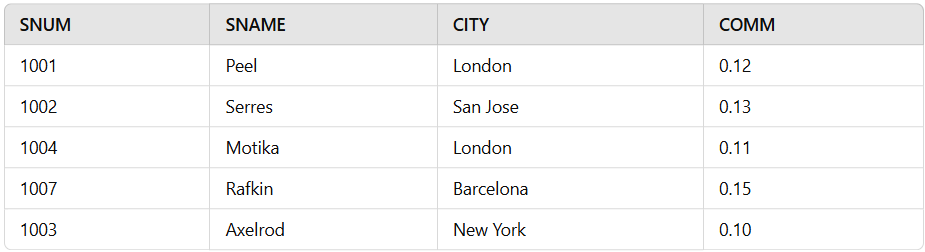
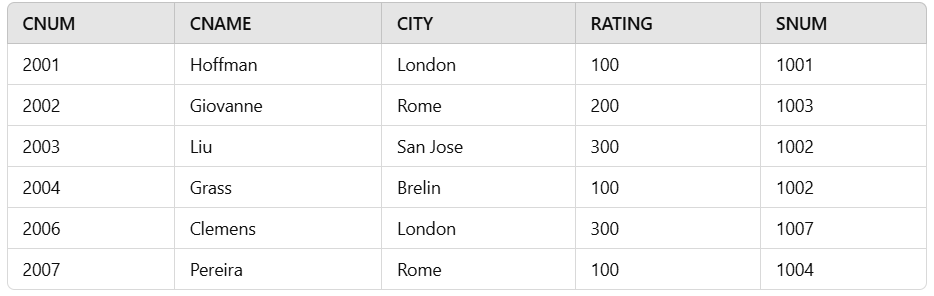
Table salespeople  
  
  
  
Table cust  
  
  
Table orders  
  
  
**Display snum,sname,city and comm of all salespeople.**SELECT SNUM, SNAME, CITY, COMM

FROM SALESPEOPLE;

**Display all snum without duplicates from all orders**.  
  
SELECT DISTINCT SNUM

FROM ORDERS;

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

SELECT SNAME, COMM

FROM SALESPEOPLE

WHERE CITY = 'London';

**All customers with rating of 100.**

SELECT CNUM, CNAME, CITY, RATING

FROM CUST

WHERE RATING = 100;

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

SELECT ONUM, AMT, ODATE

FROM ORDERS;  
**All customers excluding those with rating <= 100 unless they are located in Rome.**  
  
SELECT CNUM, CNAME, CITY, RATING

FROM CUST

WHERE (RATING > 100) OR (CITY = 'Rome');

**All salespeople either in Barcelona or in london**.  
  
SELECT SNUM, SNAME, CITY, COMM

FROM SALESPEOPLE

WHERE CITY IN ('Barcelona', 'London');

**All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded)**  
  
SELECT SNUM, SNAME, CITY, COMM

FROM SALESPEOPLE

WHERE COMM > 0.10 AND COMM < 0.12;

All customers with NULL values in city column.  
SELECT CNUM, CNAME, CITY, RATING

FROM CUST

WHERE CITY IS NULL;

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

SELECT ONUM, AMT, ODATE

FROM ORDERS

WHERE ODATE IN ('03-OCT-94', '04-OCT-94');

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

SELECT CNUM, CNAME, CITY, RATING

FROM CUST

WHERE SNUM IN (1001, 1004); -- Peel has SNUM 1001 and Motika has SNUM 1004

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

SELECT CNUM, CNAME, CITY, RATING

FROM CUST

WHERE SNUM IN (1001, 1004); -- Peel has SNUM 1001 and Motika has SNUM 1004

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

SELECT ONUM, AMT, ODATE

FROM ORDERS

WHERE AMT > 0 OR AMT IS NOT NULL;

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

SELECT COUNT(DISTINCT SNUM)

FROM ORDERS;

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

SELECT SNUM, MAX(AMT) AS MAX\_ORDER, ODATE

FROM ORDERS

GROUP BY SNUM, ODATE;

**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;

**Which day had the highest total amount ordered.**

SELECT ODATE, SUM(AMT) AS TOTAL\_AMOUNT

FROM ORDERS

GROUP BY ODATE

ORDER BY TOTAL\_AMOUNT DESC

FETCH FIRST 1 ROWS ONLY;

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

SELECT COUNT(\*) AS ORDER\_COUNT

FROM ORDERS

WHERE ODATE = '03-OCT-94';

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

SELECT COUNT(DISTINCT CITY) AS DISTINCT\_CITIES

FROM CUST

WHERE CITY IS NOT NULL;

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

SELECT CNUM, MIN(AMT) AS SMALLEST\_ORDER

FROM ORDERS

GROUP BY CNUM;

