

Five Strategies to Enhance Your DAX (or M) Toolbox

Brian Julius
@BrianJ on eDNA Forum
July 2020





Enterprise DNA Forum

5 Strategies to Enhance Your DAX (or M) Toolbox



ambepat

May 19

Hi @Melissa thanks for the explanation.

What will you do to achieve the same result just by using Calculate and not Sumx? What would you do differently.

Also I have been learning DAX now for almost a year and I am now transitioning to understanding the engine and asking why? It is a bit challenging given i sometimes try to read articles from Alberto and Marco in SQLBI. Those articles sometimes are really tough to understand but I am just keeping at it to push myself to continue to learn and grow.

Can I just ask how many years have you been working with DAX as you seem like you really know the fine details into the language and what are some tips you could advise on how to improve daily?

Thanks again for your explanations. Really helpful.

   Reply



← ALBERTO'S BLOG



7 reasons DAX is not easy

Posted on Jun 20, 2020

DAX is simple, but it is not easy. I cannot remember how many times I read this sentence on social media, attributed to me. For good reason: I have repeated this sentence at almost every classroom course I delivered over the years, at conferences, user group meetings... it is my mantra.

The sentence itself is not mine. It came as the answer to a question I routinely ask my students at the end of a training: "What do you think of DAX now?". The course was in Dublin, around 10 years ago, and a student answered that. I stole his sentence, and it became a mantra for both



Alberto Ferrari

[PROFILE](#) [BLOG](#)

BLOG ARCHIVES

- 2020 (1), 2019 (1), 2018 (2), 2012 (7), 2011 (31), 2010 (6), 2009 (7), 2008 (4), 2007 (12), 2006 (6)



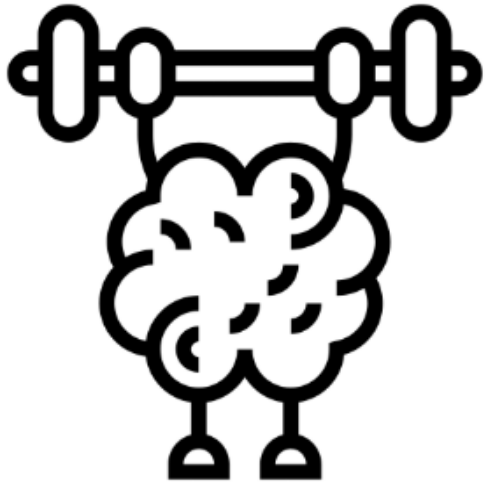
Enterprise DNA Forum

5 Strategies to Enhance Your DAX (or M) Toolbox

Reason #2: Practice makes perfect

DAX is our job: we spend all day writing and teaching DAX. All day, every day. **This is how you become an expert: by practicing.** With DAX, that means practicing a lot!

When it comes to Power BI, most users do not spend their entire time preparing reports day in, day out. This makes perfect sense: a report is meant to produce numbers, and users are expected to reason off of those numbers, not just produce them. Any regular user would need DAX when preparing a report, but this is a tiny fraction of that person's daily job. So what happens is that you have a few days to prepare a report, and then weeks, maybe months go by during which you do not write



5 Strategies to Enhance Your DAX (or M) Toolbox

STRATEGY #1: Organize your tools = Format Your DAX !!!

- wheniseeunformatteddaxitlookslikesomeoneiswritingwithnocapitalizationpunctuationorspacingicantevenbegin to figure out what the dax function is doing or not doing because icant read or organize it in my mind once you get used to looking at formatted dax the unformatted version will just look like gibberish to you as marco russo says frequently its not dax unless its formatted its the simplest thing on this list that you can do but probably the most beneficial and the best thing is it costs nothing and will actually end up saving you a lot of time
- When I see unformatted DAX, it looks like someone is writing with no capitalization, punctuation or spacing. I can't even begin to figure out what the DAX function is doing (or not doing) because I can't read or organize it in my mind. Once you get used to looking at formatted DAX, the unformatted version will just look like gibberish to you. As Marco Russo says frequently, "it's not DAX unless it's formatted". It's the simplest thing on this list that you can do, but probably the most beneficial, and the best thing is it costs nothing, and will actually end up saving you lots of time.

