**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

2008 Peter Barcelona 100 1007

**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

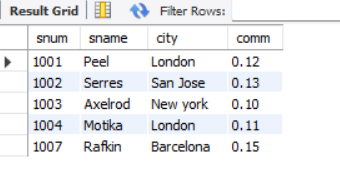
3011 9891.88 06-OCT-94 2006 1001

**QUERIES**

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

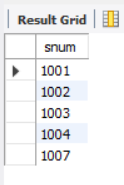
Select snum, sname, city, comm

from salespeople;



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

SELECT DISTINCT snum FROM orders;

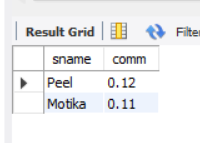


1. **Display names and commissions of all salespeople in london.**

Select sname,comm

from salespeople

where city = ‘London’;

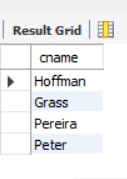


1. **All customers with rating of 100.**

Select cname

from cust

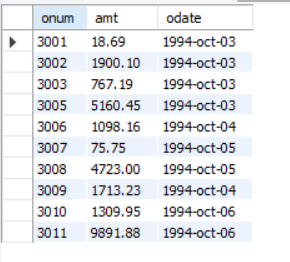
where rating = 100;



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

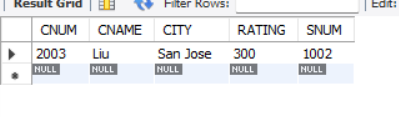
Select onum, amt, odate

from orders;



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

Select \* from cust where city=’San Jose’ AND rating > 200;



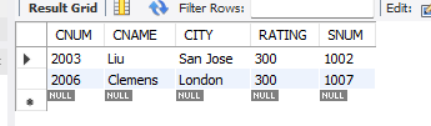
1. **All customers who were either located in San Jose or had a rating above 200.**

Select \*

from cust

where city = ‘San Jose’ OR

rating > 200;

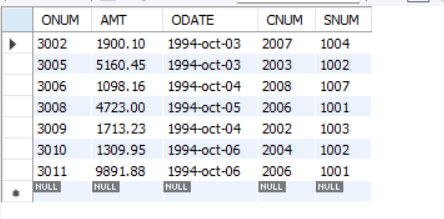


1. **All orders for more than $1000.**

Select \*

from orders

where amt > 1000;



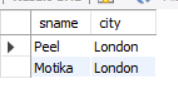
1. **Names and citires of all salespeople in london with commission above 0.10.**

Select sname, city

from salepeople

where comm > 0.10 and

city = ‘London’;



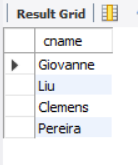
1. **All customers excluding those with rating <= 100 unless they are located in Rome.**

Select cname

from cust

where rating <= 100 or

city = ‘Rome’;

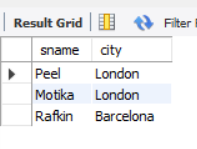


1. **All salespeople either in Barcelona or in london.**

Select sname, city

from salespeople

where city in (‘Barcelona’,’London’);

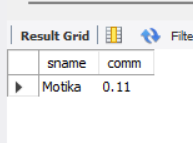
****

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

Select sname, comm

from salespeople

where comm > 0.10 and comm < 0.12;

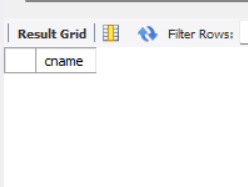


1. **All customers with NULL values in city column.**

Select cname

from cust

where city is null;

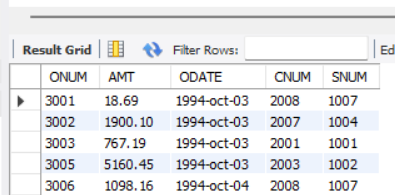


1. **All orders taken on Oct 3Rd and Oct 4th 1994.**

Select \*

from orders

where odate in (‘1994-oct-03’,’ 1994-oct-04’’);

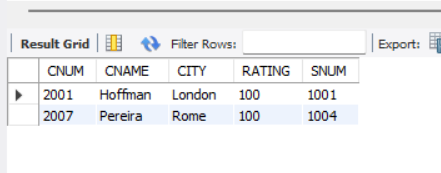


1. **All customers serviced by peel or Motika.**

SELECT c.\*

FROM cust c

JOIN salespeople s ON c.snum = s.snum

WHERE s.sname IN ('Peel', 'Motika');

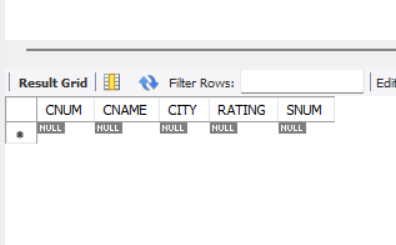
1. **All customers whose names begin with a letter from A to B.**

