating' FROM CUST WHERE RATING < 200; 

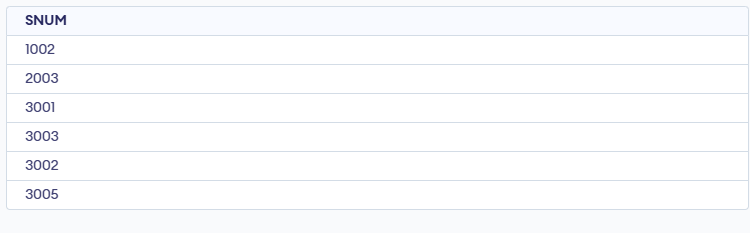
## 85. Write command that produces the name and number of each salesperson and each customer with more than one current order. Put the result in alphabetical order.

SELECT S.SNAME AS NAME, S.SNUM AS NUM FROM SALESPEOPLE S  
WHERE S.SNUM IN (SELECT SNUM FROM ORDERS GROUP BY SNUM HAVING COUNT(\*) > 1)  
UNION  
SELECT C.CNAME, C.CNUM FROM CUST C  
WHERE C.CNUM IN (SELECT CNUM FROM ORDERS GROUP BY CNUM HAVING COUNT(\*) > 1)  
ORDER BY NAME;



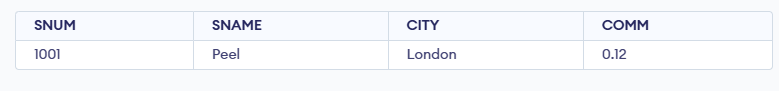
## 86. Form a union of three queries. Have the first select the snums of all salespeople in San Jose, then second the cnums of all customers in San Jose and the third the onums of all orders on Oct. 3. Retain duplicates between the last two queries, but eliminate redundancies between either of them and the first.

SELECT SNUM FROM SALESPEOPLE WHERE CITY = 'San Jose'  
UNION  
SELECT CNUM FROM CUST WHERE CITY = 'San Jose'  
UNION ALL  
SELECT ONUM FROM ORDERS WHERE ODATE = '1994-10-03';



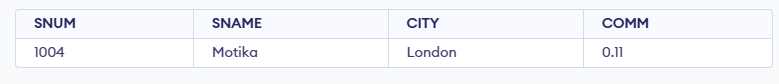
## 87. Produce all the salesperson in London who had at least one customer there.

SELECT DISTINCT S.\* FROM SALESPEOPLE S  
JOIN CUST C ON S.SNUM = C.SNUM  
WHERE S.CITY = 'London' AND C.CITY = 'London';



## 88. Produce all the salesperson in London who did not have customers there.

SELECT \* FROM SALESPEOPLE S  
WHERE S.CITY = 'London' AND NOT EXISTS (  
 SELECT \* FROM CUST C WHERE S.SNUM = C.SNUM AND C.CITY = 'London'  
);



## 89. We want to see salespeople matched to their customers without excluding those salesperson who were not currently assigned to any customers. (Use OUTER join and UNION)

SELECT S.SNAME, C.CNAME FROM SALESPEOPLE S  
LEFT JOIN CUST C ON S.SNUM = C.SNUM  
UNION  
SELECT S.SNAME, C.CNAME FROM SALESPEOPLE S  
RIGHT JOIN CUST C ON S.SNUM = C.SNUM;

