

# Introduction to Advanced Editor

DATA TRANSFORMATION IN POWER BI



**Maarten Van den Broeck**

Content Developer at DataCamp

# What is the advanced editor?

- The advanced editor allows you to view and edit the underlying code of your query
- Any transformation you make to your data is translated into M code (also referred to as M Language) - the language of Power Query
- DAX code allows you to analyze your data, M code allows you to transform and load it

Query2

Display Options ▾



```
let
    Source = OData.Feed("https://services.odata.org/V4/Northwind/Northwind.svc/", null, [Implementation="2.0"]),
    Alphabetical_list_of_products_table = Source{[Name="Alphabetical_list_of_products",Signature="table"]}[Data],
    #"Removed Other Columns" = Table.SelectColumns(Alphabetical_list_of_products_table,{"ProductID", "ProductName", "CategoryID", "CategoryName"},
    #"Sorted Rows" = Table.Sort(#"Removed Other Columns",{{"UnitsInStock", Order.Descending}}),
    #"Added Index" = Table.AddIndexColumn(#"Sorted Rows", "OverallStockRank", 1, 1, Int64.Type),
    #"Grouped Rows" = Table.Group(#"Added Index", {"CategoryID", "CategoryName"}, {{"CategoryAverageStock", each List.Average([UnitsInStock])}, {"CategoryTotalStock", each List.Sum([UnitsInStock])}},
    #"Expanded Grouped" = Table.ExpandTableColumn(#"Grouped Rows", "Grouped", {"ProductID", "ProductName", "UnitsInStock"}, {"ProductID", "ProductName", "UnitsInStock", "OverallStockRank", "CategoryAverageStock", "CategoryTotalStock"}),
    #"Added Custom" = Table.AddColumn(#"Expanded Grouped", "StockPercentDeviation", each [UnitsInStock] / [CategoryAverageStock]),
    #"Sorted Rows1" = Table.Sort(#"Added Custom",{{"StockPercentDeviation", Order.Ascending}}),
    #"Filtered Rows" = Table.SelectRows(#"Sorted Rows1", each [StockPercentDeviation] < 0.1)
in
    #"Filtered Rows"
```

## APPLIED STEPS

- Source ✖
- Navigation ✖
- Removed Other Columns ✖
- Sorted Rows ✖
- Added Index ✖
- Grouped Rows ✖
- Expanded Grouped ✖
- Added Custom ✖
- Sorted Rows1 ✖
- ✖ Filtered Rows ✖

# The difference between M code and DAX

## DAX Code

*Power BI*

- Data Analysis eXpressions
- Used to create metrics and analyze data
- Create calculations without changing data
- Filters and summarizes columns
- Not case-sensitive

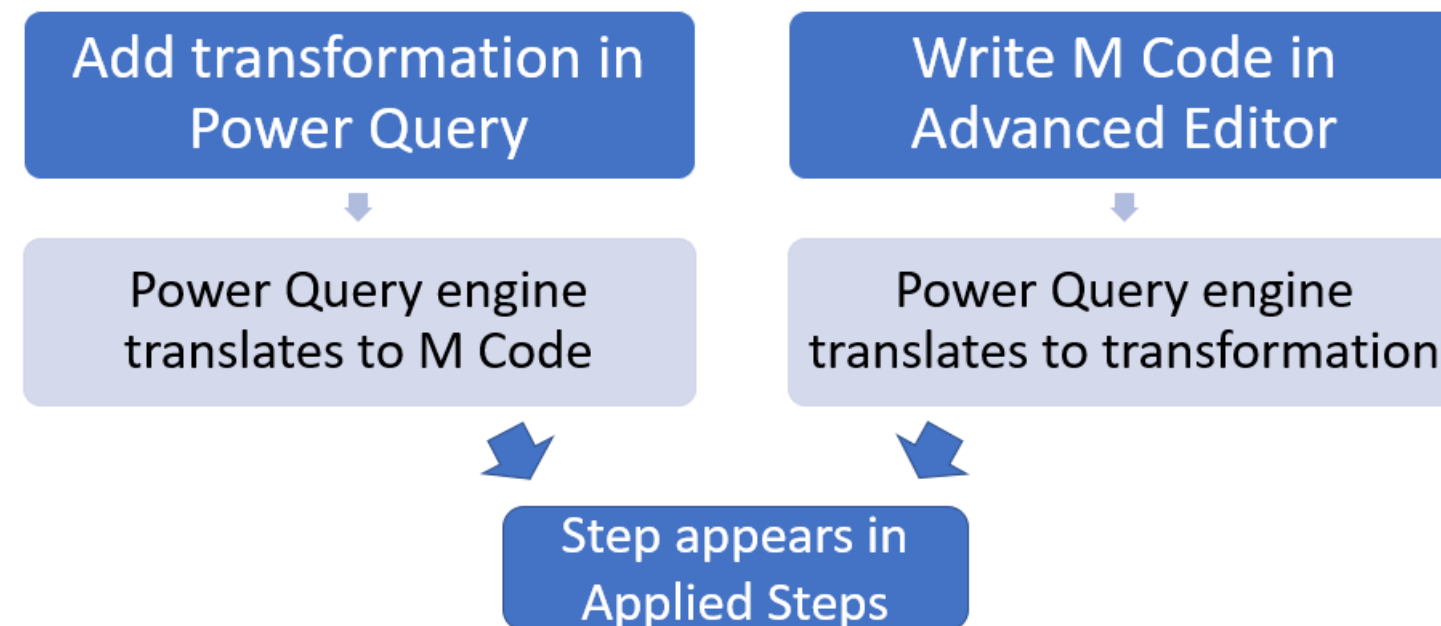
## M Language

*Power Query*

- Data Mashup
- Used to load, transform, and create data
- Remove or rename columns
- Pivot, unpivot, transpose, group your data
- Case-sensitive

# M Language and the applied steps

- Any transformation you make is automatically converted to the appropriate M code, and vice versa
- M code and Power Query transformations share a 1:1 relationship



# Writing M language

- Write your own M language allows you to:
  - Store variables for use in the query
  - Implement custom functions
  - Make advanced transformations
  - Add comments with `//`

# Intro to M language

Value types in M language:

- Number = 123
- Text = "DataCamp"
- Logical = true
- Date = 1/20/2022
- List = {123, "DataCamp", true}
- Table =

```
#table( {"Index", "Value"}, {"A",  
123}, {"B", "DataCamp"}, {"C", true})
```

Table created by example code:

	ABC 123 Index	ABC 123 Value
1	A	123
2	B	DataCamp
3	C	TRUE

# Let's practice!

DATA TRANSFORMATION IN POWER BI

# M Language and Advanced Editor demo

DATA TRANSFORMATION IN POWER BI



**Maarten Van den Broeck**  
Content Developer at DataCamp



# Let's practice!

DATA TRANSFORMATION IN POWER BI

# Congratulations!

DATA TRANSFORMATION IN POWER BI



**Maarten Van den Broeck**

Content Developer at DataCamp

# Here's what you learned

## Chapter 1

- Reshaping and aggregating data
- (Un)pivoting tables

## Chapter 3

- Custom columns

## Chapter 2

- Appending data
- Merging data

## Chapter 4

- Advanced Editor
- M language

**See you in the next  
course!**

**DATA TRANSFORMATION IN POWER BI**