

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

QUERIES

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

```
SQL> SELECT snum, sname, city, comm  
2  FROM 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

2. Display all snum without duplicates from all orders.

```
SQL> SELECT DISTINCT snum
  2  FROM orders;

      SNUM
-----
    1007
    1001
    1004
    1002
    1003
```

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

```
SQL> SELECT SNAME, COMM
  2  FROM SALESPEOPLE
  3 WHERE CITY = 'London';

      SNAME          COMM
-----
     Peel            .12
    Motika           .11
```

4. All customers with rating of 100.

```
SQL> SELECT *
  2  FROM CUST
  3 WHERE RATING = 100;

      CNUM  CNAME          CITY        RATING      SNUM
-----
    2001 Hoffman        London       100        1001
    2004 Grass          Brelin      100        1002
    2007 Pereira        Rome        100        1004
```

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

```
SQL> SELECT ONUM, AMT, ODATE
  2  FROM ORDERS;

      ONUM        AMT      ODATE
-----
    3001      18.69  03-OCT-94
    3003      767.19  03-OCT-94
    3002     1900.1   03-OCT-94
    3005      5160.45  03-OCT-94
    3006      1098.16  04-OCT-94
    3009      1713.23  04-OCT-94
    3007      75.75   05-OCT-94
    3008      4723.00  05-OCT-94
    3010      1309.95  06-OCT-94
    3011      9891.88  06-OCT-94
```

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

```
QL> SELECT *
  2  FROM CUST
  3 WHERE CITY = 'San Jose' AND RATING > 200;
```

CNUM	CNAME	CITY	RATING	SNUM
2003	Liu	San Jose	300	1002

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

```
QL> SELECT *
  2  FROM CUST
  3 WHERE CITY = 'San Jose' OR RATING > 200;
```

CNUM	CNAME	CITY	RATING	SNUM
2003	Liu	San Jose	300	1002
2006	Clemens	London	300	1007

8. All orders for more than \$1000.

```
SQL> SELECT *
  2  FROM ORDERS
  3 WHERE AMT > 1000;
```

ONUM	AMT	ODATE	CNUM	SNUM
3002	1900.1	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
3008	4723	05-OCT-94	2006	1001
3010	1309.95	06-OCT-94	2004	1002
3011	9891.88	06-OCT-94	2006	1001

```
7 rows selected.
```

9. Names and cities of all salespeople in London with commission above 0.10.

```
SQL> SELECT SNAME, CITY  
2  FROM SALESPEOPLE  
3  WHERE CITY = 'London' AND COMM > 0.10;
```

SNAME	CITY
Peel	London
Motika	London

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

```
SQL> SELECT *  
2  FROM CUST  
3  WHERE RATING > 100 OR CITY = 'Rome';
```

CNUM	CNAME	CITY	RATING	SNUM
2002	Giovanne	Rome	200	1003
2003	Liu	San Jose	300	1002
2006	Clemens	London	300	1007
2007	Pereira	Rome	100	1004

11. All salespeople either in Barcelona or in London.

```
SQL> SELECT *  
2  FROM SALESPEOPLE  
3  WHERE CITY IN ('Barcelona', 'London');
```

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1004	Motika	London	.11
1007	Rafkin	Barcelona	.15

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

```
SP2-0226: Invalid line number
SQL> SELECT *
  2  FROM SALESPEOPLE
  3  WHERE COMM > 0.10 AND COMM < 0.12;

      SNUM    SNAME          CITY        COMM
-----  -----
  1004 Motika        London       .11
```

13. All customers with NULL values in city column.

```
SP2-0226: Invalid line number
SQL> SELECT *
  2  FROM CUST
  3  WHERE CITY IS NULL;

no rows selected
```

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

```
SQL> --14
SQL> SELECT *
  2  FROM ORDERS
  3  WHERE ODATE IN ('03-OCT-94', '04-OCT-94');

no rows selected
```

15. All customers serviced by peel or Motika.

```
SQL> -15
SP2-0226: Invalid line number
SQL> SELECT *
  2  FROM CUST
  3  WHERE SNUM IN (
  4    SELECT SNUM FROM SALESPEOPLE
  5    WHERE SNAME IN ('Peel', 'Motika')
  6  );

```

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2007	Pereira	Rome	100	1004

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

```
SQL> --16
SQL> SELECT *
  2  FROM CUST
  3  WHERE CNAME BETWEEN 'A' AND 'Bzzzzzz';
no rows selected
```

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

```
SQL> --17
SQL> SELECT *
  2  FROM ORDERS
  3  WHERE AMT IS NOT NULL AND AMT <> 0;

```

ONUM	AMT	ODATE	CNUM	SNUM
3001	18.69	03-OCT-94	2008	1007
3003	767.19	03-OCT-94	2001	1001
3002	1900.1	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	05-OCT-94	2006	1001
3010	1309.95	06-OCT-94	2004	1002
3011	9891.88	06-OCT-94	2006	1001

10 rows selected.

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

```
SQL> --18
SQL> SELECT COUNT(DISTINCT SNUM) AS NumSalespeople
2  FROM ORDERS;

NUMSALESPEOPLE
-----
5
```

19. Largest order taken by each salesperson, datewise.

