Collapse All Code: All

Filter Data in Formulas

See Also Send Feedback

You can create filters within formulas, to restrict the values from the source data that are used in calculations. You do this by specifying a table as an input to the formula, and then defining a *filter expression*. The filter expression you provide is used to query the data and return only a subset of the source data. The filter is applied dynamically each time that you update the results of the formula, depending on the current context of your data. This section describes how to create filters within Data Analysis Expressions (DAX) formulas.

Creating a Filter on a Table used in a Formula

You can apply filters in formulas that take a table as input. Instead of entering a table name, you use the FILTER function to define a subset of rows from the specified table. That subset then is passed to another function, for operations such as custom aggregations.

For example, suppose you have a table of data that contains order information about resellers, and you want to calculate how much each reseller sold. However, you want to show the sales amount just for those resellers who sold multiple units of your higher-value products. The following formula, based on the DAX sample workbook, shows one example of how you can create this calculation by using a filter:

Copy Code

```
=SUMX(
      FILTER ('ResellerSales_USD', 'ResellerSales_USD'[Quantity] > 5 &&
      'ResellerSales_USD'[ProductStandardCost_USD] > 100),
      'ResellerSales_USD'[SalesAmt]
      )
```

- The first part of the formula specifies one of the PowerPivot aggregation functions, which takes a table as an argument. `SUMX` calculates a sum over a table.
- The second part of the formula, `FILTER(table, expression)`, tells `SUMX` which data to use. `SUMX` requires a table or an expression that results in a table. Here, instead of using all the data in a table, you use the `FILTER` function to specify which of the rows from the table are used.

  The filter expression has two parts: the first part names the table to which the filter applies. The second part defines an expression to use as the filter condition. In this case, you are filtering on resellers who sold more than 5 units and products that cost more than $100. The operator, `&&`, is a logical AND operator, which indicates that both parts of the condition must be true for the row to belong to the filtered subset.
- The third part of the formula tells the `SUMX` function which values should be summed. In this case you are using just the sales amount.

Note that functions such as FILTER, which return a table, never return the table or rows directly to the PowerPivot workbook, but are always embedded in another function. For more information about FILTER and other functions used for filtering, including more examples, see Filter Functions (DAX).

Note

The filter expression is affected by the *context* in which it is used. For example, if you use a filter in a measure, and the measure is used in a PivotTable or PivotChart, the subset of data that is returned may be affected by additional filters or Slicers that the user has applied in the PivotTable. For more information about context, see Context in DAX Formulas.

Filters that Remove Duplicates

In addition to filtering for specific values, you can return a unique set of values from another table or column. This can be helpful when you want to count the number of unique values in a column, or use a list of unique values for other operations. DAX provides two functions for returning distinct values: DISTINCT Function (DAX) and VALUES Function (DAX).

- The DISTINCT function examines a single column that you specify as an argument to the function, and returns a new column containing just the distinct values.
- The VALUES function also returns a list of unique values, but also returns the *Unknown member*. This is useful when you use values from two tables that are joined by a relationship, and a value is missing in one table and present in the other. For more information about the Unknown member, see **Referential Integrity, PowerPivot Relationships, and the Unknown Member**.

Both of these functions return an entire column of values; therefore, you use the functions to get a list of values that is then passed to another function. For example, you could use the following formula to get a list of the distinct products sold by a particular reseller, using the unique product key, and then count the products in that list by using the COUNTROWS function:

Copy Code

```
=COUNTROWS(DISTINCT('ResellerSales_USD'[ProductKey]))
```

How Context Affects Filters

When you add a DAX formula to a PivotTable or PivotChart, the results of the formula can be affected by the *context*. If you are working in a PowerPivot table, the context is the current row and its values. If you are working in a PivotTable or PivotChart, the context means the set or subset of data that is defined by operations such as slicing, or filtering. The design of the PivotTable or PivotChart also imposes its own context. For example, if you create a PivotTable that groups sales by region and year, only the data that applies to those regions and years appears in the PivotTable. Therefore any measures that you add to the PivotTable are calculated in the context of the column and row headings plus any filters in the measure formula.

For more information, see the following topics:

- Context in DAX Formulas
- **Key Concepts in DAX**

Removing Filters

When working with complex formulas, you might want to know exactly what the current filters are, or you might want to modify the filter part of the formula. DAX provides several functions that enable you to remove filters, and to control which columns are retained as part of the current filter context. This section provides an overview of how these functions affect results in a formula.

Overriding All Filters with the ALL Function

You can use the `ALL` function to override any filters that were previously applied, and return all rows in the table to the function that is performing the aggregate or other operation. If you use one or more columns, instead of a table, as arguments to `ALL`, the `ALL` function returns all rows, ignoring any context filters.

Note

If you are familiar with relational database terminology, you can think of `ALL` as generating the natural left outer join of all the tables.

For example, suppose that you have the tables, Sales and Products, and you want to create a formula that will calculate the sum of sales for the current product divided by the sales for all products. You must take into consideration the fact that, if the formula is used in a measure, the user of the PivotTable might be using a Slicer to filter for a particular product, with the product name on the rows. Therefore, to get the true value of the denominator regardless of any filters or Slicers, you must add the ALL function to override any filters. The following formula is one example of how to use ALL to override the effects of previous filters:

Copy Code

```
=SUM (Sales[Amount])/SUMX(Sales[Amount], FILTER(Sales, ALL(Products)))
```

- The first part of the formula, `SUM (Sales[Amount])`, calculates the numerator.
- The sum takes into account the current context, meaning that if you add the formula into a calculated column, the row context is applied, and if you add the formula into a PivotTable as a measure, any filters applied in the PivotTable (the filter context) are applied.
- The second part of the formula, calculates the denominator. The ALL function overrides any filters that might be applied to the `Products` table.

For more information, including detailed examples, see ALL Function (DAX).

Overriding Specific Filters with the ALLEXCEPT Function

The ALLEXCEPT function also overrides existing filters, but you can specify that some of the existing filters should be preserved. The columns that you name as arguments to the ALLEXCEPT function specify which columns will continue to be filtered. If you want to override filters from most columns but not all, ALLEXCEPT is more convenient than ALL. The ALLEXCEPT function is particularly useful when you are creating PivotTables that might be filtered on many different columns, and you want to control the values that are used in the formula. For more information, including a detailed example of how to use ALLEXCEPT in a PivotTable, see ALLEXCEPT Function (DAX).

See Also

Other Resources

Filter and Sort Data

Filter Functions (DAX)

For the latest information about PowerPivot for Excel, visit the PowerPivot for Excel Online Help