5 Strategies to Enhance Your DAX (or M) Toolbox

Unformatted DAX

```
Sales LY by Rank =
CALCULATE( [Sales LY],
FILTER( VALUES( PBI_AEP_CustMaster135[CUSTOMER NAME -
CODE]),
COUNTRROWS(
FILTER( 'Customer Rank Group',
RANKX( ALL( PBI_AEP_CustMaster135[CUSTOMER NAME - CODE] ),
[Avg Total Sales], DESC ) >= 'Customer Rank Group'[Min]
&& RANKX( ALL( PBI_AEP_CustMaster135[CUSTOMER NAME - CODE]
), [Avg Total Sales], DESC ) <= 'Customer Rank Group'[Max] )
)
)
```

Formatted DAX

```
Sales LY by Rank =
CALCULATE(
[Sales LY],
FILTER(
VALUES( PBI_AEP_CustMaster135[CUSTOMER NAME - CODE] ),
COUNTRROWS(
FILTER(
'Customer Rank Group',
RANKX(
ALL( PBI_AEP_CustMaster135[CUSTOMER NAME - CODE] ),
[Avg Total Sales],
DESC
) >='Customer Rank Group'[Min] &&
RANKX(
ALL( PBI_AEP_CustMaster135[CUSTOMER NAME - CODE] ),
[Avg Total Sales],
DESC
) <='Customer Rank Group'[Max]
)
)
)
)
```

5 Strategies to Enhance Your DAX (or M) Toolbox

<https://analysthub.enterprisedna.co/dax-clean-up>

The screenshot shows the 'DAX CLEAN UP' web application. At the top left, there is a logo with the text 'DAX CLEAN UP' and a sub-header 'Cleanup and format your DAX code.' To the right of this is a 'TRAINING LOG IN' button. Below the header, the main heading reads 'Make your DAX code easy to read.' Underneath this is a large text input field with the placeholder text 'Paste your DAX Code HERE'. On the right side of the interface, there is a 'SETTINGS' panel. This panel is divided into three sections: 'SEPARATORS', 'SPACING', and 'LINES'. Each section contains several toggle switches and radio buttons to configure the formatting of DAX code. The 'SEPARATORS' section includes options for 'Automatic', 'US/UK', and 'Other'. The 'SPACING' section includes options for 'Best practice', 'Space after function', and 'No space after function'. The 'LINES' section includes options for 'Long lines' and 'Short lines'.

5 Strategies to Enhance Your DAX (or M) Toolbox

Theme Light Dark

```
1 Let
2 Source = DateTime.FixedLocalNow(),
3 #"Converted to Table" = #table(1, {{Source}}),
4 #"Renamed Columns" = Table.RenameColumns(#"Converted to Table", {"Column1", "Last Refresh"}),
5 #"Changed Type" = Table.TransformColumnTypes(#"Renamed Columns", {"Last Refresh", type datetime}),
6 #"Inserted Date" = Table.AddColumn(#"Changed Type", "Date", each DateTime.Date([Last Refresh]), type date),
7 #"Insert Time" = Table.AddColumn(#"Inserted Date", "Time", each DateTime.Time([Last Refresh]), type time),
8 #"Added Custom" = Table.AddColumn(#"Insert Time", "Local Timezone", each DateTime.AddZone( [Last Refresh], 4), type datetimetimezone )
9 in
10 #"Added Custom"
11
12
13
14
```

Power Query (M-Language) Code with Power Query Formatter
<https://powerqueryformatter.com/>

I will store your preferred options in your browser and remember them when you revisit. [Reset to default values.](#)

