sql queries

TABLE SALESPEOPLE

primary key --- snum

sname should not be null

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

primary key ----- cnum

foreign key -------- snum rferences salespeople table snum

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

primary key -onum

foreign snum

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

Problems :

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

```
Select snum, sname, city, comm
from salespeople;
```

2. Display all snum without duplicates from all orders.

```
Select distinct snum
from orders;
```

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

```
Select sname,comm
from salespeople
where city = 'London';
```

4. All customers with rating of 100.

```
Select cname
from cust
where rating = 100;
```

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

```
Select ordno, amt, odate
from orders;
```

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

```
Select cname
from cust
where rating > 200;
```

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

```
Select cname
from cust
where city = 'San Jose' or
      rating > 200;
```

8. All orders for more than $1000.

```
Select *
from orders
where amt > 1000;
```

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

```
Select sname, city
from salepeople
where comm > 0.10 and
      city = 'London';
```

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

```
Select cname
from cust
where rating <= 100 or
      city = 'Rome';
```

11. All salespeople either in Barcelona or in london.

```
Select sname, city
from salespeople
where city in ('Barcelona','London');
```

12. 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;

13. All customers with NULL values in city column.

Select cname

from cust

where city is null;

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

Select *

from orders

where odate in ('03-OCT-94','04-OCT-94');

15. All customers serviced by peel or Motika.

select cnum,cname,snum

from cust

where snum in (

select snum from salespeople

where sname in ('PEEL' ,'MOTIKA'))

Select cname

from cust, orders

```
where orders.cnum = cust.cnum and
        orders.snum in ( select snum
    from salespeople
  where sname in 'Peel','Motika'));
```

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

```
Select cname
from cust
where cname like ‘A%’ or
        cname like ‘B%’;
```

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

```
Select onum
from orders
where amt != 0 or
amt is not null;
```

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

```
Select count(distinct snum)
from orders;
```

19. Largest order taken by each salesperson, datewise.

```
Select odate, snum, max(oamt)
from orders
group by odate, snum
order by odate,snum;
```

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

```
Select odate, snum, max(oamt)
from orders
where amt > 3000
group by odate, snum
order by odate,snum;
```

21. Which day had the hightest total amount ordered.

```
Select odate, amt, snum, cnum
from orders
```

where amt = (select max(oamt)

from orders)

22. Count all orders for Oct 3rd.

Select count(*)

from orders

where odate = ‘03-OCT-94’;

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

Select count(distinct city)

from cust;

24. Select each customer’s smallest order.

Select cnum, min(oamt)

from orders

group by cnum;

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

Select min(cname)

from cust

where cname like ‘G%’;

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

Select 'For ' || to_char(odate,'dd/mm/yy') || ' there are '||

count(*) || ' Orders'

from orders

group by odate;

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

Select onum, snum, amt, amt * 0.12

from orders

order by snum;

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

```
Select 'For the city (' || city || '), the highest rating is : (' ||
max(rating) || ')'
from cust
group by city;
```

29. 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);
```

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

```
Select sname, cname
from salespeople, cust
where salespeople.city = cust.city;
```

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

```
Select cname, sname
from cust, salespeople
where cust.snum = salespeople.snum;
```

32. 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;
```

33. 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;

34. 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;
```

35. 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
```

36. 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 and
      a.cnum != b.cnum
```

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

```
Select a.cname, b.cname,a.rating
from cust a, cust b
where a.rating = b.rating and
      a.cnum != b.cnum and
            a.cnum < b.cnum;
```

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

```
Select cname, sname
from salespeople, cust
where sname in  ( select sname
   from salespeople
                  where rownum <= 3)
order by cname;
```

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

```
Select cname
from cust
where city = ( select city
                    from cust, salespeople
      where cust.snum = salespeople.snum and        sname = 'Serres');
```

```
Select cname
from cust
where city in ( select city
                     from cust, orders
                where cust.cnum = orders.cnum and
                orders.snum in ( select snum
  from salespeople
                                    where sname = 'Serres'));
```

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

