www.PragimTech.com
Pragim@PragimTech.com

SQL Server Tutorial

Part 110 - RANK and DENSE_RANK

Venkat
PRAGIM Technologies
kudvenkat@gmail.com
http://csharp-video-tutorials.blogspot.com

PragimTech.com | facebook.com/pragimtech | twitter.com/kudvenkat | 91 99456 99393

In this session we will learn

- RANK and DENSE_RANK functions in SQL Server

Link to Dot Net Basics, ASP.NET, C#, ADO.NET and SQL Server video series
http://www.youtube.com/user/kudvenkat/playlists

Suggested Videos
Part 107 - Debugging SQL Server stored procedures
Part 108 - Over clause in SQL Server
Part 109 - Row_Number function in SQL Server

RANK and DENSE_RANK in SQL Server

If you are in need of the DVD with all the videos and PPT's, please visit
http://pragimtech.com/order.aspx

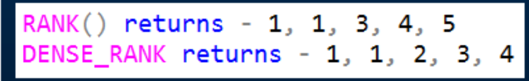
RANK and DENSE_RANK functions

- Introduced in SQL Server 2005
- Return a rank starting at 1 based on the ordering of rows imposed by the ORDER BY clause
- ORDER BY clause is required
- PARTITION BY clause is optional
- When the data is partitioned, rank is reset to 1 when the partition changes

Difference between RANK and DENSE_RANK functions

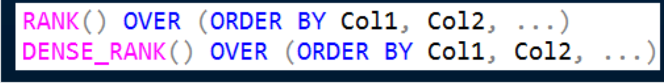
RANK function skips ranking(s) if there is a tie where as DENSE_RANK will not

For example : If you have 2 rows at rank 1 and you have 5 rows in total

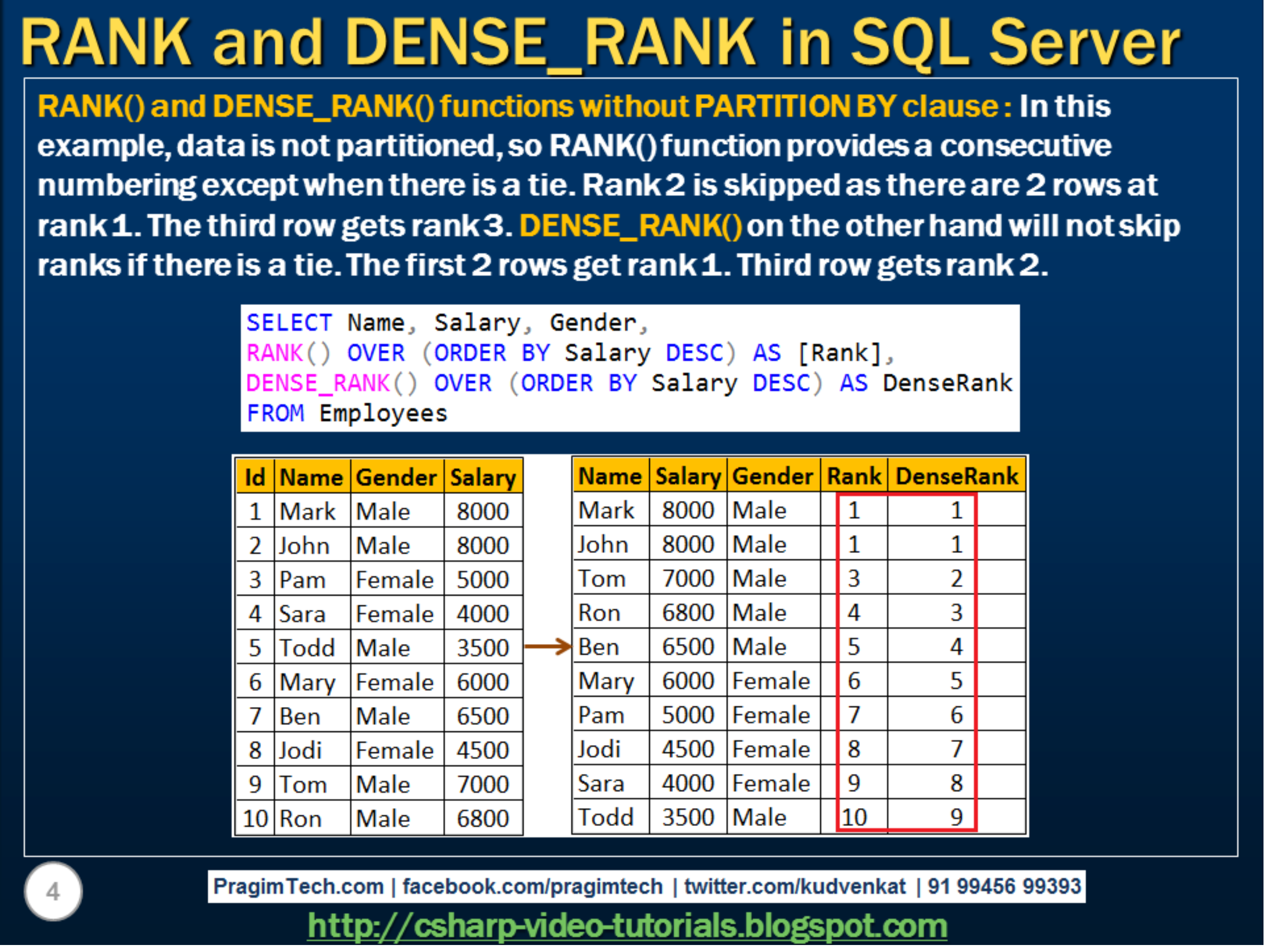


```
RANK() returns - 1, 1, 3, 4, 5
DENSE_RANK returns - 1, 1, 2, 3, 4
```

