**MYSQL:**

create database retail;

use retail;

CREATE TABLE IF NOT EXISTS salespeople (

snum INT NOT NULL,

sname VARCHAR(30) NOT NULL,

city VARCHAR(30) NOT NULL,

comm DECIMAL(4,2) NOT NULL,

PRIMARY KEY (snum)

);

INSERT INTO salespeople VALUES (1001, 'Peel', 'London', 0.12);

INSERT INTO salespeople VALUES (1002, 'Serres', 'San Jose', 0.13);

INSERT INTO salespeople VALUES (1004, 'Motika', 'London', 0.11);

INSERT INTO salespeople VALUES (1007, 'Rifkin', 'Barcelona', 0.15);

INSERT INTO salespeople VALUES (1003, 'AxelRod', 'NewYork', 0.10);

INSERT INTO salespeople VALUES (1005, 'Fran', 'London', 0.26);

--------------------------------------------------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS customer (

cnum

INT NOT NULL,

cname VARCHAR(30)

NOT NULL,

city VARCHAR(30)

NOT NULL,

rating int not null,

snum int NOT NULL,

PRIMARY KEY (cnum),

FOREIGN KEY (snum) REFERENCES salespeople(snum)

);

INSERT INTO customer VALUES (2001, 'Hoffman', 'London',100, 1001);

INSERT INTO customer VALUES (2002, 'Giovanni', 'Rome',200, 1003);

INSERT INTO customer VALUES (2003, 'Liu', 'San Jose', 200, 1002);

INSERT INTO customer VALUES (2004, 'Grass', 'Berlin',300, 1002);

INSERT INTO customer VALUES (2006, 'Clemens', 'London', 100, 1001);

INSERT INTO customer VALUES (2008, 'Cisneros', 'San Jose', 300, 1007);

INSERT INTO customer VALUES (2007, 'Pereira', 'Rome',100, 1004);

--------------------------------------------------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS orders (

onum

INT NOT NULL,

amt DECIMAL(7,2) NOT NULL,

odate Date

NOT NULL,

cnum int NOT NULL,

PRIMARY KEY (onum),

FOREIGN KEY (cnum) REFERENCES customer(cnum)

);

INSERT INTO orders VALUES (3001, 18.69, '1996-03-10', 2008);

INSERT INTO orders VALUES (3003, 767.19, '1996-03-10', 2001);

INSERT INTO orders VALUES (3002, 1900.10, '1996-03-10', 2007);

INSERT INTO orders VALUES (3005, 5160.45, '1996-03-10', 2003);

INSERT INTO orders VALUES (3006, 1098.16, '1996-03-10', 2008);

INSERT INTO orders VALUES (3009, 1713.23, '1996-04-10', 2002);

INSERT INTO orders VALUES (3007, 75.75, '1996-04-10', 2002);

INSERT INTO orders VALUES (3008, 4723.00, '1996-05-10', 2006);

INSERT INTO orders VALUES (3010, 1309.95, '1996-06-10', 2004);

INSERT INTO orders VALUES (3011, 9891.88, '1996-06-10', 2006);

--------------------------------------------------------------------------------------------------------------------

show tables ;

describe orders;

--------------------------------------------------------------------------------------------------------------------

**Queries:**

**1. List all the columns of the Salespeople table.**

1). SELECT \* from salespeople;

**2. List all customers with a rating of 100.**

SELECT cname from customer where rating = 100;

**3. Find all records in the Customer table with NULL values in the city column.**

SELECT \* from customer WHERE city IS NULL;

**\*\*4. Find the largest order taken by each salesperson on each date.**

select s.sname, sum(o.amt), o.odate from salespeople s

JOIN customer c

ON c.snum = s.snum

JOIN orders o

ON c.cnum = o.cnum

group by o.odate, s.snum

order by sum(o.amt) desc;

(or)

select s.sname, MAX(o.amt), o.odate from salespeople s

JOIN customer c

ON c.snum = s.snum

JOIN orders o

ON c.cnum = o.cnum

group by o.odate, s.snum

order by MAX(o.amt) desc;

**5. Arrange the Orders table by descending customer number.**

Select \* from orders order by cnum desc;

**6. Find which salespeople currently have orders in the Orders table.**

SELECT

a.onum, b.cnum, c.sname

FROM

orders a,

customer b,

salespeople c

WHERE

a.cnum = b.cnum AND b.snum = c.snum

ORDER by c.sname;

**7. List names of all customers matched with the salespeople serving them.**

SELECT

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

FROM

customer c,

salespeople s

WHERE

c.snum = s.snum;

