Power BI - DAX - p1 - Lecture 15

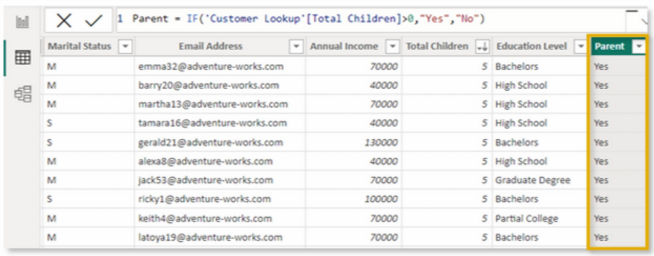
MEET DAX

Data Analysis Expressions (commonly known as DAX) is the formula language that drives the Power BI front-end. With DAX, you can:

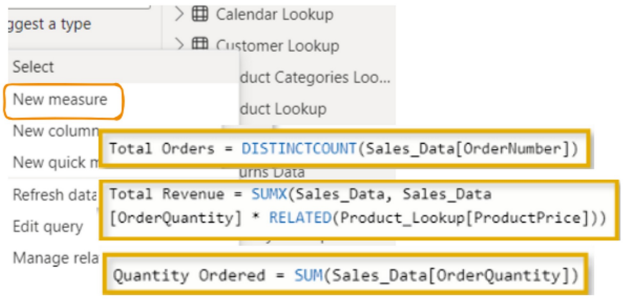
- Go beyond the capabilities of traditional spreadsheet formulas, with powerful and flexible functions built specifically to work with relational data models.
- Add calculated columns (for filtering) and measures (for aggregation) to enhance data models.

Two ways to use DAX

Calculated Columns

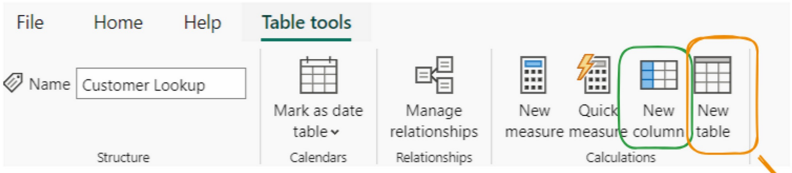


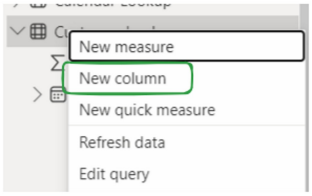
Measures

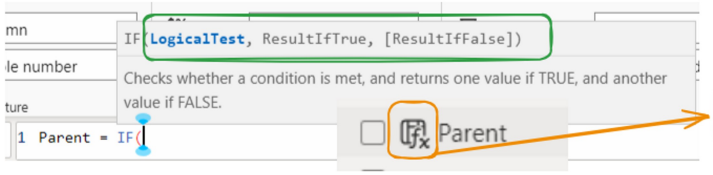



```
x = 2y + 5;
if y = 10:
x = 25.
```

```
product_price = 10
prder_quantity = 5
profit = product_price * order_quantity
```







Calculated Column

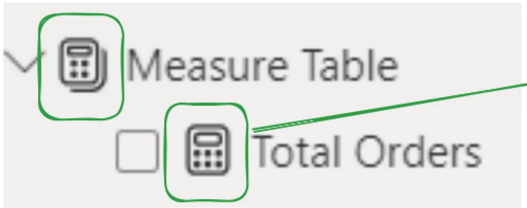
Let's Create a Measure Table where we can store all Explicit measures.

```
Parent = IF('Customer Lookup'[TotalChildren] > 0,"Yes","No")
```

Click The Enter Data & Create A Table

Rename it to "Measure Table"

→ Now, Once Measure Table is being created. We can create unlimited Explicit Measure in it.

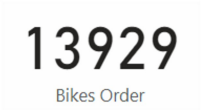


"Measure" Icon

- CALCULATE(Expression,Filter ....)

Calling out existing Explicit Measure

"Applying some filter"




```
Bikes Order = CALCULATE([Total Orders],'Product Categories Lookup'[CategoryName] = "Bikes")
```

Explicit Measure helps us to write complex calculation.

```
Mountain Bikes Orders = CALCULATE([Bikes Order],'Product Subcategories Lookup'[SubcategoryName] = "Mountain Bikes")
```

4706
Mountain Bikes Orders

CategoryName	Sum of OrderQuantity
⊞ Accessories	57809
⊟ Bikes	13929
⊞ Mountain Bikes	4706
⊞ Road Bikes	7099
⊞ Touring Bikes	2124
⊞ Clothing	12436
Total	84174

CategoryName	Sum of OrderQuantity	Total Orders	Bikes Order
⊞ Accessories	57809	57809	13929
⊞ Bikes	13929	13929	13929
⊞ Clothing	12436	12436	13929
⊞ Components			13929
Total	84174	84174	13929

M VS. DAX

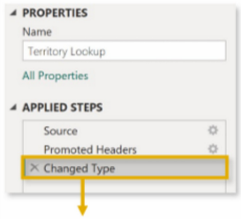
M and DAX are two distinct functional languages used within Power BI Desktop:

- M is used in the Power Query editor, and is designed specifically for extracting, transforming and loading data.

