Overview of SQL Server LEAD() function

SQL Server LEAD() is a [window function](https://www.sqlservertutorial.net/sql-server-window-functions/) that provides access to a row at a specified physical offset which follows the current row.

For example, by using the LEAD() function, from the current row, you can access data of the next row, or the row after the next row, and so on.

The LEAD() function can be very useful for comparing the value of the current row with the value of the following row.

The following shows the syntax of the LEAD() function:

LEAD(return\_value ,offset [,default])

OVER (

[PARTITION BY partition\_expression, ... ]

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

)

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

In this syntax:

 return\_value

The return value of the following row based on a specified offset. The return value must evaluate to a single value and cannot be another [window function](https://www.sqlservertutorial.net/sql-server-window-functions/).

 offset

offset is the number of rows forward from the current row from which to access data. The offset can be an expression, [subquery](https://www.sqlservertutorial.net/sql-server-basics/sql-server-subquery/), or column that evaluates to a positive integer.

The default value of offset is 1 if you don’t specify it explicitly.

 default

The function returns default if offset goes beyond the scope of the partition. If not specified, it defaults to NULL.

 PARTITION BY clause

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

If you do not specify the PARTITION BY clause, the function treats the whole result set as a single partition.

 ORDER BY clause

The ORDER BY clause specify logical order of the rows in each partition to which the LEAD() function is applied.

SQL Server LEAD() function examples

Let’s [create a new view](https://www.sqlservertutorial.net/sql-server-views/sql-server-create-view/) named sales.vw\_netsales\_brands for the demonstration:

CREATE VIEW sales.vw\_netsales\_brands

AS

SELECT

c.brand\_name,

MONTH(o.order\_date) month,

YEAR(o.order\_date) year,

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

FROM sales.orders AS o

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

INNER JOIN production.products AS p ON p.product\_id = i.product\_id

INNER JOIN production.brands AS c ON c.brand\_id = p.brand\_id

GROUP BY c.brand\_name,

MONTH(o.order\_date),

YEAR(o.order\_date);

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

The following query returns the data from the sales.vw\_netsales\_brands view:

SELECT

\*

FROM

sales.vw\_netsales\_brands

ORDER BY

year,

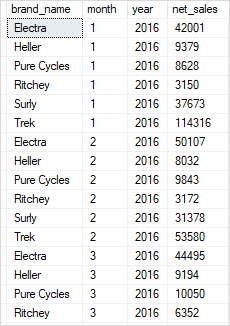
month,

brand\_name,

net\_sales;

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

Here is the output:



A) Using SQL Server LEAD() function over a result set example

The following statement uses the LEAD() function to return the net sales of the current month and the next month in the year 2017:

WITH cte\_netsales\_2017 AS(

SELECT

month,

SUM(net\_sales) net\_sales

FROM

sales.vw\_netsales\_brands

WHERE

year = 2017

GROUP BY

month

)

SELECT

month,

net\_sales,

LEAD(net\_sales,1) OVER (

ORDER BY month

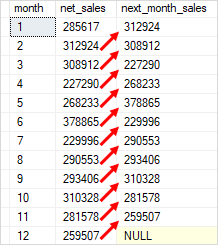
) next\_month\_sales

FROM

cte\_netsales\_2017;

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

Here is the output:



In this example:

* First, the CTE returns net sales aggregated by month.
* Then, the outer query uses the LEAD() function to return the following month sales for each month.

By doing this, you can easily compare the sales of the current month with the next month.

B) Using SQL Server LEAD() function over partitions example

The following statement uses the LEAD() function to compare the sales of the current month with the next month of each brand in the year 2018:

SELECT

month,

brand\_name,

net\_sales,

LEAD(net\_sales,1) OVER (

PARTITION BY brand\_name

ORDER BY month

) next\_month\_sales

FROM

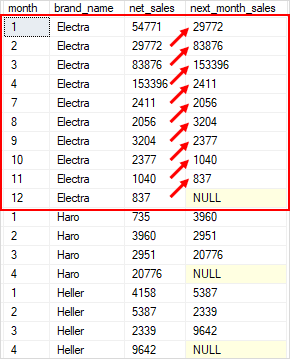
sales.vw\_netsales\_brands

WHERE

year = 2018;

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

This picture shows the output:



In this example:

* The PARTITION BY clause divided rows into partitions by brand name.
* For each partition (or brand name), the ORDER BY clause sorts the rows by month.
* For each row in each partition, the LEAD() function returns the net sales of the following row.