

NETWORKDAYS Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *date1, date2*
 - *array_holiday*
- *Examples*
 - *Example - Date diffing functions*

Calculates the number of working days between two specified dates, assuming Monday - Friday workweek. Optional list of holidays can be specified.

- Inputs can be column references or the outputs of the DATE or TIME functions.
 - See *DATE Function*.
 - See *TIME Function*.
- The first value is used as the baseline to compare the date values.
- If the first date value occurs after the second date value, a null value is returned.

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

```
networkdays(StartDate, EndDate)
```

Output: Returns the number of working days between StartDate and EndDate.

Syntax and Arguments

```
networkdays(date1,date2,[array_holiday])
```

Argument	Required?	Data Type	Description
date1	Y	datetime	Starting date to compare
date2	Y	datetime	Ending date to compare
array_holiday	N	array	An array of string values representing the valid dates of holidays.

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

date1, date2

Date values can be column references or output of the DATE function or the TIME function.

- For more information, see *DATE Function*.
- For more information, see *TIME Function*.

Date values to compared in working days.

- If `date2 > date1`, then results are positive.

- If `date2 < date1`, then a null value is returned.

If `date1` and `date2` have a specified time zone offset, the function calculates the difference including the timezone offsets.

- If `date1` does not have a specified time zone but `date2` does, the function uses the local time in the same time zone as `date2` to calculate the difference. The functions returns the difference without the time zone offset.

Usage Notes:

Required?	Data Type	Example Value
Yes	Datetime (Column reference or date output of DATE or TIME function)	LastContactDate

array_holiday

An array containing the list of holidays, which are factored in the calculation of working days.

Values in the array must be in either of the following formats:

```
[ '2020-12-24', '2020-12-25' ]
[ '2020/12/24', '2020/12/25' ]
```

Usage Notes:

Required?	Data Type	Example Value
Yes	Array	['2018-12,24','2018-12-25','2018-12-31','2019-01-01']

Examples

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

Example - Date diffing functions

This example demonstrates how to calculate the number of days between two input dates.

Functions:

Item	Description
DATEDIF Function	Calculates the difference between two valid date values for the specified units of measure.
NETWORKDAYS Function	Calculates the number of working days between two specified dates, assuming Monday - Friday workweek. Optional list of holidays can be specified.
NETWORKDAYS INTL Function	Calculates the number of working days between two specified dates. Optionally, you can specify which days of the week are working days as an input parameter. Optional list of holidays can be specified.
WORKDAY Function	Calculates the work date that is before or after a start date, as specified by a number of days. A set of holiday dates can be optionally specified.

WORKDAYINTL Function	Calculates the work date that is before or after a start date, as specified by a number of days. You can also specify which days of the week are working days and a list of holidays via parameters.
-------------------------	--

Source:

The following dataset contains two columns of dates.

- The first column values are constant. This date falls on a Monday.

Date1	Date2
2020-03-09	2020-03-13
2020-03-09	2020-03-06
2020-03-09	2020-03-16
2020-03-09	2020-03-23
2020-03-09	2020-04-10
2020-03-09	2021-03-10

Transformation:

The first transformation calculates the number of raw days between the two dates:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>datedif(Date1, Date2, day)</code>
Parameter: New column name	'datedif'

This step computes the number of working days between the two dates. Assumptions:

- Workweek is Monday - Friday.
- There are no holidays.

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>networkdays(Date1, Date2, [])</code>
Parameter: New column name	'networkDays'

For some, March 17 is an important date, especially if you are Irish. To add St. Patrick's Day to the list of holidays, you could add the following transformation:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>networkdays(Date1, Date2, ['2020-03-17'])</code>
Parameter: New column name	'networkDaysStPatricks'

In the following transformation, the NETWORKDAYSINTL function is applied so that you can specify the working days in the week:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	networkdaysintl(Date1, Date2, '1000011', [])
Parameter: New column name	'networkDaysIntl'

The following two functions enable you to calculate a specific working date based on an input date and integer number of days before or after it. In the following, the date that is five working days before the Date2 column is computed:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	workday(Date2, -5)
Parameter: New column name	'workday'

Suppose you wish to factor in a four-day workweek, in which Friday through Sunday is considered the weekend:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	workdayintl(Date2, -5, '0000111')
Parameter: New column name	'workdayintl'

Results:

Date1	Date2	workdayintl	workday	networkDaysIntl	networkDaysStPatricks	networkDays	datedif
2020-03-09	2020-03-13	2020-03-05	2020-03-06	4	5	5	4
2020-03-09	2020-03-06	2020-02-27	2020-02-28	<i>null</i>	<i>null</i>	<i>null</i>	-3
2020-03-09	2020-03-16	2020-03-15	2020-03-09	4	6	6	7
2020-03-09	2020-03-23	2020-03-12	2020-03-16	8	10	11	14
2020-03-09	2020-04-10	2020-04-02	2020-04-03	20	24	25	32
2020-03-09	2021-03-10	2021-03-02	2021-03-03	210	262	263	366