ROLLINGMAXDATE Function

Contents:

Computes the rolling maximum of date values forward or backward of the current row within the specified column. Inputs must be of Datetime type.

- If an input value is missing or null, it is not factored in the computation. For example, for the first row in the dataset, the rolling maximum of previous values is undefined.
- The row from which to extract a value is determined by the order in which the rows are organized based on the `order` parameter.
- If you are working on a randomly generated sample of your dataset, the values that you see for this function might not correspond to the values that are generated on the full dataset during job execution.
- The function takes a column name and two optional integer parameters that determine the window backward and forward of the current row.
  - The default integer parameter values are `-1` and `0`, which computes the rolling function from the current row back to the first row of the dataset.
- This function works with the Window transform. See *Window Transform*.

For more information on a non-rolling version of this function, see *MAXDATE Function*.

Wrangle vs. SQL: This function is part of Wrangle , a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

Basic Usage

Column example:

```
rollingmaxdate(myDate)
```

Output: Returns the rolling maximum of all values in the `myDate` column.

Rows before example:

```
rollingmaxdate(myDate, 3)
```

Output: Returns the rolling maximum of the current row and the three previous row values in the `myDate` column.

Rows before and after example:

```
rollingmaxdate(myDate, 3, 2)
```

Output: Returns the rolling maximum of the three previous row values, the current row value, and the two rows after the current one in the `myDate` column.

Syntax and Arguments

```
rollingmaxdate(col_ref, rowsBefore_integer, rowsAfter_integer) order: order_col [group: group_col]
```

Argument	Required?	Data Type	Description
col_ref	Y	string	Name of column whose values are applied to the function
rowsBefore_integer	N	integer	Number of rows before the current one to include in the computation
rowsAfter_integer	N	integer	Number of rows after the current one to include in the computation

For more information on the `order` and `group` parameters, see *Window Transform*.

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

col_ref

Name of the column whose values are used to compute the function. Inputs must be Datetime values.

Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference to Datetime values)	`transactionDate`

rowsBefore_integer, rowsAfter_integer

Integers representing the number of rows before or after the current one from which to compute the rolling function, including the current row. For example, if the first value is `5`, the current row and the five rows before it are used in the computation. Negative values for `rowsAfter_integer` compute the rolling function from rows preceding the current one.

- `rowBefore=0` generates the current row value only.
- `rowBefore=-1` uses all rows preceding the current one.
- If `rowsAfter` is not specified, then the value `0` is applied.
- If a `group` parameter is applied, then these parameter values should be no more than the maximum number of rows in the groups.

Usage Notes:

Required?	Data Type	Example Value
No	Integer	4

Examples

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

Example - Rolling date functions

This example describes how to use the rolling computational functions:

- `ROLLINGMINDATE` - Computes the rolling minimum of Date values forward or backward of the current row within the specified column. Inputs must be of Datetime type. See *ROLLINGMINDATE Function*.
- `ROLLINGMAXDATE` - Computes the rolling maximum of date values forward or backward of the current row within the specified column. Inputs must be of Datetime type. See *ROLLINGMAXDATE Function*.
- `ROLLINGMODEDATE` - Computes the rolling mode (most common value) forward or backward of the current row within the specified column. Input values must be of Datetime data type. See *ROLLINGMODEDATE Function*.

Source:

The following table contains an unordered list of orders:

myDate	prodId	orderDollars
2020-03-13	p001	1445
2020-03-06	p002	712
2020-03-16	p003	1374
2020-03-23	p001	1675
2020-04-09	p002	1005
2020-08-09	p003	984
2020-05-02	p001	1395
2020-06-14	p002	1866
2020-07-16	p003	824
2020-09-02	p001	1785
2020-08-31	p002	697
2020-10-22	p003	1513
2020-03-17	p001	768
2020-03-21	p002	1893
2020-03-23	p003	1122
2020-04-06	p001	805
2020-05-09	p002	1752
2021-01-09	p003	616
2020-08-18	p001	1563
2020-09-12	p002	730
2020-10-04	p003	587
2021-02-15	p001	1979
2021-02-22	p002	134
2021-03-14	p003	938

Transformation:

You can use the following Window transformation to calculate the rolling minimum, maximum, and mode dates for the last five orders for each product identifier:

Transformation Name	`Window`
Parameter: Formula1	`ROLLINGMINDATE(orderDate, 4, 0)`
Parameter: Formula2	`ROLLINGMAXDATE(orderDate, 4, 0)`
Parameter: Formula3	`ROLLINGMODEDATE(orderDate, 4, 0)`
Parameter: Group by	`prodId`
Parameter: Order by	`prodId`

You can use the following transformation to rename the generated `window` columns:

Transformation Name	`Rename columns`
Parameter: Option	`Manual rename`
Parameter: Column	`window1`
Parameter: New column name	`rollingMinDate`
Parameter: Parameter: Column	`window2`
Parameter: New column name	`rollingMaxDate`
Parameter: Parameter: Column	`window3`
Parameter: New column name	`rollingModeDate`

Results:

orderDate	prodId	orderDollars	rollingMinDate	rollingMaxDate	rollingModeDate
3/16/20	p003	1374	3/16/20	3/16/20	3/16/20
8/9/20	p003	984	3/16/20	8/9/20	3/16/20
7/16/20	p003	824	3/16/20	8/9/20	3/16/20
10/22/20	p003	1513	3/16/20	10/22/20	3/16/20
3/23/20	p003	1122	3/16/20	10/22/20	3/16/20
1/9/21	p003	616	3/23/20	1/9/21	3/23/20
10/4/20	p003	587	3/23/20	1/9/21	3/23/20
3/14/21	p003	938	3/23/20	3/14/21	3/23/20
3/13/20	p001	1445	3/13/20	3/13/20	3/13/20
3/23/20	p001	1675	3/13/20	3/23/20	3/13/20
5/2/20	p001	1395	3/13/20	5/2/20	3/13/20
9/2/20	p001	1785	3/13/20	9/2/20	3/13/20
3/17/20	p001	768	3/13/20	9/2/20	3/13/20
4/6/20	p001	805	3/17/20	9/2/20	3/17/20
8/18/20	p001	1563	3/17/20	9/2/20	3/17/20
2/15/21	p001	1979	3/17/20	2/15/21	3/17/20
3/6/20	p002	712	3/6/20	3/6/20	3/6/20

4/9/20	p002	1005	3/6/20	4/9/20	3/6/20
6/14/20	p002	1866	3/6/20	6/14/20	3/6/20
8/31/20	p002	697	3/6/20	8/31/20	3/6/20
3/21/20	p002	1893	3/6/20	8/31/20	3/6/20
5/9/20	p002	1752	3/21/20	8/31/20	3/21/20
9/12/20	p002	730	3/21/20	9/12/20	3/21/20
2/22/21	p002	134	3/21/20	2/22/21	3/21/20