**TOPICS IN SQL**

[**https://csharp-video-tutorials.blogspot.com/p/free-sql-server-video-tutorials-for.html**](https://csharp-video-tutorials.blogspot.com/p/free-sql-server-video-tutorials-for.html)

**constraints:** [**https://www.youtube.com/watch?v=cKNQRAMoQHw**](https://www.youtube.com/watch?v=cKNQRAMoQHw)

|  |
| --- |
|  |
| 1. [Connecting to SQL Server](http://csharp-video-tutorials.blogspot.com/2012/08/connecting-to-sql-server-using-ssms.html) Done |
| 1. [Working with databases](http://csharp-video-tutorials.blogspot.co.uk/2012/08/creating-altering-and-dropping-database.html) Done |
| 1. [Creating and working with tables](http://csharp-video-tutorials.blogspot.co.uk/2012/08/creating-and-working-with-tables-part-3.html) Done |
| 1. [Default Constraint](http://csharp-video-tutorials.blogspot.com/2012/08/default-constraint-in-sql-server-part-4.html) Done |
| 1. [Cascading referential integrity](http://csharp-video-tutorials.blogspot.com/2012/08/cascading-referential-integrity.html) Done |
| 1. [Check Constraint](http://csharp-video-tutorials.blogspot.co.uk/2012/08/check-constraint-in-sql-server-part-6.html) Done |
| 1. [Identity Column](http://csharp-video-tutorials.blogspot.com/2012/08/identity-column-in-sql-server-part-7.html) |
| 1. [Retrieving identity column value](http://csharp-video-tutorials.blogspot.com/2012/08/how-to-get-last-generated-identity.html) |
| 1. [Unique key constraint](http://csharp-video-tutorials.blogspot.com/2012/08/unique-key-constraint-part-9.html) Done |
| 1. [All about select statement](http://csharp-video-tutorials.blogspot.com/2012/08/select-statement-part-10.html) |
| 1. [Group By](http://csharp-video-tutorials.blogspot.com/2012/08/group-by-part-11.html) |
| 1. [Basic Joins](http://csharp-video-tutorials.blogspot.com/2012/08/joins-in-sql-server-part-12.html) |
| 1. [Advanced Joins](http://csharp-video-tutorials.blogspot.com/2012/08/advanced-joins-part-13.html) |
| 1. [Self join](http://csharp-video-tutorials.blogspot.com/2012/08/self-join-in-sql-server-part-14.html) |
| 1. [Different ways to replace NULL](http://csharp-video-tutorials.blogspot.com/2012/08/different-ways-to-replace-null-in-sql.html) |
| 1. [Coalesce function](http://csharp-video-tutorials.blogspot.com/2012/08/coalesce-function-in-sql-server-part-16.html) |
| 1. [Union and Union All](http://csharp-video-tutorials.blogspot.com/2012/08/union-and-union-all-in-sql-server-part.html) |
| 1. [Stored Procedures](http://csharp-video-tutorials.blogspot.com/2012/08/stored-procedures-part-18.html) |
| 1. [Stored Procedures with output parameters](http://csharp-video-tutorials.blogspot.com/2012/08/stored-procedures-with-output.html) |
| 1. [Difference between output parameters and return values](http://csharp-video-tutorials.blogspot.com/2012/08/stored-procedure-output-parameters-or.html) |
| 1. [Advantages of stored procedures](http://csharp-video-tutorials.blogspot.com/2012/08/advantages-of-using-stored-procedures.html) |
| 1. [Built in string functions](http://csharp-video-tutorials.blogspot.com/2012/08/built-in-string-functions-in-sql-server.html) |
| 1. [Left, Reight, Charindex and Substring functions](http://csharp-video-tutorials.blogspot.co.uk/2012/08/left-right-charindex-and-substring.html) |
| 1. [Replicate, Space, Patindex, Replace and Stuff functions](http://csharp-video-tutorials.blogspot.com/2012/08/replicate-space-patindex-replace-and.html) |
| 1. [DateTime functions](http://csharp-video-tutorials.blogspot.co.uk/2012/09/datetime-functions-in-sql-server-part-25.html) |
| 1. [IsDate, Day, Month, Year and DateName functions](http://csharp-video-tutorials.blogspot.com/2012/09/isdate-day-month-year-and-datename.html) |
| 1. [DatePart, DateAdd and DateDiff functions](http://csharp-video-tutorials.blogspot.com/2012/09/datepart-dateadd-and-datediff-functions.html) |
| 1. [Convert and Cast functions](http://csharp-video-tutorials.blogspot.com/2012/09/cast-and-convert-functions-in-sql.html) |
| 1. [Mathematical functions](http://csharp-video-tutorials.blogspot.com/2012/09/mathematical-functions-in-sql-server.html) |
| 1. [Scalar User Defined functions](http://csharp-video-tutorials.blogspot.com/2012/09/scalar-user-defined-functions-in-sql.html) |
| 1. [Inline Table Valued functions](http://csharp-video-tutorials.blogspot.com/2012/09/inline-table-valued-functions-part-31.html) |
| 1. [Multi-Statement Table Valued functions](http://csharp-video-tutorials.blogspot.com/2012/09/multi-statement-table-valued-functions.html) |
| 1. [Important concepts related to functions in sql server](http://csharp-video-tutorials.blogspot.com/2012/09/important-concepts-related-to-functions.html) |

