

```
create table account(accno int primary key auto_increment,  
accname varchar(100));
```

```
insert into account(accname) values('a'),('b');
```

```
alter table account auto_increment=10000;
```

7. Redo 4,5,6,7 and 8 queries from 10 queries assignment of  
emp and dep- 30 mins  
without using limit and offset

Q.4

```
Select * from (  
Select ename,deptname,salary,row_number() over(order by  
salary desc) rn  
From emp join dept  
On emp.deptid=dept.deptid) as t  
Where rn=1;
```

– CTE- Common Table Expressions or With Queries

```
With tab  
as(  

```

```
Select ename,deptname,salary,row_number() over(order by
salary desc) rn
From emp join dept
On emp.deptid=dept.deptid)
Select * from tab where rn=1;
```

Q.5

```
Select * from (
Select ename,deptname,salary,row_number() over(order by
salary) rn
From emp join dept
On emp.deptid=dept.deptid) as t
Where rn=1;
```

– CTE- Common Table Expressions or With Queries

```
With tab
as(
Select ename,deptname,salary,row_number() over(order by
salary) rn
From emp join dept
On emp.deptid=dept.deptid)
Select * from tab where rn=1;
```

Q6.

```
Select * from (
```

```
Select ename,deptname,salary,row_number() over(order by
salary desc) rn
From emp join dept
On emp.deptid=dept.deptid) as t
Where rn=2;
```

```
create table Salespeople(snum int,sname varchar(100),city
varchar(100
),comm float);
```

```
insert into salespeople values(1001, 'Peel', 'London', .12);
insert into salespeople values(1002, 'Serres', 'San Jose', .13);
insert into salespeople values(1004, 'Motika', 'London', .11);
insert into salespeople values(1007, 'Rifkin', 'Barcelona', .15);
insert into salespeople values(1003, 'AxelRod', 'New York', .10);
insert into salespeople values(1005, 'Fran', 'London', .26);
```

```
create table customers(cnum int,cname varchar(100),city
varchar(100),
rating int,snum int);
```

```
insert into customers values(2001, 'Hoffman','London', 100,
1001);
insert into customers values(2002,'Giovanni','Rome', 200,
1003);
insert into customers values(2003,'Liu','San Jose', 200, 1002);
```

```
insert into customers values(2004,'Grass','Berlin', 300, 1002);
insert into customers values(2006,'Clemens','London', 100,
1001);
insert into customers values(2008,'Cisneros','San Jose', 300
,1007);
insert into customers values(2007,'Pereira','Rome', 100, 1004);
```

```
create table orders(onum int,amt float,odate date,cnum
int,snum int);
insert into orders values(3001, 18.69 ,'96-10-03' ,2008 ,1007);
insert into orders values(3003, 767.19 ,'96-10-03' ,2001 ,1001);
insert into orders values(3002, 1900.10, '96-10-03' ,2007
,1004);
insert into orders values(3005, 5160.45, '96-10-03' ,2003
,1002);
insert into orders values(3006, 1098.16, '96-10-03' ,2008
,1007);
insert into orders values(3009, 1713.23, '96-10-04' ,2002
,1003);
insert into orders values(3007, 75.75, '96-10-04' ,2002 ,1003);
insert into orders values(3008, 4723.00 ,'96-10-05' ,2006
,1001);
insert into orders values(3010, 1309.95 ,'96-10-06' ,2004
,1002);
insert into orders values(3011, 9891.88, '96-10-06' ,2006
,1001);
```

```
create table Salespeople(
snum int,
sname varchar(10),
city varchar(10),
```

```
comm float  
);
```

```
insert into Salespeople values(1001, 'Peel', 'London', .12);  
insert into Salespeople values(1002, 'Serres', 'San Jose', .13);  
insert into Salespeople values(1004, 'Motika', 'London', .11);  
insert into Salespeople values(1007, 'Rifkin', 'Barcelona', .15);  
insert into Salespeople values(1003, 'AxelRod', 'New York',  
.10);  
insert into Salespeople values(1005, 'Fran', 'London', .26);
```

Table 2:

```
create table Customers(  
cnum int,  
cname varchar(10),  
city varchar(10),  
rating int,  
snum int  
);
```

```
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);
```

