



UP & RUNNING WITH

MICROSOFT POWER BI DESKTOP



COURSE STRUCTURE

This is a **project-based course** designed for students looking for a practical, hands-on, and highly engaging approach to learning Power BI Desktop for business intelligence

Course resources include:

-  **Downloadable PDF eBook** (200+ pages) containing all course slides, assignments and reference materials
-  **Quizzes and Assignments** to reinforce key concepts and simulate real-world scenarios, with step-by-step solution videos
-  Complete **Bonus Project** to test your abilities and apply the skills developed throughout the course to a brand-new data set

COURSE OUTLINE

| | | |
|----------|---|---|
| 1 | Introducing Power BI Desktop | <i>Installing Power BI Desktop, exploring the Power BI workflow, comparing Power BI vs. Excel, etc.</i> |
| 2 | Connecting & Shaping Data | <i>Connecting to data, shaping & transforming tables, using profiling tools, editing, merging & appending queries, etc.</i> |
| 3 | Creating a Data Model | <i>Building relational models, creating table relationships, understanding cardinality and filter flow, etc.</i> |
| 4 | Calculating Measures with DAX | <i>Understanding DAX syntax, adding calculated columns and measures, writing common formulas and functions, etc.</i> |
| 5 | Visualizing Data with Dashboards | <i>Inserting charts and visuals, customizing formats, editing interactions, applying filters and bookmarks, etc.</i> |
| 6 | Optimizing Power BI Performance | <i>Exploring common Power BI optimization tools within the Optimize and External tools menus</i> |

COURSE PROJECT

THE SITUATION

You've just been hired as a Business Intelligence Analyst by **AdventureWorks***, a global manufacturing company that produces cycling equipment and accessories

THE BRIEF

The management team needs a way to **track KPIs** (*sales, revenue, profit, returns*), **compare regional performance**, **analyze product-level trends**, and **identify high-value customers**.

All you've been given is a **folder of raw csv files**, which contain information about transactions, returns, products, customers, and sales territories.

THE OBJECTIVE

Use Power BI Desktop to:

- Connect and transform the raw data
- Build a relational data model
- Create calculated columns and measures with DAX
- Design an interactive dashboard to visualize the data

SETTING EXPECTATIONS

1 What you see on your screen **may not always match mine**

- *Power BI Desktop features are updated frequently, with new versions released each month*
- **NOTE:** *Power BI is currently only compatible with PC/Windows (not available for Mac)*

2 This course is designed to help you build **foundational skills**

- *Our goal is to help you build a deep foundational understanding of the Power BI desktop workflow; some topics may be simplified, and we won't cover some advanced tools (M code, advanced DAX, R/Python visuals, etc.)*

3 This is a **hands-on** and **project-based** learning experience

- *You will get the most value out of this course if you follow along closely with the demos and assignments; we'll be working through the entire BI workflow to create a professional-quality dashboard from scratch*

4 We will not cover **Power BI Service** as part of this course

- *This course focuses on Power BI Desktop specifically; online sharing and collaboration features (app.powerbi.com) require a separate account and are covered in-depth in a separate course*

INTRODUCING POWER BI



MEET POWER BI



In this section we'll **introduce Power BI Desktop**, review the download and installation process, adjust default settings, and explore the Power BI interface and workflow.

TOPICS WE'LL COVER:

Introducing Power BI

Power BI vs. Excel

Installation Options

Adjusting Settings

Interface & Workflow

Helpful Resources

GOALS FOR THIS SECTION:

- Download and install Power BI Desktop, and adjust the settings for our course project
- Understand the role that Power BI plays within the broader Microsoft ecosystem
- Explore core components of the Power BI Desktop interface
- Review the business intelligence workflow that we'll follow as we build our course project

MEET POWER BI



Microsoft Power BI is a self-service business intelligence platform, which includes both desktop and web-based applications for connecting, modeling, and visualizing data

Learn more at powerbi.microsoft.com



Figure 1: Magic Quadrant for Analytics and Business Intelligence Platforms



As of January 2023

© Gartner, Inc

WHY POWER BI?



Connect, transform and load millions of rows of data

- *Access data from virtually anywhere (database tables, flat files, web, cloud services, folders, etc.), and create fully automated workflows to extract, transform and load data for analysis*



Build relational models to blend data from multiple sources

- *Create table relationships to analyze holistic performance across an entire relational data model*



Define complex calculations using Data Analysis Expressions (DAX)

- *Enhance datasets and enable advanced analytics with powerful and portable DAX expressions*



Bring data to life with interactive reports and dashboards

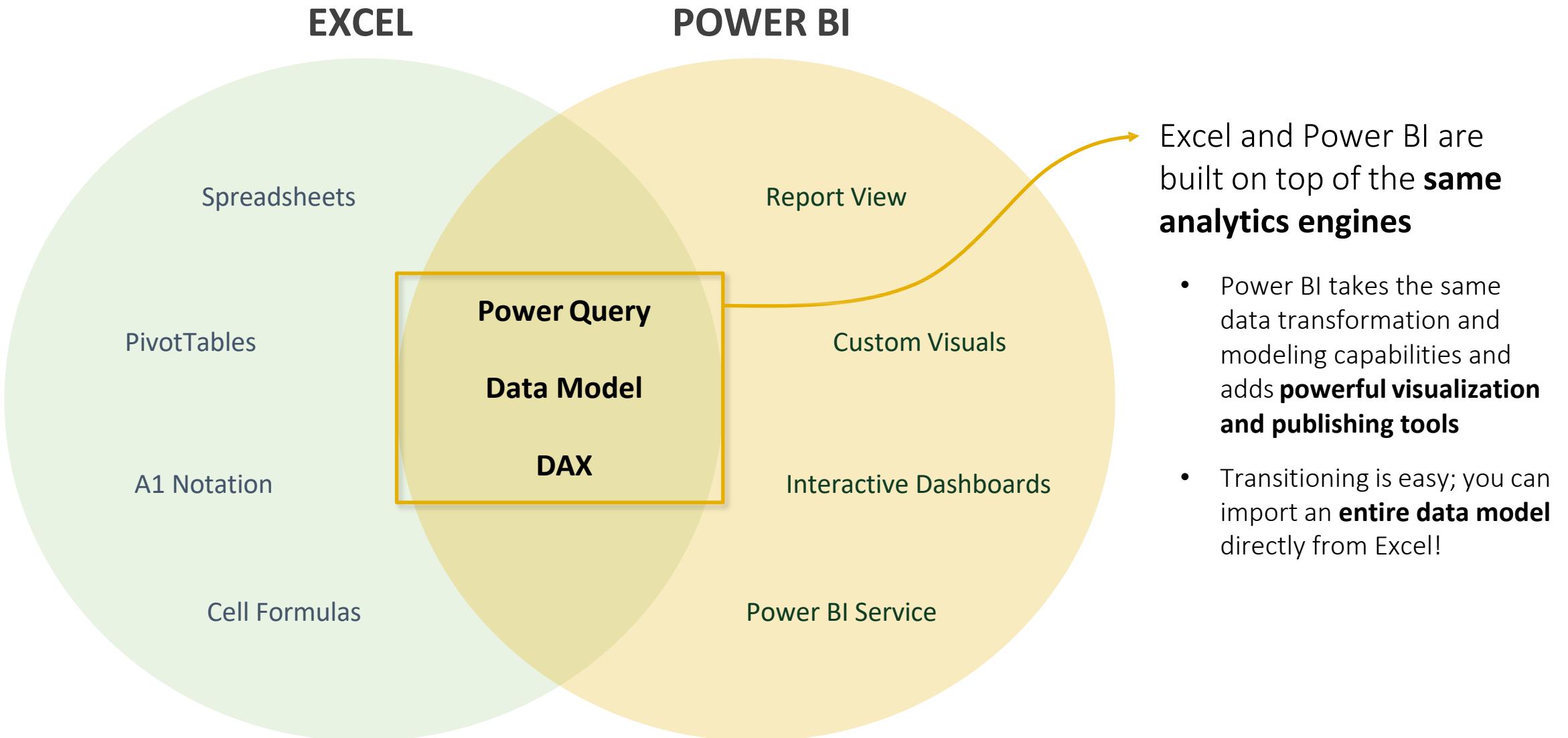
- *Build professional-quality reports and dashboards with best-in-class visualization tools*



Develop a versatile, in-demand skill set

- *Power BI is the industry leader in self-service BI, and the skills you build in this course will be highly transferrable*

EXCEL VS. POWER BI



INSTALLING POWER BI DESKTOP

1) Download from Microsoft store

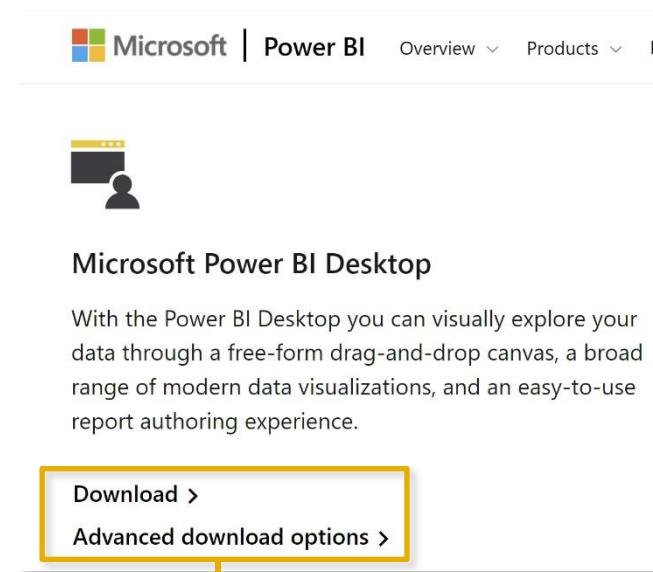
apps.microsoft.com



- Windows handles **automatic updates**
- Updates only elements that have been changed
- Doesn't require administrator access

2) Download manually from web

powerbi.microsoft.com/downloads



- **No automatic updates** (allows version control)
- Downloads an executable installation file
- Administrator access may be required

3) Install as part of Microsoft 365

microsoft.com/en-us/microsoft-365

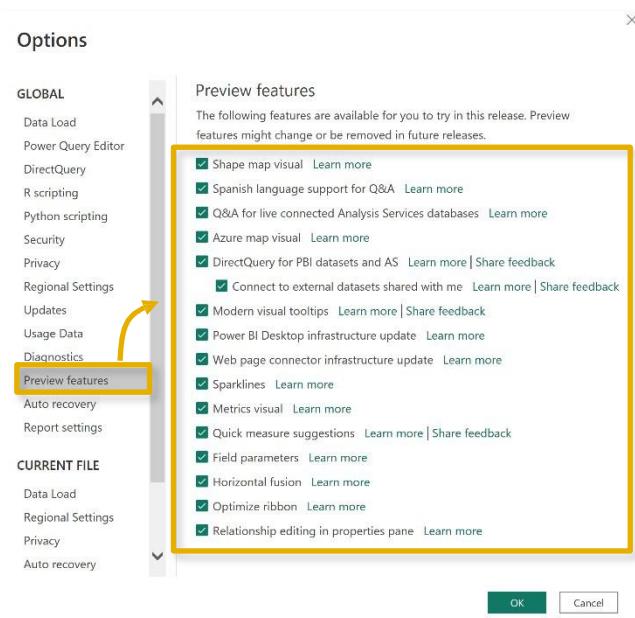


- Power BI Desktop is included as part of select enterprise Office/Microsoft 365 subscriptions
- If your company uses a compatible version of Microsoft 365, talk to an admin about getting access to Power BI



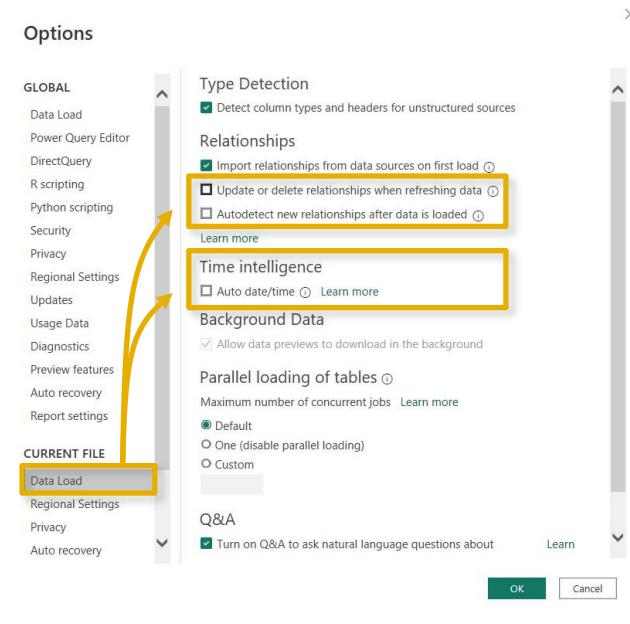
POWER BI SETTINGS

Global > Preview Features



Select **all available preview features** by default (these change with each monthly release)

