

Excel DAX - Working with Text and Dates

DAX can be used in the scenarios involving working along with text, extracting and composing date and time values or creating values based on a condition. You can do the following with DAX –

- Create a key column in a table by concatenation.
- Compose a date based on date parts extracted from a text date.
- Define a custom date format.
- Change data types using a formula.
 - Convert real numbers to integers.
 - Convert real numbers, integers, or dates to strings.
 - Convert strings to real numbers or dates.

Creating a Key Column by Concatenation

Data Model in PowerPivot allows only a single key column. It does not support composite keys that you might find in the external data sources. Hence, if any composite keys exist in a table in the data source, you need to combine them into a single key column for the table in the Data Model.

You can use the DAX function CONCATENATE to combine two columns into a single column in a table in the Data Model. DAX function CONCATENATE joins two text strings into one text string. The joined items can be text, numbers, or Boolean values represented as text or a combination of those items. You can also use a column reference if the column contains appropriate values.

```
= CONCATENATE ([Column1], [Column2])
```

DAX CONCATENATE function accepts only two arguments. If any of the arguments is not of text data type, it will be converted to text. DAX CONCATENATE function returns the concatenated string.

Date Based on Date Parts Extracted from a Text Date

Data Model in Power Pivot supports a data type datetime for date and time values. The DAX functions that work on date and/or time values require the datetime data type for the arguments.

If your data source contains dates of a different format, you need to first extract the date parts using a DAX formula and combine those parts to constitute a valid DAX datetime data type.

You can use the following DAX functions to extract and compose dates –

DATE – Returns the specified date in datetime format.

DATEVALUE – Converts a date in the form of text to a date in datetime format.

TIMEVALUE – Converts a time in text format to a time in datetime format.

Defining a Custom Date Format

Suppose the dates in your data source are not represented by a standard format. You can define a custom date format to ensure that the values are handled correctly. DAX FORMAT function enables you to convert a value to text according to the specified format.

```
FORMAT (<value>, <format_string>)
```

FORMAT function returns a string containing value formatted as defined by format_string.

You can either use pre-defined Date and Time formats or you can create user-defined Date and Time formats for the argument format_string of the FORMAT function.

Following are the predefined date and time format names. If you use strings other than these predefined strings, they will be interpreted as a custom date and time format.

S. No.	Format_String & Description
1	"General Date" Displays a date and/or time. For example, 2/10/2015 10:10:32 AM
2	"Long Date" or "Medium Date" Displays a date according to long date format. For example, Wednesday, March 07, 2016
3	"Short Date" Displays a date using short date format. For example, 2/03/2016
4	"Long Time" Displays a time using long time format. Typically includes hours, minutes and seconds. For example, 10:10:32 AM
5	"Medium Time" Displays a time in 12-hour format. For example, 09:30 PM
6	"Short Time" Displays a time in 24-hour format. For example, 14:15

Alternatively, you can use the characters in the following table to create user-defined date/time formats.

S. No.	Character & Description
1	: Time separator. Time separator. Separates hours, minutes, and seconds when time values are formatted.
2	/ Date separator. Separates the day, month, and year when date values are formatted.
3	% Used to indicate that the following character should be read as a single-letter format without regard to any trailing letters. Also used to indicate that a single-letter format is read as a userdefined format.

Following are the details of the various characters.

