Introduction to SQL Server NTILE() function

The SQL Server NTILE() is a [window function](https://www.sqlservertutorial.net/sql-server-window-functions/) that distributes rows of an ordered partition into a specified number of approximately equal groups, or buckets. It assigns each group a bucket number starting from one. For each row in a group, the NTILE() function assigns a bucket number representing the group to which the row belongs.

The syntax of the NTILE() function is as follows:

NTILE(buckets) OVER (

[PARTITION BY partition\_expression, ... ]

ORDER BY sort\_expression [ASC | DESC], ...

)

Code language: SQL (Structured Query Language) (sql)

Let’s examine the syntax in detail:

buckets

The number of buckets into which the rows are divided. The buckets can be an expression or [subquery](https://www.sqlservertutorial.net/sql-server-basics/sql-server-subquery/) that evaluates to a positive integer. It cannot be a window function.

PARTITION BY **clause**

The PARTITION BY clause distributes rows of a result set into partitions to which the NTILE() function is applied.

ORDER BY clause

The ORDER BY clause specifies the logical order of rows in each partition to which the NTILE() is applied.

If the number of rows is not divisible by the buckets, the NTILE() function returns groups of two sizes with the difference by one. The larger groups always come before the smaller group in the order specified by the ORDER BY in the OVER() clause.

On the other hand, if the total of rows is divisible by the buckets, the function divides evenly the rows among buckets.

SQL Server NTILE() function illustration

The following statement [creates a new table](https://www.sqlservertutorial.net/sql-server-basics/sql-server-create-table/) named ntile\_demo that stores 10 integers:

CREATE TABLE sales.ntile\_demo (

v INT NOT NULL

);

INSERT INTO sales.ntile\_demo(v)

VALUES(1),(2),(3),(4),(5),(6),(7),(8),(9),(10);

SELECT \* FROM sales.ntile\_demo;

Code language: SQL (Structured Query Language) (sql)

This statement uses the NTILE() function to divide ten rows into three groups:

SELECT

v,

NTILE (3) OVER (

ORDER BY v

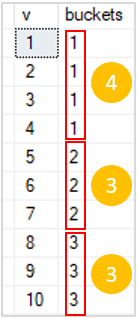
) buckets

FROM

sales.ntile\_demo;

Code language: SQL (Structured Query Language) (sql)

Here is the output:



As clearly shown in the output, the first group has four rows and the other two groups have three rows.

The following statement uses the NTILE() function to distribute rows into five buckets:

SELECT

v,

NTILE (5) OVER (

ORDER BY v

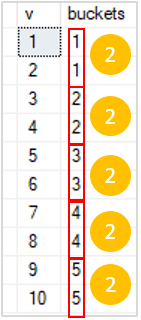
) buckets

FROM

sales.ntile\_demo;

Code language: SQL (Structured Query Language) (sql)

The output is as follows:



As you can see, the output has five groups with the same number of rows in each.

SQL Server NTILE() function examples

Let’s create a view to demonstrate the NTILE() function.

The following statement [creates a view](https://www.sqlservertutorial.net/sql-server-views/sql-server-create-view/) that returns the net sales in 2017 by months.

CREATE VIEW sales.vw\_netsales\_2017 AS

SELECT

c.category\_name,

DATENAME(month, o.shipped\_date) month,

CONVERT(DEC(10, 0), SUM(i.list\_price \* quantity \* (1 - discount))) net\_sales

FROM

sales.orders o

INNER JOIN sales.order\_items i ON i.order\_id = o.order\_id

INNER JOIN production.products p on p.product\_id = i.product\_id

INNER JOIN production.categories c on c.category\_id = p.category\_id

WHERE

YEAR(shipped\_date) = 2017

GROUP BY

c.category\_name,

DATENAME(month, o.shipped\_date);

Code language: SQL (Structured Query Language) (sql)

SELECT

category\_name,

month,

net\_sales

FROM

sales.vw\_netsales\_2017

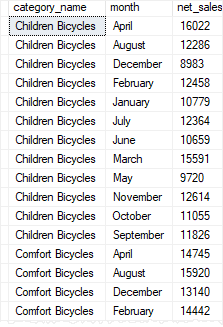
ORDER BY

category\_name,

net\_sales;

Code language: CSS (css)

Here is the result:



Using SQL Server NTILE() function over a query result set example

The following example uses the NTILE() function to distribute the months to 4 buckets based on net sales:

WITH cte\_by\_month AS(

SELECT

month,

SUM(net\_sales) net\_sales

FROM

sales.vw\_netsales\_2017

GROUP BY

month

)

SELECT

month,

FORMAT(net\_sales,'C','en-US') net\_sales,

NTILE(4) OVER(

ORDER BY net\_sales DESC

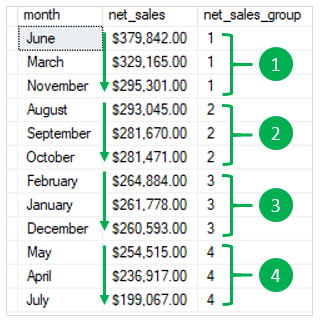
) net\_sales\_group

FROM

cte\_by\_month;

Code language: SQL (Structured Query Language) (sql)

Here is the output:



Using SQL Server NTILE() function over partitions example

This example uses the NTILE() function to divide the net sales by month into 4 groups for each product category:

SELECT

category\_name,

month,

FORMAT(net\_sales,'C','en-US') net\_sales,

NTILE(4) OVER(

PARTITION BY category\_name

ORDER BY net\_sales DESC

) net\_sales\_group

FROM

sales.vw\_netsales\_2017;

Code language: SQL (Structured Query Language) (sql)

The following picture shows the partial output:

