

How to Work with 17+ Date and Time Functions in DAX — DAX in Power BI —

We are going to look at 23 date and time functions in more detail with examples.



Photo by [Debby Hudson](#) on [Unsplash](#)

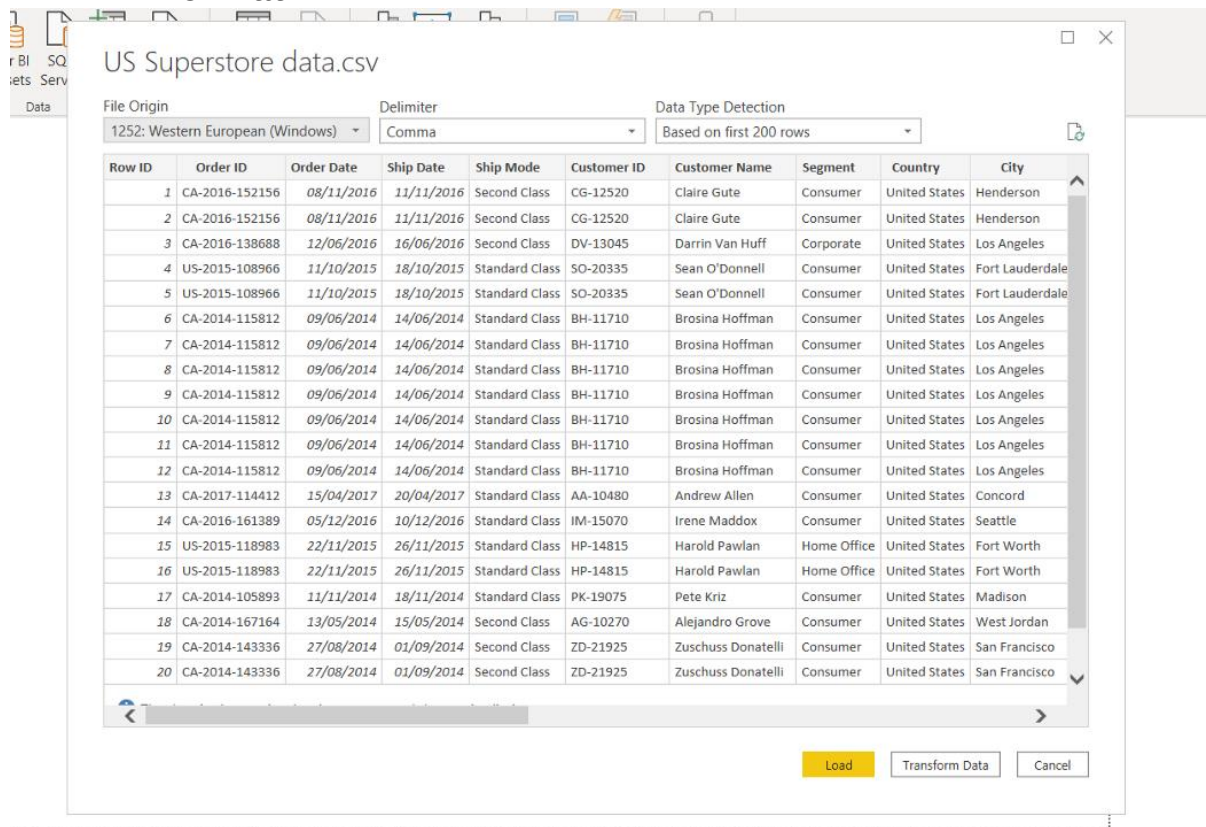
As per Microsoft documentation, there are 23 date and time functions are available in DAX. These functions are more or less similar to excel date-time functions. The only difference is that the DAX function can use DateTime data type and can take values from a column as an argument.

In this blog, we are going to explore them with some hands-on examples.

Get Data

At first import, the US Super Store data from [Kaggle](#) and save it to Power BI Desktop.

