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

SQL> select a.cname,b.onum from customer a, orders b where a.cnum=b.cnum and a.cname='cisneros' order by a.cname;

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

SQL> select cname,rating from customers where rating > (select avg(rating) from customers);

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

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

select \* from customer a, orders b where a.customer\_id=b.customer\_id and b.ord\_date=’oct-03';

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

select salesman\_id,name

from salesman a

where 1 <

(select count(\*)

from customer

where salesman\_id=a.salesman\_id);

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

Select onum, a.cnum, a.snum, b.snum

from orders a, cust b

where a.cnum = b.cnum and

a.snum != b.snum;

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

SELECT \*

FROM orders a

WHERE purch\_amt > (SELECT AVG(purch\_amt)

FROM orders b

WHERE b.customer\_id = a.customer\_id);

**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 ord\_date,

SUM(purch\_amt)

FROM orders a

GROUP BY ord\_date

HAVING SUM(purch\_amt) > (SELECT MAX(purch\_amt) + 2000

FROM orders b

WHERE a.ord\_date = b.ord\_date);

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

select cnum,cname,rating from customer where rating>=(select max(rating) from customer);

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

Select distinct cname

from cust a, salespeople b

where a.city = b.city and

a.snum != b.snum;

Select cname

from cust

where cname in ( select cname

from cust a, salespeople b

where a.city = b.city and

a.snum != b.snum );

**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 \* from cust

where 2 < (select count(\*)

from cust

where city = 'San Jose');

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

SELECT \*

FROM salesman

WHERE salesman\_id IN (

SELECT DISTINCT salesman\_id

FROM customer a

WHERE EXISTS (

SELECT \*

FROM customer b

WHERE b.salesman\_id=a.salesman\_id

AND b.cust\_name<>a.cust\_name));

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

SELECT salesman\_id,name, city

FROM salesman a

WHERE 1 <

(SELECT COUNT(\*)

FROM customer

WHERE salesman\_id=a.salesman\_id);

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

SELECT \*

FROM salesman

WHERE salesman\_id IN (

SELECT DISTINCT salesman\_id

FROM customer a

WHERE NOT EXISTS (

SELECT \* FROM customer b

WHERE a.salesman\_id=b.salesman\_id

AND a.cust\_name<>b.cust\_name));

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

SELECT \*

FROM salesman

WHERE salesman\_id IN (SELECT salesman\_id

FROM customer

WHERE customer\_id IN (SELECT customer\_id

FROM orders

GROUP BY customer\_id

HAVING COUNT(\*) > 1));

SELECT \*

FROM salesman a

WHERE EXISTS (SELECT \*

FROM customer b

WHERE a.salesman\_id=b.salesman\_id

AND 1 < (SELECT COUNT(\*)

FROM orders

WHERE orders.customer\_id=b.customer\_id));

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

Select a.snum

from salespeople a

where exists ( select b.snum

from cust b

where b.rating = 300 and

a.snum = b.snum)

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

Select a.snum

from salespeople a, cust b

where b.rating = 300 and

a.snum = b.snum;

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

select a.sname,a.city,b.cname,b.city as cus\_city from salespeople a,customer b where a.snum=b.snum and a.city!=b.city;

**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 a.cnum, max(c.cname)

from orders a, cust c

where a.cnum = c.cnum

group by a.cnum,a.snum

having count(\*) < ( select count(\*)

from orders b

where a.snum = b.snum)

order by a.cnum;

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

Select sname

from salespeople

where snum in ( select snum from cust

where salespeople.city = cust.city and

salespeople.snum = cust.snum);

Select sname

from salespeople

where snum = any ( select snum

from cust

where salespeople.city = cust.city and

salespeople.snum = cust.snum);

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

Select sname

from salespeople

where sname < any ( select cname

from cust

where salespeople.snum = cust.snum);

Select sname

from salespeople

where exists ( select cname

from cust

where salespeople.snum = cust.snum and

salespeople.sname < cust.cname);

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

select \* from customer where rating>(select max(rating) from customer where city='Rome');

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

SELECT \*

FROM Orders

WHERE purch\_amt > ANY

(SELECT purch\_amt

FROM orders

WHERE ord\_date='06/10');

**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 purch\_amt < ANY

(SELECT purch\_amt

FROM orders a, customer b

WHERE a.customer\_id=b.customer\_id

AND b.city='San Jose');

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

Select \* from cust

where rating > any (select rating from cust

where city = 'Paris');

Select \*

from cust a

where not exists ( select b.rating from cust b

where b.city != 'Paris' and

b.rating > a.rating);

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

Select cname, sname

from cust, salespeople

where rating >= any ( select rating

from cust

where snum = (select snum

from salespeople

where sname = 'Serres'))

and sname != 'Serres'

and salespeople.snum(+) = cust.snum;

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

Select sname

from salespeople

where snum in ( select snum

from cust

where salespeople.city != cust.city and

salespeople.snum = cust.snum);

Select sname

from salespeople

where snum = any ( select snum

from cust

where salespeople.city != cust.city and

salespeople.snum = cust.snum);

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

Select onum, amt

from orders

where amt > any ( select amt

from orders, cust

where city = ‘London’ and

orders.cnum = cust.cnum);

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

Select sname, cname

from cust, salespeople

where cust.city = 'London' and

salespeople.city = 'London' and

cust.snum = salespeople.snum;

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

Select a.amt, a.odate, b.amt, b.odate

from orders a, orders b

where (a.amt, b.amt) in (select max(amt), min(amt)

from orders

group by snum);

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

Select snum, city, 'Customer Present'

from salespeople a

where exists ( select snum from cust

where a.snum = cust.snum and

a.city = cust.city)

UNION

select snum, city, 'Customer Not Present'

from salespeople a

where exists ( select snum from cust c

where a.snum = c.snum and

a.city != c.city and

c.snum not in ( select snum

from cust

where a.snum = cust.snum and

a.city = cust.city));

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

Select a.cname, decode(a.city,b.city,'Matched','Not Matched')

from cust a, salespeople b

where a.snum = b.snum;

**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, cities, rating, ‘Higher Rating’

from cust

where rating >= 200

UNION

Select cname, cities, rating, ‘Lower Rating’

from cust

where rating < 200;

**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 'Customer Number ' || cnum "Code ",count(\*)

from orders

group by cnum

having count(\*) > 1

UNION

select 'Salesperson Number '||snum,count(\*)

from orders

group by snum

having count(\*) > 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 'Customer Number ' || cnum "Code "

from cust

where city = 'San Jose'

UNION

select 'Salesperson Number '||snum

from salespeople

where city = 'San Jose'

UNION ALL

select 'Order Number '|| onum

from Orders

where odate = '03-OCT-94';

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

Select snum, sname

from salespeople

where snum in ( select snum

from cust

where cust.snum = salespeople.snum and

cust.city = 'London')

and city = ‘London’;

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

Select snum, sname

from salespeople

where snum in ( select snum

from cust

where cust.snum = salespeople.snum and

cust.ci

**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 sname, cname

from cust, salespeople

where cust.snum(+) = salespeople.snum;

Select sname, cname

from cust, salespeople

where cust.snum = salespeople.snum

UNION

select distinct sname, 'No Customer'

from cust, salespeople

where 0 = (select count(\*) from cust

where cust.snum = salespeople.snum);