The **RANK() function** is a [window function](https://www.geeksforgeeks.org/window-functions-in-sql/) could be used in [SQL Server](https://www.geeksforgeeks.org/sql-tutorial/) to calculate a rank for each row within a partition of a result set.

The same rank is assigned to the rows in a partition which have the same values. The rank of the first row is 1. The ranks may not be consecutive in the **RANK() function** as it adds the number of repeated rows to the repeated rank to calculate the rank of the next row.

**Syntax :** 

RANK() OVER (

[PARTITION BY expression, ]

ORDER BY expression (ASC | DESC) );

**Example –**   
Let us create a table geek\_demo that has only column Name :

CREATE TABLE geek\_demo (Name VARCHAR(10) );

Now, insert some rows into the sales.rank\_demo table :   
INSERT INTO geek\_demo (Name)

VALUES('A'), ('B'), ('B'), ('C'), ('C'), ('D'), ('E');

Select data from the geek\_demo table : 

SELECT \*

FROM sales.geek\_demo;

| Name |
| --- |
| A |
| B |
| B |
| C |
| C |
| D |
| E |

Let us use RANK() to assign ranks to the rows in the result set of geek\_demo table :

SELECT Name,

RANK () OVER (

ORDER BY Name

) AS Rank\_no

FROM geek\_demo;

**Output –**

| Name | Rank\_no |
| --- | --- |
| A | 1 |
| B | 2 |
| B | 2 |
| C | 4 |
| C | 4 |
| D | 6 |
| E | 7 |