- Let's start with the **Get Data** option under the **Home** tab. As this is a CSV file, select the **Text/CSV** option from the drop-down list
- Select the file named **US Superstore data.csv**
- After selecting the file, data will be displayed in the below format



US Superstore data.csv

File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City
1	CA-2016-152156	08/11/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
2	CA-2016-152156	08/11/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
3	CA-2016-138688	12/06/2016	16/06/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles
4	US-2015-108966	11/10/2015	18/10/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale
5	US-2015-108966	11/10/2015	18/10/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale
6	CA-2014-115812	09/06/2014	14/06/2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
7	CA-2014-115812	09/06/2014	14/06/2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
8	CA-2014-115812	09/06/2014	14/06/2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
9	CA-2014-115812	09/06/2014	14/06/2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
10	CA-2014-115812	09/06/2014	14/06/2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
11	CA-2014-115812	09/06/2014	14/06/2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
12	CA-2014-115812	09/06/2014	14/06/2014	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles
13	CA-2017-114412	15/04/2017	20/04/2017	Standard Class	AA-10480	Andrew Allen	Consumer	United States	Concord
14	CA-2016-161389	05/12/2016	10/12/2016	Standard Class	IM-15070	Irene Maddox	Consumer	United States	Seattle
15	US-2015-118983	22/11/2015	26/11/2015	Standard Class	HP-14815	Harold Pawlan	Home Office	United States	Fort Worth
16	US-2015-118983	22/11/2015	26/11/2015	Standard Class	HP-14815	Harold Pawlan	Home Office	United States	Fort Worth
17	CA-2014-105893	11/11/2014	18/11/2014	Standard Class	PK-19075	Pete Kriz	Consumer	United States	Madison
18	CA-2014-167164	13/05/2014	15/05/2014	Second Class	AG-10270	Alejandro Grove	Consumer	United States	West Jordan
19	CA-2014-143336	27/08/2014	01/09/2014	Second Class	ZD-21925	Zuschuss Donatelli	Consumer	United States	San Francisco
20	CA-2014-143336	27/08/2014	01/09/2014	Second Class	ZD-21925	Zuschuss Donatelli	Consumer	United States	San Francisco

Buttons: Load, Transform Data, Cancel

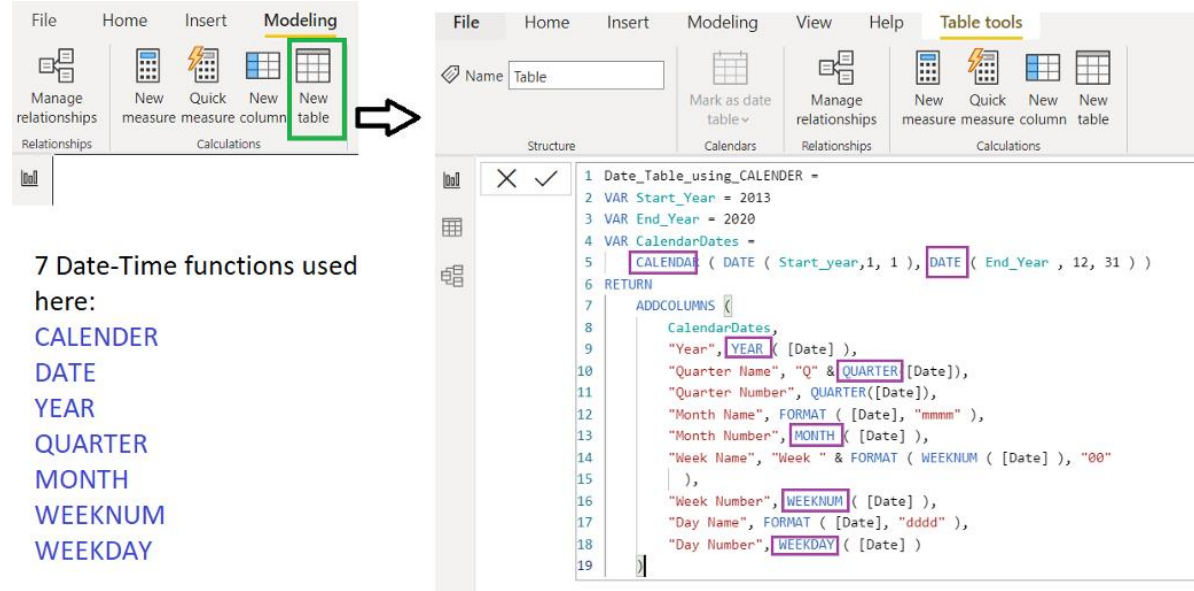
Image by Author

- Click on Load and save data.

Use of 8 functions to make a Date table

You are aware that you can create an automatic date table using the DAX function and some additional date-time functions can help you to make a proper date dimension table.

Date table with CALENDER



The screenshot shows the Power BI Desktop interface. On the left, the 'Modeling' tab is active, and the 'New table' button is highlighted with a green box. An arrow points from this button to the DAX formula editor on the right. The formula editor shows the following DAX code:

```
1 Date_Table_using_CALENDER =
2 VAR Start_Year = 2013
3 VAR End_Year = 2020
4 VAR CalendarDates =
5     CALENDAR ( DATE ( Start_Year, 1, 1 ), DATE ( End_Year, 12, 31 ) )
6 RETURN
7     ADDCOLUMNS (
8         CalendarDates,
9         "Year", YEAR ( [Date] ),
10        "Quarter Name", "Q" & QUARTER ( [Date] ),
11        "Quarter Number", QUARTER ( [Date] ),
12        "Month Name", FORMAT ( [Date], "mmmm" ),
13        "Month Number", MONTH ( [Date] ),
14        "Week Name", "Week " & FORMAT ( WEEKNUM ( [Date] ), "00" ),
15        "Week Number", WEEKNUM ( [Date] ),
16        "Day Name", FORMAT ( [Date], "dddd" ),
17        "Day Number", WEEKDAY ( [Date] )
18    )
```

Image by Author

From the above example, you can observe that one date table has been created using CALENDAR, DATE, YEAR, QUARTER, MONTH, WEEKNUM and WEEKDAY date-time functions.

CALENDAR() → It returns a table with a single date column. It takes the start date and end date as arguments.

DATE() → It takes arguments as year, month, day and returns specified date with DateTime format.

YEAR() → It takes a date as an argument and returns a year value which is a four-digit integer.

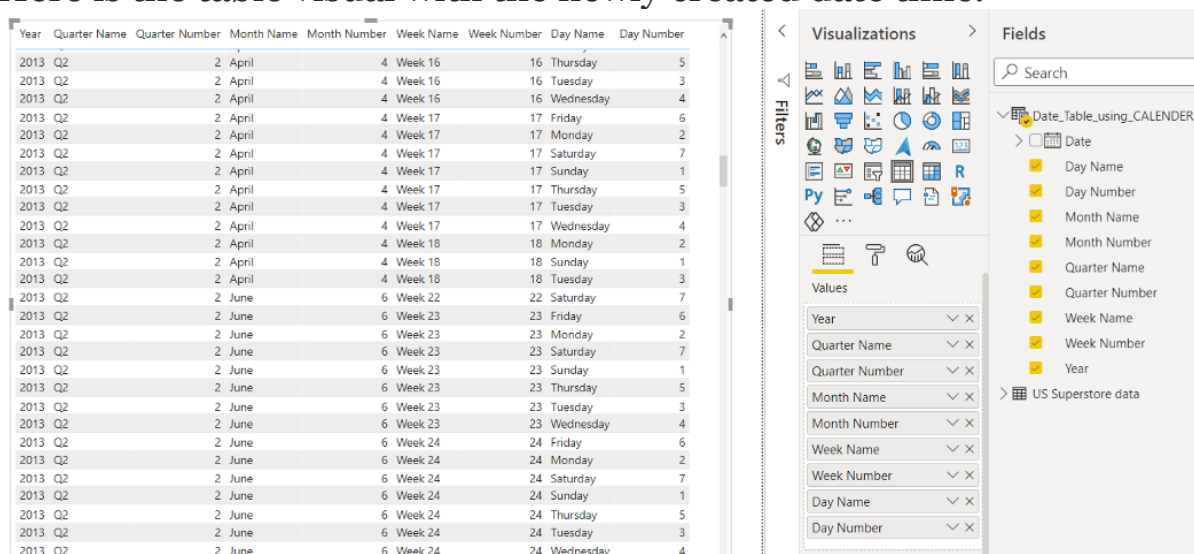
QUARTER() -> It takes a date as an argument and returns the quarter value which is a number from 1 to 4.

MONTH() -> It takes a date as an argument and returns the month number which is from 1 to 12.

WEEKNUM() → It takes a date as an argument and returns the week number for the specified date and year according to the return_type value.

WEEKDAY() → It takes a date as an argument and returns a number from 1 to 7 based on the day of the week.

Here is the table visual with the newly created date time.



The screenshot displays a table visual in Power BI with the following columns: Year, Quarter Name, Quarter Number, Month Name, Month Number, Week Name, Week Number, Day Name, and Day Number. The data spans from 2013 Q2 to 2013 Q2, showing various dates and their corresponding week and day information. To the right of the table is the 'Fields' pane, which shows a search bar and a list of fields under the 'Date' category. The fields listed are: Day Name, Day Number, Month Name, Month Number, Quarter Name, Quarter Number, Week Name, Week Number, and Year. All these fields are currently selected (checked). Below the 'Fields' pane is the 'Visualizations' pane, which shows a list of visual types (Bar, Line, Pie, etc.) and a 'Values' section with a list of fields (Year, Quarter Name, Quarter Number, Month Name, Month Number, Week Name, Week Number, Day Name, Day Number) that can be added to the visual.

