

Demystifying DAX in **Power BI**

Khilitchandra Prajapati

PASS Chapter, Singapore

www.bigator.com

Vice President, BI & Data Analytics – UOB, Singapore



Agenda

- Overview of Power BI
- Introduction to DAX
- Functions & Capabilities
- Demo & QA (Tips & Tricks)



Overview of Power BI

- Software As A Service (SaaS)
 - Central Reporting under 1 Umbrella
 - Supports Self-Service BI + Paginated Reports
 - Seamless integration (Keep expanding horizons)
 - Available any Platform, Any device
 - Cloud & On-Premise versions
-
- Find out more: <https://powerbi.microsoft.com/en-us/>



Introduction to DAX

- DAX - Data Analysis Expressions
- Language used for querying:
 - Power Pivot (came in 2010)
 - **Power BI**
 - SSAS Tabular Models
- DAX is Simple but not Easy 😊
- Designed for Data Modeling & Business Calculations
- Functional programming language
 - That means - Execution of code is just a Function call
 - Collection of Functions and Operators

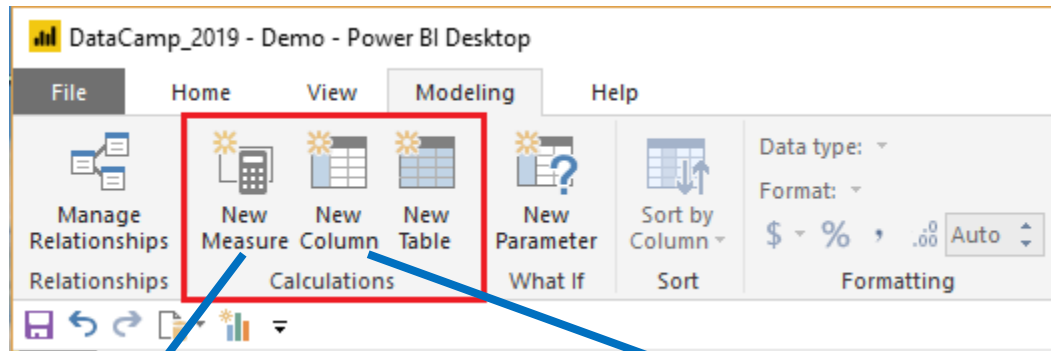


Introduction to DAX ...

- Power BI Desktop -> M & **DAX**
- Both are expression based languages
- M is the query language that can be viewed in Power Query Editor
- DAX can be written different places (we will see)



Start with DAX



Measures:



- May not work Row by Row
- Generally use Tables & Aggregators
- Does not have current Row concept
- E.g.
 - $\text{TotalSales} = \text{UnitPrice} * \text{Quantity}$
 - $\text{TotalSales} \%$

Columns:

- Computed using DAX & stored within model
- Computed only Once
- Always for current Row (each Row)
- E.g. $\text{TotalSales} = \text{UnitPrice} * \text{Quantity}$



Location & Naming

- Calculated/Derived Columns:
 - Belong to a Table
 - Can not leave boundary of Table
 - Icon:  TotalSalesColumn
 - Should be named as Table[Column]
- Calculated/Derived Measures:
 - Do not belong to a Table
 - Can be moved to any Table (Ideally should be in 1 place in a Model)
 - Icon:  TotalSalesMeasure
 - Should be named as [Measure]



Location & Naming - Demo

Quick Tips:

- Keep all your Measures in 1 Table
- Move existing Measures to Measure Table by changing [Home Table](#)
- Use Alt + Enter to write DAX in multi-lines
- Use // to put comments in code
- Quotes can be omitted when object name does not have spaces in it
- Table name can be omitted when referring to single table objects
- Bracket for columns **can not** be omitted



When to use What ?

- Calculated/Derived Columns:
 - slice/dice Or filter values in Report
- Calculated/Derived Measures:
 - Calculate percentages, ratios
 - Complex aggregations
 - Aggregation with cross table filters
- Notes – Choose wisely
 - Columns are Memory hungry
 - Measures are CPU hungry



Datatypes in DAX

- Number, Non-Numbers (Simple)
- Why ?
 - Functional programming language
 - Function for Numbers, Function for Non-Numbers
- Datatypes are strongly typed But Operators are not

Number Types:

- Integer (64 bit)
- Decimal (Floating point)
- Currency (Money)
- Date (Datetime)
- True/False (Boolean)

Non-Number Types:

- String
- Binary Objects



Aggregation Function

- SUM, AVERAGE, MIN, MAX
- SUMX, AVERAGEX, MINX, MAXX (iterators)
- Only for Numeric columns/values
- X (iterators) takes 2 parameters:
 - Table to iterate
 - Formula to evaluate for each row
- E.g.
 - SUM(Sales[Amount])
 - SUM(Sales[Quantity] * Sales[UnitPrice])
 - SUMX(Sales, Sales[Quantity] * Sales[UnitPrice])



Counting Function

- COUNT – Only for Numeric values
- COUNTA – Counts anything except Blanks
- COUNTBLANK – Count Blanks
- COUNTROWS - Count rows in Table
- DISTINCTCOUNT – you know 😊 its same



Logical & Informational Functions

- Logical:
 - AND, OR, NOT, IF, IFERROR
- Informational:
 - ISBLANK
 - ISNUMBER
 - ISTEXT
 - ISNONTEXT
 - ISERROR



Functions ...

Quick Tips:

- **MIN, MAX** – Used as Aggregator as well as for Comparisons
 - E.g. **Compute Maximum amount of Sales**
 - MAX (Sales[Amount])
 - **Compute Maximum between values**
 - MAX (Sales[Amount], Sales[GrossMargin])
- **DEVIDE** – takes care of 0 denomination and errors
 - E.g. DEVIDE (Sales[GrossMargin] / Sales[Amount], 0)
- **Variables** – To re-use the expression in code multiple times & places
 - E.g.

```
FinalQuantity =  
    Var TotalQuantity = SUM ( Sales[Quantity] )  
RETURN  
    IF (  
        TotalQuantity > 100, TotalQuantity * 2.5, TotalQuantity * 1.5  
    )
```



RELATED, RELATEDTABLE

- RELATED
 - Follows the defined relationships and returns the value of column/expression
 - Returned result is 1 value
- RELATEDTABLE
 - Follows the relationships and returns all the rows with respect to current row
 - Returned result may be multi value
- Lets see in [Demo](#)

Use RELATED for	1 to 1 relationships
Use RELATEDTABLE for	1 to * relationships



Time Intelligence Functions

- Last Year, Last Month, MTD, YTD, Running Totals
 - Quick Measure
-
- Lets see in [Demo](#)



Categories of Functions

- [Date and time functions \(DAX\)](#) - These functions in DAX are similar to date and time functions in Microsoft Excel. However, DAX functions are based on the datetime data types used by Microsoft SQL Server.
- [Time-intelligence functions \(DAX\)](#) - These functions help you create calculations that use built-in knowledge about calendars and dates. By using time and date ranges in combination with aggregations or calculations, you can build meaningful comparisons across comparable time periods for sales, inventory, and so on.
- [Filter functions \(DAX\)](#) - These functions help you return specific data types, look up values in related tables, and filter by related values. Lookup functions work by using tables and relationships between them. Filtering functions let you manipulate data context to create dynamic calculations.
- [Information functions \(DAX\)](#) - These functions look at a table or column provided as an argument to another function and tells you whether the value matches the expected type. For example, the ISERROR function returns TRUE if the value you reference contains an error.
- [Logical functions \(DAX\)](#) - These functions return information about values in an expression. For example, the TRUE function lets you know whether an expression that you are evaluating returns a TRUE value.
- [Math and Trig functions \(DAX\)](#) - Mathematical functions in DAX are similar to Excel's mathematical and trigonometric functions. However, there are some differences in the numeric data types used by DAX functions.
- [Other functions \(DAX\)](#) - These functions perform unique actions that cannot be defined by any of the categories most other functions belong to.
- [Parent and Child functions \(DAX\)](#) - These Data Analysis Expressions (DAX) functions help users manage data that is presented as a parent/child hierarchy in their data models.
- [Statistical functions \(DAX\)](#) - These functions perform aggregations. In addition to creating sums and averages, or finding minimum and maximum values, in DAX you can also filter a column before aggregating or create aggregations based on related tables.
- [Text functions \(DAX\)](#) - With these functions, you can return part of a string, search for text within a string, or concatenate string values. Additional functions are for controlling the formats for dates, times, and numbers.



Helpful Resources

- <https://curbal.com/blogglossary>
- <https://docs.microsoft.com/en-us/sql/analysis-services/tabular-models/dax-formula-compatibility-in-directquery-mode-ssas-2016?view=sql-server-2017>
- <https://docs.microsoft.com/en-us/power-bi/desktop-quickstart-learn-dax-basics>





Thank you

Enjoy rest of the Event,
Have a great weekend