```
Select cname from cust
 where snum in (select snum from cust
          group by snum
          having count(snum) > 1);
```

```
Select distinct a.cname
from cust a ,cust b
where a.snum = b.snum and a.rowid != b.rowid;
```

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

```
Select a.sname, b.sname
from salespeople a, salespeople b
where a.snum > b.snum and
```

```
    a.city = b.city;
```

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

```
Select c.cname, a.onum, b.onum
from orders a, orders b, cust c
where a.cnum = b.cnum and
        a.onum > b.onum and
                c.cnum = a.cnum;
```

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

```
Select cname, city
from cust
where rating = (select rating
                                from cust
          where cname = 'Hoffman')
and cname != 'Hoffman';
```

44. Extract all the orders of Motika.

```
Select Onum
from orders
where snum = ( select snum
  from salespeople
  where sname = ‘Motika’);
```

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

```
Select onum, sname, cname, amt
from orders a, salespeople b, cust c
where a.snum = b.snum and
        a.cnum = c.cnum and
        a.snum = ( select snum
                        from orders
```

```
                    where cnum = ( select cnum
                                   from cust
                                   where cname = 'Hoffman'));
```

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

```
Select *
from orders
where amt > ( select avg(oamt)
              from orders
                   where odate = '03-OCT-94');
```

47. Find average commission of salespeople in london.

```
Select avg(comm)
from salespeople
where city = ‘London’;
```

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

```
Select snum, cnum
from orders
where cnum in (select cnum
  from cust
              where city = 'London');
```

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

```
Select comm
from salespeople
where snum in (select snum
              from cust
              where city = ‘London’);
```

50. 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');
```

51. 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');
```

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

```
Select onum, odate
from orders

where cnum = ( select cnum
                from cust
                where cname = 'Cisnerous');
```

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

```
Select max(b.cname), max(b.rating), a.cnum
from orders a, cust b
where a.cnum = b.cnum
group by a.cnum
having count(a.cnum) > ( select avg(count(cnum))
    from orders
                        group by cnum);
```

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

```
Select snum,sum(oamt)
from orders
group by snum
having sum(oamt) > ( select max(oamt)
      from orders);
```

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

```
Select cname
from cust a, orders b
where a.cnum = b.cnum and
```

odate = ‘03-OCT-94’;

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

```
Select sname, snum
from salespeople
where snum in ( select snum
                from cust
                group by snum
                having count(snum) > 1 );
```

57. 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;
```

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

```
select onum, cnum, amt
from orders a
where amt > (  select avg(oamt)
from orders b
where a.cnum = b.cnum
group by cnum);
```

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

```
Select odate, sum(amt)
from orders a
group by odate
having sum(amt) > ( select max(amt)
                    from orders b
                    where a.odate = b.odate
                    group by odate);
```

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

```
Select a.cnum, a.cname
from cust a
where a.rating = (  select max(rating)
      from cust b
            where a.city = b.city);
```

61. 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 );
```

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

```
Select * from cust
where 2 < (select count(*)
       from cust
       where city = 'San Jose');
```

63. Find salespeople no. who have multiple customers.

```
Select snum
from cust
group by snum
having count(*) > 1;
```

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

```
Select snum, sname, city
from salespeople
where snum in ( select snum
                from cust
                group by snum
                having count(*) > 1);
```

65. Find salespeople who serve only one customer.

```
Select snum
from cust
group by snum
having count(*) = 1;
```

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

```
Select snum, count(snum)
from orders
group by snum
having count(snum) > 1;
```

67. 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)
```

68. 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;
```

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

```
Select snum, sname
from salespeople
where exists ( select cnum
                from cust
                where salespeople.city = cust.city and
                      salespeople.snum != cust.snum);
```

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.

