

# CREATING COMPLEX CALCULATED **COLUMNS** WITH DAX

Swipe →

## **INTRODUCTION**

Data Analysis Expressions (DAX) is a powerful formula language used in Power BI, Excel Power Pivot, and Analysis Services Tabular models. One of its key features is the ability to create complex calculated columns that can transform and enrich your data. In this blog post, we'll explore how to leverage DAX to create sophisticated calculated columns, using a sample dataset and focusing on common types of calculations.

## **UNDERSTANDING CALCULATED COLUMNS**

Calculated columns are columns that you add to an existing table in your data model. Unlike measures, which are calculated on the fly, calculated columns are computed when the model is processed and stored in the model. This makes them ideal for row-by-row calculations that don't change based on the filter context.

## **COMMON CALCULATED COLUMNS AND EXAMPLES**

Let's explore some of the most common types of calculated columns and how to implement them.

## 1. Basic Arithmetic

**Example:** Calculate the profit for each sale, assuming a 40% profit margin.

**DAX Formula:**

*Profit = [TotalSales] \* 0.4*

Structure	Formatting	Properties	Sort	Groups
✓	1	Profit = [TotalSales] * 0.4		

OrderID	Date	Profit
1001	1/5/2022	40.00
1002	1/10/2022	100.00
1003	1/15/2022	16.00
1004	1/20/2022	54.00
1005	1/25/2022	48.00
1006	2/1/2022	20.00
1007	2/5/2022	40.00
1008	2/10/2022	20.00
1009	2/15/2022	16.00
1010	2/20/2022	72.00

This simple calculation multiplies the TotalSales by 0.4 to get the profit.



## 2. Date Manipulation

**Example:** Extract the month name from the Date column.

**DAX Formula:**

*Month Name = FORMAT([Date], "mmmm")*

Structure	Formatting	Properties	Sort	Groups
✓	1 Month Name = FORMAT([Date], "mmmm")			

OrderID	Date	Month Name
1001	1/5/2022	January
1002	1/10/2022	January
1003	1/15/2022	January
1004	1/20/2022	January
1005	1/25/2022	January
1006	2/1/2022	February
1007	2/5/2022	February
1008	2/10/2022	February

The FORMAT function is used here to convert the date into a month name. The "mmmm" argument specifies that we want the full month name.

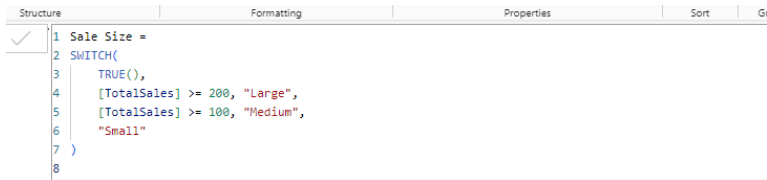


### 3. Conditional Statements

**Example:** Categorize sales into size brackets.

**DAX Formula:**

*Sale Size = SWITCH( TRUE(), [TotalSales] >= 200, "Large", [TotalSales] >= 100, "Medium", "Small")*



```
1 Sale Size =  
2 SWITCH(  
3     TRUE(),  
4     [TotalSales] >= 200, "Large",  
5     [TotalSales] >= 100, "Medium",  
6     "Small"  
7 )  
8
```

OrderID	Date	Sale Size
1001	1/5/2022	Medium
1002	1/10/2022	Large
1003	1/15/2022	Small
1004	1/20/2022	Medium
1005	1/25/2022	Medium
1006	2/1/2022	Small

The SWITCH function is used here to create categories based on the TotalSales value. It evaluates each condition in order and returns the corresponding result for the first TRUE condition.

## 4. Text Manipulation

**Example:** Create a product code by combining the first three letters of the category with the product name.

**DAX Formula:**

*Product Code = LEFT([Category], 3) & "-" & [Product]*



OrderID	Date	Product Code
1001	1/5/2022	Ele-Laptop
1002	1/10/2022	Clo-T-Shirt
1003	1/15/2022	Hom-Blender
1004	1/20/2022	Ele-Smartphone
1005	1/25/2022	Hom-Coffee Maker
1006	2/1/2022	Clo-Jeans
1007	2/5/2022	Ele-Wireless Earbuds
1008	2/10/2022	Hom-Vacuum Clean

This formula uses the LEFT function to extract the first three characters of the Category, then uses the & operator to concatenate this with a hyphen and the Product name.

## 5. Ranking

**Example:** Rank products by their total sales within each category.

### DAX Formula:

*Sales Rank* = RANKX( FILTER(ALL(Sales),Sales[Category] = EARLIER(Sales[Category])),  
[TotalSales], , DESC)

```
Structure      Formatting      Properties
1 Sales Rank =
2 RANKX(
3   FILTER(
4     ALL(Sales),
5     Sales[Category] = EARLIER(Sales[Category])
6   ),
7   [TotalSales],
8   ,
9   DESC
10 )
```

This complex formula uses several functions:

- **RANKX** to perform the ranking
- **FILTER** to create a filtered table of all sales in the same category
- **ALL** to remove any existing filters on the Sales table
- **EARLIER** to refer to the Category of the current row

OrderID	Date	Sales Rank
1001	1/5/2022	9
1002	1/10/2022	2
1003	1/15/2022	25
1004	1/20/2022	5
1005	1/25/2022	1
1006	2/1/2022	17

## 6. Running Totals

**Example:** Calculate a running total of sales by date.

**DAX Formula:**

*Running Total Sales = CALCULATE(SUM(Sales[TotalSales]),FILTER(ALL(Sales), Sales[Date] <= EARLIER(Sales[Date])))*

Structure	Formatting	Properties	Sort
✓	<pre>1 Running Total Sales = 2 CALCULATE( 3     SUM(Sales[TotalSales]), 4     FILTER( 5         ALL(Sales), 6         Sales[Date] &lt;= EARLIER(Sales[Date]) 7     ) 8 ) 9</pre>		

OrderID	Date	Running Total Sales
1001	1/5/2022	170
1002	1/10/2022	460
1003	1/15/2022	605
1004	1/20/2022	765
1005	1/25/2022	955
1006	2/1/2022	1040
1007	2/5/2022	1196

This formula uses CALCULATE to change the context of the SUM function. The FILTER function creates a table of all sales up to and including the current date, which is then summed.