Line width

Indentation

Spaces per Indentation

Number of brackets

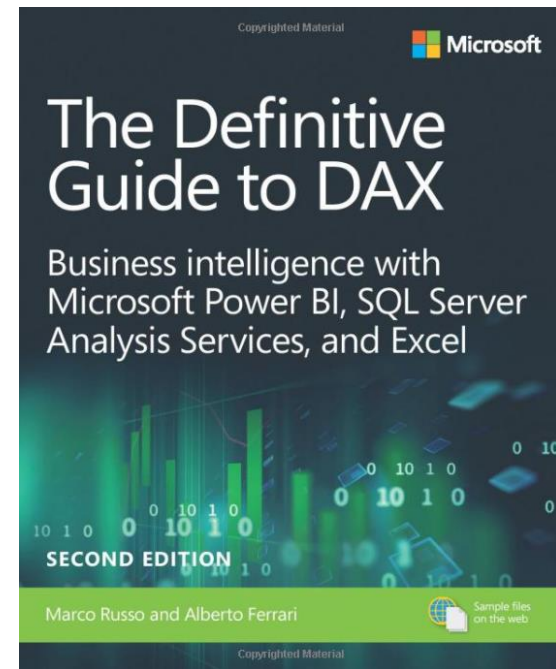
Include Comments

STRATEGY #2: Acquire New Tools = Familiarize Yourself w/ New Functions

Every day or so, familiarize yourself with a new DAX (and/or M) function

- “Familiarize”, **not** “Master”
- What does it do? How to use it? What it returns? (e.g. table, numeric value, boolean value, date, etc.)
- Goal is **recognition**
 - *“I’ve seen a function that handles this exact situation”*
- eDNA Knowledge Base is a great resource for this – in addition to all key info, provides links to forum posts and short videos that show function used in detailed real-world scenarios
- Prioritize virtual table/filter functions

Every serious DAX user needs a copy of “The Bible”



ENTERPRISE DNA

DAX Function of the Day

#	Function	Type
19	UTCTODAY	Date and time

M Function of the Day

#	Function	Type
512	Table.AddIndexColumn	Table

DAX and M function lists correspond to those available as of the June 2020 Power Bi release. Last Refresh: 6/28/2020 5:20:50 PM

- Provides a new random pair every time you reload or hit Refresh
- DAX functions hyperlinked to Enterprise DNA Knowledgebase
- M functions linked to Microsoft M Reference Guide
- Free to download at link in YouTube comments below



UTCTODAY Function (DAX)

Sen McKay



How does the UTCTODAY function (DAX) work?

The UTCTODAY function (DAX) returns the current UTC date.

UTCTODAY Formula Syntax

```
UTCTODAY(
)
```

How do you use the UTCTODAY function?

This function returns the time value 12:00:00 PM for all dates.

Related Blog Posts



Related Support Forum Posts

- Dax Calculation question
- Dax not working
- Malsj/Blanks DAX Measure

Considerations when using the UTCTODAY function

The result of the UTCTODAY function changes only when the column that contains the formula is refreshed. It is not updated continuously.

Related Video Tutorials



Table.AddIndexColumn

06/15/2020 • 2 minutes to read • 👤 👤 👤 👤

Syntax

```
Table.AddIndexColumn(table as table, newColumnName as text, optional initialValue as nullable number, optional increment as nullable number)
```

About

Appends a column named `newColumnName` to the `table` with explicit position values. An optional value, `initialValue`, the initial index value. An optional value, `increment`, specifies how much to increment each index value.