Year	Quarter Name	Quarter Number	Month Name	Month Number	Week Name	Week Number	Day Name	Day Number
2013	Q2	2	April	4	Week 16	16	Thursday	5
2013	Q2	2	April	4	Week 16	16	Tuesday	3
2013	Q2	2	April	4	Week 16	16	Wednesday	4
2013	Q2	2	April	4	Week 17	17	Friday	6
2013	Q2	2	April	4	Week 17	17	Monday	2
2013	Q2	2	April	4	Week 17	17	Saturday	7
2013	Q2	2	April	4	Week 17	17	Sunday	1
2013	Q2	2	April	4	Week 17	17	Thursday	5
2013	Q2	2	April	4	Week 17	17	Tuesday	3
2013	Q2	2	April	4	Week 17	17	Wednesday	4
2013	Q2	2	April	4	Week 18	18	Monday	2
2013	Q2	2	April	4	Week 18	18	Sunday	1
2013	Q2	2	April	4	Week 18	18	Tuesday	3
2013	Q2	2	June	6	Week 22	22	Saturday	7
2013	Q2	2	June	6	Week 23	23	Friday	6
2013	Q2	2	June	6	Week 23	23	Monday	2
2013	Q2	2	June	6	Week 23	23	Saturday	7
2013	Q2	2	June	6	Week 23	23	Sunday	1
2013	Q2	2	June	6	Week 23	23	Thursday	5
2013	Q2	2	June	6	Week 23	23	Tuesday	3
2013	Q2	2	June	6	Week 23	23	Wednesday	4
2013	Q2	2	June	6	Week 24	24	Friday	6
2013	Q2	2	June	6	Week 24	24	Monday	2
2013	Q2	2	June	6	Week 24	24	Saturday	7
2013	Q2	2	June	6	Week 24	24	Sunday	1
2013	Q2	2	June	6	Week 24	24	Thursday	5
2013	Q2	2	June	6	Week 24	24	Tuesday	3
2013	Q2	2	June	6	Week 24	24	Wednesday	4

Image by Author

Date table with CALENDERAUTO

With CALENDERAUTO function Power BI automatically finds the start date and end date from the data model where some date fields exist.

But it has one limitation. If your data model contains some dates that are not used for analysis, like date of birth, then it will pick those. This will give a much wider date range than you want.

