NEXT Function

Contents:

Extracts the value from a column that is a specified number of rows after the current value.

- The row from which to extract a value is determined by the order in which the rows are organized at the time that the function is executed.
- 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.
- If the next value is missing or null, this function generates a missing value.
- You can use the `group` and `order` parameters to define the groups of records and the order of those records to which this function is applied.
- This function works with the following transforms:
  - *Window Transform*
  - *Set Transform*
  - *Derive Transform*

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

```
next(myNumber, 1) order:Date
```

Output: Returns the value in the row in the `myNumber` column immediately after the current row when the dataset is ordered by `Date`.

Syntax and Arguments

```
next(col_ref, k_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
k_integer	Y	integer (positive)	Number of rows after the current one from which to extract the value

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 extract the value that is `k-integer` values after the current one.

- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference)	`myColumn`

k_integer

Integer representing the number of rows after the current one from which to extract the value.

- Value must be a positive integer. For negative values, see *PREV Function*.
- `k=1` represents the immediately following row value.
- If k is greater than or equal to the number of values in the column, all values in the generated column are missing. If a `group` parameter is applied, then this parameter should be no more than the maximum number of rows in the groups.
- If the range provided to the function exceeds the limits of the dataset, then the function generates a null value.
- If the range of the function is valid but includes missing values, the function generates a missing, non-null value.

Usage Notes:

Required?	Data Type	Example Value
Yes	Integer	4

Examples

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

Example - Examine prior order history

The following dataset contains order information for the preceding 12 months. You want to compare the current month's average against the preceding quarter.

Source:

Date	Amount
12/31/15	118
11/30/15	6
10/31/15	443
9/30/15	785
8/31/15	77
7/31/15	606

6/30/15	421
5/31/15	763
4/30/15	305
3/31/15	824
2/28/15	135
1/31/15	523

Transformation:

Using the `ROLLINGAVERAGE` function, you can generate a column containing the rolling average of the current month and the two previous months:

Transformation Name	`Window`
Parameter: Formulas	`ROLLINGAVERAGE(Amount, 3, 0)`
Parameter: Order by	`-Date`

Note the sign of the second parameter and the `order` parameter. The sort is in the reverse order of the `Date` par ameter, which preserves the current sort order. As a result, the second parameter, which identifies the number of rows to use in the calculation, must be positive to capture the previous months.

Technically, this computation does not capture the prior quarter, since it includes the current quarter as part of the computation. You can use the following column to capture the rolling average of the preceding month, which then becomes the true rolling average for the prior quarter. The `window` column refers to the name of the column generated from the previous step:

Transformation Name	`Window`
Parameter: Formulas	`NEXT(window, 1)`
Parameter: Order by	`-Date`

Note that the order parameter must be preserved. This new column, `window1`, contains your prior quarter rolling average:

Transformation Name	`Rename columns`
Parameter: Option	`Manual rename`
Parameter: Column	`window1`
Parameter: New column name	`'Amount_PriorQtr'`

You can reformat this numeric value:

Transformation Name	`Edit column with formula`
Parameter: Columns	`Amount_PriorQtr`
Parameter: Formula	`NUMFORMAT(Amount_PriorQtr, '###.00')`

You can use the following transformation to calculate the net change. This formula computes the change as a percentage of the prior quarter and then formats it as a two-digit percentage.

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`NUMFORMAT(((Amount - Amount_PriorQtr) / Amount_PriorQtr) * 100, '##.##')`
Parameter: New column name	`'NetChangePct_PriorQtr'`

Results:

> **NOTE:** You might notice that there are computed values for `Amount_PriorQtr` for February and March. These values do not factor in a full three months because the data is not present. The January value does not exist since there is no data preceding it.

Date	Amount	Amount_PriorQtr	NetChangePct_PriorQtr
12/31/15	118	411.33	-71.31
11/30/15	6	435.00	-98.62
10/31/15	443	489.33	-9.47
9/30/15	785	368.00	113.32
8/31/15	77	596.67	-87.1
7/31/15	606	496.33	22.1
6/30/15	421	630.67	-33.25
5/31/15	763	421.33	81.09
4/30/15	305	494.00	-38.26
3/31/15	824	329.00	150.46
2/28/15	135	523.00	-.74.19
1/31/15	523		