Window Transform

Contents:

The `window` transform enables you to perform summations and calculations based on a rolling window of data relative to the current row. For example, you can compute the rolling average for a specified column for the current row value and the three preceding rows. This transform is particularly useful for processing time or otherwise sequential data.

You can apply one or more functions to your `window` transform step.

- For more information on window functions, see *Window Functions*.
- You can also use the aggregation functions with this transform. See *Aggregate Functions*.

> **NOTE:** Be careful applying this transform across a large number of rows. In some cases, the application can run out of memory generating the results, and your results can fail.

Basic Usage

```
window value: ROLLINGAVERAGE(myValues,3) order: MyDate group: customerId
```

Output: Generates a new column called, `window`, which contains the result of the `ROLLINGAVERAGE` function applied from the current row in the `myValues` column across the `3` rows forward, ordered by `MyDate` and grouped by `customerId`.

Syntax and Parameters

```
window value: WINDOW_FUNCTION(arg1,arg2) order: order_col [group: group_col]
```

Token	Required?	Data Type	Description
window	Y	transform	Name of the transform
value	Y	string	Expression that evaluates to the window function call and its parameters
order	Y	string	Column or column names by which to sort the dataset before the `value` expression is applied
group	N	string	Column name or names containing the values by which to group for calculation

For more information on syntax standards, see *Language Documentation Syntax Notes*.

value

For the `window` transform, the `value` parameter contains the function call or calls, which define the set of rows to which the function is applied.

You can specify multiple window functions for the value. Each function reference must be separated by a comma. The transform generates a new column for each window function.

This transform uses a special set of functions. For more information on the available functions, see *Window Functions*.

Usage Notes:

Required?	Data Type
Yes	String (expression)

order

For the `window` transform, this parameter specifies the column on which to sort the dataset before applying the specified function. For combination sort keys, you can add multiple comma-separated columns.

> **NOTE:** The order parameter must unambiguously specify an ordering for the data, or the generated results may vary between job executions.

> **NOTE:** If you are applying a window function, it requires a primary key to identify rows in the output. Otherwise, results can be ambiguous. For more information on defining a primary key, see *Window Functions*.

> **NOTE:** If it is present, the dataset is first grouped by the `group` value before it is ordered by the values in the `order` column.

> **NOTE:** The `order` column does not need to be sorted before the `window` transform is executed on it.

> **Tip:** To sort in reverse order, prepend the column name with a dash (`-MyDate`).

Usage Notes:

Required?	Data Type
Yes	String (column name)

group

For the `window` transform, this parameter specifies the column whose values are used to group the dataset prior to applying the specified function. For combination grouping, you can specify multiple comma-separated column names.

> **NOTE:** Transforms that use the `group` parameter can result in non-deterministic re-ordering in the data grid. However, you should apply the `group` parameter, particularly on larger datasets, or your job may run out of memory and fail. To enforce row ordering, you can use the `sort` transform. For more information, see *Sort Transform*.

Usage Notes:

Required?	Data Type
No	String (column name)

Examples

> **Tip:** For additional examples, see *Common Tasks*.

See the individual functions for examples. See *Window Functions*.