```
SQL> --19
SQL> SELECT SNUM, ODATE, MAX(AMT) AS MaxAmt
2  FROM ORDERS
3  GROUP BY SNUM, ODATE;

      SNUM    ODATE        MAXAMT
-----  -----  -----
    1007 03-OCT-94      18.69
    1001 03-OCT-94     767.19
    1004 03-OCT-94    1900.1
    1002 03-OCT-94    5160.45
    1007 04-OCT-94    1098.16
    1003 04-OCT-94    1713.23
    1002 05-OCT-94     75.75
    1001 05-OCT-94    4723
    1002 06-OCT-94   1309.95
    1001 06-OCT-94   9891.88

10 rows selected.
```

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

```
SQL> --20
SQL> SELECT SNUM, MAX(AMT) AS MaxAmt
  2  FROM ORDERS
  3  WHERE AMT > 3000
  4  GROUP BY SNUM;
```

SNUM	MAXAMT
1002	5160.45
1001	9891.88

```
SQL>
```

21. Which day had the highest total amount ordered.

```
SQL> --21
SQL> SELECT ODATE
  2  FROM ORDERS
  3  GROUP BY ODATE
  4  ORDER BY SUM(AMT) DESC
  5  FETCH FIRST 1 ROW ONLY;
```

ODATE

06-OCT-94

22. Count all orders for Oct 3rd.

```
SQL> -22
SP2-0226: Invalid line number
SQL> SELECT COUNT(*)
  2  FROM ORDERS
  3  WHERE ODATE = '03-OCT-94';

COUNT(*)
-----
0
```

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

```
SQL> --23
SQL> SELECT COUNT(DISTINCT CITY)
  2  FROM CUST
  3  WHERE CITY IS NOT NULL;

COUNT(DISTINCTCITY)
-----
4
```

24. Select each customer's smallest order.

```
SQL> --24
SQL> SELECT CNUM, MIN(AMT) AS SmallestOrder
  2  FROM ORDERS
  3  GROUP BY CNUM;

  CNUM  SMALLESTORDER
-----  -----
  2008      18.69
  2001      767.19
  2007      1900.1
  2003      5160.45
  2002      1713.23
  2004      75.75
  2006      4723

  7 rows selected.
```

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

```
SQL> --25
SQL> SELECT *
  2  FROM CUST
  3  WHERE CNAME LIKE 'G%'
  4  ORDER BY CNAME ASC
  5  FETCH FIRST 1 ROW ONLY;

  CNUM  CNAME          CITY        RATING    SNUM
-----  -----  -----
  2002  Giovanne       Rome        200        1003
```

26. Get the output like “ For dd/mm/yy there are ___ orders.

```
SQL> --26
SQL> SELECT 'For ' || TO_CHAR(ODATE, 'DD/MM/YY') || ' there are ' || COUNT(*)
) || ' orders' AS Output
  2  FROM ORDERS
  3  GROUP BY ODATE;

OUTPUT
-----
For 03/10/94 there are 4 orders
For 04/10/94 there are 2 orders
For 05/10/94 there are 2 orders
For 06/10/94 there are 2 orders
```

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

```
SQL> --27
SQL> SELECT ONUM, SNUM, AMT * 0.12 AS Commission
  2  FROM ORDERS;

      ONUM      SNUM COMMISSION
-----  -----
    3001      1007    2.2428
    3003      1001   92.0628
    3002      1004   228.012
    3005      1002   619.254
    3006      1007  131.7792
    3009      1003  205.5876
    3007      1002      9.09
    3008      1001   566.76
    3010      1002   157.194
    3011      1001  1187.0256

10 rows selected.
```

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

```
SQL> --28
SQL> SELECT CITY, MAX(RATING) AS HighestRating
  2  FROM CUST
  3  GROUP BY CITY;

      CITY      HIGHESTRATING
-----  -----
London                  300
Rome                   200
San Jose                300
Brelin                  100
```

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

```
SQL> --29
SQL> SELECT ODATE, SUM(AMT) AS Total
  2  FROM ORDERS
  3  GROUP BY ODATE
  4  ORDER BY Total DESC;