- DAX is used in the Power BI front-end, and is designed specifically for analyzing relational data models

M CODE

Query Editor:




```
#"Changed Type" = Table.TransformColumnTypes(    // Adding a new step
  #"Promoted Headers",                           // after we promoted headers
  {
    {"SalesTerritoryKey", Int64.Type},           // that changes column datatypes
    {"Region", type text},
    {"Country", type text},
    {"Continent", type text}
  }
)
```

DAX

Report View:

Category Name	Total Returns	Bike Returns
Accessories	1,115	
Bikes	427	427
Clothing	267	
Total	**1,809**	**427**

```
1 Bike Returns =
2 CALCULATE(
3     [Total Returns],                                  // Counting total returns
4     'Product Categories Lookup'[Category Name] = "Bikes"    // filtered for bikes only
5 )
```

CALCULATED COLUMNS

Calculated columns allow you to add new, formula-based columns to tables in a model

- Calculated columns refer to entire tables or columns (no A1-style cell references).

- Calculated columns generate values for each row, which are visible within tables in the Data view.

- Calculated columns understand row context; they're great for defining properties based on information in each row, but generally useless for aggregation (sum, count, etc.)

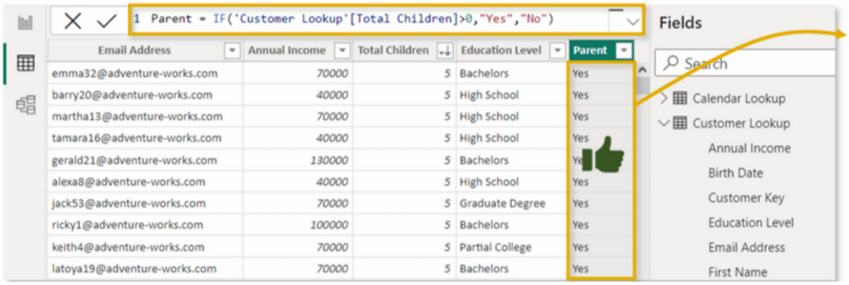
HEY THIS IS IMPORTANT!
As a rule of thumb, use calculated columns to "stamp" static, fixed values to each row in a table (or go upstream and use the Query Editor!)
DO NOT use calculated columns for aggregation — this is what measures are for!

Calculated Columns - Filtering & Grouping.
- Row Context.

Measures - Filter Context.
- Aggregation.

PRO TIP:
Calculated columns are typically used for filtering & grouping data, rather than creating aggregate numerical values

EXAMPLE: CALCULATED COLUMNS

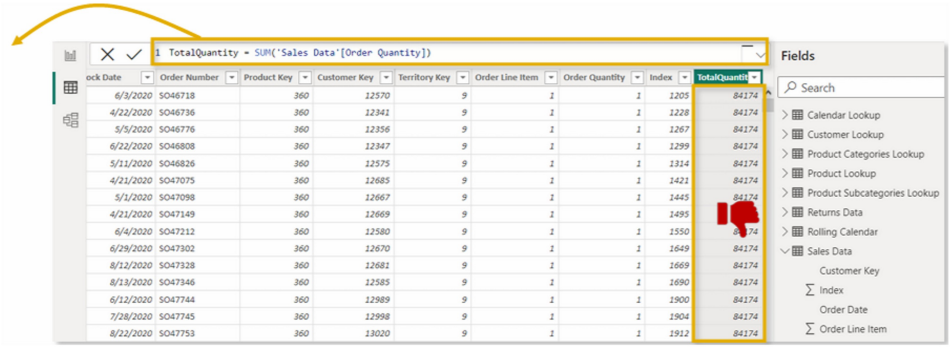


In this case we've added a calculated column named Parent, which equals "Yes" if the [Total Children] field is greater than 0, and "No" otherwise

- Since calculated columns understand row context, a new value is calculated in each row based on the value in the [Total Children] column.

- This is a valid use of calculated columns; it creates a new row "property" that we can use to filter or segment any related data within the model.