Select cname

from cust

where cname like ‘A%’ or

cname like ‘B%’;



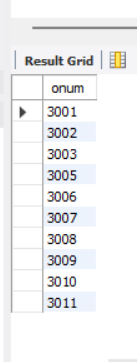
1. **All orders except those with 0 or NULL value in amt field.**

Select onum

from orders

where amt != 0 or

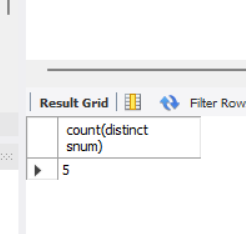
amt is not null;



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

Select count(distinct snum)

from orders;



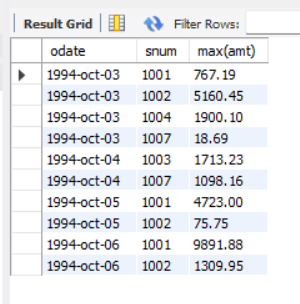
1. **Largest order taken by each salesperson, datewise.**

Select odate, snum, max(amt)

from orders

group by odate, snum

order by odate,snum;



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

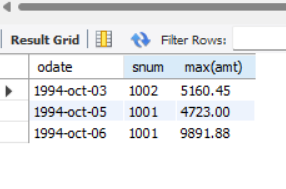
Select odate, snum, max(amt)

from orders

where amt > 3000

group by odate, snum

order by odate,snum;



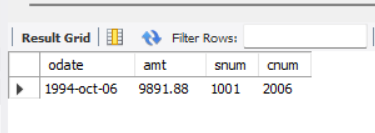
1. **Which day had the hightest total amount ordered.**

Select odate, amt, snum, cnum

from orders

where amt = (select max(amt)

from orders)

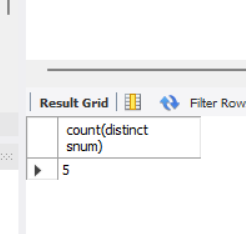


1. **Count all orders for Oct 3rd.**

Select count(\*)

from orders

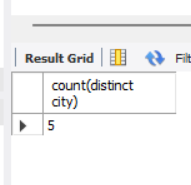
where odate = ‘1994-oct-03’;



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

Select count(distinct city)

from cust;

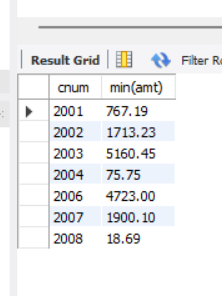


1. **Select each customer’s smallest order.**

Select cnum, min(amt)

from orders

group by cnum;

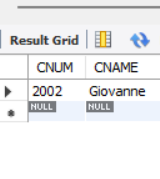


1. **First customer in alphabetical order whose name begins with G.**

Select cnum,min(cname)

from cust

where cname like ‘G%’;

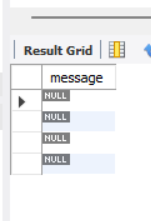


1. **Get the output like “ For dd/mm/yy there are \_\_\_ orders.**

SELECT

CONCAT('For ', DATE\_FORMAT(odate, '%d/%m/%y'), ' there are ', COUNT(\*), ' orders') AS message

FROM orders GROUP BY odate;

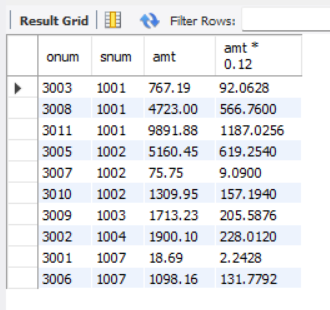


1. **Assume that each salesperson has a 12% commission. Produce order no., salesperson no., and amount of salesperson’s commission for that order.**

Select onum, snum, amt, amt \* 0.12

from orders

order by snum;



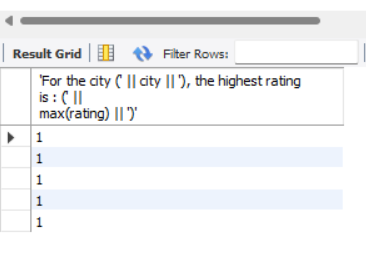
1. **Find highest rating in each city. Put the output in this form. For the city (city), the highest rating is : (rating).**

Select 'For the city (' || city || '), the highest rating is : (' ||

max(rating) || ')'

from cust

group by city;



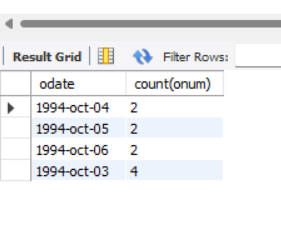
1. **Display the totals of orders for each day and place the results in descending order.**

Select odate, count(onum)

from orders

group by odate

order by count(onum);

