

Power BI - Data Modelling - p3 + DAX

ASSIGNMENT: FILTER FLOW

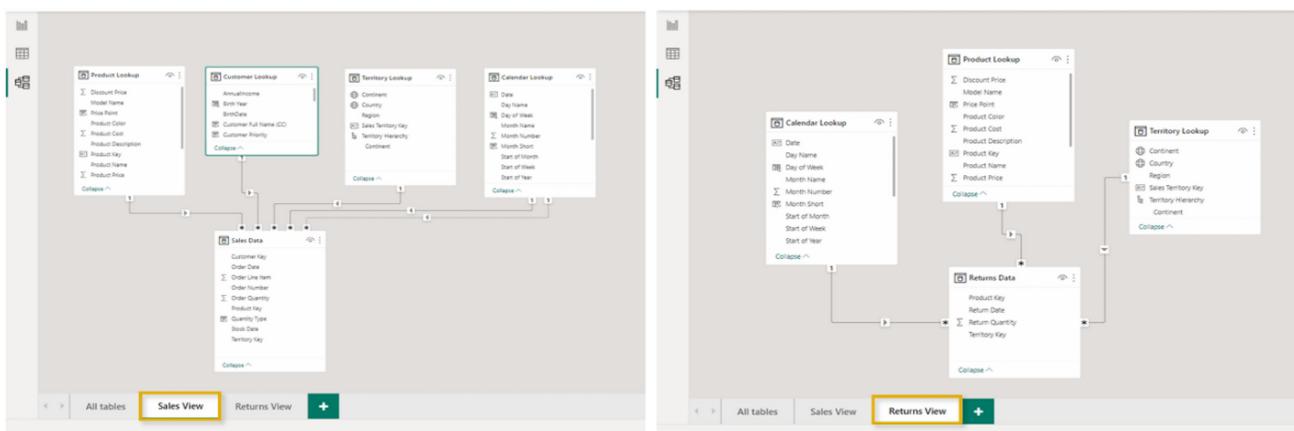
1. Replicate matrix below to diagnose what he must have done to the model

Product Key	Sum of Order Quantity	Sum of Return Quantity
324	72	3
326	65	3
328	75	4
330	51	6
332	64	2
334	63	2
336	50	1
340	56	1
342	72	1
346	24	2

- Which product is #338?
- Why didn't matrix show any orders?

2. Hide any remaining foreign keys to prevent other users from making the same mistake

MODEL LAYOUTS



--> Model layouts allow you to create custom views to show specific portions of large, complex models

- Here we've created a Sales View displaying only tables related to sales, and a Returns View displaying only tables related to returns (Note: this doesn't actually create duplicate tables)

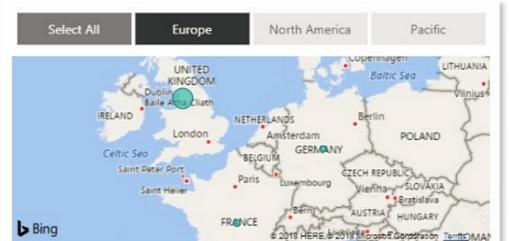
DATA FORMATS & CATEGORIES

Customize data formats from the Column tools menu in the Data view or the Properties pane in the Model view

The screenshot shows the Power BI Data view interface. The ribbon at the top has 'Table tools' selected. In the center, there's a table with columns: Region, Country, Continent, and Sales Territory Key. The 'Country' column is currently selected. The 'Column tools' ribbon tab is active. The 'Format' dropdown in the 'Formatting' group is set to 'Text'. The 'Data category' dropdown in the 'Summarization' group is set to 'Country'. A vertical list of categories is visible on the right, including Address, Place, City, County, State or Province, Postal code, Country, Continent, Latitude, Longitude, Web URL, Image URL, and Barcode. The 'Country' option is highlighted.