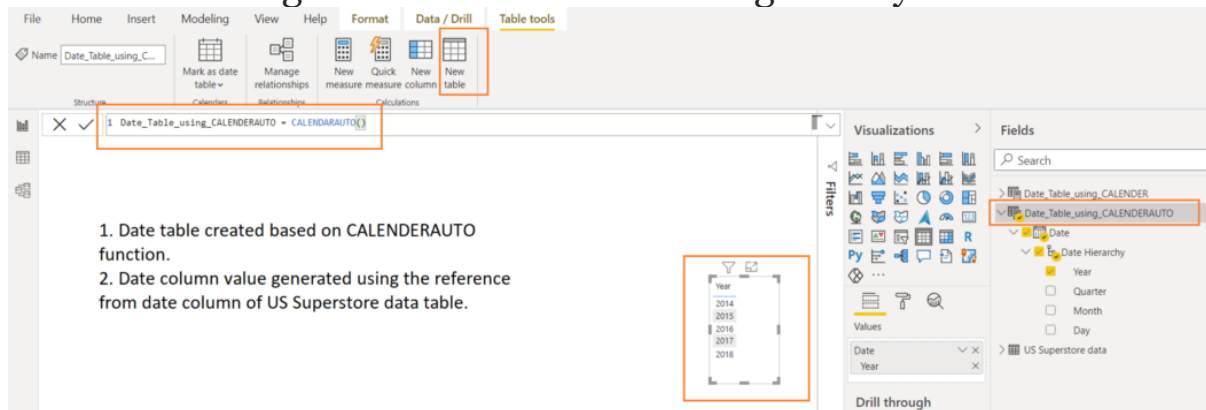


Image by Author

Some more sample of WEEKNUM and WEEKDAY



Image by Author — WEEKNUM

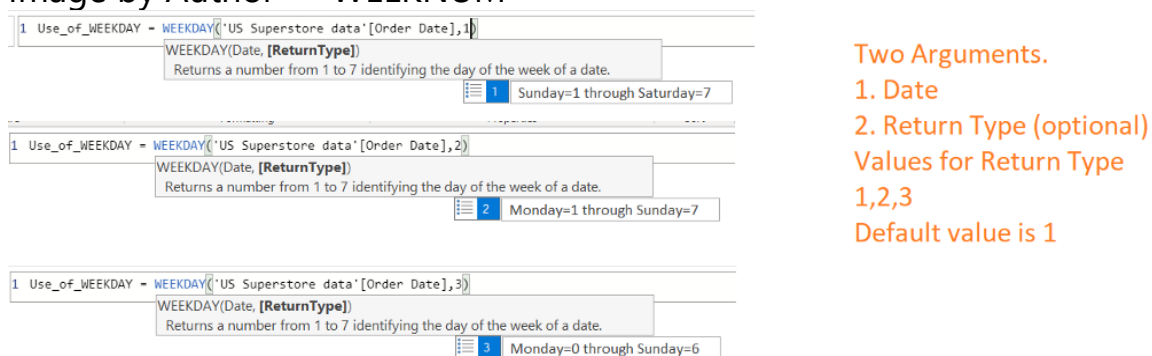


Image by Author — WEEKDAY

The TODAY, DATEDIFF, DATEVALUE, DAY Functions

Today() function is self-explanatory. It returns today's date, but not the time.

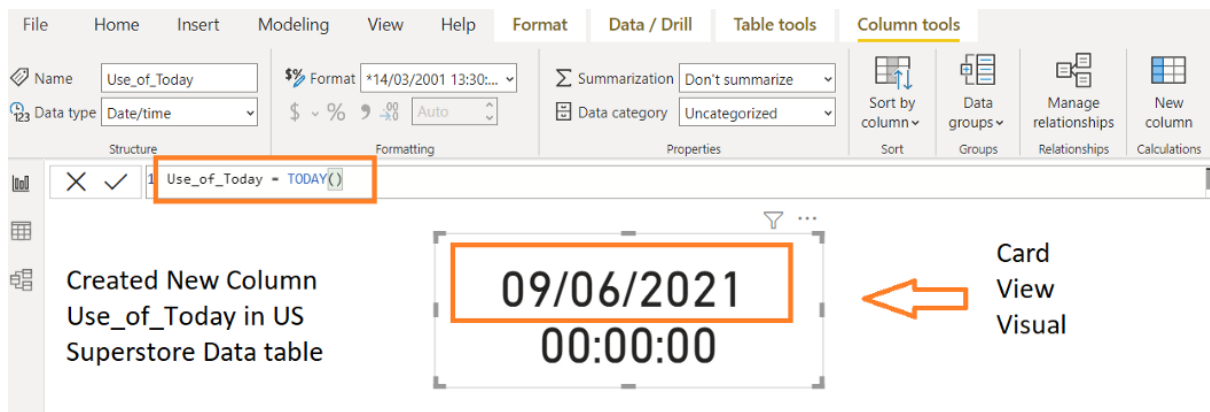
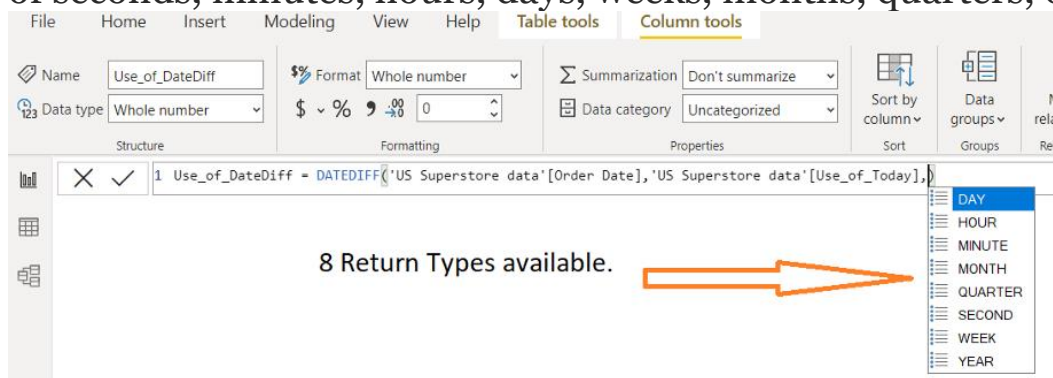


Image by Author

The **DATEDIFF** function returns the total amount of time between two specified dates. The best part is you can get the results in terms of seconds, minutes, hours, days, weeks, months, quarters, or years.



