⊟ Collapse All   ▶ Code: All

DAX Scenarios

See Also Send Feedback

This section provides links to examples that demonstrate the use of DAX formulas in the following scenarios.

Performing complex calculations

Working with text and dates

Conditional values and testing for errors

Using time intelligence

Ranking and comparing values

⊟ Getting Started

If you are new to DAX formulas, we recommend that you begin by reviewing the examples in the DAX Sample workbook. For more information about how to obtain the sample workbook, see Get Sample Data for PowerPivot

Additional Resources

You can use the following links to find videos, additional samples, and walkthroughs to help you learn DAX.

- PowerPivot team blog
- PowerPivot for Excel Online Help

⊟ Scenarios: Performing Complex Calculations

DAX formulas can perform complex calculations that involve custom aggregations, filtering, and the use of conditional values. This section provides examples of how to get started with custom calculations.

Create custom calculations for a PivotTable

CALCULATE and CALCULATETABLE are powerful, flexible functions that useful for defining measures. These functions let you change the context in which the calculation will be performed. You can also customize the type of aggregation or mathematical operation to perform. See the following topics for examples.

- CALCULATE Function (DAX)
- CALCULATETABLE Function (DAX)

Apply a filter to a formula

In most places where a DAX function takes a table as an argument, you can usually pass in a filtered table instead, either by using the FILTER function instead of the table name, or by specifying a filter expression as one of the function arguments. The following topics provide examples of how to create filters and how filters affect the results of formulas. For more information, see Filter Data in Formulas.

The FILTER function lets you specify filter criteria by using an expression, while the other functions are designed specifically to filter out blank values.

- FILTER Function (DAX)
- ALLNOBLANKROW Function (DAX)
- FIRSTNONBLANK Function (DAX)
- LASTNONBLANK Function (DAX)

Remove filters selectively to create a dynamic ratio

By creating dynamic filters in formulas, you can easily answer questions such as the following:

- What was the contribution of the current product's sales to total sales for the year?
- How much has this division contributed to total profits for all operating years, compared to other divisions?

Formulas that you use in a PivotTable can be affected by the PivotTable context, but you can selectively change the context by adding or removing filters. The example in the ALL topic shows you how to do this. To find the ratio of sales for a specific reseller over the sales for all resellers, you create a measure that calculates the value for the current context divided by the value for the ALL context.

The ALLEXCEPT topic provides an example of how to selectively clear filters on a formula. Both examples walk you through how the results change depending on the design of the PivotTable.

- ALL Function (DAX)
- ALLEXCEPT Function (DAX)

For other examples of how to calculate ratios and percentages, see the following topics:
- ISBLANK Function (DAX)
- ISERROR Function (DAX)

Using a value from an outer loop

In addition to using values from the current context in calculations, DAX can use a value from a previous loop in creating a set of related calculations. The following topic provides a walkthrough of how to build a formula that references a value from an outer loop. The EARLIER function supports up to two levels of nested loops.

- EARLIER Function (DAX)

To learn more about row context and related tables, and how to use this concept in formulas, see Context in DAX Formulas.

Back to Top

⊟ Scenarios: Working with Text and Dates

This section provides links to DAX reference topics that contain examples of common scenarios involving working with text, extracting and composing date and time values, or creating values based on a condition.

Create a key column by concatenation

PowerPivot does not allow composite keys; therefore, if you have composite keys in your data source you may need to combine them into a single key column. The following topic provides one example of how to create a calculated column based on a composite key.

- CONCATENATE Function (DAX)

Compose a date based on date parts extracted from a text date

PowerPivot uses a SQL Server date/time data type to work with dates; therefore, if your external data contains dates that are formatted differently -- for example, if your dates are written in a regional date format that is not recognized by the PowerPivot data engine, or if your data uses integer surrogate keys -- you may need to use a DAX formula to extract the date parts and then compose the parts into a valid date/time representation.