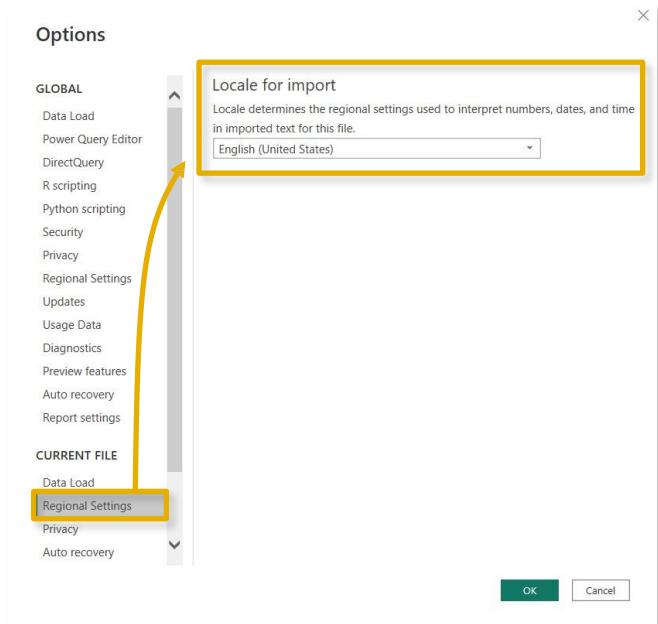
Current File > Data Load



Make sure the following options are **NOT selected**:

- Update or delete relationships when refreshing data***
- Autodetect new relationships after data is loaded***
- Time Intelligence > Auto date/time***

Current File > Regional Settings



Select "**English (United States)**" from the dropdown menu (this will align with the data in course project files)

HEY THIS IS IMPORTANT!

Options under **CURRENT FILE** need to be adjusted **every time you open a new Power BI workbook** (these settings do not persist across new .pbix files)

POWER BI WORKFLOW

Raw data is extracted and transformed in the **Power Query editor**, then loaded to the Power BI “front-end”

The screenshot shows the Power Query Editor interface with a table of 293 rows and 11 columns. The table includes columns for Product Key, Product Subcategory Key, Product SKU, Product Name, Model Name, and Product Description. The 'Product Description' column contains detailed descriptions of various bicycle components like patch kits, tires, and water bottles.



Power Query Editor

Power BI “Back-End”

The screenshot shows the Power BI desktop interface with a dashboard. The dashboard features four KPI cards: Revenue (\$24.9M), Profit (\$10.5M), Orders (25.2K), and Return Rate (2.2%). Below the KPIs are three line charts: 'Weekly Revenue' (a line with a dashed trend line), 'Monthly Revenue' (a line with a dashed trend line), and 'Monthly Orders' (a line with a dashed trend line). To the right of the charts is a table titled 'Top 10 Products' showing items like Water Bottle - 30 oz. and Patch Kit/8 Patches. Further down are sections for 'Most Ordered Product Type' (Tires and Tubes) and 'Most Returned Product Type' (Shorts).



Model View

Power BI “Front-End”

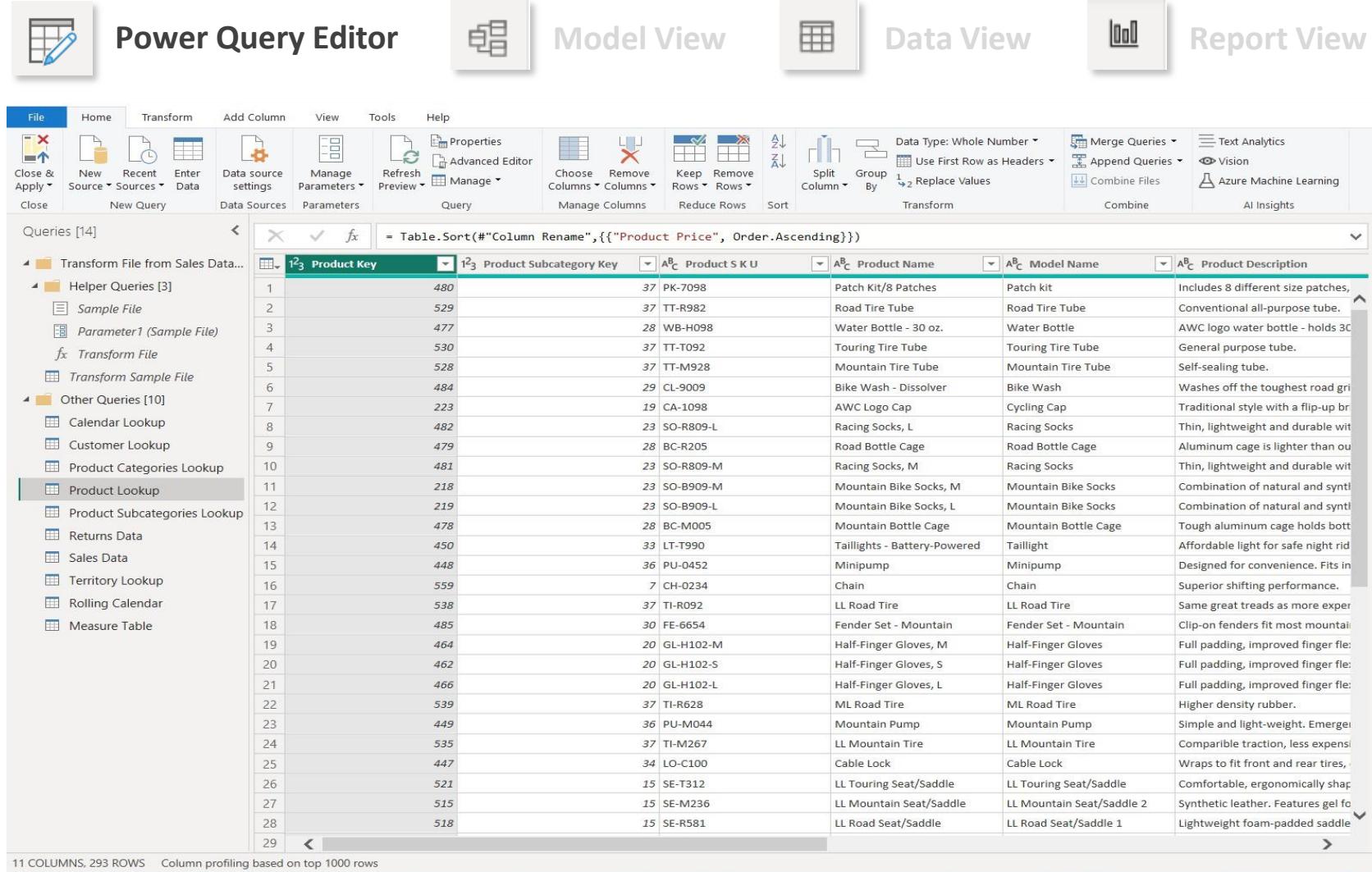


Data View



Report View

POWER BI WORKFLOW

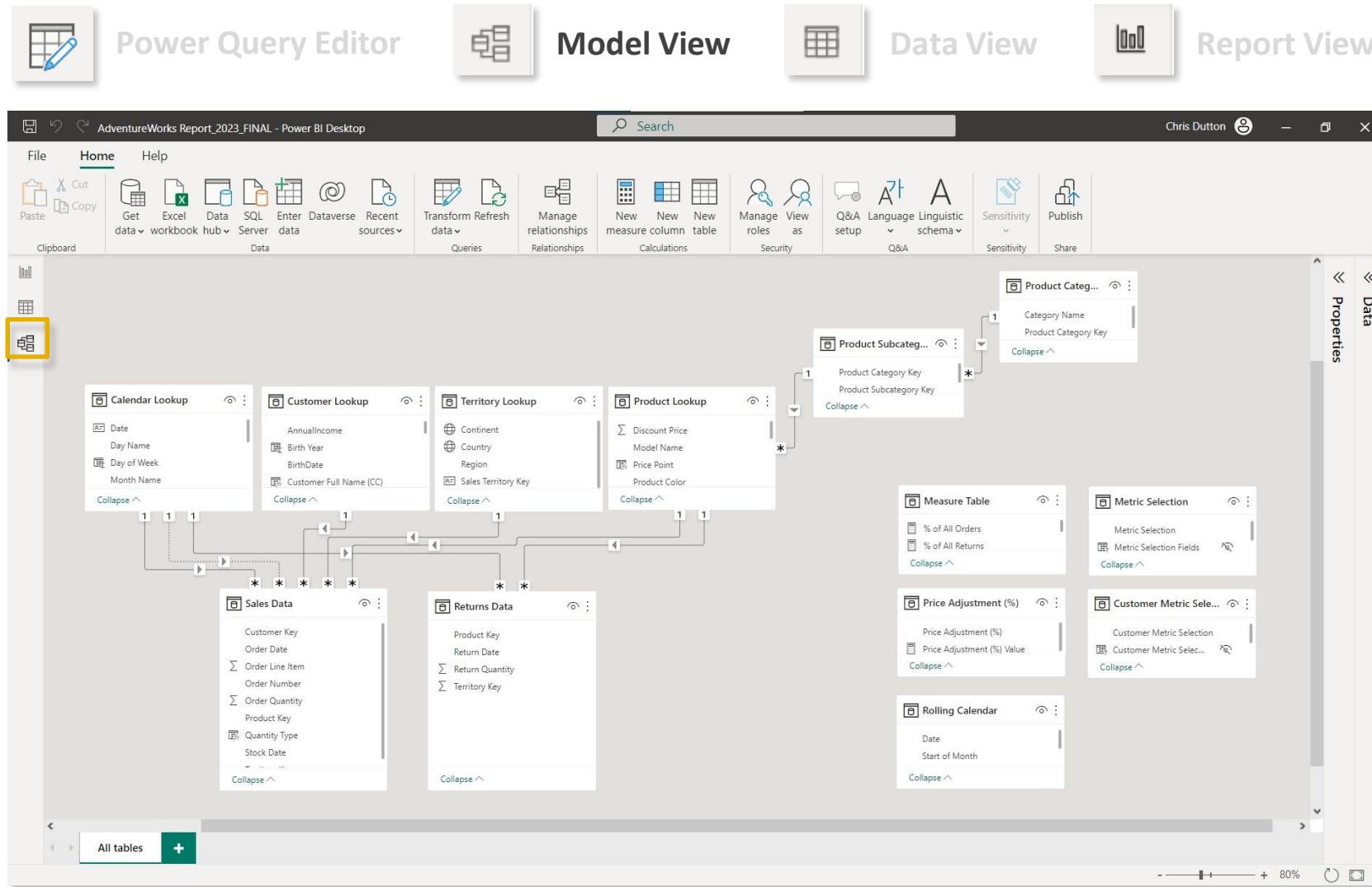


The screenshot shows the Microsoft Power Query Editor interface. The top navigation bar includes tabs for Power Query Editor, Model View, Data View, and Report View. The main area displays a table with 11 columns and 293 rows, titled "Product". The table includes columns for Product Key, Product Subcategory Key, Product SKU, Product Name, Model Name, and Product Description. The "Product Description" column contains detailed product descriptions. On the left, a sidebar lists various queries: Transform File from Sales Data..., Helper Queries [3] (including Sample File, Parameter1 (Sample File), and Transform File), Transform Sample File, Other Queries [10] (including Calendar Lookup, Customer Lookup, Product Categories Lookup, Product Lookup, Product Subcategories Lookup, Returns Data, Sales Data, Territory Lookup, Rolling Calendar, and Measure Table). The "Product Lookup" query is currently selected. The bottom status bar indicates "11 COLUMNS, 293 ROWS" and "Column profiling based on top 1000 rows".