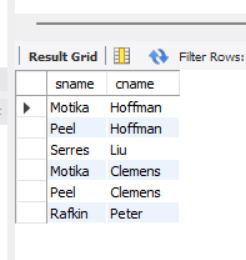


1. **All combinations of salespeople and customers who shared a city. (ie same city).**

Select sname, cname

from salespeople, cust

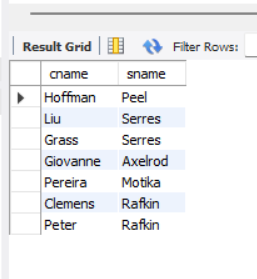
where salespeople.city = cust.city;

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

Select cname, sname

from cust, salespeople

where cust.snum = salespeople.snum;

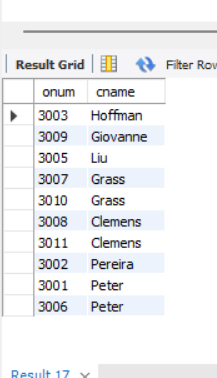


1. **List each order number followed by the name of the customer who made the order.**

Select onum, cname

from orders, cust

where orders.cnum = cust.cnum;



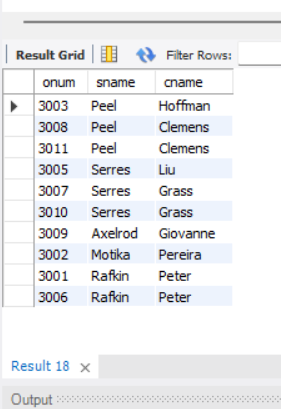
1. **Names of salesperson and customer for each order after the order number.**

Select onum, sname, cname

from orders, cust, salespeople

where orders.cnum = cust.cnum and

orders.snum = salespeople.snum;



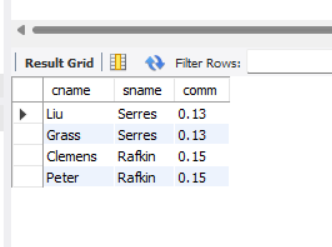
1. **Produce all customer serviced by salespeople with a commission above 12%.**

Select cname, sname, comm

from cust, salespeople

where comm > 0.12 and

cust.snum = salespeople.snum;



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

Select sname, amt \* comm

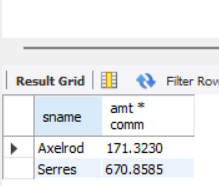
from orders, cust, salespeople

where rating > 100 and

salespeople.snum = cust.snum and

salespeople.snum = orders.snum and

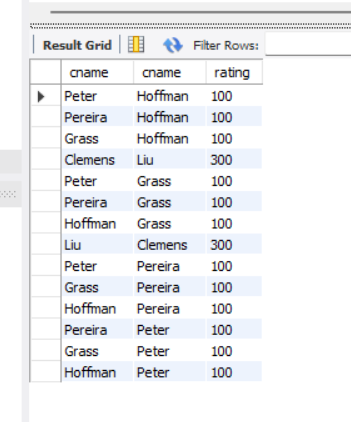
cust.cnum = orders.cnum



1. **Find all pairs of customers having the same rating.**

Select a.cname, b.cname,a.rating

from cust a, cust b

where a.rating = b.rating an a.cnum !=b.cnum

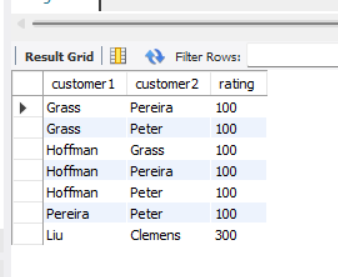
1. **Find all pairs of customers having the same rating, each pair coming once only.**

SELECT a.cname AS customer1, b.cname AS customer2, a.rating

FROM cust a

JOIN cust b ON a.rating = b.rating AND a.cnum < b.cnum

ORDER BY a.rating, customer1, customer2;

****

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

SELECT c.cname, sname

FROM cust c

JOIN (

SELECT sname

FROM salespeople

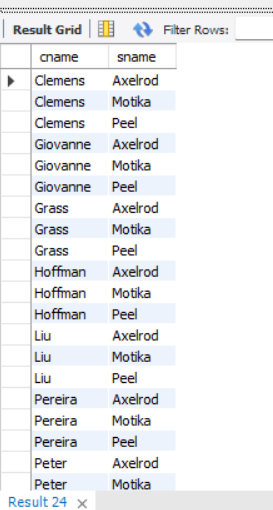
ORDER BY sname

LIMIT 3

) s\_top3

ON 1 = 1 -- Cross join

ORDER BY c.cname, s\_top3.sname;



1. **Display all customers located in cities where salesman serres has customer.**

SELECT DISTINCT c1.cname

FROM cust c1

WHERE c1.city IN (

SELECT DISTINCT c2.city

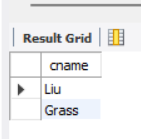
FROM cust c2

JOIN orders o ON c2.cnum = o.cnum

JOIN salespeople s ON o.snum = s.snum

WHERE s.sname = 'Serres'