## 7. Time Intelligence

**Example:** calculate the year-to-date sales.

**DAX Formula:**

*YTD Sales = CALCULATE( SUM(Sales[TotalSales]), DATESYTD(Sales[Date]) )*

Structure	Formatting	Properties	Sort	Groups
✓	1	YTD Sales = CALCULATE( SUM(Sales[TotalSales]), DATESYTD(Sales[Date]) )		

OrderID	Date	YTD Sales
1001	1/5/2022	100
1002	1/10/2022	250
1003	1/15/2022	40
1004	1/20/2022	135
1005	1/25/2022	120
1006	2/1/2022	50
1007	2/5/2022	100
1008	2/10/2022	50
1009	2/15/2022	40

The CALCULATE function is used to modify the context in which the SUM of TotalSales is calculated. DATESYTD is used to filter the dates from the beginning of the year to the current row's date.

## 8. Profit Margin

**Example:** Calculate the profit margin for each sale, assuming a fixed cost of 60% of the unit price.

**DAX Formula:**

*Profit Margin = DIVIDE( [TotalSales] - ([Quantity] \* [UnitPrice] \* 0.6), [TotalSales], 0 )*

Structure	Formatting	Properties	Sort
✓	<pre>1 Profit Margin = 2 DIVIDE( 3     [TotalSales] - ([Quantity] * [UnitPrice] * 0.6), 4     [TotalSales], 5     0 6 )</pre>		

OrderID	Date	Profit Margin
1023	5/1/2022	0.40
1024	5/5/2022	0.40
1025	5/10/2022	0.40
1026	5/15/2022	0.40
1027	5/20/2022	0.40
1028	5/25/2022	0.40
1029	6/1/2022	0.40
1030	6/5/2022	0.40
1031	6/15/2022	0.40

This formula calculates the profit margin as a percentage. It subtracts the cost (60% of total sales) from the total sales and divides by the total sales. The DIVIDE function is used to handle potential division by zero errors.

## 9. Concatenation

**Example:** Create a full product name by combining the category and product name.

**DAX Formula:**

*Full Product Name = [Category] & " - " & [Product]*

Structure	Formatting	Properties	Sort
✓	1 Full Product Name = [Category] & " - " & [Product]		

OrderID	Date	Category	Product	Full Product Name
1001	1/5/2022	Electronics	Laptop	Electronics - Laptop
1002	1/10/2022	Clothing	T-Shirt	Clothing - T-Shirt
1003	1/15/2022	Home	Blender	Home - Blender
1004	1/20/2022	Electronics	Smartphone	Electronics - Smartphone
1005	1/25/2022	Home	Coffee Maker	Home - Coffee Maker
1006	2/1/2022	Clothing	Jeans	Clothing - Jeans
1007	2/5/2022	Electronics	Wireless Earbuds	Electronics - Wireless Earbuds
1008	2/10/2022	Home	Vacuum Cleaner	Home - Vacuum Cleaner
1009	2/15/2022	Electronics	Smartwatch	Electronics - Smartwatch
1010	2/20/2022	Clothing	Dress Shirt	Clothing - Dress Shirt

This simple formula uses the & operator to concatenate the Category and Product columns with a separator.



## 10. Percentages of Total

**Example:** Calculate the percentage of total sales for each product.

**DAX Formula:**

*Percent of Total Sales = DIVIDE([TotalSales], CALCULATE(SUM(Sales[TotalSales]), ALL(Sales)))*

Structure	Formatting	Properties	Sort
✓			
1 Percent of Total Sales =			
2 DIVIDE(			
3 [TotalSales],			
4 CALCULATE(			
5 SUM(Sales[TotalSales]),			
6 ALL(Sales)			
7 )			
8 )			

OrderID	Date	Percent of Total Sales
1001	1/5/2022	1.44%
1002	1/10/2022	3.60%
1003	1/15/2022	0.58%
1004	1/20/2022	1.94%
1005	1/25/2022	1.73%
1006	2/1/2022	0.72%
1007	2/5/2022	1.44%
1008	2/10/2022	0.72%

This formula uses DIVIDE to safely perform division (avoiding divide-by-zero errors). CALCULATE and ALL are used to compute the total sales across all rows, regardless of any filters applied.

## CONCLUSION

These examples demonstrate the power and flexibility of DAX in creating complex calculated columns. By mastering these techniques and understanding the various DAX functions (like CALCULATE, FILTER, ALL, EARLIER, RANKX, DATEADD, AVERAGEX, DATESINPERIOD, DIVIDE), you can significantly enhance your data model's capabilities, making it easier to analyze and visualize your data effectively.

Remember, while calculated columns are powerful, they increase the size of your data model and are recalculated only when the model is refreshed. For calculations that need to respond dynamically to user interactions, consider using measures instead.

As you continue to work with DAX, you'll discover even more ways to leverage its capabilities for advanced data analysis and reporting. The key is to understand your data and the insights you want to derive from it, then use DAX to create the calculations that will help you achieve those insights.

---

# WAS THIS HELPFUL?

---

**DON'T FORGET TO  
SAVE THIS POST**



[WWW.FP20ANALYTICS.COM](http://WWW.FP20ANALYTICS.COM)