SQL Server PERCENT\_RANK() function overview

The PERCENT\_RANK() function is similar to the [CUME\_DIST()](https://www.sqlservertutorial.net/sql-server-window-functions/sql-server-cume_dist-function/) function. The PERCENT\_RANK() function evaluates the relative standing of a value within a partition of a result set.

The following illustrates the syntax of the SQL Server PERCENT\_RANK() function:

PERCENT\_RANK() OVER (

[PARTITION BY partition\_expression, ... ]

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

)

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

In this syntax:

PARTITION BY

The PARTITION BY clause distributes the rows into multiple partitions to which the PERCENT\_RANK() function is applied. The PARTITION BY clause is optional. If you skip it, the function will treat the whole result set as a single partition.

ORDER BY

The ORDER BY clause specifies the logic order of rows in each partition. Because PERCENT\_RANK() is order sensitive, the order\_by\_clause is required.

Return value

The result of PERCENT\_RANK() is greater than 0 and less than or equal to 1.

0 < PERCENT\_RANK() <= 1

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

The first row has a rank value of zero. Tie values evaluate to the same cumulative distribution value.

The PERCENT\_RANK() function includes NULL values by default and treats them as the lowest possible values.

SQL Server PERCENT\_RANK() examples

Let’s take some examples of using the PERCENT\_RANK() function.

The following statement [creates a new view](https://www.sqlservertutorial.net/sql-server-views/sql-server-create-view/) named sales.vw\_staff\_sales for the demonstration.

CREATE VIEW sales.vw\_staff\_sales(

staff\_id,

year,

net\_sales

) AS

SELECT

staff\_id,

YEAR(order\_date),

ROUND(SUM(quantity\*list\_price\*(1-discount)),0)

FROM

sales.orders o

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

WHERE

staff\_id IS NOT NULL

GROUP BY

staff\_id,

YEAR(order\_date);

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

Using SQL Server PERCENT\_RANK() function over a result set example

This example uses the PERCENT\_RANK() function to calculate the sales percentile of each sales staff in 2016:

SELECT

CONCAT\_WS(' ',first\_name,last\_name) full\_name,

net\_sales,

PERCENT\_RANK() OVER (

ORDER BY net\_sales DESC

) percent\_rank

FROM

sales.vw\_staff\_sales t

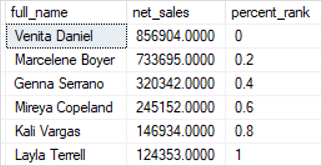
INNER JOIN sales.staffs m on m.staff\_id = t.staff\_id

WHERE

YEAR = 2016;

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

Here is the output:



To make the output more readable, you can use the FORMAT() function to format the percent rank in percentage (%):

SELECT

CONCAT\_WS(' ',first\_name,last\_name) full\_name,

net\_sales,

FORMAT(

PERCENT\_RANK() OVER (

ORDER BY net\_sales DESC

) ,

'P') percent\_rank

FROM

sales.vw\_staff\_sales t

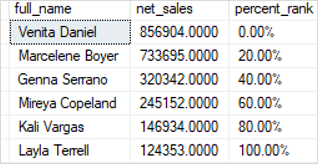
INNER JOIN sales.staffs m on m.staff\_id = t.staff\_id

WHERE

YEAR = 2016;

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

The following picture shows the new output:



Using SQL Server PERCENT\_RANK() function over partitions example

The following example uses the PERCENT\_RANK() to calculate the sales percentile for each staff in 2016 and 2017.

SELECT

year,

CONCAT\_WS(' ',first\_name,last\_name) full\_name,

net\_sales,

FORMAT(

PERCENT\_RANK() OVER (

PARTITION BY year

ORDER BY net\_sales DESC

) ,

'P') percent\_rank

FROM

sales.vw\_staff\_sales t

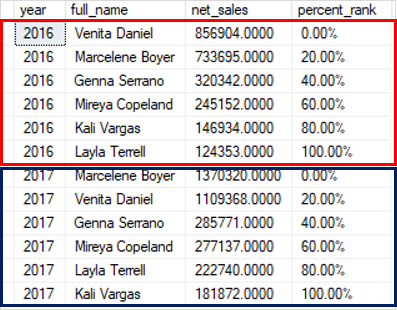
INNER JOIN sales.staffs m on m.staff\_id = t.staff\_id

WHERE

YEAR IN (2016,2017);

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

Here is the output:



In this example:

* The PARTITION BYclause distributed the rows by year into two partitions, one for 2016 and the other for 2017.
* The ORDER BY clause sorted rows in each partition by net sales from high to low.
* The PERCENT\_RANK() function is applied to each partition separately and recomputed the rank when crossing the partition’s boundary.