);



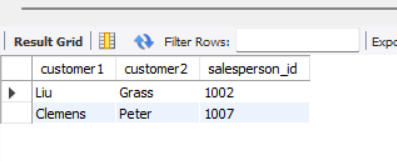
1. **Find all pairs of customers served by single salesperson.**

SELECT DISTINCT a.cname AS customer1, b.cname AS customer2, a.snum AS salesperson\_id

FROM cust a

JOIN cust b ON a.snum = b.snum AND a.cnum < b.cnum

ORDER BY a.snum, customer1, customer2;



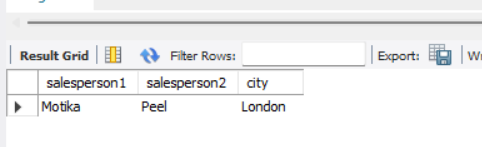
1. **Produce all pairs of salespeople which are living in the same city. Exclude combinations of salespeople with themselves as well as duplicates with the order reversed.**

SELECT a.sname AS salesperson1, b.sname AS salesperson2, a.city

FROM salespeople a

JOIN salespeople b ON a.city = b.city AND a.snum > b.snum

ORDER BY a.city, salesperson1, salesperson2;



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

SELECT

c.cname,

a.onum AS order1,

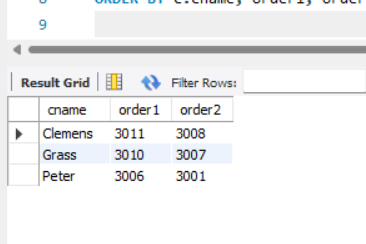
b.onum AS order2

FROM orders a

JOIN orders b ON a.cnum = b.cnum AND a.onum > b.onum

JOIN cust c ON c.cnum = a.cnum

ORDER BY c.cname, order1, order2;



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

SELECT cname, city

FROM cust

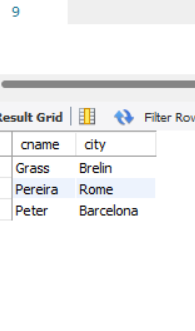
WHERE rating = (

SELECT rating

FROM cust

WHERE cname = 'Hoffman'

)

AND cname <> 'Hoffman';

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

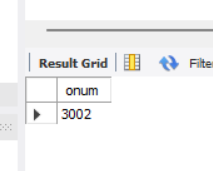
Select Onum

from orders

where snum = ( select snum

from salespeople

where sname = ‘Motika’);



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

SELECT a.onum, b.sname, c.cname, a.amt

FROM orders a

JOIN salespeople b ON a.snum = b.snum

JOIN cust c ON a.cnum = c.cnum

WHERE a.snum IN (

SELECT DISTINCT snum

FROM orders

WHERE cnum = (

SELECT cnum

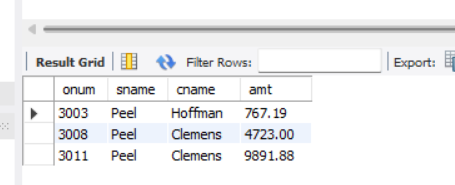
FROM cust

WHERE cname = 'Hoffman'

LIMIT 1

)

)

ORDER BY a.onum;****

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

SELECT \*

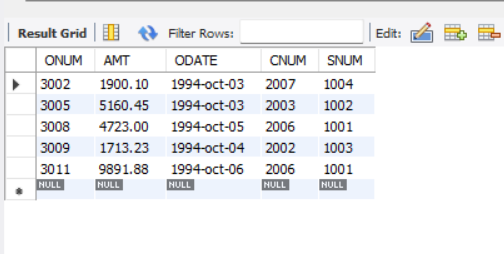
FROM orders

WHERE amt > (

SELECT AVG(amt)

FROM orders

WHERE odate = '04-OCT-94'

);

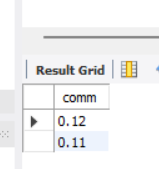
1. **Find average commission of salespeople in london.**

Select avg(comm)

from salespeople

where city = ‘London’;

Limit 1000;



1. **Find all orders attributed to salespeople servicing customers in london.**

SELECT snum, cnum, onum, amt

FROM orders

WHERE cnum IN (

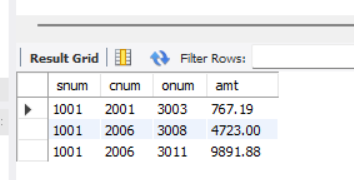
SELECT cnum

FROM cust

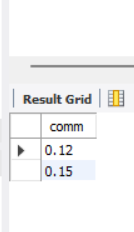
WHERE city = 'London'

)

ORDER BY snum, cnum;



