Day 11:

important window function

1)row\_number()

2)rank()

3)dense\_rank()

1)row\_number()

-it is used to generate unique sequential number

>select empid,name,dept\_name,sal,row\_number()over() from emp;

error(order by is mandetory for row\_number(),rank(),dense\_rank())

>select empid,name,dept\_name,sal,row\_number()over(order by sal) as rnk from emp;

2)rank()

rank starts with 1 and it skips the rank when tie is occured

(means if u get the same value)

it assigns same rank to same values

>select empid,name,dept\_name,sal,rank()over(order by sal) as rnk from emp;

3)dense\_rank()

it is similar to rank function, but it will not skip the rank

even if u get the same value

it assign same rank to same values

>select empid,name,dept\_name,sal,dense\_rank()over(order by sal) as rnk from emp;

dense\_rank will follow the series

normal rank will not follow the series

-Fetch employ having highest salary

>select \* from emp order by sal desc fetch first 1 rows only;

or

>select \* from (

select empid,name,dept\_name,sal,dense\_rank()over(order by sal desc) as rn from emp)

where rn=1;

-Fetch employ having 3rd highest salary

>select \* from (

select empid,name,dept\_name,sal,dense\_rank()over(order by sal desc) as rn from emp)

where rn=3;

- if employees having same salary (use dense\_rank())

-Find running sum of salary

>select empid,name,dept\_name,sal,sum(sal)over(order by empid) as running\_sum\_of\_salary from emp;

>select e.\*,sum(sal)over(order by empid) as running\_sum\_of\_salary from emp e;

ex-->

salary row\_number() rank() dense\_rank()

2000 1 1 1

3000 2 2 2

3000 3 2 2

4000 4 4 3

5000 5 5 4

5000 6 5 4

6000 7 7 5

##partition by

partition by clause is used to partitioned the data

-Number of employees in each department

>select e.,count()over(partition by deptno)

as no\_of\_employees from emp e;

-Total salary in each department

>select e.\*,sum(sal) over(partition by deptno) as total\_salary\_in\_each\_department from emp e;

-Maximum salary in each department

>select e.\*,max(sal) over(partition by deptno) as total\_salary\_in\_each\_department from emp e;

-Minimum salary in each department

>select e.\*,min(sal) over(partition by deptno) as total\_salary\_in\_each\_department from emp e;

-Fetch employ having highest salary in each department

>select \* from (

select e.\*,dense\_rank() over(partition by deptno order by sal desc) as rnk from emp e)

where rnk=1;

-Fetch 3rd highest salary from each department

>select \* from (

select e.\*,dense\_rank() over(partition by deptno order by sal desc) as rnk from emp e)

where rnk=3;

-Fetch employees who earn more than average salary of all

employees

>select \* from emp where sal>(select avg(sal) from emp);

-Fetch employees having same salary

>select \* from emp where sal in

(select sal from emp

group by sal

having count(\*)>1)

-Fetch employees having same salary using window function

>select \* from (

select e.,count()over(partition by sal) as cnt from emp e)

where cnt>1;