

ALL

Article • 06/20/2022

Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied. This function is useful for clearing filters and creating calculations on all the rows in a table.

Syntax

DAX

```
ALL( [<table> | <column>[, <column>[, <column>[,...]]]] )
```

Parameters

Term	Definition
table	The table that you want to clear filters on.
column	The column that you want to clear filters on.

The argument to the ALL function must be either a reference to a base table or a reference to a base column. You cannot use table expressions or column expressions with the ALL function.

Return value

The table or column with filters removed.

Remarks

- This function is not used by itself, but serves as an intermediate function that can be used to change the set of results over which some other calculation is performed.
- The normal behavior for DAX expressions containing the ALL() function is that any filters applied will be ignored. However, there are some scenarios where this is not the case because of *auto-exist*, a DAX technology that optimizes filtering in order to

reduce the amount of processing required for certain DAX queries. An example where auto-exist and ALL() provide unexpected results is when filtering on two or more columns of the same table (like when using slicers), and there is a measure on that same table that uses ALL(). In this case, auto-exist will *merge* the multiple filters into one and will only filter on existing combinations of values. Because of this merge, the measure will be calculated on the existing combinations of values and the result will be based on filtered values instead of all values as expected. To learn more about auto-exist and its effect on calculations, see Microsoft MVP Alberto Ferrari's [Understanding DAX Auto-Exist](#) article on sql.bi.com.

- The following table describes how you can use the ALL and ALLEXCEPT functions in different scenarios.

Function and usage	Description
ALL()	Removes all filters everywhere. ALL() can only be used to clear filters but not to return a table.
ALL(Table)	Removes all filters from the specified table. In effect, ALL(Table) returns all of the values in the table, removing any filters from the context that otherwise might have been applied. This function is useful when you are working with many levels of grouping, and want to create a calculation that creates a ratio of an aggregated value to the total value. The first example demonstrates this scenario.
ALL (Column[, Column[, ...]])	Removes all filters from the specified columns in the table; all other filters on other columns in the table still apply. All column arguments must come from the same table. The ALL(Column) variant is useful when you want to remove the context filters for one or more specific columns and to keep all other context filters. The second and third examples demonstrate this scenario.
ALLEXCEPT(Table, Column1 [Column2]...)	Removes all context filters in the table except filters that are applied to the specified columns. This is a convenient shortcut for situations in which you want to remove the filters on many, but not all, columns in a table.

- This function is not supported for use in DirectQuery mode when used in calculated columns or row-level security (RLS) rules.

Example 1

Calculate ratio of Category Sales to Total Sales

Assume that you want to find the amount of sales for the current cell, in your PivotTable, divided by the total sales for all resellers. To ensure that the denominator is the same regardless of how the PivotTable user might be filtering or grouping the data, you define a formula that uses ALL to create the correct grand total.

The following table shows the results when a new measure, **All Reseller Sales Ratio**, is created using the formula shown in the code section. To see how this works, add the field, **CalendarYear**, to the **Row Labels** area of the PivotTable, and add the field, **ProductCategoryName**, to the **Column Labels** area. Then, drag the measure, **All Reseller Sales Ratio**, to the **Values** area of the Pivot Table. To view the results as percentages, use the formatting features of Excel to apply a percentage number formatting to the cells that contains the measure.

Row Labels	Accessories	Bikes	Clothing	Components	Grand Total
2005	0.02%	9.10%	0.04%	0.75%	9.91%
2006	0.11%	24.71%	0.60%	4.48%	29.90%
2007	0.36%	31.71%	1.07%	6.79%	39.93%
2008	0.20%	16.95%	0.48%	2.63%	20.26%
Grand Total	0.70%	82.47%	2.18%	14.65%	100.00%

Formula

DAX

```
= SUMX(ResellerSales_USD,  
ResellerSales_USD[SalesAmount_USD])/SUMX(ALL(ResellerSales_USD),  
ResellerSales_USD[SalesAmount_USD])
```

The formula is constructed as follows:

1. The numerator, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, is the sum of the values in `ResellerSales_USD[SalesAmount_USD]` for the current cell in the PivotTable, with context filters applied on `CalendarYear` and `ProductCategoryName`.
2. For the denominator, you start by specifying a table, `ResellerSales_USD`, and use the `ALL` function to remove all context filters on the table.