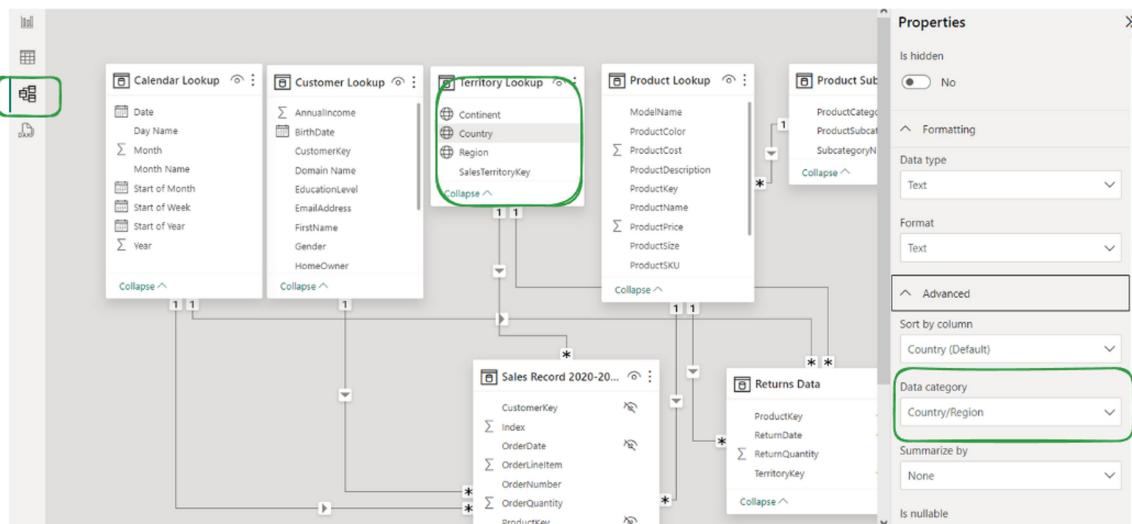
Assign data categories for geospatial fields, URLs or barcodes

- This is commonly used to help Power BI map location-based fields like addresses, countries, cities, coordinates, zip codes, etc.



Territory Lookup

- Continent
 - Country
 - Region
- SalesTerritoryKey
- [Collapse ^](#)



HIERARCHIES

Hierarchies are groups of columns that reflect multiple levels of granularity

- For example, a Geography hierarchy might include Country, State and City fields
- Hierarchies are treated as a single item in tables and reports, allowing users to "drill up" and "drill down" through each level

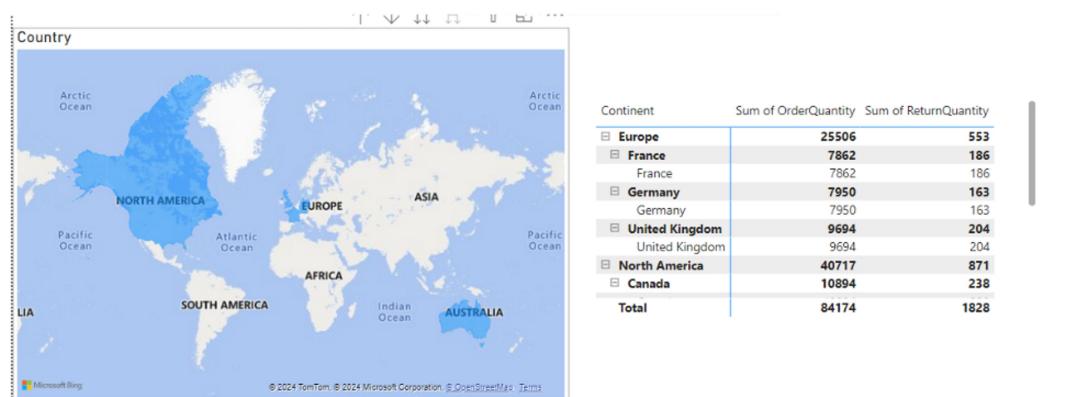
The figure consists of three screenshots from the Power BI Data pane:

- Screenshot 1 (Left):** Shows the context menu for the "Product Price" field, with the "Create hierarchy" option highlighted.
- Screenshot 2 (Middle):** Shows the "Territory Lookup" node expanded, with the "Territory Hierarchy" node selected and highlighted.
- Screenshot 3 (Right):** Shows the context menu for the "Country" field, with the "Add to hierarchy" option highlighted.

In the Data pane, right-click a field and select Create hierarchy

This hierarchy contains "Continent", and is named "Territory Hierarchy"

Right-click another field (like "Country") and select Add to Hierarchy (or drag it in!)



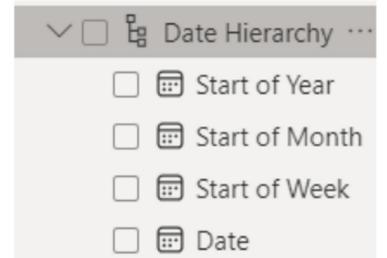
ASSIGNMENT: HIERARCHIES

1. Create a new hierarchy based on the Start of Year field, and name it "Date Hierarchy".

2. Right-click or drag to add fields until your hierarchy contains the following (in this order):

- Start of Year
- Start of Month
- Start of Week
- Date

3. Add your new hierarchy to the matrix visual (on rows) and practice drilling up and down between each level of granularity



Start of Year	Sum of OrderQuantity	Sum of ReturnQuantity
01-01-2020	2630	86
01-01-2020	184	4
01-02-2020	165	4
01-03-2020	198	9
01-04-2020	204	14
01-05-2020	206	11
01-06-2020	212	4
01-07-2020	247	3
01-08-2020	278	6
01-09-2020	196	2
01-10-2020	223	11
01-11-2020	191	5
01-12-2020	326	13
01-01-2021	36230	770
01-01-2021	242	8
01-02-2021	267	8
Total	84174	1828