|  |  |
| --- | --- |
| |  | | --- | | 1. [Temporary tables](http://csharp-video-tutorials.blogspot.com/2012/09/temporary-tables-in-sql-server-part-34.html) |  1. [**Indexes**](http://csharp-video-tutorials.blogspot.com/2012/09/indexes-in-sql-server-part-35.html) |
| 1. [**Clustered and NonClustered Indexes**](http://csharp-video-tutorials.blogspot.co.uk/2012/09/clustered-and-non-clustered-indexes.html) |
| 1. [**Unique and Non-Unique Indexes**](http://csharp-video-tutorials.blogspot.com/2012/09/unique-and-non-unique-indexes-part-37.html) |
| 1. [**Advantages and disadvantages of indexes**](http://csharp-video-tutorials.blogspot.com/2012/09/advantages-and-disadvantages-of-indexes.html) |
| 1. [**Views**](http://csharp-video-tutorials.blogspot.com/2012/09/views-in-sql-server-part-39.html) |
| 1. [**Updateable Views**](http://csharp-video-tutorials.blogspot.com/2012/09/updateable-views-part-40.html) |
| 1. [**Indexed Views**](http://csharp-video-tutorials.blogspot.com/2012/09/indexed-views-in-sql-server-part-41.html) |
| 1. [**View Limitations**](http://csharp-video-tutorials.blogspot.com/2012/09/limitations-of-views-part-42.html) |
| 1. [**DML Triggers**](http://csharp-video-tutorials.blogspot.com/2012/09/dml-triggers-part-43.html) |
| 1. [**DML After Update Trigger**](http://csharp-video-tutorials.blogspot.com/2012/09/after-update-trigger-part-44.html) |
| 1. [**Instead of insert trigger**](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-insert-trigger-part-45.html) |
| 1. [**Instead of update trigger**](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-update-triggers-part-46.html) |
| 1. [**Instead of delete trigger**](http://csharp-video-tutorials.blogspot.com/2012/09/instead-of-delete-trigger-part-47.html) |
| 1. [**Derived table and CTE's**](http://csharp-video-tutorials.blogspot.com/2012/09/derived-table-and-cte-in-sql-server.html) |
| 1. [**Common Table Expressions**](http://csharp-video-tutorials.blogspot.com/2012/09/common-table-expressions-part-49.html) |
| 1. [**Updatable CTE**](http://csharp-video-tutorials.blogspot.com/2012/09/updatable-cte-part-50.html) |
| 1. [**Recursive CTE**](http://csharp-video-tutorials.blogspot.com/2012/09/recursive-cte-part-51.html) |
| 1. [**Normalization & First Normal Form**](http://csharp-video-tutorials.blogspot.com/2012/09/database-normalization-part-52.html) |
| 1. [**2NF and 3NF**](http://csharp-video-tutorials.blogspot.com/2012/09/second-normal-form-and-third-normal.html) |
| 1. [**Pivot Operator**](http://csharp-video-tutorials.blogspot.com/2012/10/pivot-operator-in-sql-server-part-54.html) |
| 1. [**Error handling in SQL Server 2000**](http://csharp-video-tutorials.blogspot.com/2012/10/error-handling-in-sql-server-2000-part.html) |
| 1. [**Error handling in SQL Server 2005 and later versions**](http://csharp-video-tutorials.blogspot.com/2012/10/error-handling-in-sql-server-2005-and_6.html) |
| 1. [**Transactions in SQL Server**](http://csharp-video-tutorials.blogspot.com/2012/10/transactions-in-sql-server-part-57.html) |
| 1. [**Transaction ACID Tests**](http://csharp-video-tutorials.blogspot.com/2012/10/transaction-acid-tests-part-58.html) |
| 1. [**Subqueries in sql**](http://csharp-video-tutorials.blogspot.com/2013/01/subqueries-in-sql-part-59.html) |
| 1. [**Correlated subquery**](http://csharp-video-tutorials.blogspot.com/2013/01/correlated-subquery-in-sql-part-60.html) |
| 1. [**Creating a large table with random data for performance testing**](http://csharp-video-tutorials.blogspot.com/2013/01/creating-large-table-with-random-data.html) |
| 1. [**What to choose for performance - SubQuery or Joins**](http://csharp-video-tutorials.blogspot.com/2013/01/what-to-choose-for-performance.html) |
| 1. [**Cursors in sql server**](http://csharp-video-tutorials.blogspot.com/2013/01/cursors-in-sql-server-part-63.html) |