**8. Find the names and numbers of all salespeople who had more than one customer.**

SELECT

s.snum, s.sname, COUNT(c.cnum)

FROM

customer c,

salespeople s

WHERE

c.snum = s.snum

GROUP BY s.snum, s.sname

HAVING COUNT(c.cnum) > 1;

**9. Count the orders of each of the salespeople and output the results in descending order.**

SELECT

COUNT(a.onum), b.snum

FROM

orders a,

customer b,

salespeople c

WHERE

a.cnum = b.cnum AND b.snum = c.snum

group by b.snum;

**10. List the Customer table if and only if one or more of the customers in the Customer table are**

**located in San Jose.**

select \* from customer where city ='San Jose';

**11. Match salespeople to customers according to what city they lived in.**

**~~\*\*~~** ~~select a.cname, b.sname, b.city from customer a ,salespeople b where a.snum = b.snum;~~

select a.cname, b.sname, b.city from customer a ,salespeople b where a.city = b.city ;

**12. Find the largest order taken by each salesperson.**

select s.sname, max(o.amt) from salespeople s

JOIN customer c

ON c.snum = s.snum

JOIN orders o

ON c.cnum = o.cnum

group by s.sname;

**13. Find customers in San Jose who have a rating above 200.**

select \* from customer where city ='San Jose' AND rating > 200;

**14. List the names and commissions of all salespeople in London.**

select sname, comm from salespeople WHERE city LIKE 'London';

**15. List all the orders of salesperson Motika from the Orders table.**

select s.sname, o.onum, o.amt from salespeople s

JOIN customer c

ON c.snum = s.snum

JOIN orders o

ON c.cnum = o.cnum

WHERE s.sname LIKE "Motika";

**16. Find all customers with orders on October 3.**

select o.onum, o.odate, c.cname from customer c

JOIN orders o

ON c.cnum = o.cnum

WHERE o.odate = "1996-03-10";

**17. Give the sums of the amounts from the Orders table, grouped by date, eliminating all those**

**dates where the SUM was not at least 2000.00 above the MAX amount.**

select onum, SUM(amt) from orders

GROUP by odate

having SUM(amt) > 2000;

**18. Select all orders that had amounts that were greater than at least one of the orders from**

**October 6.**

select onum, amt from orders

WHERE

amt > (select min(amt) from orders WHERE odate = "1996-06-10");

**19. Write a query that uses the EXISTS operator to extract all salespeople who have customers**

**with a rating of 300.**

Select \* from salespeople

where exists

( select rating from customer

where customer.snum=salespeople.snum

and rating=300);

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

SELECT

cname, rating

FROM

customer

WHERE rating IN

(select rating from customer group by rating)

order by rating;

**21. Find all customers whose CNUM is 1000 above the SNUM of Serres.**

select cnum from customer WHERE cnum > (1000 +(select snum from salespeople WHERE sname = "Serres"));

**22. Give the salespeople’s commissions as percentages instead of decimal numbers.**

select sname, comm, (comm \* 100) AS percentage from salespeople;

**23. Find the largest order taken by each salesperson on each date, eliminating those MAX orders**

**which are less than $3000.00 in value.**

select s.sname, MAX(o.amt), o.odate from salespeople s

JOIN customer c

ON c.snum = s.snum

JOIN orders o

ON c.cnum = o.cnum

group by o.odate, s.snum

HAVING MAX(o.amt)>3000;

**24. List the largest orders for October 3, for each salesperson.**

select s.sname, SUM(o.amt), o.odate from salespeople s

JOIN customer c

ON c.snum = s.snum

JOIN orders o

ON c.cnum = o.cnum

WHERE o.odate LIKE "1996-03-10"

group by o.odate, s.snum

**25. Find all customers located in cities where Serres (SNUM 1002) has customers.**

SELECT \* FROM customer

WHERE

city = (SELECT city FROM salespeople WHERE sname LIKE 'Serres') AND snum = '1002';

**26. Select all customers with a rating above 200.00.**

select \* from customer Where rating > 200.00;

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

SELECT

count(distinct c.snum) AS ActiveStaff

FROM

customer c,

orders o

WHERE

c.cnum = o.cnum;

**28. Write a query that produces all customers serviced by salespeople with a commission above**

**12%. Output the customer’s name and the salesperson’s rate of commission.**

SELECT

c.cname, s.sname, s.comm

FROM

salespeople s

JOIN

customer c ON s.snum = c.snum

WHERE

s.comm > (12 / 100);