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

SELECT CNUM, CNAME, CITY, RATING

FROM CUST

WHERE CNAME LIKE 'G%'

ORDER BY CNAME

FETCH FIRST 1 ROWS ONLY;

**Get the output like “ For dd/mm/yy there are \_\_\_ orders.**  
SELECT 'For ' || TO\_CHAR(ODATE, 'DD/MM/YY') || ' there are ' || COUNT(\*) || ' orders' AS ORDER\_COUNT

FROM ORDERS

GROUP BY ODATE;

**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 \* 0.12 AS COMMISSION

FROM ORDERS;  
  
**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) AS RATING\_INFO

FROM CUST

GROUP BY CITY;

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

SELECT ODATE, SUM(AMT) AS TOTAL\_AMOUNT

FROM ORDERS

GROUP BY ODATE

ORDER BY TOTAL\_AMOUNT DESC;

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

SELECT S.SNUM, S.SNAME, C.CNUM, C.CNAME, C.CITY

FROM SALESPEOPLE S

JOIN CUST C ON S.CITY = C.CITY;

**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;

**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;

**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;

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

SELECT C.CNUM, C.CNAME, C.CITY, C.RATING

FROM CUST C

JOIN SALESPEOPLE S ON C.SNUM = S.SNUM

WHERE S.COMM > 0.12;

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

SELECT O.ONUM, O.AMT, O.AMT \* 0.12 AS COMMISSION

FROM ORDERS O

JOIN CUST C ON O.CNUM = C.CNUM

WHERE C.RATING > 100;

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

SELECT C1.CNAME AS CUSTOMER1, C2.CNAME AS CUSTOMER2, C1.RATING

FROM CUST C1, CUST C2

WHERE C1.RATING = C2.RATING AND C1.CNUM < C2.CNUM;

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

SELECT C1.CNAME AS CUSTOMER1, C2.CNAME AS CUSTOMER2, C1.RATING

FROM CUST C1, CUST C2

WHERE C1.RATING = C2.RATING AND C1.CNUM < C2.CNUM;

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

SELECT C.CNUM, S1.SNUM AS SALESPERSON1, S2.SNUM AS SALESPERSON2, S3.SNUM AS SALESPERSON3

FROM CUST C

JOIN SALESPEOPLE S1 ON C.SNUM = S1.SNUM

JOIN SALESPEOPLE S2 ON C.SNUM = S2.SNUM

JOIN SALESPEOPLE S3 ON C.SNUM = S3.SNUM

WHERE S1.SNUM <> S2.SNUM AND S1.SNUM <> S3.SNUM AND S2.SNUM <> S3.SNUM;

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

SELECT C.CNUM, C.CNAME, C.CITY

FROM CUST C

WHERE C.CITY IN (SELECT C.CITY

FROM CUST C

JOIN SALESPEOPLE S ON C.SNUM = S.SNUM

WHERE S.SNAME = 'Serres');

**Find all pairs of customers served by single salesperson**SELECT C1.CNAME AS CUSTOMER1, C2.CNAME AS CUSTOMER2, C.SNUM

FROM CUST C1, CUST C2

JOIN CUST C ON C1.SNUM = C2.SNUM

WHERE C1.CNUM < C2.CNUM;

**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 AS SALESPERSON1, S2.SNAME AS SALESPERSON2, S1.CITY

FROM SALESPEOPLE S1, SALESPEOPLE S2

WHERE S1.CITY = S2.CITY AND S1.SNUM < S2.SNUM;

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

SELECT O1.ONUM AS ORDER1, O2.ONUM AS ORDER2, C.CNAME

FROM ORDERS O1, ORDERS O2

JOIN CUST C ON O1.CNUM = C.CNUM AND O2.CNUM = C.CNUM

WHERE O1.ONUM < O2.ONUM;

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

SELECT C.CNAME, C.CITY

FROM CUST C

WHERE C.RATING = (SELECT RATING FROM CUST WHERE CNAME = 'Hoffman');

**Extract all the orders of Motika.**

SELECT O.ONUM, O.AMT, O.ODATE

FROM ORDERS O

JOIN SALESPEOPLE S ON O.SNUM = S.SNUM

WHERE S.SNAME = 'Motika';  
**All orders credited to the same salesperson who services Hoffman.**

SELECT O.ONUM, O.AMT, O.ODATE

FROM ORDERS O

JOIN CUST C ON O.CNUM = C.CNUM

WHERE C.CNAME = 'Hoffman' AND O.SNUM = (SELECT SNUM FROM CUST WHERE CNAME = 'Hoffman');

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

SELECT O.ONUM, O.AMT, O.ODATE

FROM ORDERS O

WHERE O.ODATE = '04-OCT-94' AND O.AMT > (SELECT AVG(AMT) FROM ORDERS WHERE ODATE = '04-OCT-94');