Syntax :



```
RANK() OVER (ORDER BY Col1, Col2, ...)
DENSE_RANK() OVER (ORDER BY Col1, Col2, ...)
```



RANK and DENSE_RANK in SQL Server

RANK() and DENSE_RANK() functions without PARTITION BY clause : In this example, data is not partitioned, so RANK() function provides a consecutive numbering except when there is a tie. Rank 2 is skipped as there are 2 rows at rank 1. The third row gets rank 3. **DENSE_RANK()** on the other hand will not skip ranks if there is a tie. The first 2 rows get rank 1. Third row gets rank 2.

```
SELECT Name, Salary, Gender,
RANK() OVER (ORDER BY Salary DESC) AS [Rank],
DENSE_RANK() OVER (ORDER BY Salary DESC) AS DenseRank
FROM Employees
```

Id	Name	Gender	Salary
1	Mark	Male	8000
2	John	Male	8000
3	Pam	Female	5000
4	Sara	Female	4000
5	Todd	Male	3500
6	Mary	Female	6000
7	Ben	Male	6500
8	Jodi	Female	4500
9	Tom	Male	7000
10	Ron	Male	6800

Name	Salary	Gender	Rank	DenseRank
Mark	8000	Male	1	1
John	8000	Male	1	1
Tom	7000	Male	3	2
Ron	6800	Male	4	3
Ben	6500	Male	5	4
Mary	6000	Female	6	5
Pam	5000	Female	7	6
Jodi	4500	Female	8	7
Sara	4000	Female	9	8
Todd	3500	Male	10	9

RANK and DENSE_RANK in SQL Server

RANK() and DENSE_RANK() functions with PARTITION BY clause : Notice when the partition changes from Female to Male Rank is reset to 1

```
SELECT Name, Salary, Gender,
RANK() OVER (PARTITION BY Gender ORDER BY Salary DESC) AS [Rank],
DENSE_RANK() OVER (PARTITION BY Gender ORDER BY Salary DESC) AS DenseRank
FROM Employees
```

Id	Name	Gender	Salary
1	Mark	Male	8000
2	John	Male	8000
3	Pam	Female	5000
4	Sara	Female	4000
5	Todd	Male	3500
6	Mary	Female	6000
7	Ben	Male	6500
8	Jodi	Female	4500
9	Tom	Male	7000
10	Ron	Male	6800