ODATE          TOTAL
-----
06-OCT-94      11201.83
03-OCT-94      7846.43
05-OCT-94      4798.75
04-OCT-94      2811.39
```

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

```
SQL> --30
SQL> SELECT S.SNAME, C.CNAME, S.CITY
  2  FROM SALESPEOPLE S, CUST C
  3  WHERE S.CITY = C.CITY;

SNAME          CNAME          CITY
-----          -----          -----
Peel           Hoffman        London
Motika         Hoffman        London
Serres         Liu            San Jose
Peel           Clemens        London
Motika         Clemens        London
```

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

```
SQL> --31
SQL> SELECT C.CNAME, S.SNAME
  2  FROM CUST C
  3  JOIN SALESPEOPLE S ON C.SNUM = S.SNUM;

CNAME          SNAME
-----          -----
Hoffman        Peel
Giovanne       Axelrod
Liu            Serres
Grass           Serres
Clemens         Rafkin
Pereira        Motika

6 rows selected.
```

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

```
SQL> --32 List each order number fol
who made the order.
SQL> SELECT O.ONUM, C.CNAME
  2  FROM ORDERS O
  3  JOIN CUST C ON O.CNUM = C.CNUM;

    ONUM  CNAME
    -----
  3003 Hoffman
  3002 Pereira
  3005 Liu
  3009 Giovanne
  3007 Grass
  3008 Clemens
  3010 Grass
  3011 Clemens

8 rows selected.
```

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

```
SQL> SELECT O.ONUM, S.SNAME, C.CNAME  
2  FROM ORDERS O  
3  JOIN SALESPEOPLE S ON O.SNUM = S.SNUM  
4  JOIN CUST C ON O.CNUM = C.CNUM;
```

ONUM	SNAME	CNAME
3003	Peel	Hoffman
3009	Axelrod	Giovanne
3005	Serres	Liu
3007	Serres	Grass
3010	Serres	Grass
3008	Peel	Clemens
3011	Peel	Clemens
3002	Motika	Pereira

8 rows selected.

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

```
SQL> SELECT C.*  
2  FROM CUST C  
3  JOIN SALESPEOPLE S ON C.SNUM = S.SNUM  
4  WHERE S.COMM > 0.12;
```

SNUM	CNUM	CNAME	CITY	RATING
1002	2003	Liu	San Jose	300
1002	2004	Grass	Brelin	100
1007	2006	Clemens	London	300

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

```

SQL> SELECT O.ONUM, O.SNUM, O.AMT * S.COMM AS COMM_AMT
  2  FROM ORDERS O
  3  JOIN CUST C ON O.CNUM = C.CNUM
  4  JOIN SALESPEOPLE S ON O.SNUM = S.SNUM
  5  WHERE C.RATING > 100;

```

ONUM	SNUM	COMM_AMT
3008	1001	566.76
3011	1001	1187.0256
3005	1002	670.8585
3009	1003	171.323

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

```

SQL> SELECT C1.CNAME AS Cust1, C2.CNAME AS Cust2, C1.RATING
  2  FROM CUST C1
  3  JOIN CUST C2 ON C1.RATING = C2.RATING AND C1.CNUM <> C2.CNUM;

CUST1          CUST2          RATING
-----          -----          -----
Grass           Hoffman        100
Pereira         Hoffman        100
Clemens         Liu            300
Hoffman         Grass           100
Pereira         Grass           100
Liu             Clemens         300
Hoffman         Pereira         100
Grass           Pereira         100

```

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

```

SQL> SELECT C1.CNAME AS Cust1, C2.CNAME AS Cust2, C1.RATING
  2  FROM CUST C1
  3  JOIN CUST C2 ON C1.RATING = C2.RATING AND C1.CNUM < C2.CNUM;

CUST1          CUST2          RATING
-----          -----          -----
Hoffman         Grass           100
Liu             Clemens         300
Hoffman         Pereira         100
Grass           Pereira         100

```

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

```
QL> SELECT C.CNAME, S1.SNAME AS Sales1, S2.SNAME AS Sales2,
  S Sales3
  2  FROM CUST C, SALESPEOPLE S1, SALESPEOPLE S2, SALESPEOPLE
  3  WHERE S1.SNUM < S2.SNUM AND S2.SNUM < S3.SNUM;
```

NAME	SALES1	SALES2
ALES3		
offman	Peel	Serres
otika		
iovanne	Peel	Serres
otika		
iu	Peel	Serres
otika		
NAME	SALES1	SALES2
ALES3		
rass	Peel	Serres
otika		
lemens	Peel	Serres
otika		
ereira	Peel	Serres
otika		
NAME	SALES1	SALES2
ALES3		
offman	Peel	Serres
afkin		
iovanne	Peel	Serres

CNAME	SALES1	SALES2
SALES3		
Grass	Axelrod	Motika
Rafkin		
Clemens	Axelrod	Motika
Rafkin		
Pereira	Axelrod	Motika
Rafkin		

60 rows selected.

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