Created two calculated columns

```
Use_of_DateDiff_Day = DATEDIFF('US Superstore data'[Order Date], 'US Superstore data'[Use_of_Today], DAY)
```

```
Use_of_DateDiff_Year = DATEDIFF('US Superstore data'[Order Date], 'US Superstore data'[Use_of_Today], YEAR)
```

Order Date	Use_of_Today	Use_of_DateDiff_Year	Use_of_DateDiff_Day
01 January 2017	09 June 2021	4	1620
02 January 2017	09 June 2021	4	1619
03 January 2017	09 June 2021	4	1618
06 January 2017	09 June 2021	4	1615
07 January 2017	09 June 2021	4	1614
08 January 2017	09 June 2021	4	1613

Created report using two new columns

Image by Author

The **DATEVALUE** function considers a date value that is in the text format and converts to date in date-time format. If you are not mentioning any year, by default it will consider the current year.

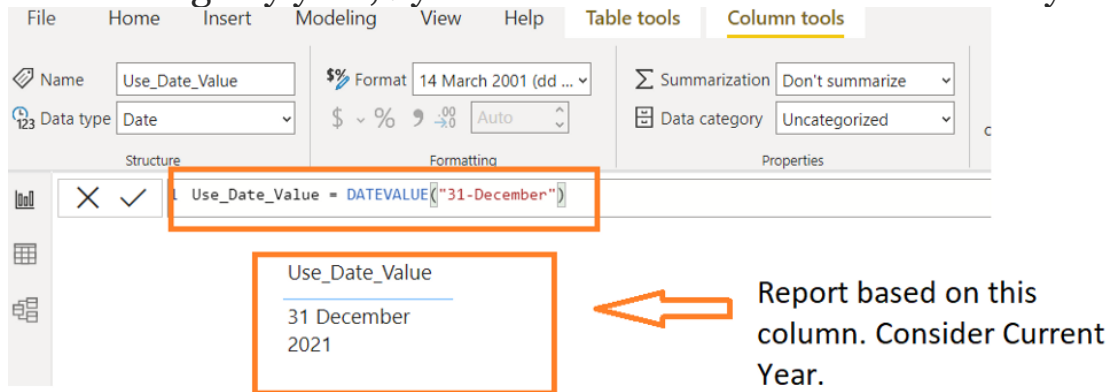


Image by Author

The **DAY** function is very simple. It returns the day number from 1 to 31.