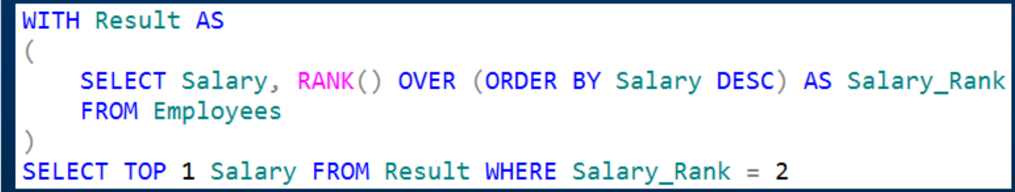
→

Name	Salary	Gender	Rank	DenseRank
Mary	6000	Female	1	1
Pam	5000	Female	2	2
Jodi	4500	Female	3	3
Sara	4000	Female	4	4
Mark	8000	Male	1	1
John	8000	Male	1	1
Tom	7000	Male	3	2
Ron	6800	Male	4	3
Ben	6500	Male	5	4
Todd	3500	Male	6	5

RANK and DENSE_RANK in SQL Server

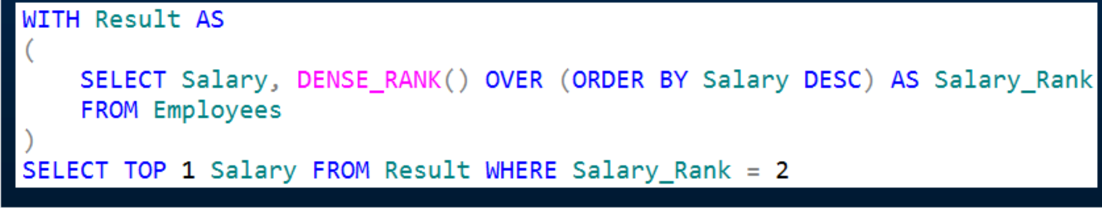
Use case for RANK and DENSE_RANK functions : Both these functions can be used to find Nth highest salary. However, which function to use depends on what you want to do when there is a tie.

Since we have 2 Employees with the FIRST highest salary. Rank() function will not return any rows for the SECOND highest Salary.



```
WITH Result AS
(
    SELECT Salary, RANK() OVER (ORDER BY Salary DESC) AS Salary_Rank
    FROM Employees
)
SELECT TOP 1 Salary FROM Result WHERE Salary_Rank = 2
```

Though we have 2 Employees with the FIRST highest salary. Dense_Rank() function returns, the next Salary after the tied rows as the SECOND highest Salary

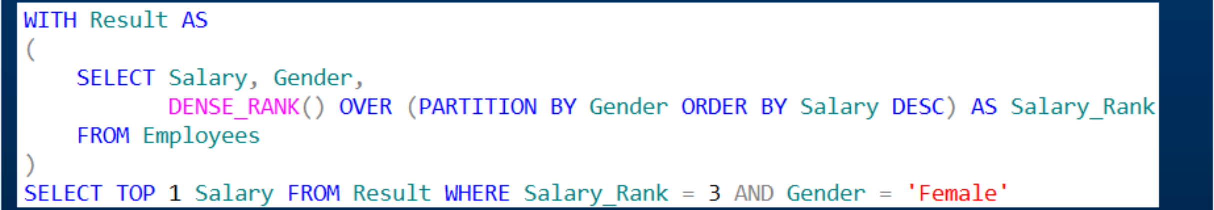


```
WITH Result AS
(
    SELECT Salary, DENSE_RANK() OVER (ORDER BY Salary DESC) AS Salary_Rank
    FROM Employees
)
SELECT TOP 1 Salary FROM Result WHERE Salary_Rank = 2
```