```
SQL> SELECT *
  2  FROM CUST
  3  WHERE CITY IN (
  4    SELECT DISTINCT CITY FROM CUST WHERE SNUM = (SELECT SNUM FROM S
ALESPEOPLE WHERE SNAME = 'Serres')
  5  );
```

SNUM	CNUM	CNAME	CITY	RATING
1002	2003	Liu	San Jose	300
1002	2004	Grass	Brelin	100

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

```
SQL> --40.      Find all pairs of customers served by single salespers
on.
SQL> SELECT C1.CNAME AS Cust1, C2.CNAME AS Cust2, C1.SNUM
  2  FROM CUST C1
  3  JOIN CUST C2 ON C1.SNUM = C2.SNUM AND C1.CNUM < C2.CNUM;
```

CUST1	CUST2	SNUM
Liu	Grass	1002

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.

```
SQL> --41
SQL> SELECT S1.SNAME AS Sales1, S2.SNAME AS Sales2, S1.CITY
  2  FROM SALESPEOPLE S1
  3  JOIN SALESPEOPLE S2 ON S1.CITY = S2.CITY AND S1.SNUM < S2.SNUM;
```

SALES1	SALES2	CITY
Peel	Motika	London

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

```
SQL> --42
SQL> SELECT 01.ONUM AS Order1, 02.ONUM AS Order2, C.CNAME
  2  FROM ORDERS 01
  3  JOIN ORDERS 02 ON 01.CNUM = 02.CNUM AND 01.ONUM < 02.ONUM
  4  JOIN CUST C ON 01.CNUM = C.CNUM;

      ORDER1      ORDER2  CNAME
-----  -----
      3007        3010  Grass
      3008        3011  Clemens
```

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

```
SQL> --43
SQL> SELECT *
  2  FROM CUST
  3  WHERE RATING = (
  4    SELECT RATING FROM CUST WHERE CNAME = 'Hoffman'
  5  );

      SNUM      CNUM  CNAME          CITY      RATING
-----  -----
      1001    2001  Hoffman        London      100
      1002    2004  Grass         Brelin      100
      1004    2007  Pereira       Rome       100
```

44. Extract all the orders of Motika.

```
SQL> --44
SQL> SELECT *
  2  FROM ORDERS
  3  WHERE SNUM = (SELECT SNUM FROM SALESPEOPLE WHERE SNAME = 'Motika'
);

      ONUM      AMT ODATE      CNUM      SNUM
-----  -----
      3002  1900.1 03-OCT-94    2007    1004
```

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

```
SQL> --45
SQL> SELECT *
  2  FROM ORDERS
  3  WHERE SNUM = (
  4    SELECT SNUM FROM CUST WHERE CNAME = 'Hoffman'
  5  );
```

ONUM	AMT	ODATE	CNUM	SNUM
3003	767.19	03-OCT-94	2001	1001
3008	4723	05-OCT-94	2006	1001
3011	9891.88	06-OCT-94	2006	1001

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

```
SQL> --46
SQL> SELECT *
  2  FROM ORDERS
  3  WHERE AMT > (
  4    SELECT AVG(AMT) FROM ORDERS WHERE ODATE = '04-OCT-94'
  5  );
```

no rows selected

47. Find average commission of salespeople in london.

```
SQL> --47
SQL> SELECT AVG(COMM)
  2  FROM SALESPEOPLE
  3  WHERE CITY = 'London';
```

AVG(COMM)
.115

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

```
SQL> --48
SQL> SELECT *
2  FROM ORDERS
3  WHERE SNUM IN (
4    SELECT SNUM FROM CUST WHERE CITY = 'London'
5  );
```

ONUM	AMT	ODATE	CNUM	SNUM
3003	767.19	03-OCT-94	2001	1001
3008	4723	05-OCT-94	2006	1001
3011	9891.88	06-OCT-94	2006	1001
3001	18.69	03-OCT-94	2008	1007
3006	1098.16	04-OCT-94	2008	1007

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

```
SQL> --49
SQL> SELECT DISTINCT S.COMM
2  FROM SALESPEOPLE S
3  JOIN CUST C ON S.SNUM = C.SNUM
4  WHERE C.CITY = 'London';

      COMM
-----
     .12
     .15
```

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