1

Data is loaded & transformed in the **Power Query Editor**

POWER BI WORKFLOW



1 Data is loaded & transformed in the **Power Query Editor**

2 Data models are configured in the **Model View**

POWER BI WORKFLOW

Power Query Editor Model View Data View Report View

AdventureWorks Report_2023_FINAL - Power BI Desktop

File Home Help Table tools

Name Customer Lookup

Structure

Prefix Gender Occupation CustomerKey FirstName LastName BirthDate MaritalStatus EmailAddress AnnualIncome TotalChildren Education

11206 Blake Flores Friday, September 24, 1948 M blake60@adventure-works.com \$60,000 2 Partie

12093 Levi Chandra Monday, November 08, 1948 M levi1@adventure-works.com \$60,000 2 Partie

12108 James Walker Thursday, February 23, 1950 M james96@adventure-works.com \$60,000 2 Partie

12117 Cameron Yang Friday, August 03, 1951 M cameron23@adventure-works.com \$60,000 2 Partie

13233 Richard Coleman Sunday, December 05, 1948 M richard61@adventure-works.com \$60,000 2 Partie

13235 Jonathan Robinson Wednesday, September 22, 1948 M jonathan72@adventure-works.com \$60,000 2 Partie

13236 Robert Wang Tuesday, May 25, 1948 M robert36@adventure-works.com \$60,000 2 Partie

13370 Blake Ross Monday, March 22, 1948 M blake51@adventure-works.com \$60,000 2 Partie

13372 Isaac Edwards Sunday, November 28, 1948 M isaac24@adventure-works.com \$60,000 2 Partie

13375 Jason Kumar Sunday, April 11, 1948 M jason26@adventure-works.com \$60,000 2 Partie

13376 Jerry Rai Friday, September 03, 1948 M jerry19@adventure-works.com \$60,000 2 Partie

13397 Adam Mitchell Sunday, April 06, 1951 M adam41@adventure-works.com \$60,000 2 Partie

14405 Dylan Walker Thursday, June 01, 1950 M dylan53@adventure-works.com \$60,000 2 Partie

14407 Thomas Bryant Sunday, June 04, 1950 M thomas20@adventure-works.com \$60,000 2 Partie

14415 William Davis Friday, November 16, 1951 M william21@adventure-works.com \$60,000 2 Partie

15325 Elijah Hayes Thursday, May 20, 1948 M elijah24@adventure-works.com \$60,000 2 Partie

15331 Jacob Taylor Friday, August 26, 1949 M jacob2@adventure-works.com \$60,000 2 Partie

15332 Jason Sharma Friday, April 08, 1949 M jason27@adventure-works.com \$60,000 2 Partie

15336 Marco Garcia Thursday, September 21, 1950 M marco15@adventure-works.com \$60,000 2 Partie

15339 Noah Zhang Saturday, July 01, 1950 M noah21@adventure-works.com \$60,000 2 Partie

15343 Carson Barnes Friday, September 07, 1951 M carson2@adventure-works.com \$60,000 2 Partie

15902 Robert Diaz Tuesday, April 25, 1950 M robert33@adventure-works.com \$60,000 2 Partie

15908 David Wilson Thursday, September 20, 1951 M david65@adventure-works.com \$60,000 2 Partie

16806 Louis Zhao Friday, August 04, 1950 M louis4@adventure-works.com \$60,000 2 Partie

16811 Luis Zhang Thursday, April 19, 1951 M luis23@adventure-works.com \$60,000 2 Partie

16813 Carson Diaz Saturday, April 14, 1951 M carson21@adventure-works.com \$60,000 2 Partie

17144 Luis Griffin Friday, February 20, 1948 M luis20@adventure-works.com \$60,000 2 Partie

17149 Steven Richardson Friday, April 22, 1949 M steven19@adventure-works.com \$60,000 2 Partie

17155 Samuel Lewis Thursday, April 21, 1949 M samuel71@adventure-works.com \$60,000 2 Partie

Table: Customer Lookup (18,148 rows)

1

Data is loaded & transformed in the **Power Query Editor**



2

Data models are configured in the **Model View**



3

Table features & calculations are added in the **Data View**

POWER BI WORKFLOW

The screenshot shows the Power BI Desktop interface with the 'Home' tab selected. The main area displays several data cards and visualizations:

- Revenue:** \$24.9M
- Profit:** \$10.5M
- Orders:** 25.2K
- Return Rate:** 2.2%

Below these are two line charts:

- Weekly Revenue:** A line chart showing revenue from January 2020 to January 2022, with a significant upward trend.
- Monthly Revenue:** A card showing \$1.83M, with a note: "Prev Month: \$1.77M (+3.31%)".

Other visualizations include:

- Orders by Category:** A bar chart showing orders for Accessories (17.0K), Bikes (13.9K), and Clothing (7.0K).
- Top 10 Products:** A table listing products with their order counts, revenue, and return percentages.
- Most Ordered Product Type:** Tires and Tubes.
- Most Returned Product Type:** Shorts.

The ribbon menu at the top includes tabs for Power Query Editor, Model View, Data View, and Report View. The Power Query Editor tab is currently active.

1 Data is loaded & transformed in the **Power Query Editor**

2 Data models are configured in the **Model View**

3 Table features & calculations are added in the **Data View**

4 Visuals & reports are designed in the **Report View**

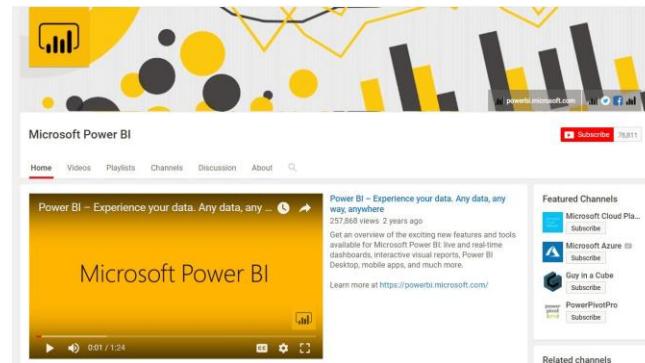
HELPFUL RESOURCES



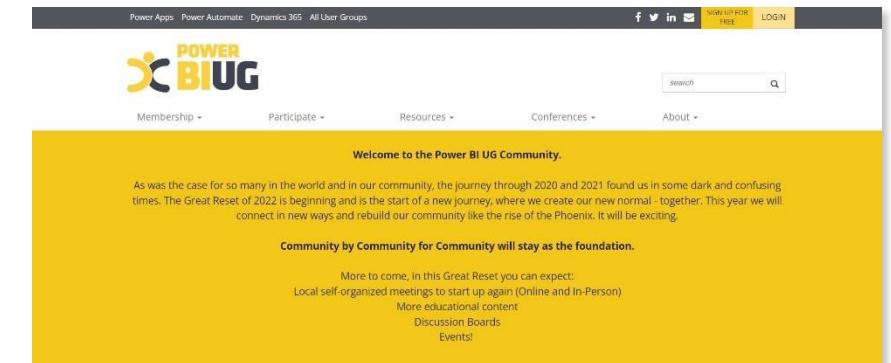
The **Help** tab includes documentation, training videos, sample files, templates, and links to support blogs and communities



The **Microsoft Power BI blog** (powerbi.microsoft.com/blog) publishes monthly summaries to showcase new features



The **Microsoft Power BI YouTube Channel** publishes demos, feature summaries, and advanced tutorials (check out “**Guy in a Cube**” too!)



Power BI User Groups (Power BIUG) are communities of users, which include both local meet-ups and helpful online forums (pbiusergroup.com)

MONTHLY UPDATES

Power BI is updated monthly, so you may notice ongoing changes to settings, options, tools, etc.
Reference the links below to stay up-to-date on product updates and new feature releases:



Power BI Desktop

<https://docs.microsoft.com/en-us/power-bi/fundamentals/desktop-latest-update>



Power BI Service

<https://docs.microsoft.com/en-us/power-bi/fundamentals/service-whats-new>



Power Platform

<https://learn.microsoft.com/en-us/dynamics365/release-plans/>

CONNECTING & SHAPING DATA

CONNECTING & SHAPING DATA



In this section we'll connect to source files and cover some of the most common techniques for **extracting**, **cleaning**, and **shaping data** to prepare it for modeling and analysis

TOPICS WE'LL COVER:

Intro to Power Query

Data Connectors

The Query Editor

Connection Modes

Data QA & Profiling

Table Transformations

Calendar Tools

Combining Queries

GOALS FOR THIS SECTION:

- Explore Power BI's query editor and understand the role that Power Query plays in the larger BI workflow
- Introduce different types of connectors and connectivity modes available for getting data into Power BI
- Review tools for checking data quality and key profiling metrics like column distribution, empty values, errors and outliers
- Transform tables using text, numerical and date/time tools, pivot and group records, and create new conditional columns
- Practice combining, modifying and refreshing queries



FRONT-END VS. BACK-END

Power BI Desktop essentially has two distinct environments: a **front-end** and a **back-end**

- The **front-end** includes the **Data, Model & Report** views, where most of the modeling, analysis and visualization takes place
- The **back-end** includes the **Power Query Editor**, where raw data is extracted, transformed, and loaded to the front-end (ETL)

BACK-END

- **Connect & extract** data using pre-built connectors
- **Profile & QA** the data to explore, clean and prepare it for modeling and analysis
- **Transform & shape** tables to add new features, modify values, group records, or sort and filter columns
- **Merge or append** queries to join and combine them prior to loading to the front-end
- Perform **advanced transformations** using custom M code (out of scope for this course)

FRONT-END

- **Build data models** by creating table relationships between primary and foreign keys
- **Add calculated measures & columns** using Data Analysis Expressions (DAX)
- **Design reports** to visualize the data and create interactive, dynamic dashboards
- **Publish & share** your Power BI workbooks using Power BI Service (cloud application)



TYPES OF DATA CONNECTORS

The screenshot shows the Microsoft Power BI ribbon. The 'Home' tab is selected, and the 'Insert' tab is active. A yellow box highlights the 'Get data' button in the 'Insert' tab's ribbon, which has a yellow arrow pointing to it from the left. To the right of the ribbon is the 'Get Data' dialog box. The 'Get Data' dialog box has a search bar at the top and a sidebar with categories: All, All, File, Database, Power Platform, Azure, Online Services, and Other. The main pane lists various data sources under the 'All' category, including Excel Workbook, Text/CSV, XML, JSON, Folder, PDF, Parquet, SharePoint folder, SQL Server database, Access database, SQL Server Analysis Services database, Oracle database, IBM Db2 database, IBM Informix database (Beta), IBM Netezza, and MySQL database. The 'Excel Workbook' item is currently selected.

Power BI can connect to virtually **any** type of source data, including (*but not limited to*):

- **Flat files & Folders** (csv, text, xlsx, etc.)
- **Databases** (SQL, Access, Oracle, IBM, etc.)
- **Power Platform** (Datasets, Datamarts, Dataflows, Dataverse, etc.)
- **Azure** (Azure SQL, Analysis Services, Databricks, etc.)
- **Online Services** (SharePoint, GitHub, Dynamics 365, Google Analytics, Salesforce, Power BI Service, etc.)
- **Other** (Web feeds, R scripts, Spark, Hadoop, etc.)



POWER QUERY EDITOR

The screenshot illustrates the Power Query Editor interface with several key components highlighted:

- Queries Pane (list of all queries)**: Located on the left side, showing a tree view of 13 queries, with "Customer Lookup" selected.
- Table Preview**: The main area displaying a preview of a table with 28 rows and 7 columns, including columns like Customer Key, Prefix, First Name, Last Name, Birth Date, Marital Status, and Gender.
- Formula Bar (this is "M" code)**: The formula bar at the top showing the current query: `= Table.RemoveColumns(#"Filtered Rows", {"Custom"})`.
- Query Editing Tools**: A callout pointing to the ribbon toolbar, which includes options like Transform, Add Column, View, Tools, and Help, along with specific transformation icons such as Conditional Column, Index Column, Duplicate Column, Merge Columns, Extract, Parse, Statistics, Standard, Scientific, Text Analytics, Vision, Azure Machine Learning, and AI Insights.
- Properties**: The Properties pane on the right showing the query's name ("Customer Lookup") and a list of applied steps.
- Applied Steps**: The Applied Steps pane listing the sequence of transformations performed on the data, including Promoted Headers, Changed Type, Column Rename, Capitalized Each Word, Inserted FullName Column, Inserted Text Before Delimiter, Renamed Columns, Inserted Text Between Delimiter, Renamed Columns1, Replaced Value, Capitalized Each Word1, Inserted Year, Renamed Columns2, Added Conditional Column, Filtered Rows, and Removed Columns.



QUERY EDITING TOOLS

The **HOME** tab includes **general settings** and **common table transformation tools**

The screenshot shows the Power BI ribbon with the 'Home' tab selected. The tabs across the top are File, Home, Transform, Add Column, View, Tools, and Help. Under the Home tab, there are sections for Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Properties, Manage, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, Data Type (Whole Number), Use First Row as Headers, Merge Queries, Append Queries, Combine Files, Combine, and AI Insights.

The **TRANSFORM** tab includes tools to **modify existing columns** (splitting/grouping, transposing, extracting text, etc.)