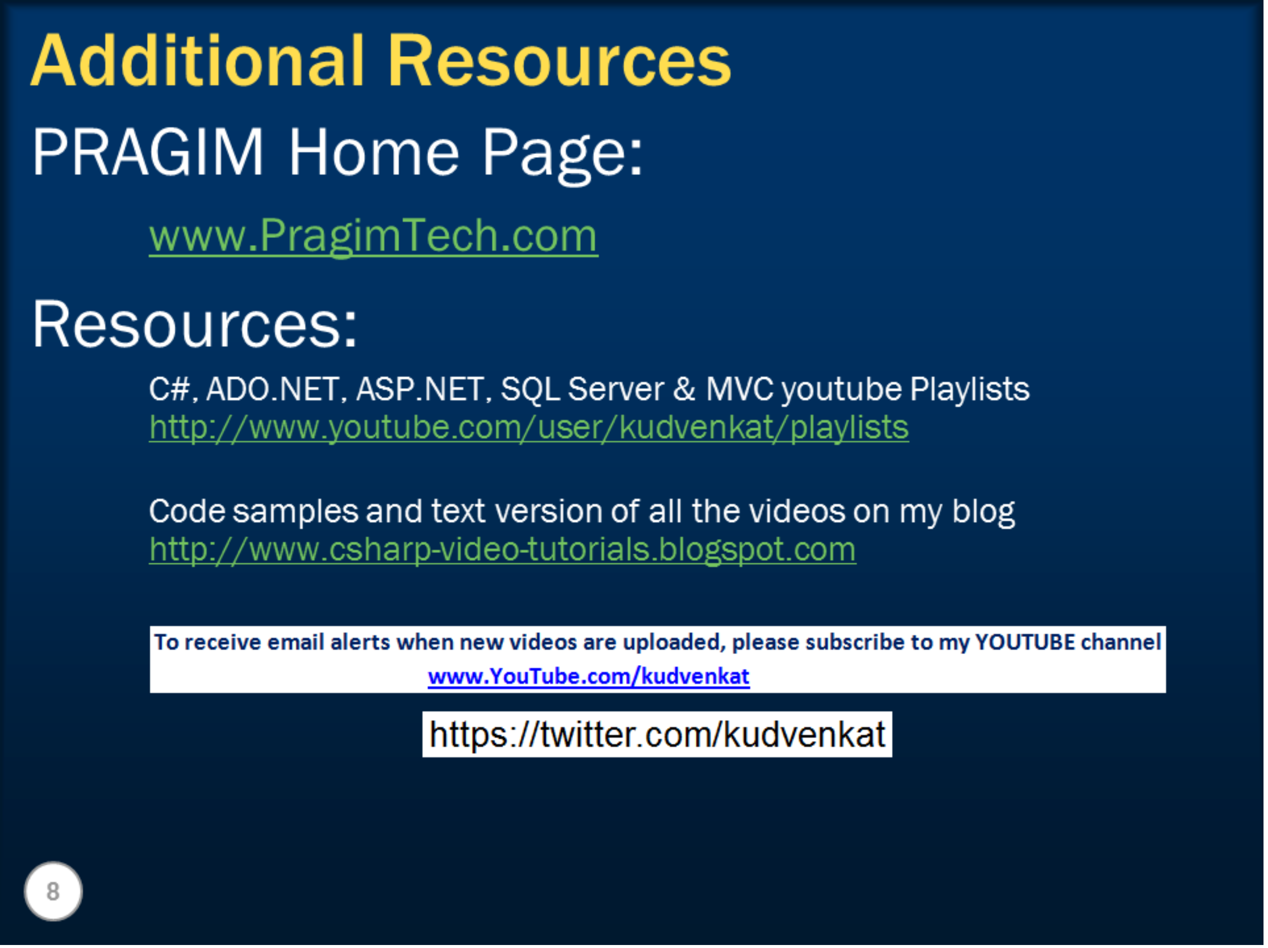
RANK and DENSE_RANK in SQL Server

You can also use RANK and DENSE_RANK functions to find the Nth highest Salary among Male or Female employee groups.

The following query finds the 3rd highest salary amount paid among the Female employees group



```sql
WITH Result AS
(
    SELECT Salary, Gender,
           DENSE_RANK() OVER (PARTITION BY Gender ORDER BY Salary DESC) AS Salary_Rank
    FROM Employees
)
SELECT TOP 1 Salary FROM Result WHERE Salary_Rank = 3 AND Gender = 'Female'
```



Additional Resources

PRAGIM Home Page:

www.PragimTech.com

Resources:

C#, ADO.NET, ASP.NET, SQL Server & MVC youtube Playlists
http://www.youtube.com/user/kudvenkat/playlists

Code samples and text version of all the videos on my blog
http://www.csharp-video-tutorials.blogspot.com

To receive email alerts when new videos are uploaded, please subscribe to my YOUTUBE channel
www.YouTube.com/kudvenkat

https://twitter.com/kudvenkat