

Row_num

ROW NUMBER

--It will return the sequential number of row starting at 1

--Order by clause is required.

--PARTITION BY clause is optional

--When the data is partitioned, row number reset to 1 when the partition changes.

--syntax

--ROW_NUMBER() OVER(ORDER BY Col1,col2)

select *, ROW_NUMBER() over (order by Esal) as RowNo from Info_3;

---RANK,DENSE RANK AND ROW NUMBER

--Rank() and Dense_Rank()

--It will return a rank starting at 1 based on ordering of rows and imposed by order by clause.

--Order by clause is required mandatory.

--PARTITION BY Clause is optional.

--Rank Syntax: RANK() OVER (ORDER BY col1,col2,....coln ASC/DESC
[PARTITION BY Col1,col2...coln])

--Dense_Rank Syntax: DENSE_RANK() OVER (ORDER BY col1,col2,....coln
ASC/DESC [PARTITION BY Col1,col2...coln])

--example

--Marks =496,496,495,494,494,490

--rank = 1,1,3,4,4,6

--Dense_rank = 1,1,2,3,3,4

--Example:

--[sal] = [1000,1000,2000,3000,4000]

--Rank() -- [1,1,3,4,5]

--Dense_rank() --[1,1,2,3,4] -- school level mark inside the class

select * ,rank() over (order by Esal) as rank1 from Info_3;

select * ,dense_rank() over (order by Esal) as denserank from info_3;

--Q. What is the difference between Rank() and Dense_Rank()

--Rank() -- Rank function skips ranking if there is same value or number.

--Dense_Rank() --It will not skips ranking if there is same value or number.

--2nd highest salary by using rank()

select * ,rank() over (order by salary) as rank1 from over_Test where
rank() over (order by salary) = 2

select * ,dense_rank() over (order by salary) as denserank from
over_Test where dense_rank() over (order by salary)=2

--The above query will through an exception in

--i.e. Windowed functions can only appear in the SELECT or ORDER BY
clauses.

--In order to avoid this kind of exception or Error in SQL we have to use CTE i.e. COMMON TABLE EXPRESSION

--CTE (Common Table Expression)

--It is temporary result set.

--It will store the temporary results to make use of that in your main query.

--It can be referred within a SELECT, INSERT, UPDATE and DELETE statements that immediately follows the CTE.

--Only DML type of operation we can perform on CTE

--Syntax

--With CTE_NAME (COL1, COL2 ...etc)

--AS

--CTE_Query

Q. Find out 4th max sal

```
with new_rank as
(select Esal, dense_rank() over (order by Esal desc) as ranking from
info_3)
select * from new_rank where ranking=4;
```

Q. Find out 3rd min sal

```
with new_rank as
(select Esal, dense_rank() over (order by Esal asc) as ranking from
info_3)
select * from new_rank where ranking=3;
```

Q. Find out 65th max salary.

Q. Find out 30th min salary.

Q. Display top 5 salary from table.

Q. Display salary from 3rd max to 6th max.

Q. Display salary other than 2nd min to 4th min.

Partition by

```
select *, ROW_NUMBER() over (partition by salary order by salary ) as  
RowNo from over_Test
```

Q. create table over_Test(EMPID int, FirstName varchar(20),Gender
varchar(2),salary int)

```
insert into over_Test values(1,'Mohini','F',1000)
```

```
insert into over_Test values(2,'Rohit','M',2000)
```

```
insert into over_Test values(3,'Amit','M',4000)
```

```
insert into over_Test values(4,'Sonal','F',5000)
```

```
insert into over_Test values(5,'Minal','F',6000)
```

```
insert into over_Test values(6,'Amar','M',3600)
```

```
insert into over_Test values(7,'Shital','F',4500)
```

```
insert into over_Test values(8,'Sohil','M',6000)
```

```
insert into over_Test values(9,'praveen','F',9000)
```

```
insert into over_Test values(10,'Mithali','F',9000)
```

```
insert into over_Test values(11,'seema','F',9000)
```

```
insert into over_Test values(12,'meena','F',10000)
```

```
select * from over_Test
```