| 1. [**Replacing cursors using joins**](http://csharp-video-tutorials.blogspot.com/2013/01/replacing-cursors-using-joins-in-sql.html) |
| 1. [**List all tables in a sql server database using a query**](http://csharp-video-tutorials.blogspot.com/2013/06/part-65-list-all-tables-in-sql-server.html) |
| 1. [**Writing re-runnable sql server scripts**](http://csharp-video-tutorials.blogspot.com/2013/06/writing-re-runnable-sql-server-scripts.html) |
| 1. [**Alter database table columns without dropping table**](http://csharp-video-tutorials.blogspot.com/2013/08/part-67-alter-database-table-columns.html) |
| 1. [**Optional parameters in sql server stored procedures**](http://csharp-video-tutorials.blogspot.com/2013/09/part-68-optional-parameters-in-sql.html) |
| 1. [**Merge in SQL Server**](http://csharp-video-tutorials.blogspot.com/2014/09/part-69-merge-in-sql-server.html) |
| 1. [**SQL Server concurrent transactions**](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-concurrent-transactions.html) |
| 1. [**SQL Server dirty read example**](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-dirty-read-example.html) |
| 1. [**SQL Server lost update problem**](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-lost-update-problem.html) |
| 1. [**Non repeatable read example in sql server**](http://csharp-video-tutorials.blogspot.com/2015/08/non-repeatable-read-example-in-sql.html) |
| 1. [**Phantom reads example in sql server**](http://csharp-video-tutorials.blogspot.com/2015/08/phantom-reads-example-in-sql-server.html) |
| 1. [**Snapshot isolation level in sql server**](http://csharp-video-tutorials.blogspot.com/2015/08/snapshot-isolation-level-in-sql-server.html) |
| 1. [**Read committed snapshot isolation level in sql server**](http://csharp-video-tutorials.blogspot.com/2015/08/read-committed-snapshot-isolation-level.html) |
| 1. [**Difference between snapshot isolation and read committed snapshot**](http://csharp-video-tutorials.blogspot.com/2015/08/difference-between-snapshot-isolation.html) |
| 1. [**SQL Server deadlock example**](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-deadlock-example.html) |
| 1. [**SQL Server deadlock victim selection**](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-deadlock-victim-selection.html) |
| 1. [**Logging deadlocks in sql server**](http://csharp-video-tutorials.blogspot.com/2015/08/logging-deadlocks-in-sql-server.html) |
| 1. [**SQL Server deadlock analysis and prevention**](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-deadlock-analysis-and.html) |
| 1. [**Capturing deadlocks in sql profiler**](http://csharp-video-tutorials.blogspot.com/2015/08/capturing-deadlocks-in-sql-profiler.html) |
| 1. [**SQL Server deadlock error handling**](http://csharp-video-tutorials.blogspot.com/2015/08/sql-server-deadlock-error-handling.html) |
| 1. [**Handling deadlocks in ado.net**](http://csharp-video-tutorials.blogspot.com/2015/08/handling-deadlocks-in-adonet.html) |
| 1. [**Retry logic for deadlock exceptions**](http://csharp-video-tutorials.blogspot.com/2015/08/retry-logic-for-deadlock-exceptions.html) |
| 1. [**How to find blocking queries in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/how-to-find-blocking-queries-in-sql.html) |
| 1. [**SQL Server except operator**](http://csharp-video-tutorials.blogspot.com/2015/09/sql-server-except-operator.html) |
| 1. [**Difference between except and not in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/difference-between-except-and-not-in.html) |
| 1. [**Intersect operator in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/intersect-operator-in-sql-server.html) |
| 1. [**Difference between union intersect and except in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/difference-between-union-intersect-and.html) |
| 1. [**Cross apply and outer apply in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/cross-apply-and-outer-apply-in-sql.html) |