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



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

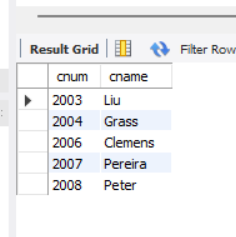
**.**

Select cnum, cname from cust

where cnum > ( select snum+1000

from salespeople

where sname = 'Serres');



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

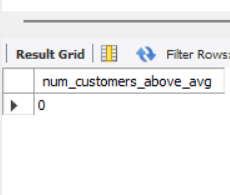
Select cnum, rating

from cust

where rating > ( select avg(rating)

from cust

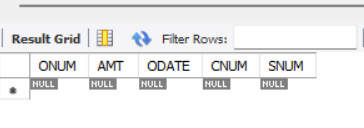
where city = 'San Jose');

****

1. **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 customer WHERE cname = 'Cisnerous');

****

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

SELECT cname, rating

FROM customer

WHERE cnum IN (

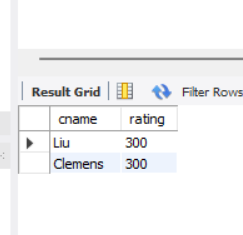
SELECT cnum

FROM orders

GROUP BY cnum

HAVING AVG(amt) > (SELECT AVG(amt) FROM orders)

);

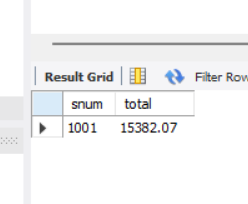
****

1. **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) AS total

FROM orders

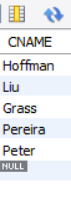
GROUP BY snum

HAVING SUM(amt) > (SELECT MAX(amt) FROM orders);

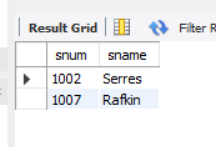
1. **Find all customers with order on 3rd Oct.**

SELECT cname FROM orders

WHERE odate = '2022-10-03';

****

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

**** SELECT snum, sname

FROM salespeople

WHERE snum IN (

SELECT snum

FROM cust

GROUP BY snum

HAVING COUNT(\*) > 1

);

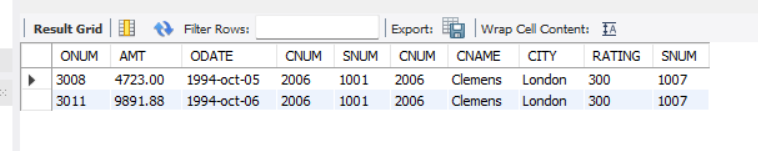
1. **Check if the correct salesperson was credited with each sale.**

SELECT \*

FROM orders o

JOIN cust c ON o.cnum = c.cnum

WHERE o.snum <> c.snum;



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

SELECT \*

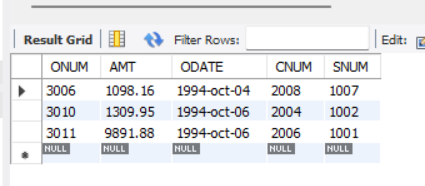
FROM orders o

WHERE amt > (

SELECT AVG(amt)

FROM orders

WHERE cnum = o.cnum

); ****

1. **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)

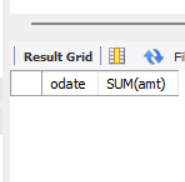
FROM orders

GROUP BY odate

HAVING SUM(amt) >= (

SELECT MAX(amt) FROM orders

) + 2000;

****

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

SELECT cname, cnum

FROM cust c

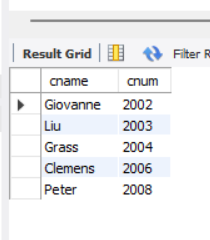
WHERE rating = (

SELECT MAX(rating)

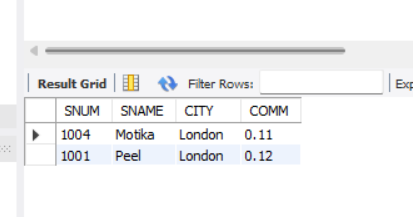
FROM cust

WHERE city = c.city

);



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



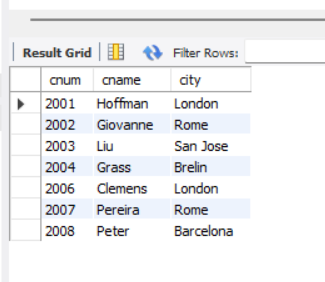
SELECT DISTINCT s.\*

FROM salespeople s

JOIN cust c ON s.city = c.city

WHERE s.snum <> c.snum;

1. **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 cnum, cname, city

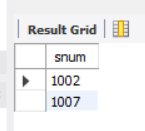
FROM cust

WHERE EXISTS (

SELECT 1 FROM cust WHERE city = 'San Jose'

);