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

SELECT AVG(COMM) AS AVG\_COMM

FROM SALESPEOPLE

WHERE CITY = 'London';

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

SELECT O.ONUM, O.AMT, O.ODATE

FROM ORDERS O

JOIN CUST C ON O.CNUM = C.CNUM

JOIN SALESPEOPLE S ON O.SNUM = S.SNUM

WHERE S.CITY = 'London';

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

SELECT S.SNUM, S.SNAME, S.COMM

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

WHERE C.CITY = 'London';

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

SELECT C.CNUM, C.CNAME

FROM CUST C

WHERE C.CNUM = (SELECT SNUM + 1000 FROM SALESPEOPLE WHERE SNAME = 'Serres');

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

SELECT COUNT(\*) AS CUSTOMER\_COUNT

FROM CUST

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

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

SELECT O.ONUM, O.AMT, O.ODATE

FROM ORDERS O

JOIN CUST C ON O.CNUM = C.CNUM

WHERE C.CNAME = 'Cisnerous';

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

SELECT C.CNAME, C.RATING

FROM CUST C

WHERE C.RATING > (SELECT AVG(RATING) FROM CUST);

**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 O.SNUM, SUM(O.AMT) AS TOTAL\_AMOUNT

FROM ORDERS O

GROUP BY O.SNUM

HAVING SUM(O.AMT) > (SELECT MAX(AMT) FROM ORDERS);

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

SELECT C.CNAME, O.ONUM, O.AMT, O.ODATE

FROM ORDERS O

JOIN CUST C ON O.CNUM = C.CNUM

WHERE O.ODATE = '03-OCT-94';

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

SELECT S.SNUM, S.SNAME

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

GROUP BY S.SNUM, S.SNAME

HAVING COUNT(C.CNUM) > 1;

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

SELECT O.ONUM, O.SNUM, C.SNUM AS CORRECT\_SNUM

FROM ORDERS O

JOIN CUST C ON O.CNUM = C.CNUM

WHERE O.SNUM <> C.SNUM;

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

SELECT O.ONUM, O.AMT, O.ODATE

FROM ORDERS O

JOIN CUST C ON O.CNUM = C.CNUM

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

**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) AS TOTAL\_AMOUNT

FROM ORDERS

GROUP BY ODATE

HAVING SUM(AMT) >= (SELECT MAX(AMT) FROM ORDERS) + 2000;

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

SELECT C.CNUM, C.CNAME

FROM CUST C

WHERE C.RATING = (SELECT MAX(RATING) FROM CUST WHERE CITY = C.CITY);

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

SELECT S.SNAME

FROM SALESPEOPLE S

JOIN CUST C ON S.CITY = C.CITY

WHERE S.SNUM <> C.SNUM;  
  
SELECT S.SNAME

FROM SALESPEOPLE S

WHERE EXISTS (

SELECT 1

FROM CUST C

WHERE C.CITY = S.CITY AND S.SNUM <> C.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 C.CNUM, C.CNAME, C.CITY

FROM CUST C

WHERE C.CITY = 'San Jose';

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

SELECT S.SNUM

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

GROUP BY S.SNUM

HAVING COUNT(C.CNUM) > 1;

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

SELECT S.SNUM, S.SNAME, S.CITY

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

GROUP BY S.SNUM, S.SNAME, S.CITY

HAVING COUNT(C.CNUM) > 1;

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

SELECT S.SNUM, S.SNAME

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

GROUP BY S.SNUM, S.SNAME

HAVING COUNT(C.CNUM) = 1;

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

SELECT S.SNUM, S.SNAME

FROM SALESPEOPLE S

JOIN ORDERS O ON S.SNUM = O.SNUM

GROUP BY S.SNUM, S.SNAME

HAVING COUNT(O.ONUM) > 1;

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

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE EXISTS (

SELECT 1

FROM CUST C

WHERE C.SNUM = S.SNUM AND C.RATING = 300

);

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

SELECT DISTINCT S.SNAME

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

WHERE C.RATING = 300;

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

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE EXISTS (

SELECT 1

FROM CUST C

WHERE C.CITY = S.CITY AND C.SNUM <> S.SNUM

);

**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 C.CNUM, C.CNAME, C.CITY

FROM CUST C

WHERE EXISTS (

SELECT 1

FROM CUST C2

WHERE C2.SNUM = C.SNUM AND C2.CNUM <> C.CNUM

AND EXISTS (

SELECT 1

FROM ORDERS O

WHERE O.CNUM = C2.CNUM

)

);

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

-- Using ANY

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE S.CITY = ANY (SELECT C.CITY FROM CUST C WHERE C.SNUM = S.SNUM);

-- Using IN

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE S.CITY IN (SELECT C.CITY FROM CUST C WHERE C.SNUM = S.SNUM);