Example 1

Add an index column named "Index" to the table.

```
powerquery-m Copy
Table.AddIndexColumn(
    Table.FromRecords({
        [CustomerID = 1, Name = "Bob", Phone = "123-4567"],
        [CustomerID = 2, Name = "Jim", Phone = "987-6543"],
        [CustomerID = 3, Name = "Paul", Phone = "543-7890"],
        [CustomerID = 4, Name = "Ringo", Phone = "232-1550"]
    })
)
```

5 Strategies to Enhance Your DAX (or M) Toolbox

STRATEGY #3: Borrow Tools from Others = Utilize Existing Code/Patterns

- Pattern = “reusable solution to a commonly occurring problem”
- Some established sources:
 - [Daxpatterns.com \(SQLBI\)](http://Daxpatterns.com)
 - [Quick Measures Gallery \(Microsoft\)](#)
- Two new ones:
 - [DAX Patterns forum category \(Enterprise DNA\)](#)
 - [M Code Showcase \(Enterprise DNA\)](#)
- Major caveat:
 - Don’t use code you don’t fully understand, even if it seems to “work”
 - Conversely, presents a great learning opportunity

Don't borrow this:



If you only know how to operate this:



5 Strategies to Enhance Your DAX (or M) Toolbox

STRATEGY #4: Regularly Build Something New

THE PROBLEM

- For most of us, the DAX and M we do for work supports a limited number of reports and analyses
- After a burst of initial learning, we can become stagnant

THE SOLUTION

- Regularly build something new that challenges you to expand your PBI skillset
- User forums are a never-ending source of interesting and challenging problems at all skill levels
 - Data Challenges are extremely useful in this regard too



ENTERPRISE DNA Percentage Contribution to Total Sales
 DAX Calculations

D diego

Hi @upwardD

Thanks for sharing, your posts are really interesting to learn

Attached the solution.

1. Your summarized table does not work because is isolated, so I used the model tables.
2. It was missing a CALCULATE to compute the ranking,

Customer Area	Transaction	Rank	Sales QTYs	% over Total
England	55424	1	31,64 %	
Product 5	3458	1	1,97 %	
Product 12	3406	2	1,94 %	
Product 9	3125	3	1,78 %	
Product 20	3113	4	1,78 %	
Product 13	2881	5	1,65 %	

Year: 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020

5 Strategies to Enhance Your DAX (or M) Toolbox

THE STRATEGY “Deliberate Practice”

Ericsson, K. A., Krampe, R. T., and Tesch-Römer, C. (1993). [The role of deliberate practice in the acquisition of expert performance](#). *Psychol. Rev.* 100, 363–406.

Has been applied to fields of surgery, acting, chess, writing, **computer programming**, ballet, music, aviation, firefighting, and many others.

“Consistently and overwhelmingly, the evidence showed that experts are always made, not born.”

K. Anders Ericsson, [The Cambridge Handbook of Expertise and Expert Performance](#), 2006.

Characteristics of Deliberate Practice

- Designed to continually stretch an individual just beyond his or her current abilities
- High repetition
- Continual feedback on results
- Highly mentally demanding/difficult
- Goals are based on the process, not the outcome