```
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;
```

71. 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);
```

72. 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);
```

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

```
Select a.cname
from cust a
where city = 'Rome' and
        rating > ( select max(rating)
                from cust
                where city != 'Rome');
```

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

```
Select onum, amt
from orders
where odate != '06-oct-94' and
            amt > ( select min(amt)
                        from orders
                    where odate = '06-oct-94');
```

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

```
Select onum, amt
from orders
```

```
where amt < any ( select amt
            from orders, cust
            where city = 'San Jose' and
                orders.cnum = cust.cnum);
```

```
Select onum, amt
from orders
where amt < ( select max(amt)
            from orders, cust
            where city = 'San Jose' and
                orders.cnum = cust.cnum);
```

76. 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);
```

77. 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;
```

78. 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);
```

79. 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);
```

80. 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;
```

81. 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);
```

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

```
Select 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));
```

83. 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;
```

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

```
Select cname, cities, rating, ‘Higher Rating’
from cust
where rating >= 200
UNION

Select cname, cities, rating, ‘Lower Rating’
from cust
where rating < 200;
```

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

```
Select '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;
```

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.

```
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';
```

87. 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’;
```

88. 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.city = 'London')
                and city = 'London';
```

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)

```
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);
```

90. Insert into table emp1 empno, sal and deptno from emp table.

If table emp1 is created then

```
insert into emp1 ( select empno,sal,deptno
                   from emp);
```

IF table is not created then

```
Create table emp1 as ( select empno,sal,deptno
                       from emp);
```

91. Update Salary of all employees by 10%.

```
Update emp
set sal = sal + 0.10 * sal;
```

92. Delete all rows from emp for deptno = 10.

```
Delete from emp
where deptno = 10;
```

93. Select list of all jobs which have an annual average salary greater than that managers.

```
Select job,avg(sal)
from emp
group by job
having avg(sal) > ( select avg(sal)
                    from emp
                        where job = 'MANAGER');
```

94. Select list of all employees who have atleast one other employee reporting to them.

```
Select  a.job, a.ename, a.empno, a.deptno
```

```
from emp a
where exists ( select *
                from emp b
                where a.empno = b.mgr);
```

95. Select all employees with correponding level numbers.

```
Column orgn_chart format a21
Select lpad(' ',3*level)|| ename orgn_charts,level,empno,job,mgr
from emp
connect by prior empno = mgr
start with name = 'KING';
```

96. Select average salary for employee at each level.

```
Select level,avg(sal)
from emp
connect by prior empno = mgr
start with name = 'KING'
group by level
order by level;
```

97. Display organization chart for only those employee who work under ‘JONES’.

```
Column orgn_chart format a21
Select lpad(' ',3*level)|| ename orgn_charts,level,empno,job,mgr
from emp
connect by prior empno = mgr
start with name = 'JONES';
```

98. Display organization chart for only those employee who work under ‘JONES’ and ‘BLAKE’.

```
Column orgn_chart format a21
Select lpad(' ',3*level)|| ename orgn_charts,level,empno,job,mgr
from emp
connect by prior empno = mgr
```

start with name in ('JONES',’BLAKE’);

99. List information about all the people in the organization above ‘ADAMS’.

```
Column orgn_chart format a21
Select lpad(' ',3*level)|| ename orgn_charts,empno,job,mgr
from emp
connect by empno = prior mgr
start with name = ‘ADAMS’;
```

100. List all the people who work under ‘BLAKE’ except ‘JAMES’.

```
Column orgn_chart format a21
Select lpad(' ',3*level)|| ename orgn_chart,level,empno,job,mgr
from emp
where ename != 'JAMES'
connect by prior empno = mgr
start with ename = 'BLAKE';
```

101. List all the people who work under ‘KING’ except all employees working under ‘BLAKE’.

```
Select lpad(' ',3*level)|| ename orgn_chart,level,empno,job,mgr
from emp
connect by prior empno = mgr
and ename != 'BLAKE'
start with ename = 'KING'
```