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

-- Using ANY

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE EXISTS (

SELECT 1

FROM CUST C

WHERE C.CNUM > S.SNUM

AND C.CNAME > S.SNAME

);

-- Using EXISTS

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE EXISTS (

SELECT 1

FROM CUST C

WHERE C.CNAME > S.SNAME

);

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

SELECT C.CNAME

FROM CUST C

WHERE C.RATING > ALL (SELECT RATING FROM CUST WHERE CITY = 'Rome');

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

SELECT O.ONUM

FROM ORDERS O

WHERE O.AMT > ANY (SELECT AMT FROM ORDERS WHERE ODATE = '06-OCT-94');

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

-- Using ANY

SELECT O.ONUM

FROM ORDERS O

WHERE O.AMT < ANY (SELECT AMT FROM CUST C JOIN ORDERS O2 ON C.CNUM = O2.CNUM WHERE C.CITY = 'San Jose');

-- Without ANY

SELECT O.ONUM

FROM ORDERS O

WHERE O.AMT < ALL (SELECT AMT FROM CUST C JOIN ORDERS O2 ON C.CNUM = O2.CNUM WHERE C.CITY = 'San Jose');

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

-- Using ALL

SELECT C.CNAME

FROM CUST C

WHERE C.RATING > ALL (SELECT RATING FROM CUST WHERE CITY = 'Paris');

-- Using NOT EXISTS

SELECT C.CNAME

FROM CUST C

WHERE NOT EXISTS (

SELECT 1

FROM CUST C2

WHERE C2.CITY = 'Paris' AND C2.RATING >= C.RATING

);

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

SELECT C.CNAME

FROM CUST C

WHERE C.RATING >= ANY (SELECT RATING FROM CUST WHERE CNAME = 'Serres');

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

-- Using ANY

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE NOT EXISTS (

SELECT 1

FROM CUST C

WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM

);

-- Using ALL

SELECT S.SNAME

FROM SALESPEOPLE S

WHERE NOT EXISTS (

SELECT 1

FROM CUST C

WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM

);

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

SELECT O.ONUM

FROM ORDERS O

WHERE O.AMT > ANY (SELECT AMT FROM CUST C JOIN ORD

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

SELECT S.SNAME, C.CNAME

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

WHERE C.CITY = 'London';

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

SELECT O.SNUM, MAX(O.ODATE) AS MAX\_DATE, MIN(O.ODATE) AS MIN\_DATE

FROM ORDERS O

GROUP BY O.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 S.SNAME,

CASE

WHEN EXISTS (SELECT 1 FROM CUST C WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM) THEN 'Has customers in city'

ELSE 'No customers in city'

END AS CITY\_STATUS

FROM SALESPEOPLE S;

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

SELECT S.SNAME || ' - ' ||

CASE

WHEN EXISTS (SELECT 1 FROM CUST C WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM)

THEN 'Has customers in city'

ELSE 'No customers in city'

END AS SALESPERSON\_STATUS

FROM SALESPEOPLE S;

**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 C.CNAME, C.CITY, C.RATING || ' High Rating' AS RATING\_INFO

FROM CUST C

WHERE C.RATING >= 200

UNION

SELECT C.CNAME, C.CITY, C.RATING || ' Low Rating' AS RATING\_INFO

FROM CUST C

WHERE C.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 S.SNUM, S.SNAME, C.CNUM, C.CNAME

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

JOIN ORDERS O ON C.CNUM = O.CNUM

GROUP BY S.SNUM, S.SNAME, C.CNUM, C.CNAME

HAVING COUNT(O.ONUM) > 1

ORDER BY S.SNAME, C.CNAME;

**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 S.SNUM FROM SALESPEOPLE S WHERE S.CITY = 'San Jose'

UNION

SELECT C.CNUM FROM CUST C WHERE C.CITY = 'San Jose'

UNION

SELECT O.ONUM FROM ORDERS O WHERE O.ODATE = '03-OCT-94';

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

SELECT DISTINCT S.SNAME

FROM SALESPEOPLE S

JOIN CUST C ON S.SNUM = C.SNUM

WHERE C.CITY = 'London';

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

SELECT DISTINCT S.SNAME

FROM SALESPEOPLE S

WHERE NOT EXISTS (SELECT 1 FROM CUST C WHERE S.SNUM = C.SNUM AND C.CITY = 'London');

**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.SNAME, C.CNAME

FROM SALESPEOPLE S

LEFT JOIN CUST C ON S.SNUM = C.SNUM

UNION

SELECT S.SNAME, 'No customer assigned' AS CNAME

FROM SALESPEOPLE S

WHERE NOT EXISTS (SELECT 1 FROM CUST C WHERE S.SNUM = C.SNUM);