Queries:

1. Display the name of all employees whose salary is between 500 and 1300

Select ename,salary from emp where salary between 500 and 1300;

Select ename,salary from emp where salary >= 500 and salary<= 1300;

2. Display the name of all employees who are in HR dept

Subquery

Select ename,'HR' Deptname from emp where deptid=(select deptid from dept where deptname='HR');

Join

Select ename,deptname

From emp e join dept d

On e.deptid=d.deptid

Where deptname='HR';

3. Display the name of all the dept and count of all the employees in that dept.

Select deptname,count(eid) emp\_count

From emp e right join dept d

On e.deptid=d.deptid

Group by deptname;

4. Display the name , deptname and salary of the employee whose salary is highest.

Select \* from (

Select ename,deptname,salary,rank() over(order by salary desc) rn

From emp e join dept d

On e.deptid=d.deptid) as t

Where rn=1;

Select ename,deptname,salary

From emp e join dept d

On e.deptid=d.deptid

Order by salary desc limit 1;

Select ename,deptname,salary

From emp e join dept d

On e.deptid=d.deptid

Where salary=(select max(salary) from emp);

5. Display the name , deptname and salary of the employee whose salary is lowest.

Select \* from (

Select ename,deptname,salary,rank() over(order by salary) rn

From emp e join dept d

On e.deptid=d.deptid) as t

Where rn=1;

Select ename,deptname,salary

From emp e join dept d

On e.deptid=d.deptid

Order by salary limit 1;

6. Display the name , deptname of employee whose salary is second highest.

Select ename,deptname,salary

From emp e join dept d

On e.deptid=d.deptid

Order by salary desc limit 1 offset 1;

Select \* from (

Select ename,deptname,salary,dense\_rank() over(order by salary desc) rn

From emp e join dept d

On e.deptid=d.deptid) as t

Where rn=2;

7. display the name, deptname of top five earning employees.

Select ename,deptname,salary

From emp e join dept d

On e.deptid=d.deptid

Order by salary desc limit 5;

Select \* from (

Select ename,deptname,salary,dense\_rank() over(order by salary desc) rn

From emp e join dept d

On e.deptid=d.deptid) as t

Where rn<=5;

8. Display the deptname, name of top 5 earning employees from each dept.

Select \* from (

Select ename,deptname,salary,dense\_rank() over(partition by deptname order by salary desc) rn

From emp e join dept d

On e.deptid=d.deptid) as t

Where rn<=5;

Select ename,salary,

Case

When salary>=1500 then 'Very High'

When salary>=1000 then 'High'

When salary>=500 then 'Avg'

Else 'Low' end Salary\_Band

From emp

Order by salary;

9. Display the avg salary of each dept.

Select deptname,avg(salary) avg\_sal

From emp e right join dept d

On e.deptid=d.deptid

Group by deptname;

10.Display the min and max salary of each dept.

Select deptname,min(salary) min\_sal,max(salary) max\_sal

From emp e right join dept d

On e.deptid=d.deptid

Group by deptname;

Create table emp\_rank

(eid int, ename varchar(10), salary int, deptid int);

insert into emp\_rank values(1,'a',100,1);

insert into emp\_rank values(2,'b',200,2);

insert into emp\_rank values(3,'c',100,1);

insert into emp\_rank values(4,'d',200,2);

insert into emp\_rank values(5,'e',50,1);

insert into emp\_rank values(6,'f',60,2);

insert into emp\_rank values(7,'g',70,1);

Select eid,ename,salary,deptid,

row\_number() over(order by salary desc) rn,

rank() over(order by salary desc) rnk,

dense\_rank() over(order by salary desc) drnk

From emp\_rank;

Select eid,ename,salary,deptid,

row\_number() over(partition by deptid order by salary desc) rn,

rank() over(partition by deptid order by salary desc) rnk,

dense\_rank() over(partition by deptid order by salary desc) drnk

From emp\_rank;

=CONCAT("insert into emp\_rank values(",[@EID],",'",[@ENAME],"',",[@SALARY],",",[@DEPTID],");")

|  |  |  |
| --- | --- | --- |
| Marks | | |
| **Roll** | **Sub** | **Marks** |
| 1 | M | 98 |
| 1 | E | 96 |
| 1 | H | 99 |
| 2 | M | 92 |
| 2 | E | 94 |
| 2 | H | 20 |
| 3 | M | 36 |
| 3 | E | 37 |
| 3 | H | 38 |

| Result | |
| --- | --- |
| **Roll** | **Result** |
| 1 | Pass |
| 2 | Fail |
| 3 | Fail |

| Pass if in all subjects marks are>=35 and percent is 40 or above else fail | | |
| --- | --- | --- |
|

Create table marks (roll int, sub varchar(5), marks int);

insert into marks values(1,'M',98);

insert into marks values(1,'E',96);

insert into marks values(1,'H',99);

insert into marks values(2,'M',92);

insert into marks values(2,'E',94);

Select roll ,

Case when min(marks)>=35 and avg(marks)>=40 then 'Pass'

Else 'Fail' end Result

From marks

Group by roll;

– write a query that shows all the name of dept who have no employee working in it using correlated sub query

Select deptname from dept

Where not exists (select 1 from emp where emp.deptid=dept.deptid);

– write a query that shows all the name of dept who have atleast one employee working in it using correlated sub query

Select deptname from dept

Where exists (select 1 from emp where emp.deptid=dept.deptid);

– write a correlated sub query to return name and salary of all employees whose salary is greater than average salary of all employees