```
SQL> --50
SQL> SELECT *
2  FROM CUST
3  WHERE CNUM = (
4    SELECT SNUM + 1000 FROM SALESPEOPLE WHERE SNAME = 'Serres'
5  );

      CNUM  CNAME          CITY        RATING
SNUM
-----
-----  -----
1003      2002 Giovanne       Rome         200
```

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

```
SQL> --51
SQL> SELECT COUNT(*)
  2  FROM CUST
  3  WHERE RATING > (
  4    SELECT AVG(RATING)
  5    FROM CUST
  6    WHERE CITY = 'San Jose'
  7  );
      COUNT(*)
-----
          0
```

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

```
SQL> --52
SQL> SELECT ONUM, ODATE
  2  FROM ORDERS
  3  WHERE CNUM = (
  4    SELECT CNUM
  5    FROM CUST
  6    WHERE CNAME = 'Cisnerous'
  7  );
no rows selected
```

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

```
SQL> --53
SQL> SELECT C.CNAME, C.RATING
  2  FROM CUST C
  3  WHERE C.CNUM IN (
  4    SELECT CNUM
  5    FROM ORDERS
  6    GROUP BY CNUM
  7    HAVING AVG(AMT) > (
  8      SELECT AVG(AMT)
  9      FROM ORDERS
 10    )
 11  );
      CNAME          RATING
-----
Liu                  300
Clemens              300
```

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.

```
SQL> --54
SQL> SELECT SNUM, SUM(AMT) AS TOTAL_AMT
  2  FROM ORDERS
  3  GROUP BY SNUM
  4  HAVING SUM(AMT) > (
  5    SELECT MAX(AMT)
  6    FROM ORDERS
  7  );
-----
```

SNUM	TOTAL_AMT
1001	15382.07

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

```
SQL> --55
SQL> SELECT DISTINCT C.CNAME
  2  FROM CUST C
  3  JOIN ORDERS O ON C.CNUM = O.CNUM
  4  WHERE O.ODATE = '03-OCT-94';
no rows selected
```

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

```
SQL> --56
SQL> SELECT S.SNUM, S.SNAME
  2  FROM SALESPeOPLE S
  3  JOIN CUST C ON S.SNUM = C.SNUM
  4  GROUP BY S.SNUM, S.SNAME
  5  HAVING COUNT(C.CNUM) > 1;
-----
```

SNUM	SNAME
1002	Serres

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

```
SQL> --57
SQL> SELECT O.ONUM, O.SNUM, C.SNUM AS CUST_SNUM
  2  FROM ORDERS O
  3  JOIN CUST C ON O.CNUM = C.CNUM
  4  WHERE O.SNUM <> C.SNUM;

    ONUM      SNUM  CUST_SNUM
-----  -----  -----
  3008        1001      1007
  3011        1001      1007
```

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

```
SQL> --58
SQL> SELECT O.*
  2  FROM ORDERS O
  3  JOIN (
  4    SELECT CNUM, AVG(AMT) AS AVG_AMT
  5    FROM ORDERS
  6    GROUP BY CNUM
  7  ) A ON O.CNUM = A.CNUM
  8  WHERE O.AMT > A.AVG_AMT;
```

ONUM	AMT	ODATE	CNUM	SNUM
3006	1098.16	04-OCT-94	2008	1007
3010	1309.95	06-OCT-94	2004	1002
3011	9891.88	06-OCT-94	2006	1001

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.

```
SQL> --59
SQL> SELECT ODATE, SUM(AMT) AS TOTAL_AMT
  2  FROM ORDERS
  3  GROUP BY ODATE
  4  HAVING SUM(AMT) >= (
  5    SELECT MAX(AMT) + 2000
  6    FROM ORDERS
  7  );
no rows selected
```

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

```
SQL> --60
SQL> SELECT C1.CNUM, C1.CNAME
  2  FROM CUST C1
  3  WHERE C1.RATING = (
  4      SELECT MAX(C2.RATING)
  5      FROM CUST C2
  6      WHERE C2.CITY = C1.CITY
  7  );
 
CNUM  CNAME
-----
2002  Giovanne
2003  Liu
2004  Grass
2006  Clemens
```

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

```
SQL> --61
SQL> SELECT DISTINCT S.SNAME
  2  FROM SALESPEOPLE S
  3  JOIN CUST C ON S.CITY = C.CITY
  4  WHERE S.SNUM <> C.SNUM;
 
SNAME
-----
Motika
Peel

SQL> SELECT S.SNAME
  2  FROM SALESPEOPLE S
  3  WHERE EXISTS (
  4      SELECT 1
  5      FROM CUST C
  6      WHERE C.CITY = S.CITY AND C.SNUM <> S.SNUM
  7  );
 
SNAME
-----
Motika
Peel
```

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.