| 1. [**DDL Triggers in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/ddl-triggers-in-sql-server.html) |
| 1. [**Server-scoped ddl triggers**](http://csharp-video-tutorials.blogspot.com/2015/09/server-scoped-ddl-triggers.html) |
| 1. [**SQL Server trigger execution order**](http://csharp-video-tutorials.blogspot.com/2015/09/sql-server-trigger-execution-order.html) |
| 1. [**Audit table changes in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/audit-table-changes-in-sql-server.html) |
| 1. [**Logon triggers in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/logon-triggers-in-sql-server.html) |
| 1. [**Select into in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/select-into-in-sql-server.html) |
| 1. [**Difference between where and having in sql server**](http://csharp-video-tutorials.blogspot.com/2015/09/difference-between-where-and-having-in.html) |
| 1. [**Table valued parameters in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/table-valued-parameters-in-sql-server.html) |
| 1. [**Send datatable as parameter to stored procedure**](http://csharp-video-tutorials.blogspot.com/2015/09/send-datatable-as-parameter-to-stored.html) |
| 1. [**Grouping Sets in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/grouping-sets-in-sql-server.html) |
| 1. [**Rollup in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/rollup-in-sql-server.html) |
| 1. [**Cube in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/cube-in-sql-server.html) |
| 1. [**Difference between cube and rollup in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/difference-between-cube-and-rollup-in.html) |
| 1. [**Grouping function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/grouping-function-in-sql-server.html) |
| 1. [**GROUPING\_ID function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/groupingid-function-in-sql-server.html) |
| 1. [**Debugging sql server stored procedures**](http://csharp-video-tutorials.blogspot.com/2015/09/debugging-sql-server-stored-procedures.html) |
| 1. [**Over clause in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/over-clause-in-sql-server.html) **DONE** |
| 1. [**Row\_Number function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/09/rownumber-function-in-sql-server.html) **DONE** |
| 1. [**Rank and Dense\_Rank in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/rank-and-denserank-in-sql-server.html) **DONE** |
| 1. [**Difference between rank dense\_rank and row\_number in SQL**](http://csharp-video-tutorials.blogspot.com/2015/10/difference-between-rank-denserank-and.html) **DONE** |
| 1. [**Calculate running total in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/calculate-running-total-in-sql-server.html) |
| 1. [**NTILE function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/ntile-function-in-sql-server.html) |
| 1. [**Lead and Lag functions in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/lead-and-lag-functions-in-sql-server.html) **DONE** |
| 1. [**FIRST\_VALUE function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/firstvalue-function-in-sql-server.html) **DONE** |
| 1. [**Window functions in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/window-functions-in-sql-server.html) **DONE** |
| 1. [**Difference between rows and range**](http://csharp-video-tutorials.blogspot.com/2015/10/difference-between-rows-and-range.html) **DONE** |
| 1. [**LAST\_VALUE function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/lastvalue-function-in-sql-server.html) **DONE** |
| 1. [**UNPIVOT in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/unpivot-in-sql-server.html) **DONE** |
| 1. [**Reverse PIVOT table in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/reverse-pivot-table-in-sql-server.html) **DONE** |
| 1. [**Choose function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/choose-function-in-sql-server.html) **DONE** |