Table 3:

```
create table Orders(  
  ONUM int,  
  AMT float,  
  ODATE date,  
  CNUM int,  
  SNUM int  
);
```

```
SNUM int, SNAME varchar(100), CITY varchar(100), COMM  
float);
```

```
insert into Salespeople values(1001,'Peel','London',.12);  
insert into Salespeople values(1002,'Serres','San Jose',.13);  
insert into Salespeople values(1004,'Motika','London',.11);  
insert into Salespeople values(1007,'Rifkin','Barcelone',.15);  
insert into Salespeople values(1003,'AxelRod','New York',.10);  
insert into Salespeople values(1005,'Fran','London',.26);
```

```
create table Customers(  
  CNUM int, CNAME varchar(100), CITY varchar(100), RATING  
  int, SNUM int);  
insert into Customers values(2001, 'Hoffman', 'London', 100,  
  1001);  
insert into Customers values(2002, 'Giovanni', 'Rome', 200,  
  1003);
```

```
insert into Customers values(2003, 'Liu', 'San Jose', 200, 1002);
insert into Customers values(2004, 'Grass', 'Berlin', 300, 1002);
insert into Customers values(2006, 'Clemens', 'London', 100,
1001);
insert into Customers values(2008, 'Cisneros', 'San Jose', 300,
1007);
insert into Customers values(2007, 'Pereira', 'Rome', 100,
1004);
```

```
create table Orders(
ONUM int, AMT float, ODATE date, CNUM int, SNUM int);
insert into Orders values(3001, 18.69, '1996-10-03', 2008,
1007);
insert into Orders values(3003, 767.19, '1996-10-03', 2001,
1001);
insert into Orders values(3002, 1900.10, '1996-10-03', 2007,
1004);
insert into Orders values(3005, 5160.45, '1996-10-03', 2003,
1002);
insert into Orders values(3006, 1098.16, '1996-10-03', 2008,
1007);
insert into Orders values(3009, 1713.23, '1996-10-04', 2002,
1003);
insert into Orders values(3007, 75.75, '1996-10-04', 2002,
1003);
insert into Orders values(3008, 4723.00, '1996-10-05', 2006,
1001);
insert into Orders values(3010, 1309.95, '1996-10-06', 2004,
1002);
insert into Orders values(3011, 9891.88, '1996-10-06', 2006,
1001);
```

— Queries from 1 to 50

1. List all the columns of the Salespeople table.

Desc salespeople;

2. List all customers with a rating of 100.

Select \* from customers where rating=100;

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

Select \* from customers where city is null;

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

Select sname,odate,max(amt) highest\_order  
From salespeople s , orders o  
Where s.snum=o.snum  
Group by sname, odate;

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 sname from salespeople  
Where snum in(select snum from orders);
```

```
Select distinct s.snum from salespeople s, orders o  
Where s.snum=o.snum;
```

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

```
Select cname,sname  
From customers c , salespeople s  
Where c.snum=s.snum;
```

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

```
Select sname,s.snum,count(distinct cnum) cust_count  
From customers c , salespeople s  
Where c.snum=s.snum  
Group by sname,s.snum  
Having count(distinct cnum)>1;
```

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

```
Select sname,count(onum) o_count  
From salespeople s left join orders o  
On s.snum=o.snum  
Group by sname  
Order by 2 desc;
```

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 customers where  
Exists (select 1 from customers where city='San Jose');
```

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

```
Select sname,cname,s.city  
From salespeople s, customers c  
Where c.snum=s.snum  
And s.city=c.city;
```

12. Find the largest order taken by each salesperson.

```
Select sname,max(amt) highest_order  
From salespeople s , orders o  
Where s.snum=o.snum  
Group by sname;
```

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

```
Select * from customers where rating>200 and city='San Jose';
```

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

```
Select sname,comm  
From salespeople  
Where city='London';
```

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

```
Select sname,onum,odate,amt From orders o left join  
salespeople s  
On s.snum=o.snum  
where sname='Motika';
```

16. Find all customers with orders on October 3.

```
Select distinct cname  
From customers c,orders o  
Where c.cnum=o.cnum  
And odate='1996-10-03';
```

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 odate,sum(amt) from orders  
Group by odate  
Having sum(amt)>2000+max(amt);
```

18. Select all orders that had amounts that were greater than at least one of the orders from October 6.

```
Select * from orders where amt>any  
(Select amt from orders where odate='1996-10-06');
```

```
Select * from orders where amt>  
(Select min(amt) from orders where odate='1996-10-06');
```

19. Write a query that uses the EXISTS operator to extract all salespeople who have customers with a rating of 300.

```
Select * from salespeople s
Where exists(select 1
From customers c where c.snum=s.snum and rating=300);
```

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

```
Select c1.cname c1,c2.cname c2,c1.rating
From
Customers c1 , customers c2
Where c1.cname<c2.cname
And c1.rating=c2.rating;
```

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

```
Select * from customers where cnum>
(Select snum from salespeople where
sname='Serres')+1000;
```

```
Select * from customers where cnum>
(Select snum+1000 from salespeople where
sname='Serres');
```

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

```
Select snum,sname,concat(round(comm*100,2),'%')
comm_percent 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  sname,odate,max(amt) highest_order
From salespeople s , orders o
Where s.snum=o.snum
Group by sname, odate
Having max(amt)>=3000;
```

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

```
Select  sname,max(amt) highest_order
From salespeople s , orders o
Where s.snum=o.snum and odate='1996-10-03'
Group by sname;
```

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

```
Select * from customers where city in(Select c.city from
salespeople s, customers c
Where s.snum=c.snum
And sname='Serres');
```

26. Select all customers with a rating above 200.00.

Select \* from customers where rating>200;

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

Select count(distinct snum) from orders;

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 cname,comm  
From customers c , salespeople s  
Where c.snum=s.snum and comm>0.12;

29. Find salespeople who have multiple customers.

Select sname,count(cnum)  
From salespeople s ,customers c  
Where s.snum=c.snum  
Group by sname  
Having count(cnum)>1;

30. Find salespeople with customers located in their city.

Select sname,cname,s.city  
From salespeople s, customers c  
Where c.snum=s.snum  
And s.city=c.city;

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

```
Select * 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 odate,onum,amt  
From orders  
Where cnum=(select cnum from customers where  
cname='Cisneros');
```

```
Select 'Cisneros' cname,odate,onum,amt  
From orders  
Where cnum=(select cnum from customers where  
cname='Cisneros');
```

33. Find the largest orders for Serres and Rifkin.