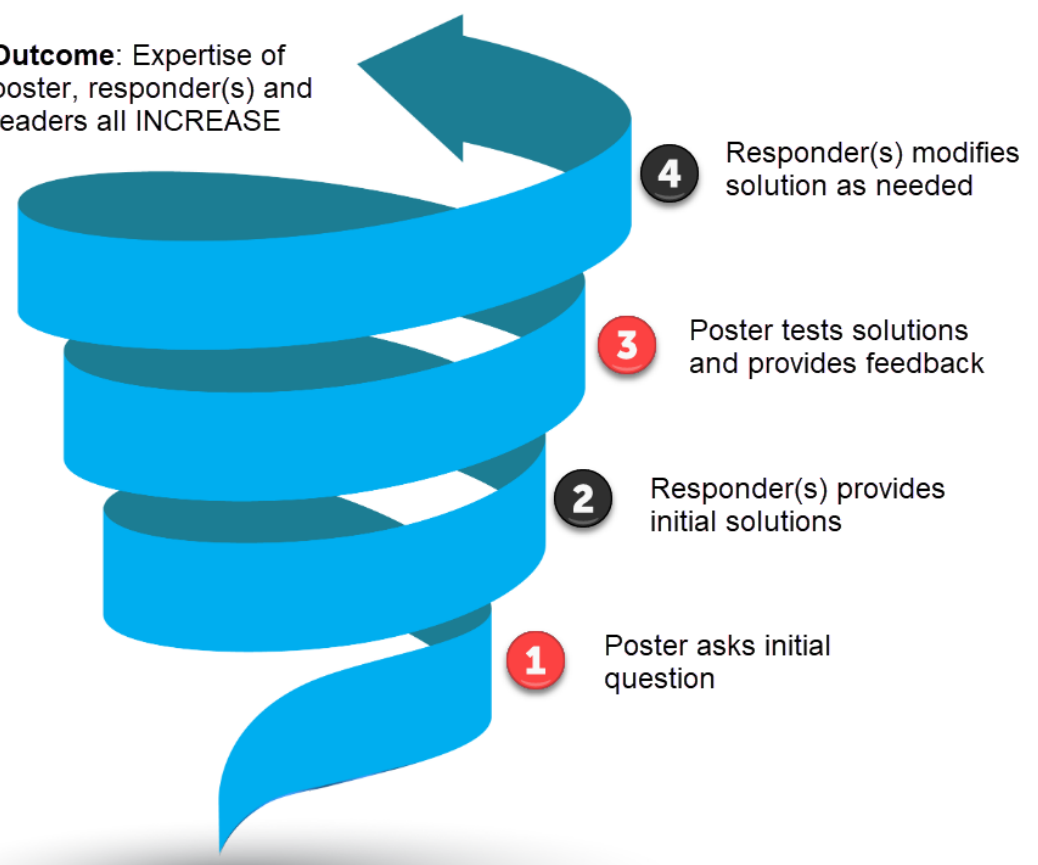
Example Goals (mine)

- Answer at least 1 question per day that is difficult **for me**
- Strive to maximize PQ/M responses
- At least 1x/week tackle a problem well beyond my current capabilities (“*swing for the fences*”)

5 Strategies to Enhance Your DAX (or M) Toolbox

The “Upward Spiral” of Community Problem Solving

Outcome: Expertise of poster, responder(s) and readers all INCREASE



RESPONSIBILITIES

- *Poster*
 - Provide clear explanation of problem, and desired outcome (w/ mockup and PBIX file)
 - Anonymize data if necessary
 - Carefully review and test solutions provided, and provide feedback
 - Make sure solution is well understood
- *Responder(s)*
 - Provide clear explanation of solution, and full solution file if possible
 - Modify solution as needed in response to feedback



Replies to Commonly Heard Responses

1. *"I'm still learning Power BI"*

We have created several articles about **ALLSELECTED** and, unfortunately, we have never been completely clear about its behavior. The reason is very simple. We fell into one of the many traps of **ALLSELECTED**: we believed we understood its behavior, but we were just so close – not there yet.

- Marco Russo and Alberto Ferrari, ["The Definitive Guide to ALLSELECTED"](#)

2. *"I don't know enough to be solving other users' problems"*

Problems are posted at ALL levels, from raw beginner to expert. Pick those that are a slight stretch for YOU.

3. *"What if my solution is bad/wrong/incomplete?"*

Hi, and welcome to the club! Some absolutely will be. Others will correct/build on your effort – these are some of the best learning experiences. Also a great way to gauge your progress over time.

5 Strategies to Enhance Your DAX (or M) Toolbox

STRATEGY #5: Upgrade the Toolbox Itself = Use a Better DAX/M Editor(s)

No single perfect solution...yet

Option	Cost	Best Use
Native PBI Editor	Free	As motivation to find a better editor
Native Editor w/ Keyboard shortcuts	Free	Basic DAX and M coding
DAX Studio	Free	Complex DAX measures (plus many other valuable features beyond editing)
Microsoft Visual Studio Code	Free	Managing/using code snippets