```
SQL> --62
SQL> SELECT CNUM, CNAME, CITY
  2  FROM CUST
  3 WHERE EXISTS (
  4   SELECT 1
  5   FROM CUST
  6   WHERE CITY = 'San Jose'
  7 );
```

CNUM	CNAME	CITY
2001	Hoffman	London
2002	Giovanne	Rome
2003	Liu	San Jose
2004	Grass	Brelin
2006	Clemens	London
2007	Pereira	Rome

6 rows selected.

63. Find salespeople no. who have multiple customers.

```
SQL> --63
SQL> SELECT SNUM
  2  FROM CUST
  3 GROUP BY SNUM
  4 HAVING COUNT(*) > 1;
```

SNUM
1002

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

```
SQL> --64
SQL> SELECT S.SNUM, S.SNAME, S.CITY
  2  FROM SALESPEOPLE S
  3 JOIN CUST C ON S.SNUM = C.SNUM
  4 GROUP BY S.SNUM, S.SNAME, S.CITY
  5 HAVING COUNT(C.CNUM) > 1;
```

SNUM	SNAME	CITY
1002	Serres	San Jose

65. Find salespeople who serve only one customer.

```
SQL> --65
SQL> SELECT S.SNUM, S.SNAME
  2  FROM SALESPEOPLE S
  3  JOIN CUST C ON S.SNUM = C.SNUM
  4  GROUP BY S.SNUM, S.SNAME
  5  HAVING COUNT(C.CNUM) = 1;

      SNUM SNAME
-----
    1001 Peel
    1003 Axelrod
    1007 Rafkin
    1004 Motika
```

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

```
SQL> --66
SQL> SELECT S.*
  2  FROM SALESPEOPLE S
  3  JOIN ORDERS O ON S.SNUM = O.SNUM
  4  GROUP BY S.SNUM, S.SNAME, S.CITY, S.COMM
  5  HAVING COUNT(O.ONUM) > 1;

      SNUM SNAME          CITY        COMM
-----
    1007 Rafkin        Barcelona   .15
    1001 Peel           London     .12
    1002 Serres         San Jose   .13
```

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

```
SQL> --67
SQL> SELECT S.SNUM, S.SNAME
  2  FROM SALESPEOPLE S
  3  WHERE EXISTS (
  4    SELECT 1
  5    FROM CUST C
  6    WHERE C.SNUM = S.SNUM AND C.RATING = 300
  7  );

      SNUM SNAME
-----
    1002 Serres
    1007 Rafkin
```

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

```
SQL> --68
SQL> SELECT DISTINCT S.SNUM, S.SNAME
  2  FROM SALESPEOPLE S
  3  JOIN CUST C ON S.SNUM = C.SNUM
  4  WHERE C.RATING = 300;

      SNUM SNAME
-----
    1002 Serres
    1007 Rafkin
```

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

```
SQL> --69
SQL> SELECT S.SNUM, S.SNAME
  2  FROM SALESPEOPLE S
  3  WHERE EXISTS (
  4    SELECT 1
  5    FROM CUST C
  6    WHERE C.CITY = S.CITY AND C.SNUM <> S.SNUM
  7  );

      SNUM SNAME
-----
    1004 Motika
    1001 Peel
```

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> --70
SQL> SELECT C1.*
  2  FROM CUST C1
  3 WHERE EXISTS (
  4   SELECT 1
  5   FROM CUST C2
  6   JOIN ORDERS O ON C2.CNUM = O.CNUM
  7   WHERE C2.SNUM = C1.SNUM AND C2.CNUM <> C1.CNUM
  8 );
```

SNUM	CNUM	CNAME	CITY	RATING
1002	2004	Grass	Brelin	100
1002	2003	Liu	San Jose	300

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

```
SQL> --71
SQL> SELECT *
  2  FROM SALESPeOPLE
  3 WHERE CITY = ANY (
  4   SELECT CITY
  5   FROM CUST
  6 );
```

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1004	Motika	London	.11
1002	Serres	San Jose	.13

```
SQL> SELECT *
  2  FROM SALESPeOPLE
  3 WHERE CITY IN (
  4   SELECT CITY
  5   FROM CUST
  6 );
```

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1004	Motika	London	.11
1002	Serres	San Jose	.13

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

```
SQL> --72
SQL> SELECT *
  2  FROM SALESPEOPLE
  3  WHERE SNAME < ANY (
  4    SELECT CNAME
  5    FROM CUST
  6  );

```

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1004	Motika	London	.11
1003	Axelrod	New York	.1

```
SQL> SELECT *
  2  FROM SALESPEOPLE S
  3  WHERE EXISTS (
  4    SELECT 1
  5    FROM CUST C
  6    WHERE C.CNAME > S.SNAME
  7  );

