Date Functions

Tableau Desktop Reference Guide

Date functions operate on date fields in order to manipulate, compare, or query information about them. If you want to refer to a date as a literal, that is input as is, surround the input date with the pound symbol, (for example: #December 20, 2016#).

Some functions use a start_of_week option which defaults based on the data source value. The syntax for date format parameters follows the International Components for Unicode (ICU) definitions.

Date Parts

Many date functions use date_part, which indicates the specific part of the date you would like to manipulate or query. Note that the date_part argument is case sensitive and must be all in lowercase. Valid date_part values are:

date _part	Values
'year'	Four-digit year
'quarter'	1–4
'month'	1–12 or "January", "February", and so on
'dayofyear'	Day of the year; Jan 1 is 1, Feb 1 is 32, and so on
'day'	1–31
'weekday'	1–7 or "Sunday", "Monday", and so on
'week'	1–52
'hour'	0–23
'minute'	0–59
'second'	0–60

Examples

These are some examples of common date functions:

Function Syntax	Purpose	Example
MONTH(date) **YEAR** and **DAY** are similar **DATEPART**(date_part, date, [start_of_week]) is also similar	Returns a number representing the year (or date_part) of a given date.	**MONTH**(**[Ship Date]**) returns 12 if the **[Ship Date]** is equal to #December 20, 2016#.

DATENAME(date_part, date, **[start_of_week]**)	Similar to **DATEPART**, but returns the date_part of the given date as a string instead of as a number.	**DATENAME**('month', **[Ship Date]**) returns "December" if the **[Ship Date]** is equal to #2016-12-20#.
DATEPARSE(format, string)	Works in the reverse of **DATENAME** by converting a string into a date/time using the given ICU format.	**DATEPARSE**("dd.MMMM.yyyy", **[Delivered On]**) returns #10/28/2015# if **[Delivered On]** is equal to the string "28.October.2015".
DATEDIFF(date_part, start_date, end_date, [start_of_week])	Returns the difference between the two dates using the units of date_ part.	**DATEDIFF**('month', **[Order Date]**, #2016-01-01#) returns 1 if the **[Order Date]** is #February 23, 2016# and -12 if the **[Order Date]** is #January 3, 2015#.
DATEADD(date_part, increment, date)	Returns a date in which the increment has been added to the given date. The type of increment is specified using the date_part	**DATEADD**('month', 6, **[Order Date])** returns #July 1, 2016# if **[Order Date]** is #January 1, 2016#.
DATETRUNC(date_part, date, [start_of_week])	Truncates or "rounds down" the given date to the level of the specified date_part.	**DATETRUNC**('month', **[Order Date]**) returns #February 1, 2005# if **[Order Date]** is #February 17, 2005#. DATETRUNC('quarter', **[Order Date]**) returns #April 1, 2015# if **[Order Date]** is #June 21, 2015#.
MIN(expression) or **MIN**(expr1, expr2) **MAX** is similar	Returns the earliest date across all records, or the earlier of the two dates for each record.	**MIN**(**[Order Date]**) returns $2012-01-03# if the oldest order date is #January 3, 2012#.

Other date functions are available and include:

Function	Purpose
TODAY, **NOW**	Return the current date or current date and time.
ISDATE	Check if a given string is a valid date.
MAKEDATE, **MAKETIME**, **MAKEDATETIME**	Returns a date, time, or datetime value constructed from the arguments given.

Performance Considerations Related to Date Functions

To improve the performance of your worksheet, take these considerations into account when using date functions:

Use the **NOW** function only if you need the time stamp level of detail, otherwise, use **TODAY** for date level calculations.

If possible, use the **DATETRUNC**, **DATEADD**, and **DATEDIFF** functions instead of more complex tests using multiple date functions, such as **YEAR** and **MONTH**.

Use **DATEPARSE** if your data has dates that are stored as strings or numeric time stamps (that is, not stored in native date formats).