102. List all the people who work under ‘KING’ except ‘ADAMS’ and ‘BLAKE’ and all employees working under ‘BLAKE’.

```
Select lpad(' ',3*level)|| ename orgn_chart,level,empno,job,mgr
from emp
where ename != ‘ADAMS’
connect by prior empno = mgr
and ename != 'BLAKE'
start with ename = 'KING'
```

103. Select max salarys of deptno 10,20 and 30 in single row.

```
Select min(decode(deptno,10,max(sal))) "Dept No 10",
       min(decode(deptno,20,max(sal))) "Dept No 20",
              min(decode(deptno,30,max(sal))) "Dept No 30"
        from emp
        group by deptno;
```

104. If supply table has three fields vendor, job, part. Find list of vendor who are supplying all part for given job.

```
Select a.vendor,a.job,count(*)
from supply a
group by a.vendor,a.job
having count(*) = ( select count(*)
                          from supply b
                    where a.job = b.job);
```

105. List all pairs of orders having same item and qty.

```
        Select a.ordid, b.ordid
    from item a, item b
    where not exists (  select itemid,qty
                        from item c
                        where c.ordid = a.ordid
                        minus
                    select itemid,qty
                        from item d
                        where d.ordid = b.ordid ) and
          not exists (  select itemid,qty
                        from item c
                        where c.ordid = b.ordid
                        minus
                            select itemid,qty
                        from item d
                        where d.ordid = a.ordid ) and
    a.ordid < b.ordid
```

group by a.ordid,b.ordid

Table name: lot_mas t

Structure:

lot_no	lot_desc	storage
a	Pentium	fabrication
a	486	fabrication
b	Pentium	fabrication
d	Pentium	fabrication
d	Pentium	fabrication
.	.	.
.	.	.
.	.	.

I want to display the out put like

lot_no	times
a	2
d	3
.	.
.	.

```
select lot_no,count(lot_no)
        from lot_mast
        group by lot_no
        having count(lot_no) > 1;
```

There are twelve records for each employee in a year. The table structure is as follow.

Empno	month_no	salary
1000	01	2000

1000	02	3500
1000	03	2500
....	..	....
....	..	....
1000	12	3450
2000	01	1900
2000	02	5000
....	..	....
....	..	....
2000	12	5450
3000	01	1900
3000	02	5000
....	..	....
....	..	....

The output should be as follows

empno	Jan	Feb	Mar	Apr	May	.....	Dec
1000	2000	3500	2500	....	....	.....	3450
2000	1900	5000	....	....	....	.....	5450
3000	1900	5000	....	....	....	.....	.....
....	....	....	....	....	....	....	....
....	....	....	....	....	....	....	....
....	....	....	....	....	....	....	....

```
select a.empno,a.salary,b.salary,c.salary,d.salary,e.salary,
f.salary,g.salary,h.salary,i.salary,j.salary,k.salary,l.salary
from emp a,emp b,emp c,emp d,emp e,emp
        f,emp g,emp h,emp i,emp j,emp,k.emp,emp l
        where a.month = 1 and a.empno = b.empno and b.month = 2
                and b.empno = c.empno and c.month = 3
```

```
and c.empno = d.empno and d.month = 4
and d.empno = e.empno and e.month = 5
and d.empno = f.empno and f.month = 6
and d.empno = g.empno and g.month = 7
    and d.empno = h.empno and h.month = 8
    and d.empno = i.empno and i.month = 9
    and d.empno = j.empno and j.month = 10
    and d.empno = k.empno and k.month = 11
    and d.empno = l.empno and l.month = 12;
```

The table structure is as follows

start_site_name	end_site_name
F1	F2
F2	F3
F3	F4
F5	F6
F6	F7
Fa	Fb
Fb	Fc
..	..
..	..

The out will be

level	start_site_name	end_site_name
1	F1	F2
2	F2	F3
3	F3	F4
1	F5	F6
2	F6	F7

1	Fa	Fb
2	Fb	Fc
..	..	..
..	..	..
..	..	..

```
select level,start_site_name,end_site_name
        from table_name
        connect by prior end_site_name = start_site_name
        start with start_site_name = F1;
```