```
Select sname,max(amt) highest_order  
From salespeople s , orders o  
Where s.snum=o.snum and sname in ('Serres','Rifkin')  
Group by sname;
```



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

```
Select SNUM, SNAME, COMM, CITY  
From salespeople;
```

35. Select all customers whose names fall in between 'A' and 'G' alphabetical range.

```
Select * from customers where left(cname,1)  
Between 'a' and 'g';
```

```
Select * from customers where  
Cname between 'a' and 'h' and cname !='H';
```

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

```
Select c1.cname c1,c2.cname c2,c1.rating  
From  
Customers c1 , customers c2  
Where c1.cname<c2.cname;
```

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='1996-10-04');
```

38. Write a select command using a correlated subquery that selects the names and numbers of all customers with ratings equal to the maximum for their city.

```
Select cname,cnum
From customers co
Where exists(select 1 from customers ci
Where co.city=ci.city
Group by ci.city
Having max(ci.rating)=co.rating);
```

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 2 desc;
```

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

```
Select rating,cname from customers where city='San
Jose';
```

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

```
Select * from orders where amt
<any(select amt from orders o , customers c
Where o.cnum=c.cnum and city='San Jose');
```

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

Select avg (amt) from Orders

Select \* from Orders

Where Amt>(Select avg (amt) from Orders);

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

Select city,max(rating) from customers 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 sname,sum(amt\*comm) total\_comm

from orders o,salespeople s , customers c

Where o.snum=s.snum

And

o.cnum=c.cnum

And

c.snum=s.snum

And rating>100

Group by sname;

Select sname,sum(amt\*comm) total\_comm

from orders o join salespeople s

On o.snum=s.snum

join customers c

on

o.cnum=c.cnum

And

```
c.snum=s.snum
where rating>100
Group by sname;
```

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

```
select count(CNUM) from customers where rating>(
select avg(rating) from customers 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.

```
Select s1.sname s1,s2.sname s2
From salespeople s1, salespeople s2;
```

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

```
select * from salespeople where city in
('Barcelona','London');
```

48. Find all salespeople with only one customer.

```
select sname,count(cnum)
from salespeople s , customers c
where s.snum=c.snum
group by sname
having count(cnum)=1;
```

49. Write a query that joins the Customer table to itself to find all pairs of customers served by a single salesperson.

```
Select c1.cname,c2.cname  
From customers c1,customers c2  
Where c1.cnum<c2.cnum  
And c1.snum=c2.snum;
```

50. Write a query that will give you all orders for more than \$1000.00

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

Custprod

CID	PID
1	1
1	5
1	7
1	9
1	12

2	2
2	6
2	12
3	3
3	7
3	8
3	9
3	10

Create table custprod(cid int, pid int);

```
insert into custprod values(1,1);
insert into custprod values(1,5);
insert into custprod values(1,7);
insert into custprod values(1,9);
insert into custprod values(1,12);
insert into custprod values(2,2);
insert into custprod values(2,6);
insert into custprod values(2,12);
insert into custprod values(3,3);
insert into custprod values(3,7);
insert into custprod values(3,8);
insert into custprod values(3,9);
insert into custprod values(3,10);
```

```

with tab
as (
select c1.cid c1,c2.cid c2,
count(case when c1.pid=c2.pid then c1.pid else null end)
num,
count(distinct c1.pid) + count(distinct c2.pid) den
from custprod c1, custprod c2
where c1.cid<c2.cid
group by c1.cid,c2.cid)
select c1,c2,num,den-num as
den,concat(round(num/(den-num)*100,2),'%') as similarity
from tab;

```

Set @@cte\_max\_recursion\_depth=5000000;

```

With recursive tab
As
(select 1 as id
Union all
Select id+1 from tab where id<2000)
Select * from tab;

```

1. Revise Day 1,2,3 Queries & Theory-1 hour
2. Revise CODDs rules, all normal forms, ACID
3. Do first 100 queries from 125 queries assignment
4. Read about Isolation Levels
5. Read about Data Models
6. Revise slides- 30 mins

## Parts

PartID	ParentPartID
--------	--------------

A	A
---	---

B	A
---	---

C	C
---	---

D	C
---	---

E	D
---	---

F	A
---	---

G	F
---	---

H	D
---	---

I	H
---	---

J	H
---	---



PartID	TopMostParent
--------	---------------

A	A
---	---

B	A
---	---

C	C
---	---

D	C
---	---

E	C
---	---

F	A
---	---

G	A
---	---

H	C
---	---

I	C
---	---

J	C
---	---