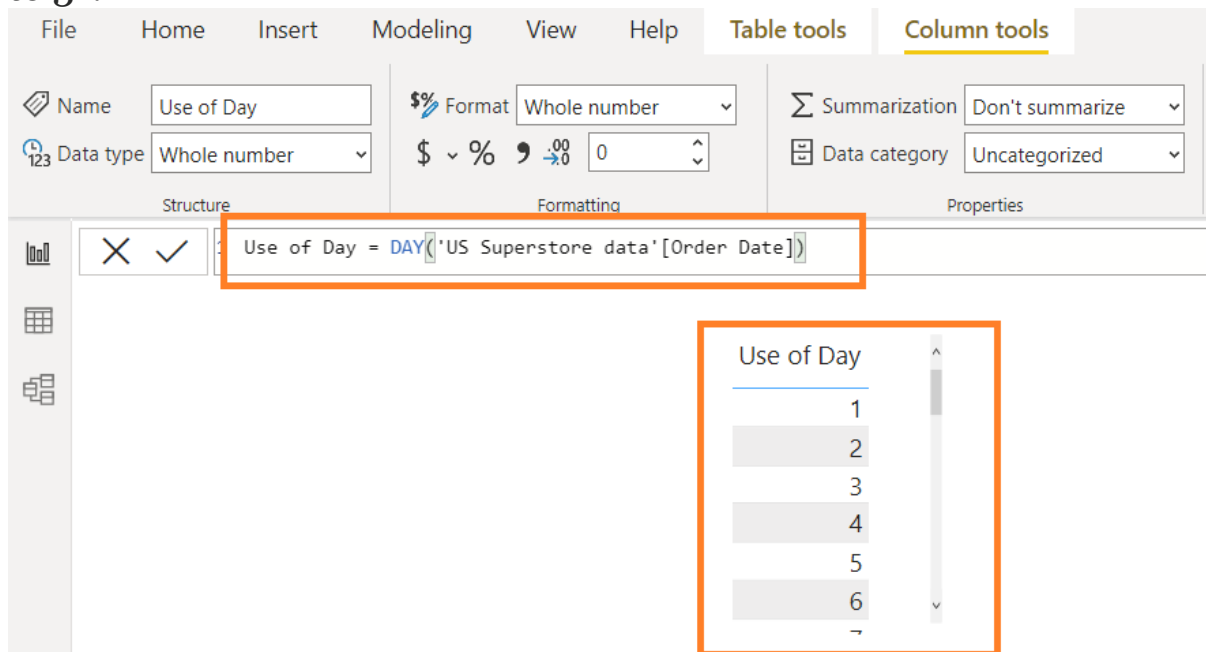


Image by Author

The NOW, HOUR, MINUTE, SECOND, TIME, TIMEVALUE functions

The **NOW** function returns the current date as well as the current time. This is the main difference with the TODAY function.

The screenshot shows the Tableau interface with the 'Format' tab selected. The 'Name' field is 'Use_of_NOW', 'Data type' is 'Date/time', and 'Format' is '*14/03/2001 13:30:...'. The 'Summarization' is 'Don't summarize' and 'Data category' is 'Uncategorized'. Below the interface, two visualizations are shown. The first is a single box labeled 'Use_of_NOW' containing the timestamp '09/06/2021 12:00:22'. An orange arrow points from this box to the text 'Displaying Current System Timestamp.'. The second visualization is a box with two columns: 'Use_of_NOW' and 'Use_of_Today'. The first column contains '09/06/2021 12:00:22' and the second contains '09 June 2021'. An orange arrow points from this box to the text 'Difference Between NOW and TODAY'.

Image by Author

As the name suggested, the **HOUR**, **MINUTE** and **SECOND** functions extract the hour, minute and second information from a given date and time value

The screenshot shows the Tableau interface with three calculated fields: 'Using_HOUR = HOUR([US Superstore data][Use_of_NOW])', 'Using_Minute = MINUTE([US Superstore data][Use_of_NOW])', and 'Using_Second = SECOND([US Superstore data][Use_of_NOW])'. An orange bracket on the right side of these fields points to the text 'Created 3 columns for HOUR, MINUTE, SECOND'.

Use_of_NOW	Using_HOUR	Using_Minute	Using_Second
09/06/2021 12:00:22	12	0	22

Report with 3 newly created columns

Image by author

The **TIME** function returns a time in DateTime format if you pass hours, minutes and seconds as numbers.

Whereas the **TIMEVALUE** returns a time in DateTime format if you pass time in text format.

The screenshot shows the DAX editor for a column named 'Use_of_Time'. The data type is set to 'Date/time'. The formula bar contains the formula: `Use_of_Time = TIME(2,20,30)`. The formula is highlighted with an orange box.

The screenshot shows the DAX editor for a column named 'Use_of_Timevalue'. The data type is set to 'Date/time'. The formula bar contains the formula: `Use_of_Timevalue = TIMEVALUE("22:30:30")`. The formula is highlighted with an orange box.

Use_of_Time	Use_of_Timevalue
30/12/1899 02:20:30	30/12/1899 22:30:30



Created report with two new columns.

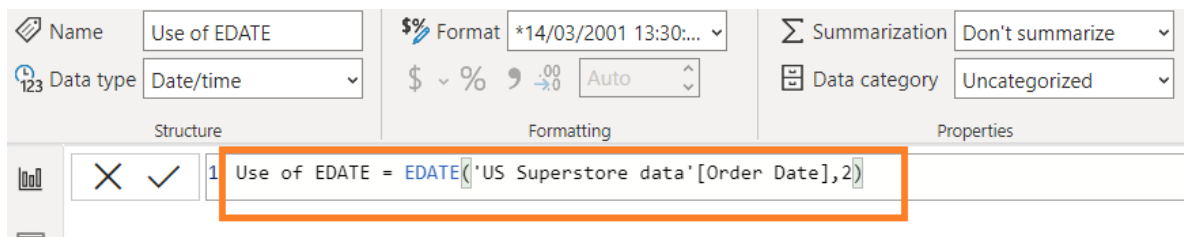
Displaying the Proper Time.

But as didn't mention any date, so it is considering Power BI default date.

Image by Author

The EDATE, EOMONTH Functions

The **EDATE** function considers two arguments, the first one is the date and the second is the number of months. It returns the date with the specified number of months before or after the given start date.