Here we're using an aggregation function (SUM) to calculate a new column named Total Quantity



ock Date	Order Number	Product Key	Customer Key	Territory Key	Order Line Item	Order Quantity	Index	TotalQuantity
6/3/2020	SO46718	360	12570	9	1	1	1205	84174
4/22/2020	SO46736	360	12341	9	1	1	1228	84174
5/5/2020	SO46776	360	12356	9	1	1	1267	84174
6/22/2020	SO46808	360	12347	9	1	1	1299	84174
5/11/2020	SO46826	360	12575	9	1	1	1314	84174
4/21/2020	SO47075	360	12685	9	1	1	1421	84174
5/1/2020	SO47098	360	12667	9	1	1	1445	84174
4/21/2020	SO47149	360	12669	9	1	1	1495	84174
6/4/2020	SO47212	360	12580	9	1	1	1550	84174
6/29/2020	SO47302	360	12670	9	1	1	1649	84174
8/12/2020	SO47328	360	12681	9	1	1	1669	84174
8/13/2020	SO47346	360	12585	9	1	1	1690	84174
6/12/2020	SO47744	360	12989	9	1	1	1900	84174
7/28/2020	SO47745	360	12998	9	1	1	1904	84174
8/22/2020	SO47753	360	13020	9	1	1	1912	84174

- Since this is an aggregation function, the same grand total is returned in every row of the table
- This is not a valid use of calculated columns; these values are statically "stamped" onto the table and can't be filtered, sliced, etc.

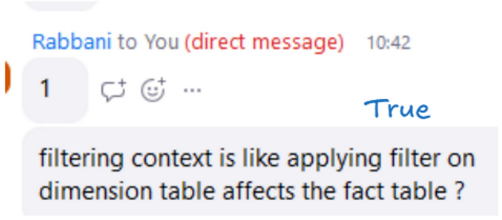
DAX MEASURES

Measures are DAX formulas used to generate new calculated values

- Like calculated columns, measures reference entire tables or columns (no A1-style cell references).
- Unlike calculated columns, measures aren't visible within tables; they can only be "seen" within a visualization like a chart or matrix (similar to a calculated field in a PivotTable).
- Measures evaluate based on filter context, which means they recalculate when the fields or filters around them change.

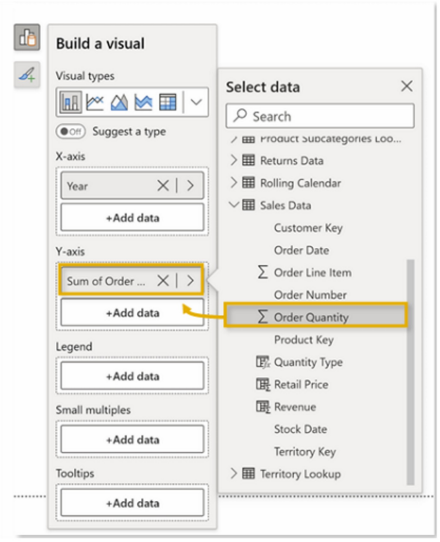
C.C - Model View
- Table View
Measures - Report View

HEY THIS IS IMPORTANT!
As a rule of thumb, use measures when a single row can't give you the answer, or when you need to aggregate values across multiple rows in a table



PRO TIP:
Use measures to create numerical, calculated values that can be analyzed in the "values" field of a report visual

IMPLICIT VS. EXPLICIT MEASURES



Example of an implicit measure

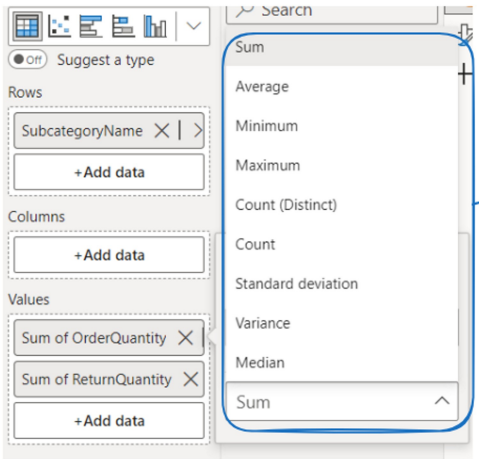
Implicit measures are created when you drag raw numerical fields into a report visual and manually select an aggregation mode (Sum, Average, Min, Max, Count, etc.)

Explicit measures are created when you actually write a DAX formula and define a new measure that can be used within the model

HEY THIS IS IMPORTANT!

Implicit measures are only accessible within the specific visualization in which they were created, and cannot be referenced elsewhere.

Explicit measures can be used anywhere in the report, and referenced by other DAX calculations to create "measure trees".



Implicit Measures.