```

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1004	Motika	London	.11
1003	Axelrod	New York	.1

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

```
SQL> --73
SQL> SELECT *
  2  FROM CUST
  3  WHERE RATING > ANY (
  4    SELECT RATING
  5    FROM CUST
  6    WHERE CITY = 'Rome'
  7  );

```

SNUM	CNUM	CNAME	CITY	RATING
1003	2002	Giovanne	Rome	200
1002	2003	Liu	San Jose	300
1007	2006	Clemens	London	300

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

```
SQL> --74
SQL> SELECT *
  2  FROM ORDERS
  3  WHERE AMT > ANY (
  4    SELECT AMT
  5    FROM ORDERS
  6    WHERE ODATE = '06-OCT-94'
  7  );
no rows selected
```

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

```
SQL> --75
SQL> SELECT *
  2  FROM ORDERS
  3  WHERE AMT < ANY (
  4    SELECT AMT
  5    FROM ORDERS O
  6    JOIN CUST C ON O.CNUM = C.CNUM
  7    WHERE C.CITY = 'San Jose'
  8  );
      ONUM      AMT ODATE        CNUM      SNUM
-----  -----  -----  -----  -----
  3001    18.69 03-OCT-94    2008    1007
  3007    75.75 05-OCT-94    2004    1002
  3003    767.19 03-OCT-94   2001    1001
  3006   1098.16 04-OCT-94   2008    1007
  3010   1309.95 06-OCT-94   2004    1002
  3009   1713.23 04-OCT-94   2002    1003
  3002   1900.1  03-OCT-94   2007    1004
  3008    4723   05-OCT-94   2006    1001
8 rows selected.

SQL> SELECT *
  2  FROM ORDERS
  3  WHERE AMT < (
  4    SELECT MAX(AMT)
  5    FROM ORDERS O
  6    JOIN CUST C ON O.CNUM = C.CNUM
  7    WHERE C.CITY = 'San Jose'
  8  );
      ONUM      AMT ODATE        CNUM      SNUM
-----  -----  -----  -----  -----
  3001    18.69 03-OCT-94    2008    1007
  3003    767.19 03-OCT-94   2001    1001
  3002   1900.1  03-OCT-94   2007    1004
  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   05-OCT-94   2006    1001
  3010   1309.95 06-OCT-94   2004    1002
8 rows selected.
```

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

```

SQL> --76
SQL> SELECT *
  2  FROM CUST
  3 WHERE RATING > ALL (
  4   SELECT RATING
  5   FROM CUST
  6   WHERE CITY = 'Paris'
  7  );

```

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2007	Pereira	Rome	100	1004
2004	Grass	Brelin	100	1002
2002	Giovanne	Rome	200	1003
2006	Clemens	London	300	1007
2003	Liu	San Jose	300	1002

6 rows selected.

```

SQL> SELECT *
  2  FROM CUST A
  3 WHERE NOT EXISTS (
  4   SELECT 1
  5   FROM CUST B
  6   WHERE B.CITY = 'Paris' AND B.RATING >= A.RATING
  7  );

```

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

6 rows selected.

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

```

SQL> --77
SQL> SELECT *
  2  FROM CUST
  3 WHERE RATING >= ANY (
  4   SELECT RATING
  5   FROM CUST
  6   WHERE SNUM = (
  7     SELECT SNUM
  8     FROM SALESPEOPLE
  9     WHERE SNAME = 'Seeres'
 10   )
 11  );

```

CNUM	CNAME	CITY	RATING	SNUM
2003	Liu	San Jose	300	1002
2006	Clemens	London	300	1007
2002	Giovanne	Rome	200	1003
2007	Pereira	Rome	100	1004
2004	Grass	Brelin	100	1002
2001	Hoffman	London	100	1001

6 rows selected.

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

```
SQL> --78
SQL> SELECT *
  2  FROM SALESPEOPLE
  3  WHERE CITY <> ANY (
  4    SELECT CITY
  5    FROM CUST
  6    WHERE CUST.SNUM = SALESPEOPLE.SNUM
  7  );
-----  
SNUM SNAME          CITY           COMM  
-----  
1003 Axelrod        New york      .1  
1002 Serres         San Jose      .13  
1007 Rafkin         Barcelona    .15  
1004 Motika          London       .11  
  
SQL> SELECT *
  2  FROM SALESPEOPLE
  3  WHERE CITY <> ALL (
  4    SELECT CITY
  5    FROM CUST
  6    WHERE CUST.SNUM = SALESPEOPLE.SNUM
  7  );
-----  
SNUM SNAME          CITY           COMM  
-----  
1004 Motika          London       .11  
1007 Rafkin          Barcelona   .15  
1003 Axelrod          New york    .1  
  
SQL>
```