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



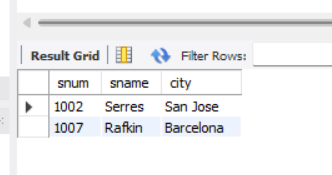
SELECT snum

FROM cust

GROUP BY snum

HAVING COUNT(\*) > 1;

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

****

SELECT s.snum, s.sname, s.city

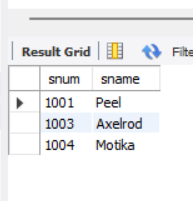
FROM salespeople s

WHERE s.snum IN (

SELECT snum FROM cust GROUP BY snum HAVING COUNT(\*) > 1

);

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

****

SELECT snum, sname

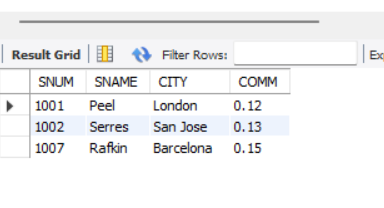
FROM salespeople

WHERE snum IN (

SELECT snum FROM cust GROUP BY snum HAVING COUNT(\*) = 1

);

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

****

SELECT s.\*

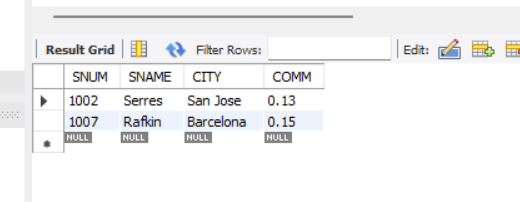
FROM salespeople s

JOIN orders o ON s.snum = o.snum

GROUP BY s.snum

HAVING COUNT(o.onum) > 1;

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

****

SELECT \*

FROM salespeople s

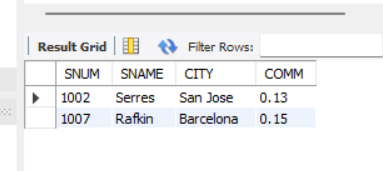
WHERE EXISTS (

SELECT 1 FROM cust c

WHERE c.snum = s.snum AND c.rating = 300

);

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

****

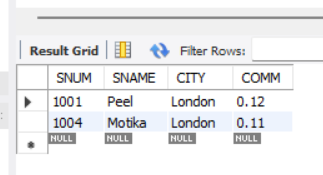
SELECT DISTINCT s.\*

FROM salespeople s

JOIN cust c ON s.snum = c.snum

WHERE c.rating = 300;

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

****

SELECT \*

FROM salespeople s

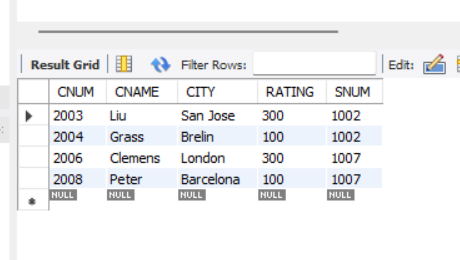
WHERE EXISTS (

SELECT 1 FROM cust c

WHERE c.city = s.city AND c.snum <> s.snum

);

1. **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.**

 SELECT \*

FROM cust c1

WHERE EXISTS (

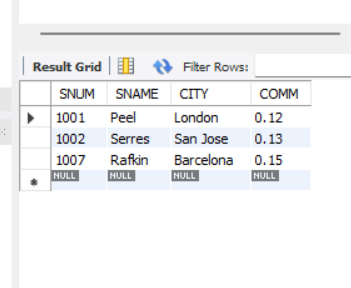
SELECT 1 FROM cust c2

JOIN orders o ON c2.cnum = o.cnum

WHERE c1.snum = c2.snum AND c1.cnum <> c2.cnum

);

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

**** SELECT \*

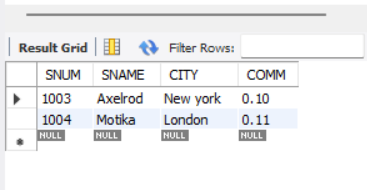
FROM salespeople s

WHERE city = ANY (

SELECT city FROM cust WHERE s.snum = cust.snum

);

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

****

SELECT \*

FROM salespeople s

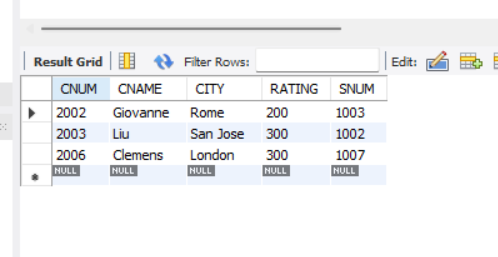
WHERE EXISTS (

SELECT 1 FROM cust c

WHERE c.snum = s.snum AND c.cname > s.sname

);