Order Date	Use of EDATE
03 January 2014	03 March 2014
04 January 2014	04 March 2014
05 January 2014	05 March 2014
06 January 2014	06 March 2014
07 January 2014	07 March 2014

Report with Order date and new column. It reflects the results.

Image by Author

Whereas the **EOMONTH** function is similar to the EDATE function, except it returns the last date of the month where the specified number of months is before or after the given start date.

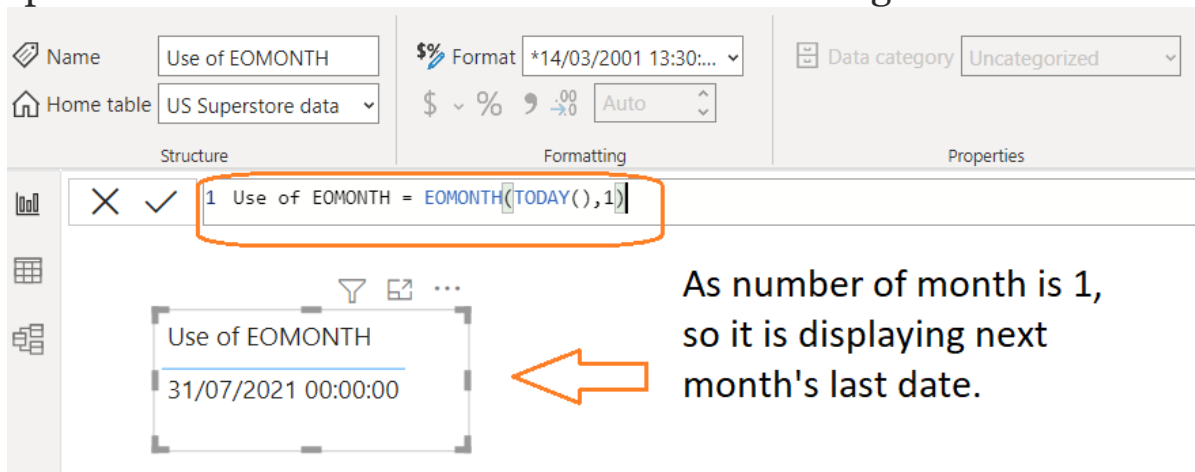


Image by Author

The UTCNOW and UTCTODAY Functions

Using the **UTCNOW** and **UTCTODAY** functions, you can get the current timestamp and current date for the UTC (Coordinated Universal Time) time zone.

Name	Use_of_UTCNOW	Format	*14/03/2001 13:30:...	Summarization	Don't summarize
Data type	Date/time	Formatting	Auto	Data category	Uncategorized
Structure					
1 Use_of_UTCNOW = UTCNOW()					

Name	Use_of_UTCTODAY	Format	*14/03/2001 13:30:...	Summarization	Don't summarize
Data type	Date/time	Formatting	Auto	Data category	Uncategorized
Structure					
1 Use_of_UTCTODAY = UTCTODAY()					



Report with two new columns.

Use_of_UTCNOW	Use_of_UTCTODAY
09/06/2021 11:43:50	09/06/2021 00:00:00

The YEARFRAC Function

The YEARFRAC function calculates the fraction of the year represented by the number of whole days between two dates.

If you want to know in detail, please refer to [Microsoft documentation](#).

The syntax of the YEARFRAC function is as follows:

YEARFRAC (<StartDate>, <EndDate> [, <Basis>])

Values for Basis →

0 — US (NASD) 30/360

1 — Actual/actual

2 — Actual/360

3 — Actual/365

4 — European 30/360

The screenshot shows the DAX Studio interface. At the top, the formula bar displays the formula: `Use_of_YEARFRAC = YEARFRAC('US Superstore data'[Order Date], TODAY(), 1)`. Below the formula bar, a table is displayed with three columns: `Order Date`, `Use_of_Today`, and `Use_of_YEARFRAC`. The table contains data for dates from December 23, 2017, to December 30, 2017, all with a `Use_of_Today` value of 09 June 2021. The `Use_of_YEARFRAC` values range from 3.44 to 3.46.

Order Date	Use_of_Today	Use_of_YEARFRAC
30 December 2017	09 June 2021	3.44
29 December 2017	09 June 2021	3.44
28 December 2017	09 June 2021	3.45
27 December 2017	09 June 2021	3.45
26 December 2017	09 June 2021	3.45
25 December 2017	09 June 2021	3.46
24 December 2017	09 June 2021	3.46
23 December 2017	09 June 2021	3.46

Image by Author

Conclusion

In this doc, we learn the 23 date and time functions of DAX with practical examples.