The screenshot shows the Power BI ribbon with the 'Transform' tab selected. The tabs across the top are File, Home, Transform, Add Column, View, Tools, and Help. Under the Transform tab, there are sections for Group By, Use First Row as Headers, Transpose, Reverse Rows, Count Rows, Detect Data Type, Rename, Data Type (Whole Number), Replace Values, Unpivot Columns, Fill, Move, Pivot Column, Convert to List, Split Column, Format, Extract, Parse, Statistics (Standard, Scientific, Information), Trigonometry, Rounding, Date, Time, Duration, Run R script, and Run Python script.

The **ADD COLUMN** tools **create new columns** (based on conditional rules, text operations, calculations, dates, etc.)

The screenshot shows the Power BI ribbon with the 'Add Column' tab selected. The tabs across the top are File, Home, Transform, Add Column, View, Tools, and Help. Under the Add Column tab, there are sections for Column From Examples, Custom Column, Invoke Custom Function, Conditional Column, Index Column, Duplicate Column, Format (From Text, From Number, From Date & Time), Merge Columns, Statistics (Standard, Scientific, Information), Trigonometry, Rounding, Date, Time, Duration, Text Analytics, Vision, Azure Machine Learning, and AI Insights.



BASIC TABLE TRANSFORMATIONS

The screenshot shows the Power BI desktop ribbon with several transformation tools highlighted by yellow arrows:

- Sort values (A-Z, Low-High, etc.)**: Points to the Sort button in the Transform ribbon tab.
- Change data type (date, \$, %, text, etc.)**: Points to the Data Type dropdown in the Transform ribbon tab.
- Promote headers**: Points to the Use First Row as Headers dropdown in the Transform ribbon tab.
- Choose or remove columns**: Points to the Manage Columns button in the Transform ribbon tab, which opens a context menu:
 - Remove Columns
 - Remove Other Columns
- Keep or remove rows**: Points to the Remove Rows button in the Transform ribbon tab, which opens a context menu:
 - Remove Top Rows
 - Remove Bottom Rows
 - Remove Alternate Rows
 - Remove Duplicates
 - Remove Blank Rows
 - Remove Errors
- Duplicate, move or rename columns**: Points to the context menu for a column header, which includes options like Copy, Remove, and Change Type.
- Tip: Right-click column headers to access common tools**: Provides a tip for using the context menu for column headers.

Transform ribbon tab

- File
- Home
- Transform
- Add Column
- View
- Tools
- Help

Transform ribbon tab buttons

- Close & Apply
- New Source
- Recent Sources
- Enter Data
- Data source settings
- Manage Parameters
- Refresh Preview
- Advanced Editor
- Properties
- Manage
- Query
- Choose Columns
- Remove Columns
- Keep Rows
- Remove Rows
- Reduce Rows
- Sort
- Split Column
- Group By
- Data Type: Whole Number
- Use First Row as Headers
- Merge Queries
- Append Queries
- Combine Files
- Combine
- AI Insights

Context menu for a column header (e.g., Order)

- Copy
- Remove
- Remove Other Columns
- Duplicate Column
- Add Column From Examples...
- Remove Duplicates
- Remove Errors
- Change Type
- Transform
- Replace Values...
- Replace Errors...
- Group By...
- Fill
- Unpivot Columns
- Unpivot Other Columns
- Unpivot Only Selected Columns
- Rename...
- Move
- Drill Down
- Add as New Query

ASSIGNMENT: TABLE TRANSFORMATIONS





NEW MESSAGE

From: **Ethan T. Langer (Analytics Manager)**
Subject: **Welcome aboard!**

Hello, and welcome to the team! We're excited that you'll be helping us develop our new internal reports in Power BI. Looks like you've already gotten started, but we have some new data to add to the model. Could you please create two new queries to connect to the **Product Category Lookup** and **Product Subcategory Lookup** files attached, and help with a few modifications to the product table? Thanks!
-ETL

[Product Category Lookup](#)
[Product Subcategory Lookup](#)

[Reply](#) [Forward](#)

Key Objectives

1. Create queries to connect to the two new .csv files
2. Name your queries **Product Category Lookup** and **Product Subcategory Lookup**
3. Confirm that column headers have been promoted and that all data types are correct
4. Add a new column to extract all characters before the dash (" - ") in the **Product SKU** column, and name it "**SKU Type**"
5. Update the **SKU Type** calculation above to return all characters before *second* dash, instead of the first
6. Replace zeros (**0**) in the **Product Style** column with "**NA**"
7. Close and load to your data model



SOLUTION: TABLE TRANSFORMATIONS

  NEW MESSAGE

From: **Ethan T. Langer (Analytics Manager)**

Subject: **Welcome aboard!**

Hello, and welcome to the team!

We're excited that you'll be helping us develop our new internal reports in Power BI. Looks like you've already gotten started, but we have some new data to add to the model.

Could you please create two new queries to connect to the **Product Category Lookup** and **Product Subcategory Lookup** files attached, and help with a few modifications to the product table?

Thanks!

-ETL

 [Product Category Lookup](#)
[Product Subcategory Lookup](#)

Solution Preview

File Home Transform Add Column View Tools Help

New Source Recent Enter Data Data source settings Data Sources Manage Parameters Refresh Advanced Editor Properties Manage Columns Manage Rows Split Column By Group Replace Values Transform

Quer es [] = Table.TransformColumnTypes

Territory Lookup Product Lookup Product Category Lookup Product Subcategories Lookup

| 1 | 2 | 3 | 4 |
|---|-------------|---|---|
| 1 | Bikes | | |
| 2 | Components | | |
| 3 | Clothing | | |
| 4 | Accessories | | |

PROPERTIES
Name: ProductCategoryLookup
All Properties

APPLIED STEPS
Source: ProductCategoryLookup
Promoted Headers
Changed Type

= Table.ReplaceValue(#"Renamed Columns","0","NA",Replacer.ReplaceText,

| ProductStyle | ProductCost | ProductPrice | SKU Type |
|--------------|-------------|--------------|----------|
| | 13.09 | 34.99 | HL-U509 |
| 1 | 12.03 | 33.64 | HL-U509 |
| 2 | 3.40 | 9.50 | SO-B909 |
| 3 | 3.40 | 9.50 | SO-B909 |
| 4 | 12.03 | 33.64 | HL-U509 |
| 5 | 5.71 | 8.64 | CA-1098 |
| 6 | 31.72 | 48.07 | LI-0192 |
| 7 | 31.72 | 48.07 | LI-0192 |
| 8 | 31.72 | 48.07 | LI-0192 |
| 9 | 31.72 | 48.07 | LI-0192 |
| 10 | 31.72 | 48.07 | LI-0192 |
| 11 | 747.97 | 1,263.46 | FR-R92R |
| 12 | 747.97 | 1,263.46 | FR-R92R |
| 13 | 747.97 | 1,263.46 | FR-R92R |
| 14 | 747.97 | 1,263.46 | FR-R92R |

PROPERTIES
Name: Product Lookup
All Properties

APPLIED STEPS
Source: Product Lookup
Promoted Headers
Changed Type
Changed to Currency
Removed Columns
Sorted Rows
Inserted Text Before Delimiter
Renamed Columns
Replaced Value



PRO TIP: STORAGE & CONNECTION MODES

Power BI Desktop supports several types of **storage** and **connection modes**:

- **Import:** Tables are stored in-memory within Power BI and queries are fulfilled by cached data (*default*)
- **DirectQuery:** Tables are connected directly to the source and queries are executed on-demand at the data source
- **Composite Model (Dual):** Tables come from a mix of Import and DirectQuery modes, or integrate multiple DirectQuery tables
- **Live Connection:** Connect to pre-published Power BI datasets in Power BI Service or Azure Analysis Services



Import

- ✓ Dataset is less than 1GB (after compression) & fast performance
- ✓ Source data does not change frequently
- ✓ No restrictions on Power Query, data modeling, and DAX functions



DirectQuery

- ✓ Dataset is too large to be stored in-memory
- ✓ Source data changes frequently and reports must reflect changes
- ✓ Company policy states that data can only be accessed from the original source



Composite Model

- ✓ Boost performance by setting appropriate storage for each table
- ✓ Combine a DirectQuery model with additional imported data
- ✓ Create a single model from two or more DirectQuery models



Live Connection

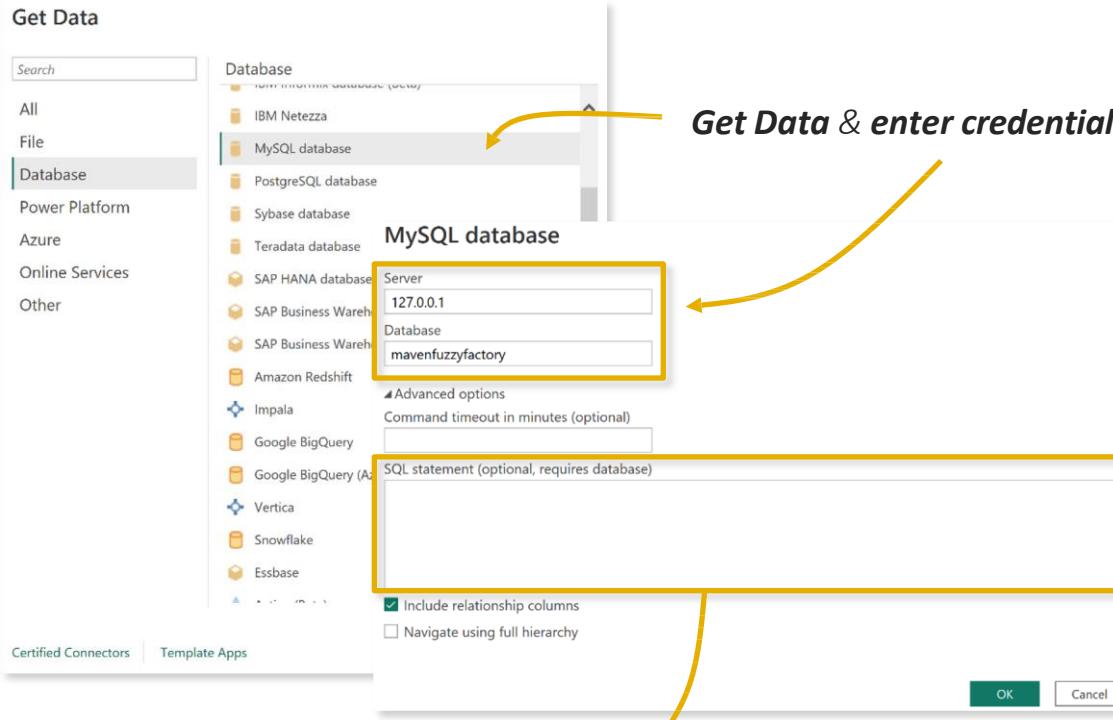
- ✓ Create one dataset that serves as a central source of truth
- ✓ Analyst teams can create different reports from the same source
- ✓ Multi-developer teams where one user builds the model and another works on visualization

Learn more: <https://learn.microsoft.com/en-us/power-bi/connect-data/service-dataset-modes-understand>



CONNECTING TO A DATABASE

Power Query can connect to data from various **database sources** including SQL Server, MS Access, MySQL, PostgreSQL, Oracle, SAP, and more



Select tables & transform

| website_session_id | created_at | user_id | is_repeat_session | utm_source | utm_campaign |
|--------------------|-------------------------|---------|-------------------|------------|--------------|
| 1 | 3/19/2012, 8:04:16 AM | 1 | 0 | gsearch | nonbrand |
| 2 | 3/19/2012, 8:16:49 AM | 2 | 0 | gsearch | nonbrand |
| 3 | 3/19/2012, 8:26:55 AM | 3 | 0 | gsearch | nonbrand |
| 4 | 3/19/2012, 8:37:33 AM | 4 | 0 | gsearch | nonbrand |
| 5 | 3/19/2012, 9:00:55 AM | 5 | 0 | gsearch | nonbrand |
| 6 | 3/19/2012, 9:05:46 AM | 6 | 0 | gsearch | nonbrand |
| 7 | 3/19/2012, 9:06:27 AM | 7 | 0 | gsearch | nonbrand |
| 8 | 3/19/2012, 9:17:17 AM | 8 | 0 | gsearch | nonbrand |
| 9 | 3/19/2012, 9:27:56 AM | 9 | 0 | gsearch | nonbrand |
| 10 | 3/19/2012, 9:35:37 AM | 10 | 0 | gsearch | nonbrand |
| 11 | 3/19/2012, 9:37:42 AM | 11 | 0 | gsearch | nonbrand |
| 12 | 3/19/2012, 9:39:57 AM | 12 | 0 | gsearch | nonbrand |
| 13 | 3/19/2012, 9:45:29 AM | 13 | 0 | gsearch | nonbrand |
| 14 | 3/19/2012, 9:45:47 AM | 14 | 0 | gsearch | nonbrand |
| 15 | 3/19/2012, 9:57:14 AM | 15 | 0 | gsearch | nonbrand |
| 16 | 3/19/2012, 10:01:35 ... | 16 | 0 | gsearch | nonbrand |
| 17 | 3/19/2012, 10:14:34 ... | 17 | 0 | gsearch | nonbrand |
| 18 | 3/19/2012, 10:16:57 ... | 18 | 0 | gsearch | nonbrand |
| 19 | 3/19/2012, 10:17:38 ... | 19 | 0 | gsearch | nonbrand |

Write custom or advanced queries with SQL statements (optional)



EXTRACTING DATA FROM THE WEB

Power Query includes a native **Web connector** for importing web-hosted files (csv, xlsx, etc.) or scraping URLs for anything that Power Query can identify as a structured table

List of asset management firms

From Wikipedia, the free encyclopedia

"Asset management company" redirects here. The term may also refer to a [bad bank](#).