- **%d** – Displays the day as a number without a leading zero (e.g. 5).
- **%dd** – Displays the day as a number with a leading zero (e.g. 05).
- **%ddd** – Displays the day as an abbreviation (e.g. Sun).
- **%dddd** – Displays the day as a full name (e.g. Sunday).
- **%M** – Displays the month as a number without a leading zero (e.g. January is represented as 1).
- **%MM** – Displays the month as a number with a leading zero (e.g. January is represented as 01).
- **%MMM** – Displays the month as an abbreviation (e.g. January is represented as Jan).
- **%MMMM** – Displays the month as a full month name (e.g. January).
- **%gg** – Displays the period/era string (e.g. A.D.).
- **%h** – Displays the hour as a number without leading zeros using the 12-hour clock (e.g. 1:15:15 PM). Use %h if this is the only character in your user-defined numeric format.
- **%hh** – Displays the hour as a number with leading zeros using the 12-hour clock (e.g. 01:15:15 PM).
- **%H** – Displays the hour as a number without leading zeros using the 24-hour clock (e.g. 13:15:15, 1:15:15). Use %H if this is the only character in your user-defined numeric format.
- **%HH** – Displays the hour as a number with leading zeros using the 24-hour clock (e.g. 13:15:15, 1:15:15).
- **%m** – Displays the minute as a number without leading zeros (e.g. 2:1:15). Use %m if this is the only character in your user-defined numeric format.
- **%mm** – Displays the minute as a number with leading zeros (e.g. 2:01:15).
- **%s** – Displays the second as a number without leading zeros (e.g. 2:15:5). Use %s if this is the only character in your user-defined numeric format.
- **%ss** – Displays the second as a number with leading zeros (e.g. 2:15:05).
- **%f** – Displays fractions of seconds. For e.g. **ff** displays hundredths of seconds, whereas **ffff** displays ten-thousandths of seconds. You can use up to seven **f** symbols in your user-defined format. Use %f if this is the only character in your user-defined numeric format.
- **%t** – Uses the 12-hour clock and displays an uppercase A for any hour before noon; displays an uppercase P for any hour between noon and 11:59 P.M. Use %t if this is the only character in your user-defined numeric format.
- **%tt** – For locales that use a 12-hour clock, displays an uppercase AM with any hour before noon; displays an uppercase PM with any hour between noon and 11:59 P.M. For locales that use a 24-hour clock, displays nothing.
- **%y** – Displays the year number (0-9) without leading zeros. Use %y if this is the only character in your user-defined numeric format.
- **%yy** – Displays the year in two-digit numeric format with a leading zero, if applicable.
- **%yyy** – Displays the year in four-digit numeric format.
- **%yyyy** – Displays the year in four-digit numeric format.
- **%z** – Displays the timezone offset without a leading zero (e.g. -8). Use %z if this is the only character in your user-defined numeric format.
- **%zz** – Displays the timezone offset with a leading zero (e.g. -08)
- **%zzz** – Displays the full timezone offset (e.g. -08:00).

As you can observe, formatting strings are case sensitive. Different formatting can be obtained by using a different case.

Changing Data Types of DAX Formula Outputs

In DAX formulas, the data type of the output is determined by the source columns and you cannot explicitly specify the data type of the result. This is because the optimal data type is determined by Power Pivot. However, you can use the implicit data type conversions performed by Power Pivot to manipulate the output data type. Otherwise, you can use certain DAX functions to convert the output data type.

Using the Implicit Data Type Conversions

- To convert a date or a number string to a number, multiply by 1.0. For example, = (TODAY()+5)*1.0. This formula calculates the current date plus 5 days and converts the result to an integer value.

- To convert a date, number or currency value to a string, concatenate the value with an empty string. For example, = Today() & ""

Using the DAX Functions for Data Type Conversions

You can use DAX functions for the following –

- Converting Real Numbers to Integers.
- Converting Real Numbers, Integers or Dates to Strings.
- Converting Strings to Real Numbers or Dates.

You will learn this in the following sections.

Converting Real Numbers to Integers

You can use the following DAX functions for converting real numbers to integers –

ROUND (<number>, <num_digits>) – Rounds a number to the specified number of digits and returns a decimal number.

CEILING (<number>, <significance>) – Rounds a number up, to the nearest integer or to the nearest multiple of significance and returns a decimal number.

FLOOR (<number>, <significance>) – Rounds a number down, toward zero, to the nearest multiple of significance and returns a decimal number.

Converting Real Numbers, Integers, or Dates to Strings

You can use the following DAX functions for converting real numbers, integers, or dates to strings –

FIXED (<number>, [<decimals>], [<no_comma>]) – Rounds a number and returns the result as text. The number of digits to the right of the decimal point is 2 or the specified number of decimals. The result is with commas or optionally with no commas.

FORMAT (<value>, <format_string>) – Converts a value to text according to the specified format.

You have already learnt about using Format function for converting dates to strings.

Converting Strings to Real Numbers or Dates

You can use the following DAX functions for converting strings to real numbers or dates –

VALUE (<text>) – Converts a text string that represents a number to a number.

DATEVALUE (date_text) – Converts a date in the form of text to a date in datetime format.

TIMEVALUE (time_text) – Converts a time in text format to a time in datetime format.