**29. Find salespeople who have multiple customers.**

SELECT

c.snum, COUNT(c.snum), s.sname

FROM

salespeople s

JOIN

customer c ON s.snum = c.snum

GROUP BY snum

HAVING COUNT(c.snum) > 1;

**30. Find salespeople with customers located in their city.**

SELECT

s.sname, s.city, c.cname, c.city

FROM

salespeople s, customer c

HAVING s.city = c.city;

**(Or)**

SELECT

\*

FROM

salespeople s

JOIN

customer c ON s.snum = c.snum

GROUP BY c.snum

HAVING s.city = c.city;

**31. Find all salespeople whose name starts with ‘P’ and the fourth character is ‘l’.**

Select sname from salespeople Where sname LIKE "P\_\_l%";

**32. Write a query that uses a subquery to obtain all orders for the customer named Cisneros.**

**Assume you do not know his customer number.**

select \* from orders Where cnum = (select cnum from customer Where cname = "Cisneros")

**33. Find the largest orders for Serres and Rifkin.**

SELECT

cnum, MAX(amt)

FROM

orders

WHERE

cnum IN (SELECT

cnum

FROM

customer

WHERE

snum IN (SELECT

snum

FROM

salespeople

WHERE

sname = 'Serres' OR sname = 'Rifkin'))

group by cnum;

**34. Extract the Salespeople table in the following order : SNUM, SNAME, COMMISSION, CITY.**

Select snum AS SNUM, sname AS SNAME, comm AS COMMISSION, city AS CITY from salespeople;

**35. Select all customers whose names fall in between ‘A’ and ‘G’ alphabetical range.**

select cname from customer Where cname BETWEEN "A%" AND "G%";

**36. Select all the possible combinations of customers that you can assign.**

select \* from salespeople cross join customer;

**37. Select all orders that are greater than the average for October 4.**

select \* from orders where amt > (select AVG(amt) from orders Where odate LIKE "1996-03-10")

**38. Write a select command using a corelated subquery that selects the names and numbers of all**

**customers with ratings equal to the maximum for their city.**

select a.cname, a.cnum, a.rating, a.city from customer a where a.rating = (select max(b.rating) from customer b Where a.city = b.city) ;

**39. Write a query that totals the orders for each day and places the results in descending order.**

select odate, sum(amt) from orders group by odate order by sum(amt) desc;

**40. Write a select command that produces the rating followed by the name of each customer in**

**San Jose.**

select rating, cname from customer Where city = "San Jose";

**41. Find all orders with amounts smaller than any amount for a customer in San Jose.**

select amt from orders WHERE amt < (select max(o.amt) from orders o JOIN customer c on o.cnum=c.cnum Where city = "San Jose");

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

select c.cname from customer c join orders o on c.cnum=o.cnum Where amt > (select avg(amt) from orders);

**43. Write a query that selects the highest rating in each city.**

select city, MAX(rating) from customer group by city;

**44. Write a query that calculates the amount of the salesperson’s commission on each order by a**

**customer with a rating above 100.00.**

SELECT s.sname, o.amt, s.comm, c.cname, c.rating

FROM

salespeople s

JOIN customer c ON s.snum = c.snum

JOIN orders o ON c.cnum = o.cnum

WHERE c.cnum IN (SELECT c1.cnum FROM customer c1 WHERE rating > 100)

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

select count(cnum) from customer Where rating > (select avg(rating) from customer where city = "San Jose");

**46. Write a query that produces all pairs of salespeople with themselves as well as duplicate rows**

**with the order reversed.**

**????????????????????????????????????????????????????????????????**

**47. Find all salespeople that are located in either Barcelona or London.**

select sname, city from salespeople Where city = "Barcelona" OR city = "London";

**48. Find all salespeople with only one customer.**

select snum from customer group by snum having count(snum) = 1 ;

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

5/11/2017 OBIEE TRAINING: sql queries

http://obieetraining11.blogspot.in/2012/06/sqlqueries.

html 6/15

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(amt)

from orders

group by snum

having sum(amt) > ( select max(amt)

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 = ‘03OCT94’

;

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(amt)

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

5/11/2017 OBIEE TRAINING: sql queries

http://obieetraining11.blogspot.in/2012/06/sqlqueries.

html 7/15

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 != '06oct94'

and

amt > ( select min(amt)

from orders

where odate = '06oct94')

;

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);

5/11/2017 OBIEE TRAINING: sql queries

http://obieetraining11.blogspot.in/2012/06/sqlqueries.

html 9/15

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 = '03OCT94'

;

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’.

5/11/2017 OBIEE TRAINING: sql queries

http://obieetraining11.blogspot.in/2012/06/sqlqueries.

html 12/15

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);