

NOW Function

Derives the timestamp for the current time in UTC time zone. You can specify a different time zone by optional parameter.

For this function, the values that you see in the Transformer grid are generated during the preview. These values will differ from the values that are generated later, when the job is executed.

NOTE: Some Datetime functions do not allow the nesting of `NOW` and `TODAY` functions. You should create a separate column containing these values.

Other differences:

- If you refresh the page for the Transformer grid, the function is recalculated.
- If you re-open the dataset in the Transformer page, the function is recalculated.
- If you have multiple versions of the function in the same dataset, you are likely to end up with different generated timestamps. The difference in their values cannot be accurately predicted.

Tip: If you wish to use the same computed value for this function across your dataset, you should generate a column containing values for the function and then base all of your other calculations off of these column values.

NOTE: If no time zone is specified, the default is UTC time zone. Time values are returned in 24-hour time.

For more information on generating the date value only for today, see *TODAY 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

Basic Example:

```
now()
```

Output: Returns the current timestamp in UTC time zone.

Example with Time Zone:

```
now('America/New York')
```

Output: Returns the current timestamp based on the time in the Eastern U.S. time zone.

Syntax and Arguments

```
now(str_timezone)
```

Argument	Required?	Data Type	Description
str_timezone	N	string	String value for the time zone for which the timestamp is calculated.

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

str_timezone

String value for the time zone to use.

Usage Notes:

Required?	Data Type	Example Value
No	String	'America/New York'

Examples

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

Example - Flight Status report

This example illustrates you to generate the date and time values for the current date and timestamp in the specified time zone.

Functions:

Item	Description
NOW Function	Derives the timestamp for the current time in UTC time zone. You can specify a different time zone by optional parameter.
TODAY Function	Derives the value for the current date in UTC time zone. You can specify a different time zone by optional parameter.
DATEDIF Function	Calculates the difference between two valid date values for the specified units of measure.

Source:

The following table includes flight arrival information for Los Angeles International airport.

FlightNumber	Gate	Arrival
1234	1	2/15/17 11:35
212	2	2/15/17 11:58
510	3	2/15/17 11:21
8401	4	2/15/17 12:08
99	5	2/16/17 12:12
116	6	2/16/17 13:32
876	7	2/15/17 16:43
9494	8	2/15/17 21:00
102	9	2/14/17 19:21
77	10	2/16/17 12:31

Transformation:

You are interested in generating a status report on today's flights. To assist, you must generate columns with the current date and time values:

Tip: You should create separate columns containing static values for `NOW` and `TODAY` functions. Avoid creating multiple instances of each function in your dataset, as the values calculated in them can vary at execution time.

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>NOW('America\Los_Angeles')</code>
Parameter: New column name	'currentTime'

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>TODAY('America\Los_Angeles')</code>
Parameter: New column name	'currentDate'

Next, you want to identify the flights that are landing today. In this case, you can use the `DATEDIF` function to determine if the `Arrival` value matches the `currentTime` value within one day:

NOTE: The `DATEDIF` function computes difference based on the difference from the first date to the second date based on the unit of measure. So, a timestamp that is 23 hours difference from the base timestamp can be within the same unit of day, even though the dates may be different (2/15/2017 vs. 2/14/2017).

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>DATEDIF(currentDate, Arrival, day)</code>
Parameter: New column name	'today'

Since you are focusing on today only, you can remove all of the rows that do not apply to today:

Transformation Name	Filter rows
Parameter: Condition	Custom formula
Parameter: Type of formula	Custom single
Parameter: Condition	<code>today <> 0</code>
Parameter: Action	Delete matching rows

Now focusing on today's dates, you can calculate the difference between the current time and the arrival time by the minute:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	DATEDIF(currentTime, Arrival, minute)
Parameter: New column name	'status'

Using the numeric values in the `status` column, you can compose the following transform, which identifies status of each flight:

Transformation Name	Edit column with formula
Parameter: Columns	<code>status</code>
Parameter: Formula	<code>if(status < -20, 'arrived', if(status > 20, 'scheduled', if(status <= 0, 'landed', 'arriving')))</code>

Results:

You now have a daily flight status report:

<code>currentDate</code>	<code>currentTime</code>	<code>FlightNumber</code>	<code>Gate</code>	<code>Arrival</code>	<code>status</code>	<code>today</code>
2017-02-15	2017-02-15 11:46:12	1234	1	2/15/17 11:35	landed	0
2017-02-15	2017-02-15 11:46:12	212	2	2/15/17 11:58	arriving	0
2017-02-15	2017-02-15 11:46:12	510	3	2/15/17 11:21	arrived	0
2017-02-15	2017-02-15 11:46:12	8401	4	2/15/17 12:08	scheduled	0
2017-02-15	2017-02-15 11:46:12	876	7	2/15/17 16:43	scheduled	0
2017-02-15	2017-02-15 11:46:12	9494	8	2/15/17 21:00	scheduled	0
2017-02-15	2017-02-15 11:46:12	102	9	2/14/17 19:21	arrived	0

The `currentDate`, `currentTime`, and `today` columns can be deleted.