DATA MODEL BEST PRACTICES

Focus on building a normalized model from the start

- Leverage relationships and make sure that each table serves a clear, distinct purpose

Organize dimension tables above data tables in your model

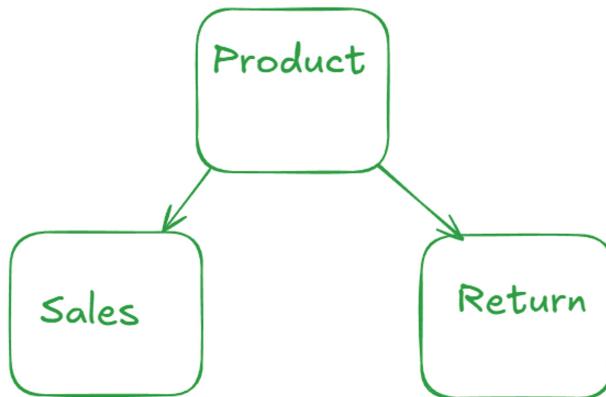
- This serves as a visual reminder that filters always flow "downstream"

Avoid complex relationships unless absolutely necessary

- Aim to use 1-to-many table relationships and one-way filters whenever possible

Hide fields from report view to prevent invalid filter context

- This forces report users to filter using primary keys from dimension tables

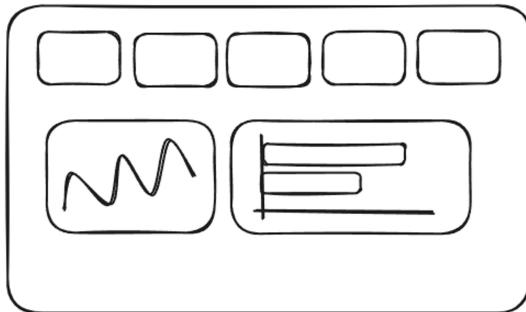


CALCULATED FIELDS WITH DAX

In this section we'll use Data Analysis Expressions (DAX) to add calculated columns & measures to our model, and introduce topics like row & filter context, iterators and more

TOPICS WE'LL COVER:

1. DAX
2. Row & Filter Context
3. Common Functions
4. Iterators
5. Columns & Measures
6. DAX Syntax
7. Calculate
8. Time Intelligence



GOALS FOR THIS SECTION:

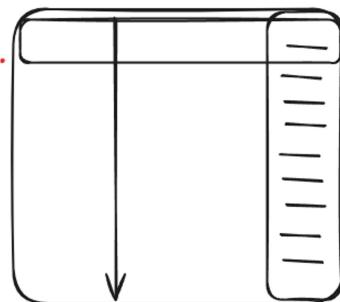
- Introduce DAX fundamentals and learn when to use calculated columns and measures.
- Understand the difference between **row context** and **filter context**, and how they impact DAX calculations.
- Learn DAX formula syntax, basic operators and common function categories (math, logical, text, date/time, filter, etc.).
- Explore nested functions, and more complex topics like iterators and time intelligence patterns.

Weekend Order

Sunday
Saturday

Order Qty.

Measure : Group By Aggregation



Weekday Order

Mon - Fri

Order Qty.

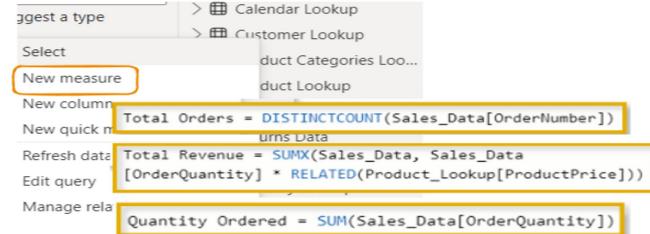
MEET DAX

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.



Marital Status	Email Address	Annual Income	Total Children	Education Level	Parent
M	emma32@adventure-works.com	70000	5	Bachelors	Yes
M	barry20@adventure-works.com	40000	5	High School	Yes
M	marta15@adventure-works.com	70000	5	High School	Yes
S	tamara16@adventure-works.com	40000	5	High School	Yes
S	gerald21@adventure-works.com	130000	5	Bachelors	Yes
M	alex8@adventure-works.com	40000	5	High School	Yes
M	jack53@adventure-works.com	70000	5	Graduate Degree	Yes
S	ricky1@adventure-works.com	100000	5	Bachelors	Yes
M	keith4@adventure-works.com	70000	5	Partial College	Yes
M	latoya19@adventure-works.com	70000	5	Bachelors	Yes



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}
    }
)
```

Report View:

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

DAX

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