SQL Server LAST\_VALUE() function overview

The LAST\_VALUE() function is a [window function](https://www.sqlservertutorial.net/sql-server-window-functions/) that returns the last value in an ordered partition of a result set.

The following shows the syntax of the LAST\_VALUE() function:

LAST\_VALUE ( scalar\_expression )

OVER (

[PARTITION BY partition\_expression, ... ]

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

)

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

In this syntax:

scalar\_expression

scalar\_expression is an expression evaluated against the value of the last row in an ordered partition of the result set. The scalar\_expression can be a column, [subquery](https://www.sqlservertutorial.net/sql-server-basics/sql-server-subquery/), or expression evaluates to a single value. It cannot be a window function.

PARTITION BY clause

The PARTITION BY clause distributes rows of the result set into partitions to which the LAST\_VALUE() function is applied. If you skip the PARTITION BY clause, the LAST\_VALUE() function will treat the whole result set as a single partition.

ORDER BY clause

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

rows\_range\_clause

The rows\_range\_clause further limits the rows within a partition by defining start and end points.

SQL Server LAST\_VALUE() function examples

We will use the sales.vw\_category\_sales\_volume view created in the FIRST\_VALUE() function tutorial to demonstrate how the LAST\_VALUE()function works.

The following query returns data from the view:

SELECT

category\_name,

year,

qty

FROM

sales.vw\_category\_sales\_volume

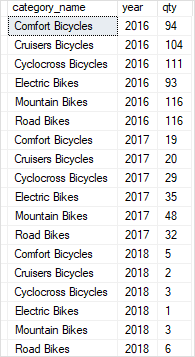
ORDER BY

year,

category\_name,

qty;

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



A) Using LAST\_VALUE() over a result set example

This example uses LAST\_VALUE() function to return category name with the highest sales volume in 2016:

SELECT

category\_name,

year,

qty,

LAST\_VALUE(category\_name) OVER(

ORDER BY qty

RANGE BETWEEN

UNBOUNDED PRECEDING AND

UNBOUNDED FOLLOWING

) highest\_sales\_volume

FROM

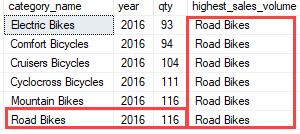
sales.vw\_category\_sales\_volume

WHERE

year = 2016;

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

Here is the output:



In this example:

* The PARTITION BY clause was not specified therefore the whole result set was treated as a single partition.
* The ORDER BY clause sorted rows in each partition by quantity (qty) from low to high.
* The RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING clause defined the frame in the partition starting from the first row and ending at the last row.

B) Using LAST\_VALUE() over partitions example

The following example uses the LAST\_VALUE() function to return product categories with the highest sales volumes in 2016 and 2017.

SELECT

category\_name,

year,

qty,

LAST\_VALUE(category\_name) OVER(

PARTITION BY year

ORDER BY qty

RANGE BETWEEN

UNBOUNDED PRECEDING AND

UNBOUNDED FOLLOWING

) highest\_sales\_volume

FROM

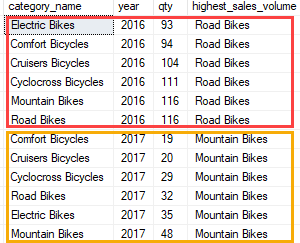
sales.vw\_category\_sales\_volume

WHERE

year IN (2016,2017);

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

The following picture shows the output:



In this example:

* The PARTITION BY clause distributed rows by year into two partitions, one for 2016 and the other for 2017.
* The ORDER BY clause sorted rows in each partition by quantity (qty) from low to high.
* The RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING clause defines the frame starting from the first row and ending at the last row of the partition.
* The LAST\_VALUE() function was applied to each partition separately. For the first partition, it returned Electric Bikes and for the second partition it returned Comfort Bicycles because these categories were the last rows in each partition.