3. You then use the SUMX function to sum the values in the ResellerSales_USD[SalesAmount_USD] column. In other words, you get the sum of ResellerSales_USD[SalesAmount_USD] for all resellers sales.

Example 2

Calculate Ratio of Product Sales to Total Sales Through Current Year

Assume that you want to create a table showing the percentage of sales compared over the years for each product category (ProductCategoryName). To obtain the percentage for each year over each value of ProductCategoryName, you need to divide the sum of sales for that particular year and product category by the sum of sales for the same product category over all years. In other words, you want to keep the filter on ProductCategoryName but remove the filter on the year when calculating the denominator of the percentage.

The following table shows the results when a new measure, **Reseller Sales Year**, is created using the formula shown in the code section. To see how this works, add the field, CalendarYear, to the **Row Labels** area of a PivotTable, and add the field, ProductCategoryName, to the **Column Labels** area. To view the results as percentages, use Excel's formatting features to apply a percentage number format to the cells containing the measure, **Reseller Sales Year**.

Row labels	Accessories	Bikes	Clothing	Components	Grand Total
2005	3.48%	11.03%	1.91%	5.12%	9.91%
2006	16.21%	29.96%	27.29%	30.59%	29.90%
2007	51.62%	38.45%	48.86%	46.36%	39.93%
2008	28.69%	20.56%	21.95%	17.92%	20.26%
Grand Total	100.00%	100.00%	100.00%	100.00%	100.00%

Formula

DAX

```
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE( SUM(ResellerSales_USD[SalesAmount_USD]), ALL(DateTime[CalendarYear]))
```

The formula is constructed as follows:

1. The numerator, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, is the sum of the values in `ResellerSales_USD[SalesAmount_USD]` for the current cell in the pivot table, with context filters applied on the columns `CalendarYear` and `ProductCategoryName`.
2. For the denominator, you remove the existing filter on `CalendarYear` by using the `ALL(Column)` function. This calculates the sum over the remaining rows on the `ResellerSales_USD` table, after applying the existing context filters from the column labels. The net effect is that for the denominator the sum is calculated over the selected `ProductCategoryName` (the implied context filter) and for all values in `Year`.

Example 3

Calculate Contribution of Product Categories to Total Sales Per Year

Assume that you want to create a table that shows the percentage of sales for each product category, on a year-by-year basis. To obtain the percentage for each product category in a particular year, you need to calculate the sum of sales for that particular product category (`ProductCategoryName`) in year *n*, and then divide the resulting value by the sum of sales for the year *n* over all product categories. In other words, you want to keep the filter on year but remove the filter on `ProductCategoryName` when calculating the denominator of the percentage.

The following table shows the results when a new measure, **Reseller Sales CategoryName**, is created using the formula shown in the code section. To see how this works, add the field, `CalendarYear` to the **Row Labels** area of the PivotTable, and add the field, `ProductCategoryName`, to the **Column Labels** area. Then add the new measure to the **Values** area of the PivotTable. To view the results as percentages, use Excel's formatting features to apply a percentage number format to the cells that contain the new measure, **Reseller Sales CategoryName**.

Row Labels	Accessories	Bikes	Clothing	Components	Grand Total
2005	0.25%	91.76%	0.42%	7.57%	100.00%
2006	0.38%	82.64%	1.99%	14.99%	100.00%
2007	0.90%	79.42%	2.67%	17.01%	100.00%

Row Labels	Accessories	Bikes	Clothing	Components	Grand Total
2008	0.99%	83.69%	2.37%	12.96%	100.00%
Grand Total	0.70%	82.47%	2.18%	14.65%	100.00%

Formula

DAX

```
= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE( SUM(  
ResellerSales_USD[SalesAmount_USD]),  
ALL(ProductCategory[ProductCategoryName]))
```

The formula is constructed as follows:

1. The numerator, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, is the sum of the values in `ResellerSales_USD[SalesAmount_USD]` for the current cell in the PivotTable, with context filters applied on the fields, `CalendarYear` and `ProductCategoryName`.
2. For the denominator, you use the function, `ALL(Column)`, to remove the filter on `ProductCategoryName` and calculate the sum over the remaining rows on the `ResellerSales_USD` table, after applying the existing context filters from the row labels. The net effect is that, for the denominator, the sum is calculated over the selected Year (the implied context filter) and for all values of `ProductCategoryName`.

See also

[Filter functions](#)

[ALL function](#)

[ALLEXCEPT function](#)

[FILTER function](#)