An **asset management company** (AMC) is an asset management / investment management company/firm that invests the pooled funds of retail investors in securities in line with the stated investment objectives. For a fee, the company/firm provides more diversification, liquidity, and professional management consulting service than is normally available to individual investors. The diversification of portfolio is done by investing in such securities which are inversely correlated to each other. Money is collected from investors by way of floating various collective investment schemes, e.g. mutual fund schemes. In general, an AMC is a company that is engaged primarily in the business of investing in, and managing, portfolios of securities. A study by consulting firm Casey Quirk, which is owned by Deloitte, found that asset management firms ended 2020 with record highs in both revenue and assets under management.^[1]

Largest companies [edit]

The following is a list of the top 20 asset managers in the world (as of 2022), ranked by total assets under management (AUM)^[2]:

| Rank | Firm/company | Country | AUM (billion USD) |
|------|------------------------------|----------------|-------------------|
| 1 | BlackRock | United States | 9.570 |
| 2 | Vanguard Group | United States | 8.100 |
| 3 | Fidelity Investments | United States | 4.283 |
| 4 | UBS | Switzerland | 4.380 |
| 5 | State Street Global Advisors | United States | 4.020 |
| 6 | Morgan Stanley | United States | 3.230 |
| 7 | JPMorgan Chase | United States | 2.960 |
| 8 | Crédit Agricole | France | 2.875 |
| 9 | Allianz | Germany | 2.760 |
| 10 | Capital Group | United States | 2.700 |
| 11 | Goldman Sachs | United States | 2.394 |
| 12 | BNY Mellon | United States | 2.266 |
| 13 | Amundi | France | 2.251 |
| 14 | PIMCO | United States | 2.000 |
| 15 | Legal & General | United Kingdom | 1.866 |
| 16 | Edward Jones Investments | United States | 1.700 |
| 17 | PGIM | United States | 1.620 |
| 18 | Deutsche Bank | Germany | 1.615 |
| 19 | Bank of America | United States | 1.571 |
| 20 | Invesco | United States | 1.556 |

https://en.wikipedia.org/wiki/List_of_asset_management_firms

Navigator

Display Options

- HTML Tables [8]
 - Largest companies[edit]
 - Table 1
 - Table 2
 - Table 3
 - Table 4
 - Table 5
 - Table 6
 - Table 7
- Suggested Tables [4]
 - Table 8
 - Table 9
 - Table 10
 - Table 11
- Text [2]
 - HTML Code
 - Displayed Text

Table View Web View

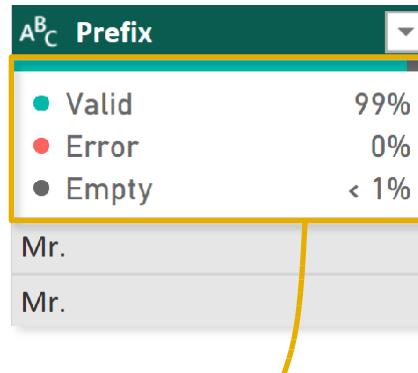
Largest companies[edit]

| Rank | Firm/company | Country | AUM (billion USD) |
|------|------------------------------|----------------|-------------------|
| 1 | BlackRock | United States | 10010 |
| 2 | Charles Schwab | United States | 8140 |
| 3 | Vanguard Group | United States | 8100 |
| 4 | UBS | Switzerland | 4380 |
| 5 | Fidelity Investments | United States | 4283 |
| 6 | State Street Global Advisors | United States | 4020 |
| 7 | Morgan Stanley | United States | 3230 |
| 8 | JPMorgan Chase | United States | 2960 |
| 9 | Allianz | Germany | 2760 |
| 10 | Capital Group | United States | 2700 |
| 11 | Goldman Sachs | United States | 2394 |
| 12 | BNY Mellon | United States | 2266 |
| 13 | Amundi | France | 2251 |
| 14 | PIMCO | United States | 2000 |
| 15 | Legal & General | United Kingdom | 1866 |
| 16 | Prudential Financial | United States | 1620 |
| 17 | Deutsche Bank | Germany | 1615 |
| 18 | Bank of America | United States | 1571 |
| 19 | Invesco | United States | 1556 |
| 20 | T. Rowe Price | United States | 1552 |

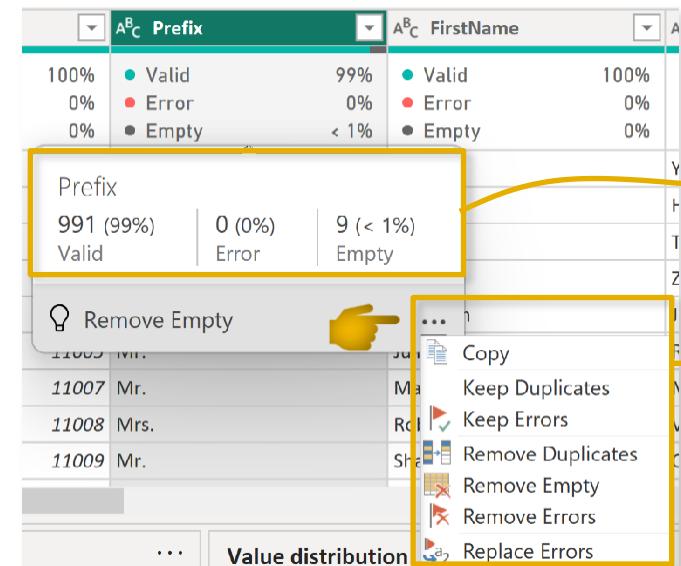


DATA PROFILING: COLUMN QUALITY

Profiling tools like **column quality**, **column distribution**, and **column profile** allow you to explore the quality, composition, and distribution of your data before loading it into the Power BI front-end



Column quality shows the percentage of values within a column that are **valid**, contain **errors**, or are **empty**



Hover over the column quality box to see the **number of records** in each category

Click the **options menu** to remove duplicates, errors or empty values

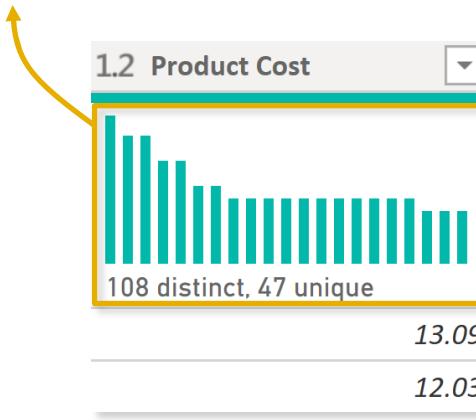


PRO TIP: Profiling tools are a great way to **quickly find and address common data quality issues in one place**, instead of having to manually apply multiple tools or filters



DATA PROFILING: COLUMN DISTRIBUTION

Column distribution provides a sample distribution of the data in a column



Suggested action based on column distribution results

Order Date

911 (1%) Distinct

3 (< 1%) Unique

Remove Duplicates

...
Copy
Keep Duplicates
Keep Errors
Remove Duplicates
Remove Empty
Remove Errors
Replace Errors

| | Order Date | Stock Date | O |
|----|------------|------------|---|
| 12 | 12/13/2021 | SO612 | |
| 11 | 9/24/2021 | SO612 | |
| 10 | 9/4/2021 | SO612 | |
| 9 | 9/28/2021 | SO613 | |
| 8 | 10/21/2021 | SO613 | |
| 7 | 1/1/2022 | SO612 | |
| 6 | 1/1/2022 | SO612 | |
| 5 | 1/1/2022 | SO612 | |
| 4 | 1/1/2022 | SO612 | |
| 3 | 1/1/2022 | SO612 | |
| 2 | 1/1/2022 | SO612 | |
| 1 | 1/1/2022 | SO612 | |

Hover over the column quality box to see the **number of distinct & unique records**

Click the **options menu** to remove duplicates, errors or empty values



DATA PROFILING: COLUMN PROFILE

Column profile provides a more holistic view of the data in a column, including a sample distribution and profiling statistics

Column statistics provide more detailed profiling metrics, including:

Count = 293

(total number of values in column)

Distinct Count = 119

(total number of distinct values, whether they appear once or multiple times)

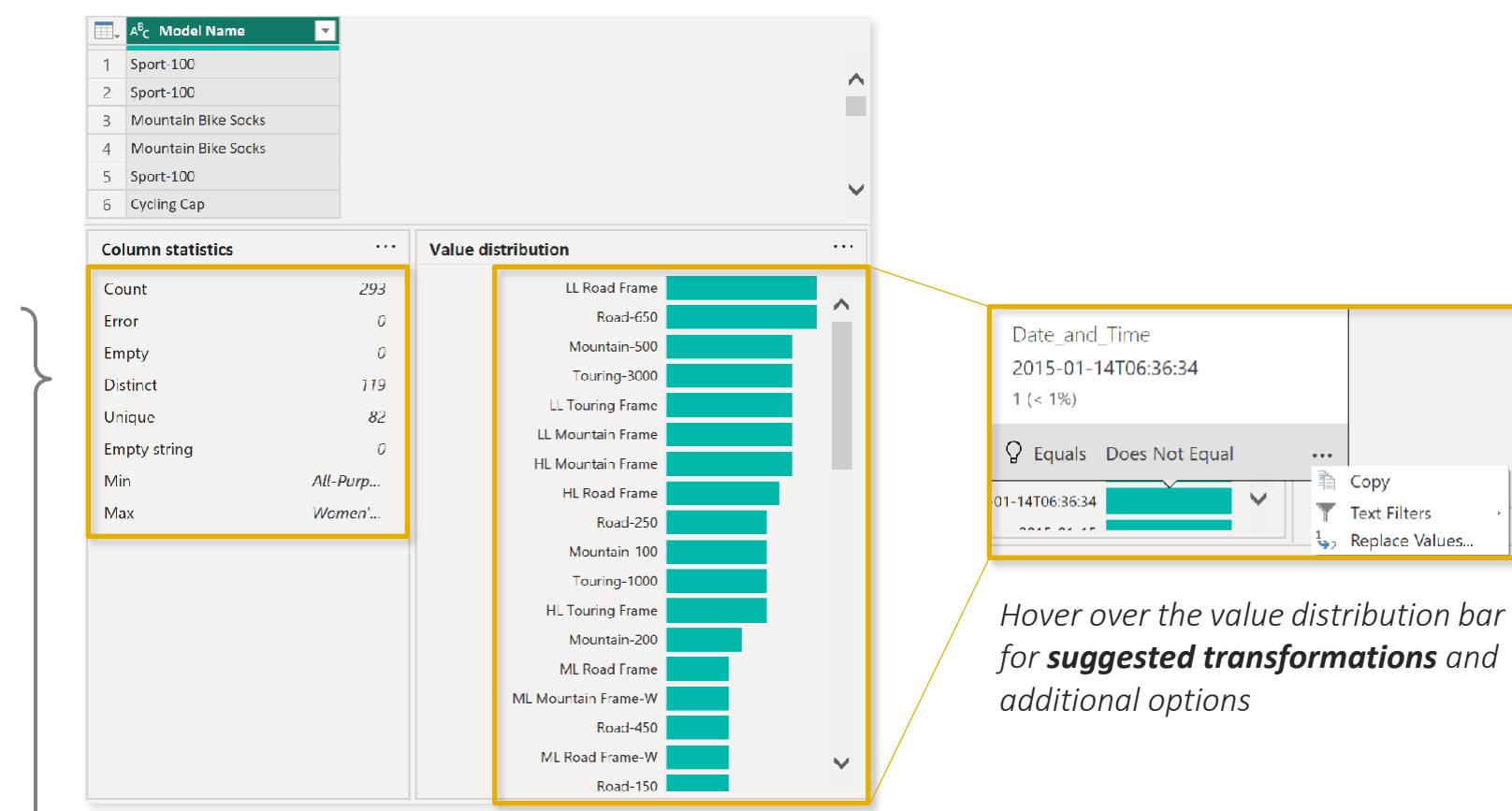
Unique = 82

(total number of values that appear exactly once)

Min & Max

(lowest and highest observed values)

Note: Typically only useful for numerical values





TEXT TOOLS

The screenshot shows the Power BI ribbon with the 'Transform' tab selected. The ribbon includes tabs for File, Home, Transform, Add Column, View, Tools, and Help. Under the Transform tab, there are several icons: Group By, Use First Row as Headers, Transpose, Reverse Rows, Detect Data Type, Rename, Data Type: Date, Replace Values, Unpivot Columns, Fill, Move, Pivot Column, Convert to List, and Any Column.

A yellow box highlights the 'Split Column' option in the Transform ribbon. A callout box below it lists various splitting methods:

- By Delimiter
- By Number of Characters
- By Positions
- By Lowercase to Uppercase
- By Uppercase to Lowercase
- By Digit to Non-Digit
- By Non-Digit to Digit

Split a text column based on a specific delimiter, number of characters, or other attributes

A yellow box highlights the 'Format' option in the Transform ribbon. A callout box below it lists text formatting options:

- lowercase
- UPPERCASE
- Capitalize Each Word
- Trim
- Clean
- Add Prefix
- Add Suffix

A yellow box highlights the 'Text Column' option in the Transform ribbon. A callout box to its right lists text extraction methods:

- Length
- First Characters
- Last Characters
- Range
- Text Before Delimiter
- Text After Delimiter
- Text Between Delimiters

Extract characters from text based on fixed lengths, first/last characters, ranges or delimiters

HEY THIS IS IMPORTANT!

You can access many tools from both the **Transform** and **Add Column** menus - the difference is whether you want to **ADD** a new column or **OVERWRITE** an existing one

Format a text column to upper, lower or proper case, or add a prefix or suffix

Tip: Use "Trim" to eliminate leading & trailing spaces, or "Clean" to remove non-printable characters



ASSIGNMENT: TEXT TOOLS

  NEW MESSAGE

From: **Ethan T. Langer** (*Analytics Manager*)
Subject: **Customer domains**

Hi!
We're looking to better understand where our customers may be coming from, based on their email domains.
Could you please create a new column in the customer table that will allow us do this?

Thanks!
-ETL

Reply Forward

Key Objectives

1. Duplicate the email address column and name it **“Domain Name”**
2. In the new column, remove all text/characters except for the domain name
3. Use transformation steps to clean up and capitalize the domain names (i.e. **“Adventure Works”**)
4. Save & Apply changes

SOLUTION: TEXT TOOLS





NEW MESSAGE

From: **Ethan T. Langer** (*Analytics Manager*)
Subject: **Customer domains**

Hi!

We're looking to better understand where our customers may be coming from, based on their email domains.

Could you please create a new column in the customer table that will allow us do this?

Thanks!
-ETL

Solution Preview

► PROPERTIES

Name

All Properties

► APPLIED STEPS

- Source
- Promoted Headers
- Changed Type
- Changed Type1
- Capitalized Each Word
- Customer Full Name
- Duplicated Column
- Renamed Columns
- Extracted Text After Delimiter
- Extracted Text Before Delimiter
- Replaced Value
- Capitalized Each Word1



NUMERICAL TOOLS

The screenshot shows the Power BI ribbon with the 'Transform' tab selected. The 'Number Column' section of the ribbon is highlighted with a yellow box and a callout pointing to a separate box containing 'Is Even', 'Is Odd', and 'Sign' tools. Below the ribbon, four groups of tools are shown in boxes:

- Statistics**: Sum, Minimum, Maximum, Median, Average, Standard Deviation, Count Values, Count Distinct Values.
- Standard**: Add, Multiply, Subtract, Divide, Integer-Divide, Modulo, Percentage, Percent Of.
- Scientific**: Absolute Value, Power, Square Root, Exponent, Logarithm, Factorial.
- Trigonometry**: Sine, Cosine, Tangent, Arcsine, Arccosine, Arctangent.

Statistics functions allow you to evaluate basic stats for a selected column (sum, min/max, average, count, count distinct, etc.)

Note: These tools return a **SINGLE** value, and are commonly used to explore a table rather than prepare it for loading

Standard, Scientific and Trigonometry tools allow you to apply standard operations (addition, multiplication, division, etc.) or more advanced calculations (power, logarithm, sine, tangent, etc.) to each value in a column

Note: Unlike the Statistics tools, these are applied to each row in the table

Information tools allow you to define binary flags (1/0 or TRUE/FALSE) to mark rows as even, odd, positive or negative



ASSIGNMENT: NUMERICAL TOOLS



 NEW MESSAGE

From: **Ethan T. Langer** (*Analytics Manager*)
Subject: **Need some stats for leadership**

Hi again,

Leadership is asking us to validate some high-level stats about our products and customers. Can you please help me answer the following questions?

We don't really need to store these values anywhere, so make sure to restore the tables back to their original state once you're done pulling the stats.

Thank you!
-ETL

[Reply](#) [Forward](#)

Key Objectives

1. What is our average product cost?
2. How many colors do we sell our products in?
3. How many distinct customers do we have?
4. What is the maximum annual customer income?
5. Return the tables to their original state



SOLUTION: NUMERICAL TOOLS

  NEW MESSAGE

From: **Ethan T. Langer** (Analytics Manager)
Subject: **Need some stats for leadership**

Hi again,

Leadership is asking us to validate some high-level stats about our products and customers. Can you please help me answer the following questions?

We don't really need to store these values anywhere, so make sure to restore the tables back to their original state once you're done pulling the stats.

Thank you!
-ETL

Reply Forward

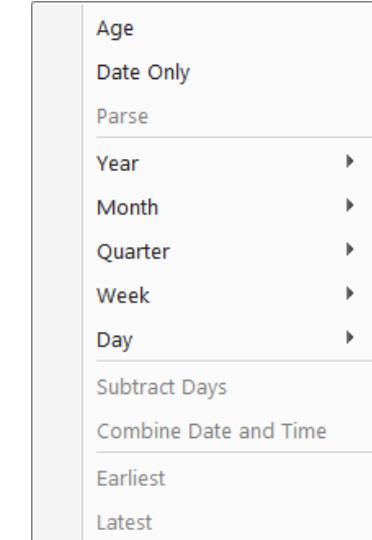
Solution Preview

1. What is our average product cost? (**\$413.66**)
2. How many colors do we sell our products in? (**10**)
3. How many distinct customers do we have? (**18,148**)
4. What is the maximum annual customer income? (**\$170k**)
5. Return the tables to their original state



DATE & TIME TOOLS

The screenshot shows the Power BI ribbon with the "Add Column" tab selected. The "Date & Time" tools group is highlighted with a yellow box. It contains three icons: "Date", "Time", and "Duration". Below these icons is a dropdown menu labeled "From Date & Time" which is also highlighted with a yellow box. A yellow arrow points from the "From Date & Time" dropdown to a detailed view of its options.



Date & Time tools are relatively straight-forward, and include the following options:

- **Age**: Difference between the current date and the date in each row
- **Date Only**: Removes the time component from a date/time field
- **Year/Month/Quarter/Week/Day**: Extracts individual components from a date field (time-specific options include Hour, Minute, Second, etc.)
- **Earliest/Latest**: Evaluates the earliest or latest date from a column as a single value (can only be accessed from the “Transform” menu)

Note: You will almost always want to perform these operations from the “Add Column” menu to build out new fields, rather than transforming an individual date/time column

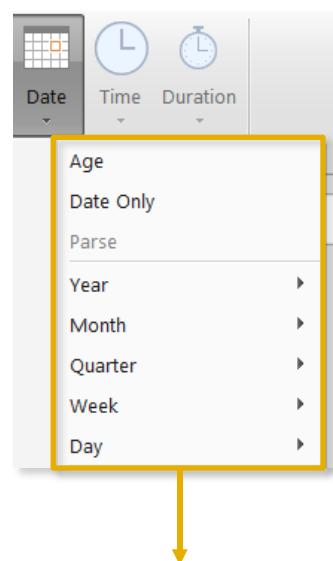


PRO TIP: Load up a table containing a **single date column** and use Date tools to build out an **entire calendar table**



CREATING A CALENDAR TABLE

| | Date |
|----|-----------|
| 1 | 1/1/2020 |
| 2 | 1/2/2020 |
| 3 | 1/3/2020 |
| 4 | 1/4/2020 |
| 5 | 1/5/2020 |
| 6 | 1/6/2020 |
| 7 | 1/7/2020 |
| 8 | 1/8/2020 |
| 9 | 1/9/2020 |
| 10 | 1/10/2020 |
| 11 | 1/11/2020 |
| 12 | 1/12/2020 |
| 13 | 1/13/2020 |
| 14 | 1/14/2020 |
| 15 | 1/15/2020 |
| 16 | 1/16/2020 |
| 17 | 1/17/2020 |
| 18 | 1/18/2020 |
| 19 | 1/19/2020 |
| 20 | 1/20/2020 |
| 21 | 1/21/2020 |
| 22 | 1/22/2020 |
| 23 | 1/23/2020 |
| 24 | 1/24/2020 |
| 25 | 1/25/2020 |
| 26 | 1/26/2020 |
| 27 | 1/27/2020 |
| 28 | 1/28/2020 |



Use the **Date** options in the **Add Column** menu to quickly build out an entire calendar table from a list of dates

| Date | Day Name | Start of Week | Start of Month | Month Name |
|-----------|-----------|---------------|----------------|------------|
| 1/1/2020 | Wednesday | 12/29/2019 | 1/1/2020 | January |
| 1/2/2020 | Thursday | 12/29/2019 | 1/1/2020 | January |
| 1/3/2020 | Friday | 12/29/2019 | 1/1/2020 | January |
| 1/4/2020 | Saturday | 12/29/2019 | 1/1/2020 | January |
| 1/5/2020 | Sunday | 1/5/2020 | 1/1/2020 | January |
| 1/6/2020 | Monday | 1/5/2020 | 1/1/2020 | January |
| 1/7/2020 | Tuesday | 1/5/2020 | 1/1/2020 | January |
| 1/8/2020 | Wednesday | 1/5/2020 | 1/1/2020 | January |
| 1/9/2020 | Thursday | 1/5/2020 | 1/1/2020 | January |
| 1/10/2020 | Friday | 1/5/2020 | 1/1/2020 | January |
| 1/11/2020 | Saturday | 1/5/2020 | 1/1/2020 | January |
| 1/12/2020 | Sunday | 1/12/2020 | 1/1/2020 | January |
| 1/13/2020 | Monday | 1/12/2020 | 1/1/2020 | January |
| 1/14/2020 | Tuesday | 1/12/2020 | 1/1/2020 | January |
| 1/15/2020 | Wednesday | 1/12/2020 | 1/1/2020 | January |
| 1/16/2020 | Thursday | 1/12/2020 | 1/1/2020 | January |
| 1/17/2020 | Friday | 1/12/2020 | 1/1/2020 | January |
| 1/18/2020 | Saturday | 1/12/2020 | 1/1/2020 | January |
| 1/19/2020 | Sunday | 1/19/2020 | 1/1/2020 | January |
| 1/20/2020 | Monday | 1/19/2020 | 1/1/2020 | January |
| 1/21/2020 | Tuesday | 1/19/2020 | 1/1/2020 | January |
| 1/22/2020 | Wednesday | 1/19/2020 | 1/1/2020 | January |
| 1/23/2020 | Thursday | 1/19/2020 | 1/1/2020 | January |
| 1/24/2020 | Friday | 1/19/2020 | 1/1/2020 | January |
| 1/25/2020 | Saturday | 1/19/2020 | 1/1/2020 | January |
| 1/26/2020 | Sunday | 1/26/2020 | 1/1/2020 | January |
| 1/27/2020 | Monday | 1/26/2020 | 1/1/2020 | January |
| 1/28/2020 | Tuesday | 1/26/2020 | 1/1/2020 | January |



CHANGE TYPE WITH LOCALE

| | A ^B C Date |
|----|-------------------------------|
| 1 | 1.2 Decimal Number |
| 2 | \$ Fixed decimal number |
| 3 | 1 ² 3 Whole Number |
| 4 | % Percentage |
| 5 | Date/Time |
| 6 | Date |
| 7 | Time |
| 8 | Date/Time/Timezone |
| 9 | Duration |
| 10 | A ^B C Text |
| 11 | True/False |
| 12 | Binary |
| 13 | Using Locale... |
| 14 | |

Change Type with Locale

Change the data type and select the locale of origin.

Data Type
Date
Locale
English (United States)

Sample input values:
3/29/2016
Tuesday, March 29, 2016
March 29
March 2016

| | Date |
|----|-----------|
| 1 | 1/1/2023 |
| 2 | 2/1/2023 |
| 3 | 3/1/2023 |
| 4 | 4/1/2023 |
| 5 | 5/1/2023 |
| 6 | 6/1/2023 |
| 7 | 7/1/2023 |
| 8 | 8/1/2023 |
| 9 | 9/1/2023 |
| 10 | 10/1/2023 |
| 11 | 11/1/2023 |
| 12 | 12/1/2023 |
| 13 | Error |
| 14 | Error |

| | Date |
|----|-----------|
| 1 | 1/1/2020 |
| 2 | 1/2/2020 |
| 3 | 1/3/2020 |
| 4 | 1/4/2020 |
| 5 | 1/5/2020 |
| 6 | 1/6/2020 |
| 7 | 1/7/2020 |
| 8 | 1/8/2020 |
| 9 | 1/9/2020 |
| 10 | 1/10/2020 |
| 11 | 1/11/2020 |
| 12 | 1/12/2020 |
| 13 | 1/13/2020 |
| 14 | 1/14/2020 |

1) Left click the data type icon in the column header and select the **Using Locale...** option

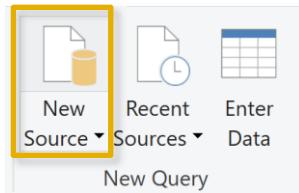
2) Select **Date** as the data type and **English (United States)** as the locale for all datasets in this course (regardless of your actual location)

3) Confirm that the **data type is correctly recognized**. You should see a calendar icon next to the column name in the header and no errors in the column

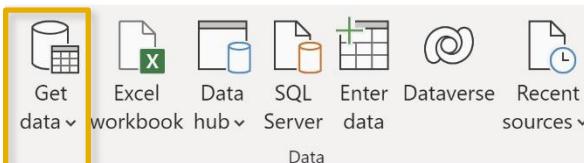


PRO TIP: ROLLING CALENDARS

- 1 Create a new **blank query** & name it “*Rolling Calendar*”

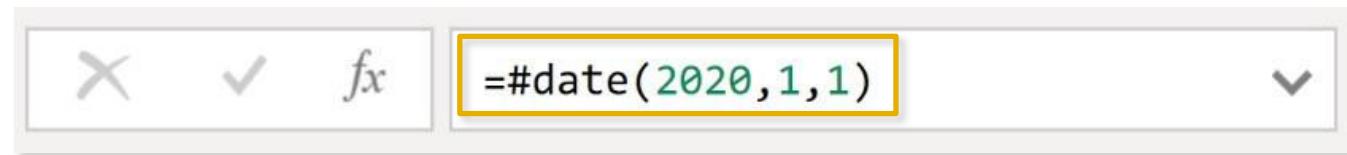


Power Query: New Source > Blank Query



Front end: Get Data > Blank Query

- 2 In the formula bar, type a “**literal**” to generate a start date:



Format as: YYYY, MM, DD

- 3 Click the **fx** icon to **add a custom step**, and enter the following formula to generate a list of dates between the start date and the current day:

```
= List.Dates(  
    Source,  
    Number.From(DateTime.LocalNow()) - Number.From(Source),  
    #duration(1, 0, 0, 0)  
)
```

Note: If your first applied step is named something other than “**Source**”, use that name in your formula (this is common for non-US users)



PRO TIP: ROLLING CALENDARS

4

Convert the resulting list into a **Table** and set the data type as a **Date**

The screenshot shows the 'List Tools' ribbon in Power BI. The 'File' tab is selected. In the 'Tools' section, the 'To Table' button is highlighted with a yellow arrow. Below it, a dropdown menu titled 'Column1' lists various data types: Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time, Date, Time, Date/Time/Timezone, Duration, Text, True/False, Binary, and Using Locale... The 'Date' option is highlighted with a yellow box.

5

Rename the column to “**Date**” and add calculated date columns (year, month, quarter, etc.) using the **Add Column** tools

| | Date | Year | Start of Quarter | Start of Month |
|----|-----------|------|------------------|----------------|
| 1 | 1/1/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 2 | 1/2/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 3 | 1/3/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 4 | 1/4/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 5 | 1/5/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 6 | 1/6/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 7 | 1/7/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 8 | 1/8/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 9 | 1/9/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 10 | 1/10/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 11 | 1/11/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 12 | 1/12/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 13 | 1/13/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 14 | 1/14/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 15 | 1/15/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 16 | 1/16/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 17 | 1/17/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 18 | 1/18/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 19 | 1/19/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 20 | 1/20/2020 | 2020 | 1/1/2020 | 1/1/2020 |
| 21 | 1/21/2020 | 2020 | 1/1/2020 | 1/1/2020 |



ASSIGNMENT: CALENDAR TABLES

  NEW MESSAGE

From: **Ethan T. Langer** (*Analytics Manager*)
Subject: **New date fields**

Hi,

We need to add a few fields to our calendar table to help us analyze sales trending over time.

Could you please add the following columns when you get a chance?

Thanks!
-ETL

Reply Forward

Key Objectives

Add the following columns to the calendar table:

1. **Month Name** (e.g. “January”)
2. **Month Number** (e.g. “1”)
3. **Start of Year** (e.g. “1/1/2020”)
4. **Year** (e.g. “2020”)

SOLUTION: CALENDAR TABLES



  NEW MESSAGE

From: **Ethan T. Langer** (*Analytics Manager*)
Subject: **New date fields**

Hi,

We need to add a few fields to our calendar table to help us analyze sales trending over time.

Could you please add the following columns when you get a chance?

Thanks!
-ETL

Solution Preview

◀ PROPERTIES

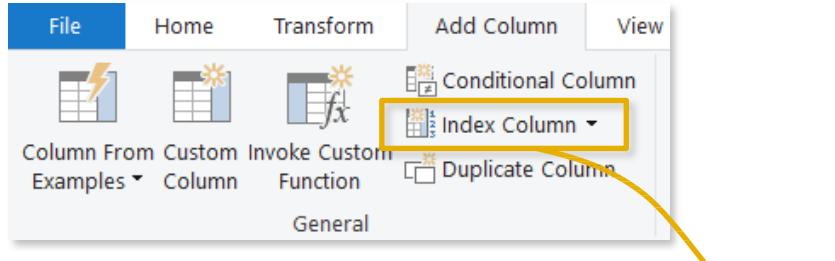
Name _____

1 APPLIED STEPS

- Source
Promoted Headers
Changed Type
Inserted Day Name
Inserted Start of Week
Inserted Start of Month
Inserted Month Name
Inserted Start of Year
Inserted Year
Inserted Month
Renamed Columns



INDEX COLUMNS



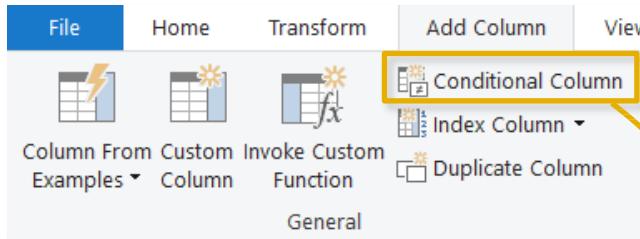
Index Columns contain a list of sequential values that can be used to identify each unique row in a table (*typically starting from 0 or 1*)

These are often used to create **unique IDs** that can be used to form relationships between tables (*more on that later!*)

| Index | Order Date | Stock Date | Order Number | Product Key |
|-------|------------|------------|--------------|-------------|
| 1 | 1/1/2020 | 9/21/2019 | SO45080 | 332 |
| 2 | 1/1/2020 | 12/5/2019 | SO45079 | 312 |
| 3 | 1/1/2020 | 10/29/2019 | SO45082 | 350 |
| 4 | 1/1/2020 | 11/16/2019 | SO45081 | 338 |
| 5 | 1/2/2020 | 12/15/2019 | SO45083 | 312 |
| 6 | 1/2/2020 | 10/12/2019 | SO45084 | 310 |
| 7 | 1/2/2020 | 12/18/2019 | SO45086 | 314 |
| 8 | 1/2/2020 | 10/9/2019 | SO45085 | 312 |
| 9 | 1/3/2020 | 10/3/2019 | SO45093 | 312 |
| 10 | 1/3/2020 | 9/29/2019 | SO45090 | 310 |
| 11 | 1/3/2020 | 12/11/2019 | SO45088 | 345 |
| 12 | 1/3/2020 | 10/24/2019 | SO45092 | 313 |
| 13 | 1/3/2020 | 12/16/2019 | SO45089 | 351 |
| 14 | 1/3/2020 | 10/26/2019 | SO45091 | 314 |
| 15 | 1/3/2020 | 9/11/2019 | SO45087 | 350 |
| 16 | 1/3/2020 | 9/11/2019 | SO45094 | 310 |
| 17 | 1/4/2020 | 10/30/2019 | SO45096 | 312 |
| 18 | 1/4/2020 | 10/30/2019 | SO45097 | 313 |
| 19 | 1/4/2020 | 9/15/2019 | SO45098 | 310 |
| 20 | 1/4/2020 | 12/7/2019 | SO45095 | 344 |



CONDITIONAL COLUMNS



Conditional Columns allow you to define new fields based on logical rules and conditions (IF/THEN statements)

Here we're creating a conditional column named **Quantity Type**, which is based on **Order Quantity**:

- If Order Quantity =1, Quantity Type = “**Single Item**”
- Else If Order Quantity >1, Quantity Type = “**Multiple Items**”
- Else; Quantity Type = “**Other**”

The dialog box is titled "Add Conditional Column" and contains the following configuration:

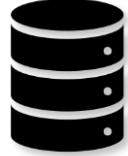
- New column name: QuantityType
- Clause 1: If Order Quantity equals ABC 123, Then Single Item
- Clause 2: Else If Order Quantity is greater than ABC 123, Then Multiple Items
- Clause 3: Else ABC 123, Other

Buttons at the bottom: OK and Cancel.

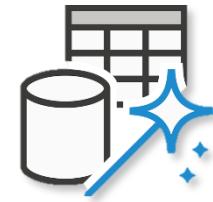


CALCULATED COLUMN BEST PRACTICES

As a best practice, table transformations and column calculations should ideally happen **as close to the original data source as possible**, to optimize performance and speed



Data Source



Power Query



Power BI Front-End



Published Reports

UPSTREAM

DOWNSTREAM

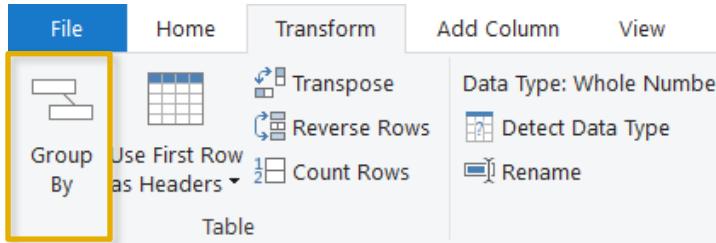


HEY THIS IS IMPORTANT!