Select ename,salary from emp o where exists (select 1 from emp i having o.salary>avg(i.salary))

;

Create view vw\_emp\_dept

as

Select ename,deptname,salary

From emp join dept

On emp.deptid=dept.deptid;

Delete from vw\_emp\_dept;

Update vw\_emp\_dept set salary=100 ,deptname='HR1'

Where salary=500;

Update vw\_emp\_dept set salary=100

Where salary=500;

Explain Select \* from vw\_emp\_dept;

show create view vw\_emp\_dept;

show create table emp;

Create table test\_ins(c1 int not null,c2 int not null,c3 int, c4 int, c5 int);

Create view vw\_test\_ins

As

Select c2,c3,c4,c5 from test\_ins;

Delete from vw\_test\_ins;

Insert into vw\_test\_ins values(1,1,1,1);

Desc information\_schema.tables;

Select table\_schema, table\_name,table\_type from information\_schema.tables

Where table\_schema=database()

And table\_type='VIEW';

Create view vw\_emp\_sal\_bonus

as

Select ename,salary basectc, salary\*1.1 totalctc

From emp;

Delete from vw\_emp\_sal\_bonus where ename='a1';

Update vw\_emp\_sal\_bonus set ename='b' where ename='a1';

Insert into emp (ename) values('xyz');

Update vw\_emp\_sal\_bonus set totalctc=1000 where ename='a1';

Insert into vw\_emp\_sal\_bonus (ename) values('xyz');

show variables like 'datadir';

Select salary,count(\*) from emp

Group by salary having count(\*)>1;

Insert into emp values(46,'test',20001,2);

select ename, salary,(select sum(salary) from emp i

Where i.salary> o.salary

Or (i.salary= o.salary

and

i.eid<=o.eid)

) running\_salary\_total

from emp o

Order by salary desc,eid;

Select ename,salary,sum(salary) over(order by salary desc

Rows between unbounded preceding and current row) running\_total\_salary from emp;

select ename, salary,(select max(salary) from emp i

Where i.salary< o.salary

Or (i.salary= o.salary

and

i.eid>o.eid)

) next\_high\_sal

from emp o

Order by salary desc,eid;

Select ename,salary,sum(salary) over(order by salary desc

Rows between 1 following and 1 following) next\_high\_sal from emp;

Select ename,salary,max(salary) over(order by salary desc

Rows between 1 following and 2 following) next\_high\_sal from emp;

Select ename,salary,lead(salary,1,0) over(order by salary desc ) next\_high\_sal from emp;

Select ename,salary,lag(salary,1,0) over(order by salary desc ) prev\_high\_sal from emp;

Select ename,salary,sum(salary) over(order by salary desc

Rows between 1 preceding and 1 preceding) prev\_high\_sal from emp;

select ename, salary,(select min(salary) from emp i

Where i.salary> o.salary

Or (i.salary= o.salary

and

i.eid<o.eid)

) prev\_high\_sal

from emp o

Order by salary desc,eid;

1 preceding

2 preceding

3 preceding

…

n preceding

Unbounded preceding

Current row

1 following

2 following

3 following

..

n following

Unbounded following

| custprod | |
| --- | --- |
| **cid** | **pid** |
| 1 | 1 |
| 1 | 4 |
| 1 | 6 |
| 1 | 9 |
| 1 | 10 |
| 2 | 2 |
| 2 | 4 |
| 2 | 10 |
| 2 | 18 |
| 3 | 1 |
| 3 | 6 |
| 3 | 9 |
| 3 | 13 |
| 3 | 14 |

| 1 | 1,4,6,9,10 |
| --- | --- |
| 2 | 2,4,10,18 |
| 3 | 1,6,9,13,14 |

| Num | count of common products between the two customers |
| --- | --- |
| Den | Count of all the products between the two customers |

| **C1** | **C2** | **Similarity** | **Num** | **Den** |
| --- | --- | --- | --- | --- |
| 1 | 2 | 29% | 2 | 7 |
| 2 | 3 | 0% | 0 | 9 |
| 1 | 3 | 43% | 3 | 7 |

Create table custprod (cid int, pid int);

insert into custprod values(1,1);

insert into custprod values(1,4);

insert into custprod values(1,6);

insert into custprod values(1,9);

insert into custprod values(1,10);

insert into custprod values(2,2);

insert into custprod values(2,4);

insert into custprod values(2,10);

insert into custprod values(2,18);

insert into custprod values(3,1);

insert into custprod values(3,6);

insert into custprod values(3,9);

insert into custprod values(3,13);

insert into custprod values(3,14);

Select c1,c2,num,den,concat(round(num\*100/den,2),'%') as similarity from(

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) - count(case when c1.pid=c2.pid then c1.pid else null end) den

From custprod c1 , custprod c2

Where c1.cid<c2.cid

Group by c1.cid,c2.cid) as t;

Assignment to be done till Wed Morning

1. Revise all topics covered on Day 1 and Day 2 from Slides
2. Redo all the Day 1 and Day 2 queries
3. Revise CODD Rules
4. Revise all Normal forms
5. Complete all 125 queries from PDF assignment
6. Complete all 15 queries from Nov19 doc file
7. Complete all 20 queries from Nov20 doc file
8. Do Query number 8 from 10 queries assignment without using ranking functions
9. Practice thoroughly all ranking, aggregate function with over clause and windowing functions
10. Read PPT on Isolation Levels
11. Read PPT on Data Models