| 1. [**IIF function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/iif-function-in-sql-server.html)  **DONE** |
| 1. [**TRY\_PARSE function in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/tryparse-function-in-sql-server-2012.html)DONE |
| 1. [**TRY\_CONVERT function in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/tryconvert-function-in-sql-server-2012.html) DONE |
| 1. [**EOMONTH function in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/eomonth-function-in-sql-server-2012.html) DONE |
| 1. [**DATEFROMPARTS function**](http://csharp-video-tutorials.blogspot.com/2015/10/datefromparts-function-in-sql-server.html)  **DONE** |
| 1. [**Difference between DateTime and SmallDateTime in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/difference-between-datetime-and.html) |
| 1. [**DateTime2FromParts function in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/datetime2fromparts-function-in-sql.html) **DONE** |
| 1. [**Difference between DateTime and DateTime2 in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/difference-between-datetime-and_21.html) |
| 1. [**Offset fetch next in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/offset-fetch-next-in-sql-server-2012.html) **DONE** |
| 1. [**Identifying object dependencies in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/identifying-object-dependencies-in-sql.html) **DONE** |
| 1. [**sys.dm\_sql\_referencing\_entities in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/sysdmsqlreferencingentities-in-sql.html) |
| 1. [**sp\_depends in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/spdepends-in-sql-server.html) |
| 1. [**Sequence object in SQL Server 2012**](http://csharp-video-tutorials.blogspot.com/2015/10/sequence-object-in-sql-server-2012.html) |
| 1. [**Difference between sequence and identity in SQL Server**](http://csharp-video-tutorials.blogspot.com/2015/10/difference-between-sequence-and.html) |
| 1. [**Guid in SQL Server**](http://csharp-video-tutorials.blogspot.com/2017/03/guid-in-sql-server.html) |
| 1. [**How to check GUID is null or empty in SQL Server**](http://csharp-video-tutorials.blogspot.com/2017/03/how-to-check-guid-is-null-or-empty-in.html) |
| 1. [**Dynamic SQL in SQL Server**](http://csharp-video-tutorials.blogspot.com/2017/03/dynamic-sql-in-sql-server.html) |
| 1. [**Implement search web page using ASP.NET and Stored Procedure**](http://csharp-video-tutorials.blogspot.com/2017/03/implement-search-web-page-using-aspnet.html) |
| 1. [**Implement search web page using ASP.NET and Dynamic SQL**](http://csharp-video-tutorials.blogspot.com/2017/04/implement-search-web-page-using-aspnet.html) |
| 1. [**Prevent sql injection with dynamic sql**](http://csharp-video-tutorials.blogspot.com/2017/04/prevent-sql-injection-with-dynamic-sql.html) |
| 1. [**Dynamic SQL in Stored Procedure**](http://csharp-video-tutorials.blogspot.com/2017/04/dynamic-sql-in-stored-procedure.html) |
| 1. [**Sql server query plan cache**](http://csharp-video-tutorials.blogspot.com/2017/04/sql-server-query-plan-cache.html) |
| 1. [**exec vs sp\_executesql in sql server**](http://csharp-video-tutorials.blogspot.com/2017/04/exec-vs-spexecutesql-in-sql-server.html) |
| 1. [**Dynamic sql table name variable**](http://csharp-video-tutorials.blogspot.com/2017/04/dynamic-sql-table-name-variable.html) |
| 1. [**Quotename function in SQL Server**](http://csharp-video-tutorials.blogspot.com/2017/04/quotename-function-in-sql-server.html) |
| 1. [**Dynamic SQL vs Stored Procedure**](http://csharp-video-tutorials.blogspot.com/2017/05/dynamic-sql-vs-stored-procedure.html) |
| 1. [**Dynamic sql output parameter**](http://csharp-video-tutorials.blogspot.com/2017/05/dynamic-sql-output-parameter.html) |
| 1. [**Temp tables in dynamic sql**](http://csharp-video-tutorials.blogspot.com/2017/05/temp-tables-in-dynamic-sql.html) |