1. **Select customers who have a greater rating than any customer in rome.**



SELECT \*

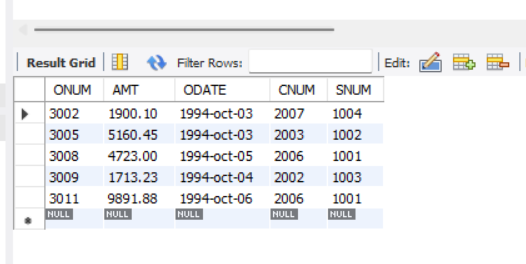
FROM cust

WHERE rating > ANY (

SELECT rating FROM cust WHERE city = 'Rome'

);

1. **Select all orders that had amounts that were greater that atleast one of the orders from Oct 6th.**

****

SELECT \*

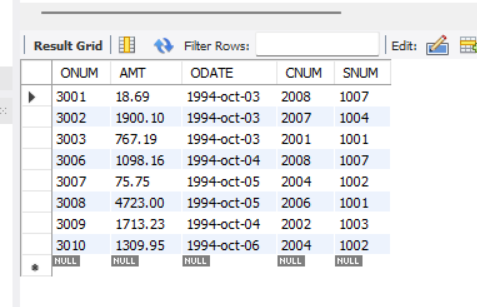
FROM orders

WHERE amt > ANY (

SELECT amt FROM orders WHERE odate = '1994-oct-06'

);

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

****

SELECT \*

FROM orders

WHERE amt < ANY (

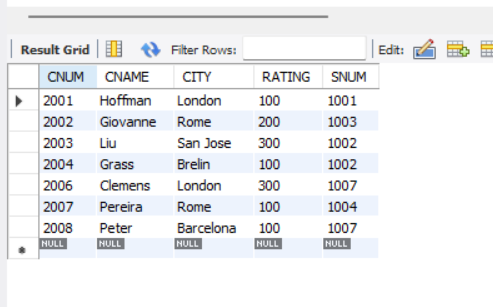
SELECT amt FROM orders

JOIN cust ON orders.cnum = cust.cnum

WHERE cust.city = 'San Jose'

);

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



SELECT \*

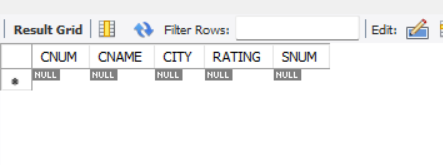
FROM cust

WHERE rating > ALL (

SELECT rating FROM cust WHERE city = 'Paris'

);

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



SELECT \*

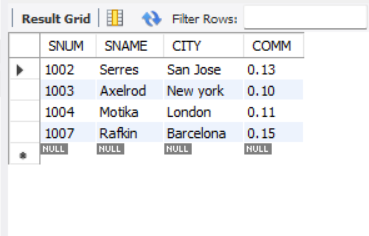
FROM cust

WHERE rating >= ANY (

SELECT rating FROM cust WHERE cname LIKE '%Seeres%'

);

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



SELECT \*

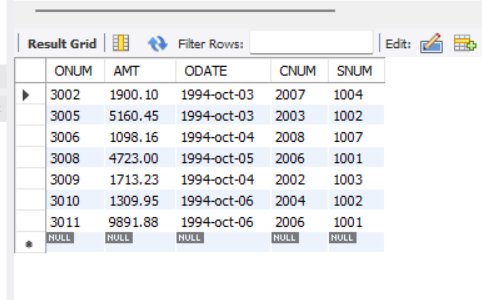
FROM salespeople s

WHERE city <> ANY (

SELECT city FROM cust WHERE s.snum = cust.snum

);

1. **Find all orders for amounts greater than any for the customers in London.**

****

SELECT \*

FROM orders

WHERE amt > ANY (

SELECT amt FROM orders

JOIN cust ON orders.cnum = cust.cnum

WHERE cust.city = 'London'

);

1. **Find all salespeople and customers located in london.**

SELECT snum AS cnum, sname AS name, city, 'Salesperson' AS role

FROM salespeople

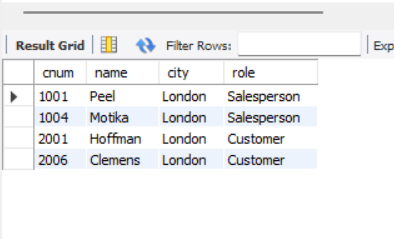
WHERE city = 'London'

UNION

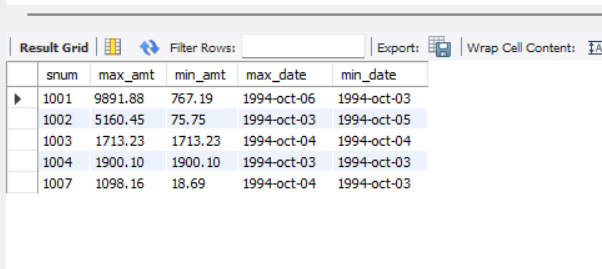
SELECT cnum AS cnum, cname AS name, city, 'Customer' AS role

FROM cust

WHERE city = 'London';



1. **For every salesperson, dates on which highest and lowest orders were brought.**

****

SELECT s.snum,

MAX(o.amt) AS max\_amt,

MIN(o.amt) AS min\_amt,

(SELECT o1.odate FROM orders o1 WHERE o1.snum = s.snum AND o1.amt = MAX(o.amt)) AS max\_date,

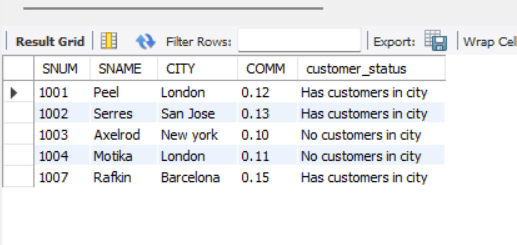
(SELECT o2.odate FROM orders o2 WHERE o2.snum = s.snum AND o2.amt = MIN(o.amt)) AS min\_date

FROM salespeople s

JOIN orders o ON s.snum = o.snum

GROUP BY s.snum;

1. **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.\*,

CASE

WHEN EXISTS (SELECT 1 FROM cust c WHERE c.snum = s.snum AND c.city = s.city)

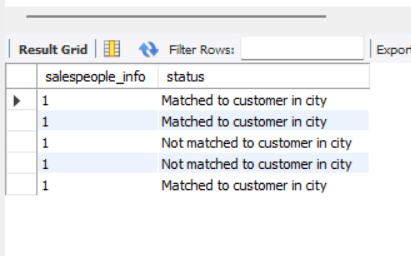
THEN 'Has customers in city'

ELSE 'No customers in city'

END AS customer\_status

FROM salespeople s;

1. **Append strings to the selected fields, indicating weather or not a given salesperson was matched to a customer in his city.**

****

SELECT s.sname || ' (' || s.snum || ')' AS salespeople\_info,

CASE

WHEN EXISTS (

SELECT 1 FROM cust c WHERE c.snum = s.snum AND c.city = s.city

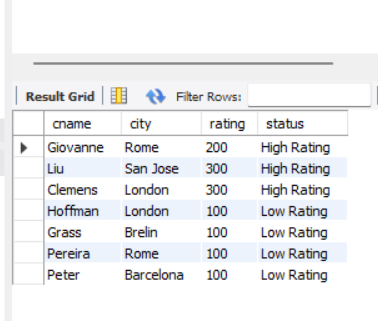
) THEN 'Matched to customer in city'

ELSE 'Not matched to customer in city'

END AS status

FROM salespeople s;

1. **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

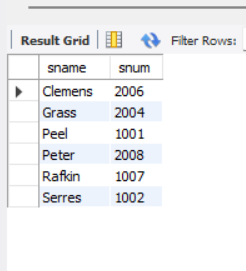
UNION

SELECT cname, city, rating, 'Low Rating'

FROM cust

WHERE rating < 200;

1. **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, s.snum

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

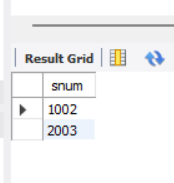
WHERE c.cnum IN (

SELECT cnum FROM orders GROUP BY cnum HAVING COUNT(\*) > 1

)

ORDER BY 1;

1. **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 eliminates and redundancies between either of them and the first.**



SELECT snum FROM salespeople WHERE city = 'San Jose'

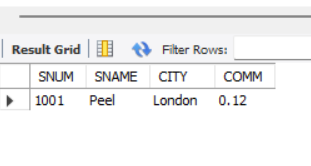
UNION

SELECT cnum FROM cust WHERE city = 'San Jose'

UNION ALL

SELECT onum FROM orders WHERE odate = '2022-10-03';

1. **Produce all the salesperson in London who had at least one customer there.**

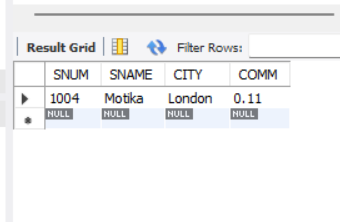


SELECT DISTINCT s.\*

FROM salespeople s

JOIN cust c ON s.city = 'London' AND c.city = 'London' AND s.snum = c.snum;

1. **Produce all the salesperson in London who did not have customers there.**



SELECT \*

FROM salespeople s

WHERE city = 'London' AND NOT EXISTS (

SELECT 1 FROM cust c

WHERE c.city = 'London' AND c.snum = s.snum

);

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

SELECT s.snum, s.sname, s.city, c.cnum, c.cname, c.city

FROM salespeople s

LEFT JOIN cust c ON s.snum = c.snum AND s.city = c.city

UNION

SELECT s.snum, s.sname, s.city, NULL, NULL, NULL

FROM salespeople s

WHERE s.snum NOT IN (

SELECT DISTINCT snum FROM cust

);