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

```
SQL> --79
SQL> SELECT *
  2  FROM SALESPEOPLE S
  3  WHERE NOT EXISTS (
  4    SELECT 1
  5    FROM CUST C
  6    WHERE C.CITY = S.CITY AND C.SNUM = S.SNUM
  7  );
-----  
SNUM SNAME          CITY           COMM  
-----  
1003 Axelrod        New york      .1  
1007 Rafkin         Barcelona    .15  
1004 Motika          London       .11
```

80. Find all salespeople and customers located in london.

```
SQL> --80
SQL> SELECT C.CNUM, C.CNAME, C.CITY, S.SNUM, S.SNAME, S.CITY
  2  FROM CUST C
  3  JOIN SALESPEOPLE S ON C.CITY = S.CITY AND C.SNUM = S.SNUM;
-----  
CNUM CNAME          CITY           SNUM  
-----  
SNAME          CITY  
-----  
Peel            London          1001  
2001 Hoffman     London          1001  
Serres          San Jose        1002  
2003 Liu          San Jose        1002
```

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

```
SQL> --81
SQL> SELECT C.CNUM, C.CNAME, C.CITY, S.SNAME, S.CITY
  2  FROM CUST C
  3  JOIN SALESPEOPLE S ON C.SNUM = S.SNUM
  4  WHERE C.CITY <> S.CITY;

      CNUM CNAME          CITY        SNAME
-----+
CITY
-----+
      2002 Giovanne       Rome       Axelrod
New York

      2004 Grass           Brelin     Serres
San Jose

      2006 Clemens         London     Rafkin
Barcelona

      CNUM CNAME          CITY        SNAME
-----+
CITY
-----+
      2007 Pereira         Rome       Motika
London
```

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

```
SQL> --82
SQL> SELECT DISTINCT S.SNUM, S.SNAME
  2  FROM SALESPEOPLE S
  3  JOIN CUST C ON S.CITY = C.CITY
  4  WHERE S.COMM > 0.10;

      SNUM SNAME
-----+
      1001 Peel
      1004 Motika
      1002 Serres
```

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

```
SQL> --83
SQL> SELECT DISTINCT S.SNAME
  2  FROM SALESPEOPLE S
  3  LEFT JOIN CUST C ON S.SNUM = C.SNUM
  4  WHERE S.COMM > 0.12 OR S.CITY = C.CITY;

SNAME
-----
Peel
Serres
Rafkin
```

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'.

```
SQL> --84
SQL> SELECT C.*  
2  FROM CUST C  
3  JOIN SALESPEOPLE S ON C.SNUM = S.SNUM  
4  WHERE S.COMM > 0.10;
```

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2003	Liu	San Jose	300	1002
2004	Grass	Brelin	100	1002
2006	Clemens	London	300	1007
2007	Pereira	Rome	100	1004

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.

```
SQL> --85
SQL> SELECT DISTINCT S.SNAME
2  FROM SALESPEOPLE S
3  JOIN CUST C ON S.SNUM = C.SNUM
4  WHERE C.RATING >= 200;
```

SNAME

```
-----  
Serres  
Rafkin  
Axelrod
```

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 eliminates and redundancies between either of them and the first.

```
SQL> --86
SQL> SELECT C.CNAME
2  FROM CUST C
3  JOIN SALESPEOPLE S ON C.SNUM = S.SNUM
4  WHERE S.COMM > 0.12;
```

CNAME

```
-----  
Liu  
Grass  
Clemens
```

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

```
SQL> --87
SQL> SELECT C.CNAME
  2  FROM CUST C
  3  JOIN SALESPEOPLE S ON C.SNUM = S.SNUM
  4  WHERE C.CITY <> S.CITY;

CNAME
-----
Giovanne
Grass
Clemens
Pereira
```

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

```
SQL> --88
SQL> SELECT DISTINCT S.SNAME
  2  FROM SALESPEOPLE S
  3  JOIN CUST C ON S.SNUM = C.SNUM
  4  WHERE C.CITY = 'San Jose';

SNAME
-----
Serres
```

89. 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)

```
SQL> --89
SQL> SELECT *
  2  FROM CUST
  3  WHERE RATING > ALL (
  4      SELECT RATING
  5      FROM CUST
  6      WHERE CITY = 'Rome'
  7  );
```

CNUM	CNAME	CITY	RATING	SNUM
2006	Clemens	London	300	1007
2003	Liu	San Jose	300	1002