In SQL Server we have different categories of **window functions**

**Aggregate functions -** AVG, SUM, COUNT, MIN, MAX etc..

**Ranking functions -** RANK, DENSE\_RANK, ROW\_NUMBER etc..

**Analytic functions -** LEAD, LAG, FIRST\_VALUE, LAST\_VALUE etc...

**OVER** Clause defines the partitioning and ordering of a rows (i.e a window) for the above functions to operate on. Hence these functions are called window functions. The OVER clause accepts the following three arguments to define a window for these functions to operate on.

**ORDER BY :** Defines the logical order of the rows

**PARTITION BY :** Divides the query result set into partitions. The window function is applied to each partition separately.

**ROWSor RANGE clause :** Further limits the rows within the partition by specifying start and end points within the partition.

The default for **ROWS**or **RANGE**clause is

RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

**Rank and Dense\_Rank functions**

Introduced in SQL Server 2005

Returns 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

**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. Let me explain with an example.  
  
**If there are 2 employees with the FIRST highest salary, there are 2 different business cases**

If your business case is, not to produce any result for the SECOND highest salary, then use RANK function

If your business case is to return the next Salary after the tied rows as the SECOND highest Salary, then use DENSE\_RANK function

Since we have 2 Employees with the FIRST highest salary.

Rank() function will not return any rows for the SECOND highest Salary..

**Similarities between RANK, DENSE\_RANK and ROW\_NUMBER functions**

Returns an increasing integer value starting at 1 based on the ordering of rows imposed by the ORDER BY clause (if there are no ties)

ORDER BY clause is required

PARTITION BY clause is optional

When the data is partitioned, the integer value is reset to 1 when the partition changes

So all the 3 functions RANK, DENSE\_RANK and ROW\_NUMBER produce the same increasing integer value when ordered by Salary column.

You will only see the difference when there ties (duplicate values in the column used in the ORDER BY clause).

**Difference between RANK, DENSE\_RANK and ROW\_NUMBER functions**

**ROW NUMBER** : Returns an increasing unique number for each row starting at 1, even if there are duplicates.

**RANK** : Returns an increasing unique number for each row starting at 1. When there are duplicates, same rank is assigned to all the duplicate rows, but the next row after the duplicate rows will have the rank it would have been assigned if there had been no duplicates. So RANK function skips rankings if there are duplicates.

**DENSE\_RANK** : Returns an increasing unique number for each row starting at 1. When there are duplicates, same rank is assigned to all the duplicate rows but the DENSE\_RANK function will not skip any ranks. This means the next row after the duplicate rows will have the next rank in the sequence.

**FIRST\_VALUE function**

* Introduced in SQL Server 2012
* Retrieves the first value from the specified column
* ORDER BY clause is required
* PARTITION BY clause is optional

Syntax : FIRST\_VALUE(Column\_Name) OVER (ORDER BY Col1, Col2, ...)

FIRST\_VALUE function example WITHOUT partitions :

In the following example, FIRST\_VALUE function returns the name of the lowest paid employee from the entire table.

**FIRST\_VALUE function example WITH partitions** : In the following example, FIRST\_VALUE function returns the name of the lowest paid employee from the respective partition.

**LAST\_VALUE function not working as expected** : In the following example, LAST\_VALUE function does not return the name of the highest paid employee. This is because we have not specified an explicit value for ROWS or RANGE clause. As a result it is using it's default value RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

**LAST\_VALUE function working as expected :**In the following example, LAST\_VALUE function returns the name of the highest paid employee as expected. Notice we have set an explicit value for ROWS or RANGE clause to ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING.