TABLES

Queries to create SALESPEOPLE table and insert values into the table-

```
CREATE TABLE SALESPEOPLE(SNUM NUMBER(5) PRIMARY KEY,  SNAME
CHAR(10),  CITY CHAR(20),  COMM DECIMAL(8,3));
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, 'Rafkin','Barcelona',0.15);
INSERT INTO SALESPEOPLE VALUES (1003, 'Axelrod', 'New York',0.1);
```

Queries to create CUST table and insert values into the table –

```
CREATE TABLE CUST(CNUM NUMBER(5) PRIMARY KEY,  CNAME CHAR(20),
CITY CHAR(20), RATING NUMBER(3),SNUM NUMBER(4));
INSERT INTO CUST VALUES (2001, 'Hoffman', 'London',100,1001);
INSERT INTO CUST VALUES (2002, 'Giovanne', 'Rome',200,1003);
INSERT INTO CUST VALUES (2003, 'Liu', 'San Jose',300,1002);
INSERT INTO CUST VALUES (2004, 'Grass', 'Brelin',100,1002);
INSERT INTO CUST VALUES (2006, 'Clemens', 'London',300,1007);
INSERT INTO CUST VALUES (2007, 'Pereia', 'Rome',100,1004);
```

Queries to create ORDERS table and insert values into the table –

```
CREATE TABLE ORDERS(ONUM NUMBER(5) PRIMARY KEY,  AMT NUMBER(5,2),
ODATE Date, CNUM NUMBER(4),SNUM NUMBER(4));
INSERT INTO ORDERS VALUES (3001,18.69,'03-OCT-94',2008,1007);
INSERT INTO ORDERS VALUES (3003,767.19,'03-OCT-94',2001,1001);
INSERT INTO ORDERS VALUES (3002,1900.10,'03-OCT-94',2007,1004);
INSERT INTO ORDERS VALUES (3005,5160.45,'03-OCT-94',2003,1002);
INSERT INTO ORDERS VALUES (3006,1098.16,'04-OCT-94',2008,1007);
INSERT INTO ORDERS VALUES (3009,1713.23,'04-OCT-94',2002,1003);
INSERT INTO ORDERS VALUES (3007,75.75,'05-OCT-94',2006,1002);
INSERT INTO ORDERS VALUES (3008,4723.00,'05-OCT-94',2006,1001);
INSERT INTO ORDERS VALUES (3010,1309.95,'06-OCT-94',2004,1002);
INSERT INTO ORDERS VALUES (3011,9891.88,'06-OCT-94',2006,1001);
```

QUERIES

1.Obtain all orders for the customer named Cisnerous.

Query-
select o.onum from ORDERS o JOIN CUST c on c.cnum=o.cnum where cname='Cisnerous';

No Output (no such records exist)

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

Query-
select cname,rating from cust where rating > (select avg(rating) from cust);

Output-
Giovanne|200
Liu|300
Clemens|300

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

Query-
select distinct sum(amt),snum from ORDERS where amt>(SELECT max(onum) FROM ORDERS);

Output-
19775.33|1001

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

Query-
select cnum from ORDERS where ODATE='03-OCT-94';

Output-
2008
2001
2007

2003

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

Query-
select s.snum,s.sname from SALESPEOPLE s JOIN CUST c on s.snum=c.snum group by s.snum having count(cnum)>1;

Output-
1002|Serres

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

Query-
Select onum, o.cnum, o.snum, c.snum from orders o, cust c where o.cnum = c.cnum and o.snum != c.snum;

Output-
3007|2006|1002|1007
3008|2006|1001|1007
3011|2006|1001|1007

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

Query-
select onum from ORDERS where amt>(select avg(amt) from ORDERS);

Output-
3005
3008
3011

8.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.

Query-
select sum(amt) from ORDERS group by odate having sum(amt)>(max(amt)+2000);

Output-
7846.43

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

Query-
select cname,cnum,rating from CUST where rating in (SELECT max(rating) from CUST group by city);

Output-
Hoffman|2001|100
Giovanne|2002|200
Liu|2003|300
Grass|2004|100
Clemens|2006|300
Pereia|2007|100

10.Find all salespeople who have customers in their cities who they don't service.

Query-
select s.snum,c.snum from salespeople s join cust c on s.city = c.city where s.snum != c.snum;

Output-
1001|1007
1004|1001
1004|1007

11.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.

Query-
select cnum,cname,city from cust where city in(select city from cust where city='San Jose' group by city having count(city)>1);

No Output (no such records exist)

12.Find salespeople no. who have multiple customers.

Query-
select snum,count(snum) from cust group by snum having count(snum)>1;

Output-
1002|2

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

Query-
select c.snum,s.sname,s.city from cust c join salespeople s on c.snum=s.snum group by c.snum having count(c.snum)>1;

Output-
1002|Serres|San Jose

14.Find salespeople who serve only one customer.

Query-
select snum,sname from salespeople where snum in (select snum from cust where cnum in(select cnum from orders group by cnum having count(cnum)=1));

Output-
1001|Peel
1002|Serres
1004|Motika

15.Extract all salespeople with more than one current order.

Query-
select distinct snum from orders where cnum in (select cnum from orders group by cnum having count(cnum)>1);

Output-
1007
1002
1001

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

Query-
select s.snum,s.sname from salespeople s where exists(select * from cust c where c.snum=s.snum and c.rating=300);

Output-
1002|Serres|San Jose|0.13
1007|Rafkin|Barcelona|0.15

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

Query-
select s.snum,s.sname from salespeople s Join cust c on c.snum=s.snum where c.rating=300;

Output-
1002|Serres
1007|Rafkin

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

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

Output-
1001|Peel
1004|Motika

19.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.

Query-
select o.cnum, max(c.cname) from orders o, cust c
where o.cnum = c.cnum group by o.cnum,o.snum

having count(*) < (select count(*) from orders o1 where o.snum = o1.snum) order by o.cnum;

Output-
2001|Hoffman
2003|Liu
2004|Grass
2006|Clemens
2006|Clemens

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

Query Using IN-
Select sname from salespeople
where snum in (select snum from cust where salespeople.city = cust.city and salespeople.snum = cust.snum);

Query Using ANY-
Select sname from salespeople
where snum = any (select snum from cust where salespeople.city = cust.city and salespeople.snum = cust.snum);

Output-
Peel
Serres

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

Query Using EXISTS-
Select sname from salespeople where exists
(select cname from cust where salespeople.snum = cust.snum and salespeople.sname < cust.cname);

Query Using ANY-
Select sname from salespeople
where sname < any (select cname from cust where salespeople.snum = cust.snum);

Output-
Motika
Axelrod

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

Query-
select cname from cust where rating >(select max(rating) from cust where city='Rome');
Output-

Output-
Liu
Clemens

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

Query-
select * from orders
where odate!='06-OCT-94' and amt > (select min(amt) from orders where odate = '06-OCT-94');

Output-
3002|1900.1|03-OCT-94|2007|1004
3005|5160.45|03-OCT-94|2003|1002
3009|1713.23|04-OCT-94|2002|1003
3008|4723|05-OCT-94|2006|1001

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

Query without ANY-
Select onum, amt from orders
where amt < (select max(amt) from orders o, cust c where city = 'San Jose' and o.cnum = c.cnum);

Query with ANY-
Select onum, amt
from orders

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

Output-
3001|18.69
3003|767.19
3002|1900.1
3006|1098.16
3009|1713.23
3007|75.75
3008|4723
3010|1309.95

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

Query-
select all c1.cname,c1.cnum from cust c1
where not exists (select c2.rating from cust c2 where c2.city != 'Paris' and c2.rating > c1.rating);

Output-
Liu|2003
Clemens|2006

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

Query-
select cname from cust where rating >= ANY (select rating from cust where snum =(select snum from salespeople where sname='Serres'));

Output-
Grass
Cisneros
Liu
Giovani

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

Query using ALL-
select all snum,sname from salespeople where sname not in (select sname from salespeople where city in (select city from cust));

Query using ANY-
select snum,sname from salespeople where sname not in (select sname from salespeople where city =any(select city from cust));

Output-
1007 Rifkin
1003 AxelRod

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

Query-
select onum,amt from orders where amt >(select max(amt) from orders where snum in (select snum from cust where city ='London'));

No Output (no such records exist)

29.Find all salespeople and customers located in london.

Query-
select snum,cnum from cust where city='London';

Output-
1001|2001
1007|2006

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

Query-
select o1.amt, o1.odate, o2.amt, o2.odate from orders o1, orders o2 where (o1.amt, o2.amt) in (select max(amt), min(amt) from orders group by snum);

Output-
1900.1|03-OCT-94|1900.1|03-OCT-94
5160.45|03-OCT-94|75.75|05-OCT-94

1098.16|04-OCT-94|18.69|03-OCT-94
1713.23|04-OCT-94|1713.23|04-OCT-94
9891.88|06-OCT-94|767.19|03-OCT-94

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

Query-
```
select snum, city, 'Customer Present'
from salespeople s
where exists ( select snum from cust where s.snum = cust.snum and s.city =
cust.city)
UNION
select snum, city, 'Customer Not Present'
from salespeople s
where exists ( select snum from cust c where s.snum = c.snum and s.city !=
c.city and c.snum not in
(select snum from cust where s.snum = cust.snum and s.city = cust.city));
```

Output-
1001|London|Customer Present
1002|San Jose|Customer Present
1003|New York|Customer Not Present
1004|London|Customer Not Present
1007|Barcelona|Customer Not Present

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

Query-
```
select c.cname,s.sname,c.city,s.city,
case
when c.city=s.city then 'Matched'
else 'Not Matched'
end
from cust c,salespeople s where c.snum = s.snum;
```

Output-
Hoffman|Peel|London|London|Matched
Giovanne|Axelrod|Rome|New York|Not Matched

Liu|Serres|San Jose|San Jose|Matched
Grass|Serres|Brelin|San Jose|Not Matched
Clemens|Rafkin|London|Barcelona|Not Matched
Pereia|Motika|Rome|London|Not Matched

33.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'.

Query-
```
select cname, city, rating, 'Higher Rating'
from cust
where rating >= 200
UNION
select cname, city, rating, 'Lower Rating'
from cust
where rating < 200;
```

Output-
Clemens|London|300|Higher Rating
Giovanne|Rome|200|Higher Rating
Grass|Brelin|100|Lower Rating
Hoffman|London|100|Lower Rating
Liu|San Jose|300|Higher Rating
Pereia|Rome|100|Lower Rating

34.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.

Query-
```
select o.cnum,c.cname from orders o join cust c on c.cnum=o.cnum group by o.cnum having count(*) > 1
UNION
select o.snum,s.sname from orders o join salespeople s on s.snum=o.snum group by o.snum having count(*) > 1  order by c.cname;
```

Output-
2006|Clemens
1001|Peel

1007|Rafkin
1002|Serres

35.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.

Query-
Select snum from salespeople where city = 'San Jose'
UNION
select cnum from cust where city = 'San Jose'
UNION ALL
select onum from orders where odate = '03-OCT-94';

Output-
1002
2003
3001
3003
3002
3005

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

Query-
Select snum, sname from salespeople
where snum in (select snum from cust where cust.snum = salespeople.snum and cust.city = 'London') and city = 'London';

Output-
1001|Peel

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

Query-
Select snum, sname from salespeople

where snum not in (select snum from cust where cust.snum = salespeople.snum and cust.city='London') and city='London';

Output-
1004|Motika

38.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)

Query-
Select s.sname, c.cname from cust c join salespeople s on c.snum = s.snum
UNION
select distinct s.sname,c.cname from cust c join salespeople s on c.snum = s.snum
where 0 = (select count(*) from cust);

Output-
Axelrod|Giovanne
Motika|Pereia
Peel|Hoffman
Rafkin|Clemens
Serres|Grass
Serres|Liu