SQL for Data Science
SQL Window Functions
Learn SQL online at www.DataCamp.com

What are Window Functions?
A window function makes a calculation across multiple rows that are related to the current row. For example, a window function allows you to calculate:
- Running totals (i.e., sum values from all rows before the current row)
- 3-day moving averages (i.e., average values from 3 rows before the current row)
- Ranks

Similar to an aggregate function (GROUP BY), a window function performs the operation across multiple rows. Unlike an aggregate function, a window function does not group rows into one single row.

AGGREGATE FUNCTION | WINDOW FUNCTION
--- | ---
First | CURRENT ROW
Min | CURRENT ROW
Max | CURRENT ROW
Sum | SUM
Count | COUNT
Average | AVERAGE
Variance | VARIANCE
Standard Deviation | STDEV

Syntax
Windows can be defined in the SELECT section of the query.

To make the same window with several window functions, define a named window using the WINDOW keyword. This appears in the query after the HAVING section and before the ORDER BY section.

ORDER BY is subclauses within the ORDER BY clause. ORDER BY changes the order on which the function assigns row numbers to rows.

It is a must-have for window functions that assign sequences to rows, including RANK and ROW_NUMBER, for example, if you ORDER BY the expression prior to an ascending order, then the lowest-priced item will have the lowest rank.

Let's have the following two queries which differ only in the ORDER BY clause.

<table>
<thead>
<tr>
<th>Shop</th>
<th>Name</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

Window frame extent
A window frame is the selected set of rows in the partition over which aggregation will occur. For simplicity, they are a set of rows that are somehow related to the current row.

A window frame is defined by a lower bound and an upper bound relative to the current row. The lowest possible bound is the first row, which is known as UNBOUNDED PRECEDING. The highest possible bound is the current row, which is known as CURRENT ROW.

For example, if we only want to get 5 rows before the current row, we will specify the range using 5 PRECEDING.

Accompanying Material

Ranking window functions
There are several window functions for assigning rankings to rows. Each of these functions requires ORDER BY sub-clause within the OVER clause.

The following are the ranking window functions and their description:

- RANK() (Mandatory)
  - Assigns a sequential integer to each row within the partition of most rows.
  - Row numbers are not repeated within a partition.

- DENSE_RANK() (Mandatory)
  - Assigns the rank number to each row in the order they occur.
  - All ties receive the same rank.

- ROW_NUMBER() (Mandatory)
  - Assigns a sequential integer to each row within the partition of most rows.
  - Row numbers are repeated within a partition.

- NTILE(n) (Optional)
  - Divides the rows of a partition into n equal-sized groups.
  - Takes n, the number of desired groups.

- LAG() / LEAD() (Optional)
  - Similar to the group window function, which has the same syntax.
  - Value is calculated as the value in the partition.

- ROLLUP() / CUBE() (Optional)
  - Specifies the level of grouping at which to aggregate the data.

Value window functions
FIRST() (Mandatory) and LAST() (Mandatory) retrieve the first and last value respectively from an ordered list of rows, where the order is defined by ORDER BY.

LEAD() and LAG() are also available as window functions.

LEAD() and LAG() locate a row relative to the current row.

Aggregate window functions
Window functions available for GROUP BY, such as COUNT(), MIN(), MAX(), SUM(), and AVG() are also available as window functions.

LEAD, LAG, and COUNT are most commonly used to find the value of a previous row or the next row. For example, they are useful for calculating the year-over-year increase of business metrics like revenues.

Here is an example of using window to compare this year's sales to last year's sales.

LEAD and LAG are used to get data in the next or previous row.

LEAD and LAG allow you to specify ORDER BY in the OVER clause.

LEAD and LAG are most commonly used to find the value of a previous row or the next row. For example, they are useful for calculating the year-over-year increase of business metrics like revenues.

Here is an example of using window to compare this year's sales to last year's sales.

LEAD and LAG are used to get data in the next or previous row.

LEAD and LAG allow you to specify ORDER BY in the OVER clause.