For example, if you have a column of dates that have been represented as an integer and then imported as a text string, you can convert the string to a date/time value by using the following formula:

🗎 Copy Code

```
=DATE(RIGHT([Value1],4),LEFT([Value1],2),MID([Value1],2))
```

Value1	Result
01032009	1/3/2009
12132008	12/13/2008
06252007	6/25/2007

The following topics provide more information about the functions used to extract and compose dates.
- DATEVALUE Function (DAX)
- TIMEVALUE Function (DAX)
- DATE Function (DAX)

Define a custom date or number format

If your data contains dates or numbers that are not represented in one of the standard Windows text formats, you can define a custom format to ensure that the values are handled correctly. These formats are used when converting values to strings, or from strings. The following topics also provide a detailed list of the predefined formats that are available for working with dates and numbers.
- Pre-Defined Numeric Formats for the FORMAT Function (DAX)
- Custom Numeric Formats for the FORMAT Function (DAX)
- Pre-defined Date and Time Formats for the FORMAT Function (DAX)
- Custom Date and Time Formats for the FORMAT Function (DAX)

Change data types using a formula

In PowerPivot, the data type of the output is determined by the source columns, and you cannot explicitly specify the data type of the result, because the optimal data type is determined by PowerPivot. However, you can use the implicit data type conversions performed by PowerPivot to manipulate the output data type. For more information about type conversions, see Data Sources Supported in PowerPivot Workbooks.

- To convert a date or a number-string to a number, multiply by 1.0. For example, the following formula calculates the current date minus 3 days, and then outputs the corresponding integer value.

🗐 Copy Code

```
=(TODAY()-3)*1.0
```

- To convert a date, number, or currency value to a string, concatenate the value with an empty string. For example, the following formula returns today's date as a string.

🗐 Copy Code

```
=""& TODAY()
```

The following functions also can be used to ensure that a particular data type is returned:

Convert real numbers to integers
- ROUND Function (DAX)
- CEILING Function (DAX)
- FLOOR Function (DAX)

Convert real numbers, integers, or dates to strings
- FIXED Function (DAX)
- FORMAT Function (DAX)

Convert strings to real numbers or dates
- VALUE Function (DAX)
- DATEVALUE Function (DAX)
- TIMEVALUE Function (DAX)

Back to Top

Scenario: Conditional Values and Testing for Errors

Like Excel, DAX has functions that let you test values in the data and return a different value based on a condition. For example, you could create a calculated column that labels resellers either as **Preferred** or **Value** depending on the yearly sales amount. Functions that test values are also useful for checking the range or type of values, to prevent unexpected data errors from breaking calculations.

Create a value based on a condition

You can use nested IF conditions to test values and generate new values conditionally. The following topics contain some simple examples of conditional processing and conditional values:

- IF Function (DAX)
- OR Function (DAX)
- CONCATENATE Function (DAX)

Test for errors within a formula

Unlike Excel, you cannot have valid values in one row of a calculated column and invalid values in another row. That is, if there is an error in any part of a PowerPivot column, the entire column is flagged with an error, so that you must always correct formula errors that result in invalid values.

For example, if you create a formula that divides by zero, you might get the infinity result, or an error. Some formulas will also fail if the function encounters a blank value when it expects a numeric value. While you are developing your data model, it is best to allow the errors to appear so that you can click on the message and troubleshoot the problem. However, when you publish workbooks, you should incorporate error handling to prevent unexpected values from causing calculations to fail.

To avoid returning errors in a calculated column, you use a combination of logical and information functions to test for errors and always return valid values. The following topics provide some simple examples of how to do this in DAX:

- ISBLANK Function (DAX)
- IFERROR Function (DAX)

Back to Top

Scenarios: Using Time Intelligence

The DAX time intelligence functions include functions to help you retrieve dates or date ranges from your data. You can then use those dates or date ranges to calculate values across similar periods. The time intelligence functions also include functions that work with standard date intervals, to allow you to compare values across months, years, or quarters. You could also create a formula that compares values for the first and last date of a specified period.

For a list of all time intelligence functions, see Time Intelligence Functions (DAX). For tips on how to use dates and times effectively in a PowerPivot analysis, see Dates in PowerPivot.

Calculate cumulative sales

The following topics contain examples of how to calculate closing and opening balances. The examples let you create running balances across different intervals such as days, months, quarters, or years.

- CLOSINGBALANCEMONTH Function (DAX), CLOSINGBALANCEQUARTER Function (DAX), CLOSINGBALANCEYEAR Function (DAX)
- OPENINGBALANCEMONTH Function (DAX), OPENINGBALANCEQUARTER Function (DAX), OPENINGBALANCEYEAR Function (DAX)

Compare values over time

The following topics contain examples of how to compare sums across different time periods. The default time periods supported by DAX are months, quarters, and years.

- PREVIOUSMONTH Function (DAX), PREVIOUSQUARTER Function (DAX), PREVIOUSYEAR Function (DAX)
- TOTALMTD Function (DAX), TOTALQTD Function (DAX), TOTALYTD Function (DAX)
- PARALLELPERIOD Function (DAX)

Calculate a value over a custom date range

See the following topics for examples of how to retrieve custom date ranges, such as the first 15 days after the start of a sales promotion.

- DATESINPERIOD Function (DAX)
- DATESBETWEEN Function (DAX)
- DATEADD Function (DAX)
- FIRSTDATE Function (DAX)
- LASTDATE Function (DAX)

If you use time intelligence functions to retrieve a custom set of dates, you can use that set of dates as input to a function that performs calculations, to create custom aggregates across time periods. See the following topic for an example of how to do this:

- PARALLELPERIOD Function (DAX)

📝 Note

If you do not need to specify a custom date range, but are working with standard accounting units such as months, quarters or years, we recommend that you perform calculations by using the time intelligence functions designed for this purpose, such as TOTALQTD, TOTALMTD, TOTALQTD, etc.

Back to Top

⊟ Scenarios: Ranking and Comparing Values

To show only the top n number of items in a column or PivotTable, you have several options:

- You can use the features in Excel 2010 to create a Top filter. You can also select a number of top or bottom values in a PivotTable. The first part of this section describes how to filter for the top 10 items in a PivotTable. For more information, see the Excel documentation.
- You can create a formula that dynamically ranks values, and then filter by the ranking values, or use the ranking value as a Slicer. The second part of this section describes how to create this formula and then use that ranking in a Slicer.

There are advantages and disadvantages to each method.

- The Excel Top filter is easy to use, but the filter is solely for display purposes. If the data underlying the PivotTable changes, you must manually refresh the PivotTable to see the changes. If you need to dynamically work with rankings, you can use DAX to create a formula that compares values to other values within a column.
- The DAX formula is more powerful; moreover, by adding the ranking value to a Slicer, you can just click on the Slicer to change the number of top values that are displayed. However, the calculations are computationally expensive and this method might not be suited for tables with many rows.

Show only the top ten items in a PivotTable

To show the top or bottom values in a PivotTable

1. In the PivotTable, click the down arrow in the **Row Labels** heading.
2. In the shortcut menu, select **Value Filters**, and then select **Top 10**.
3. In the **Top 10 Filter <column name>** dialog box, choose the column to rank, and the number of values, as follows:
    a. Select **Top** to see the cells with the highest values or **Bottom** to see the cells with the lowest values.
    b. Type the number of top or bottom values that you want to see. The default is 10.
    c. Select how you want the values displayed:

Name	Description
Items	Select this option to filter the PivotTable to display only the list of the top or bottom items by their values.
Percent	Select this option to filter the PivotTable to display only the items that add up to the specified percentage.
Sum	Select this option to display the sum of the values for the top or bottom items.

1. Select the column that contains the values you want to rank.
2. Click **OK**.

Order items dynamically by using a formula

The following topic contains an example of how to use DAX to create a ranking that is stored in a calculated column. Because DAX formulas are calculated dynamically, you can always be sure that the ranking is correct even if the underlying data has changed. Also, because the formula is used in a calculated column, you can use the ranking in a Slicer and then select top 5, top 10, or even top 100 values.

- EARLIER Function (DAX).

Back to Top

⊟ See Also

Concepts

Data Analysis Expressions (DAX) Overview
Overview of Adding Calculations in PowerPivot

Other Resources

Key Concepts in DAX