This is not a strict rule or requirement but can significantly impact performance for very large or complex data models. Where you define calculations often depends on several factors (*accessibility, complexity, business requirements, etc.*), so we will practice creating columns using both Power Query and the Power BI front-end (DAX) throughout this course

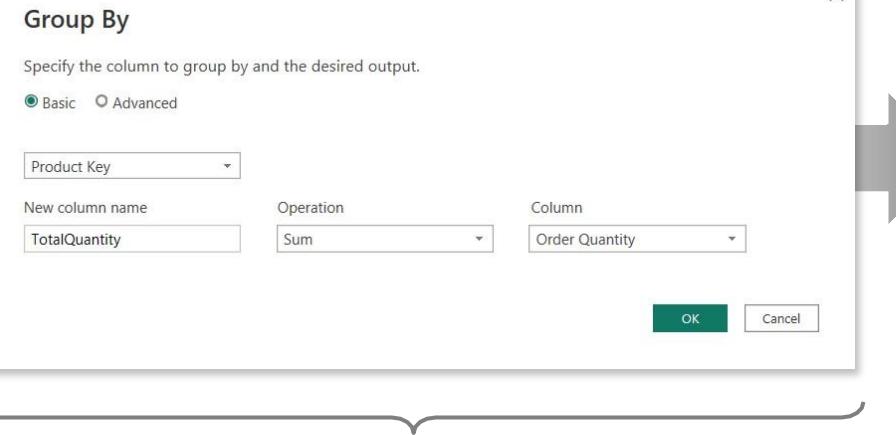


GROUPING & AGGREGATING



Group By allows you to aggregate data at a different level or “grain”
(i.e. group daily records into monthly, aggregate transactions by store, etc.)

| | Order Date | Product Key | Customer Key | Order Quantity |
|----|------------|-------------|--------------|----------------|
| 1 | 6/25/2022 | 214 | 14719 | 1 |
| 2 | 10/8/2021 | 214 | 21990 | 1 |
| 3 | 12/30/2021 | 214 | 22098 | 1 |
| 4 | 6/29/2022 | 214 | 22748 | 1 |
| 5 | 8/16/2021 | 214 | 27821 | 1 |
| 6 | 10/9/2021 | 214 | 15685 | 1 |
| 7 | 8/9/2021 | 214 | 14951 | 1 |
| 8 | 1/19/2022 | 214 | 23101 | 1 |
| 9 | 9/23/2021 | 214 | 17158 | 1 |
| 10 | 1/19/2022 | 214 | 24196 | 1 |
| 11 | 6/29/2022 | 214 | 12963 | 1 |
| 12 | 9/13/2021 | 214 | 12715 | 1 |
| 13 | 10/2/2021 | 214 | 14846 | 1 |
| 14 | 7/31/2021 | 214 | 11290 | 1 |
| 15 | 11/24/2021 | 214 | 22103 | 1 |
| 16 | 8/1/2021 | 214 | 16982 | 1 |
| 17 | 10/12/2021 | 214 | 20410 | 1 |
| 18 | 9/10/2021 | 214 | 14217 | 1 |
| 19 | 10/22/2021 | 214 | 19642 | 1 |
| 20 | 8/11/2021 | 214 | 11666 | 1 |



| | Product Key | TotalQuantity |
|----|-------------|---------------|
| 1 | 214 | 2099 |
| 2 | 215 | 1940 |
| 3 | 220 | 1995 |
| 4 | 223 | 4151 |
| 5 | 226 | 392 |
| 6 | 229 | 408 |
| 7 | 232 | 424 |
| 8 | 235 | 381 |
| 9 | 310 | 169 |
| 10 | 311 | 139 |
| 11 | 312 | 179 |
| 12 | 313 | 168 |
| 13 | 314 | 157 |
| 14 | 320 | 65 |
| 15 | 322 | 39 |
| 16 | 324 | 72 |
| 17 | 326 | 65 |

Here we're transforming a daily, transaction-level table into a summary of **Total Quantity** by **Product Key**

NOTE: Any fields not specified in the Group By settings are lost



GROUPING & AGGREGATING

| | Order Date | Product Key | Customer Key | Order Quantity |
|----|------------|-------------|--------------|----------------|
| 1 | 6/25/2022 | 214 | 14719 | 1 |
| 2 | 10/8/2021 | 214 | 21990 | 1 |
| 3 | 12/30/2021 | 214 | 22098 | 1 |
| 4 | 6/29/2022 | 214 | 22748 | 1 |
| 5 | 8/16/2021 | 214 | 27821 | 1 |
| 6 | 10/9/2021 | 214 | 15685 | 1 |
| 7 | 8/9/2021 | 214 | 14951 | 1 |
| 8 | 1/19/2022 | 214 | 23101 | 1 |
| 9 | 9/23/2021 | 214 | 17158 | 1 |
| 10 | 1/19/2022 | 214 | 24196 | 1 |
| 11 | 6/29/2022 | 214 | 12963 | 1 |
| 12 | 9/13/2021 | 214 | 12715 | 1 |
| 13 | 10/2/2021 | 214 | 14846 | 1 |
| 14 | 7/31/2021 | 214 | 11290 | 1 |
| 15 | 11/24/2021 | 214 | 22103 | 1 |
| 16 | 8/1/2021 | 214 | 16982 | 1 |
| 17 | 10/12/2021 | 214 | 20410 | 1 |
| 18 | 9/10/2021 | 214 | 14217 | 1 |
| 19 | 10/22/2021 | 214 | 19642 | 1 |
| 20 | 8/11/2021 | 214 | 11666 | 1 |

Group By

Specify the columns to group by and one or more outputs.

Basic Advanced

Product Key

Customer Key

Add grouping

New column name

TotalQuantity

Operation

Sum

Column

Order Quantity

Add aggregation

OK

Cancel

| | Product Key | Customer Key | TotalQuantity |
|----|-------------|--------------|---------------|
| 1 | 214 | | 19356 |
| 2 | 214 | | 15101 |
| 3 | 214 | | 12473 |
| 4 | 214 | | 12963 |
| 5 | 214 | | 26986 |
| 6 | 214 | | 13202 |
| 7 | 214 | | 14951 |
| 8 | 214 | | 11201 |
| 9 | 214 | | 19538 |
| 10 | 214 | | 22749 |
| 11 | 214 | | 15815 |
| 12 | 214 | | 19252 |
| 13 | 214 | | 14849 |
| 14 | 214 | | 11290 |
| 15 | 214 | | 27851 |
| 16 | 214 | | 16982 |
| 17 | 214 | | 21863 |
| 18 | 214 | | 19725 |
| 19 | 214 | | 15684 |
| 20 | 214 | | 11666 |
| 21 | 214 | | 26941 |

This time we're transforming the daily, transaction-level table into a summary of **Total Quantity** grouped by both **Product Key** and **Customer Key** (using the "Advanced" option)

NOTE: This is like creating a PivotTable in Excel and pulling in **Sum of Order Quantity** with **Product Key** and **Customer Key** as row labels

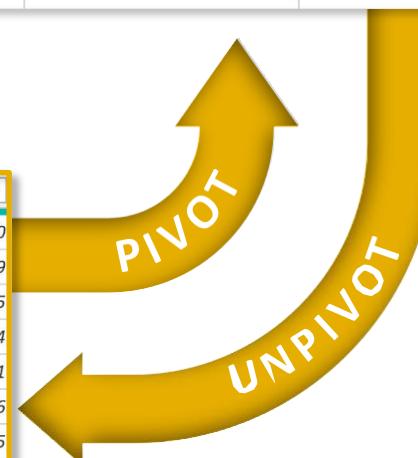


PIVOTING & UNPIVOTING

Pivoting describes the process of turning **distinct row values into columns**, and **unpivoting** describes the process of turning **distinct columns into rows**

| | Date | Product Category | North Region | Central Region | South Region |
|---|----------|------------------|--------------|----------------|--------------|
| 1 | 7/1/2022 | Bikes | 10 | 19 | 25 |
| 2 | 7/1/2022 | Components | 14 | 31 | 16 |
| 3 | 7/1/2022 | Clothing | 35 | 32 | 46 |

| | Date | Product Category | Region | Quantity Sold |
|---|----------|------------------|----------------|---------------|
| 1 | 7/1/2022 | Bikes | North Region | 10 |
| 2 | 7/1/2022 | Bikes | Central Region | 19 |
| 3 | 7/1/2022 | Bikes | South Region | 25 |
| 4 | 7/1/2022 | Components | North Region | 14 |
| 5 | 7/1/2022 | Components | Central Region | 31 |
| 6 | 7/1/2022 | Components | South Region | 16 |
| 7 | 7/1/2022 | Clothing | North Region | 35 |
| 8 | 7/1/2022 | Clothing | Central Region | 32 |
| 9 | 7/1/2022 | Clothing | South Region | 46 |



Imagine the table on a hinge; **pivoting** rotates it from **vertical** to **horizontal**, and **unpivoting** rotates it from **horizontal** to **vertical**

NOTE: **Transpose** works very similarly, but doesn't recognize unique values; instead, the entire table is transformed so that each row becomes a column and vice versa



MERGING QUERIES

The screenshot shows the 'Merge' dialog box in Power BI. On the left, a sidebar menu has 'Merge Queries' selected and highlighted with a yellow box. Below it are 'Append Queries', 'Combine Files', and 'Combine' options. The main area is titled 'Merge' with the sub-instruction 'Select a table and matching columns to create a merged table.' Two tables are listed: 'Sales Data' and 'Product Lookup'. The 'Sales Data' table has columns: Order Date, Product Key, Customer Key, Order Quantity, Index, Stock Date, Order Number, and Territory. The 'Product Lookup' table has columns: Product Key, Product Subcategory Key, Product S K U, Product Name, and Model Name. Both tables show rows for Product Key 214, 215, 218, and 219. A yellow box highlights the 'Product Key' column in both tables. Below the tables, the 'Join Kind' dropdown is set to 'Left Outer (all from first, matching from second)'. There are checkboxes for 'Use fuzzy matching to perform the merge' and 'Fuzzy matching options'. A note at the bottom says 'The selection matches 56046 of 56046 rows from the first table.' At the bottom right are 'OK' and 'Cancel' buttons.

Merging queries allows you to **join tables** based on a common column (like a lookup in Excel)

In this case we're merging the **Sales Data** table with the **Product Lookup** table, which share a common **Product Key** column

NOTE: Merging **adds columns** to an existing table/query

HEY THIS IS IMPORTANT!

Just because you can merge tables, doesn't mean you should!



In many cases, it's better to keep tables separate and define **relationships** between them in the data model (*more on that soon!*)



APPENDING QUERIES

Merge Queries ▾

Append Queries ▾ **Append**

Combine Files

Combine

Concatenate rows from two tables into a single table.

Two tables Three or more tables

First table

AdventureWorks Sales Data 2020

Second table

AdventureWorks Sales Data 2021

Appending queries allows you to **combine** or **stack** tables sharing the exact same column structure and data types

Here we're appending the **AdventureWorks Sales 2020** table to the **AdventureWorks Sales 2021** table, which is valid since they share identical table structures

NOTE: Appending **adds rows** to an existing table/query



PRO TIP: Use the **Folder** option (Get Data > More > Folder) to **append all files within a specified folder** (assuming they share the same structure); as you add new files, simply refresh the query and they will automatically append!



PRO TIP: APPENDING FILES FROM A FOLDER

The screenshot illustrates a workflow for appending files from a folder in Power BI.

Get Data Interface: The main window shows the 'Get Data' interface with various connectors like Excel, Data, and SQL Server. The 'More...' button at the bottom left is highlighted with a yellow box. The 'Folder' connector is selected and highlighted with a yellow box in the list of data sources.

Folder Selection Dialog: A modal dialog titled 'Folder' is open, showing the 'Folder path' as 'C:\Users\Branislav Poljasevic\Documents\3. PowerBI Desktop\Sales'. The 'OK' button is visible.

File Explorer View: A separate window displays the contents of the specified folder: 'C:\Users\Branislav Poljasevic\Documents\3. PowerBI Desktop\Sales'. It lists three CSV files: 'AdventureWorks Sales Data 2020.csv', 'AdventureWorks Sales Data 2021.csv', and 'AdventureWorks Sales Data 2022.csv'. The 'Transform Data' button at the bottom right is highlighted with a yellow box.

| Content | Name | Extension | Date accessed | Date modified | Date created | Attributes |
|---------|------------------------------------|-----------|-----------------------|----------------------|-----------------------|----------------|
| Binary | AdventureWorks Sales Data 2020.csv | .csv | 12/11/2022 4:17:52 PM | 11/3/2022 4:09:09 PM | 12/11/2022 6:17:52 PM | Record C:\User |
| Binary | AdventureWorks Sales Data 2021.csv | .csv | 12/11/2022 6:17:52 PM | 11/3/2022 4:06:28 PM | 12/11/2022 6:17:52 PM | Record C:\User |
| Binary | AdventureWorks Sales Data 2022.csv | .csv | 12/11/2022 6:17:52 PM | 11/3/2022 7:08:24 PM | 12/11/2022 6:17:52 PM | Record C:\User |



DATA SOURCE SETTINGS

Data Source Settings allow you to manage existing data connections, file paths and permissions

The screenshot shows the Power BI desktop ribbon with the 'Data Sources' tab selected. A yellow arrow points from the 'Data Sources' tab to the 'Data source settings' dialog window. Another yellow arrow points from the 'Change Source...' button in the bottom-left of the main window to the 'File path' field in the configuration dialog.

Data source settings

Manage settings for data sources that you have connected to using Power BI Desktop.

Data sources in current file Global permissions

Search data source settings

| |
|--|
| c:\users\branislav poljasevic\...ntureworks calendar lookup.csv |
| c:\users\branislav poljasevic\...ntureworks customer lookup.csv |
| c:\users\branislav poljasevic\... product categories lookup.csv |
| c:\users\branislav poljasevic\...ntureworks product lookup.csv |
| c:\users\branislav poljasevic\...oduct subcategories lookup.csv |
| c:\users\branislav poljasevic\...vtureworks returns data.csv |
| c:\users\branislav poljasevic\...tureworks territory lookup.csv |
| c:\users\branislav poljasevic\...top update\raw data\sales data |

Change Source... Export PBIDS Edit Permissions... Clear Permissions Close

Comma-Separated Values

Basic Advanced

File path: C:\Users\Branislav Poljasevic\Desktop\PBI Desktop Update\Raw Data\Adv...

Open file as: Csv Document

File origin: 1252: Western European (Windows)

Line breaks: Apply all line breaks

Delimiter: Comma

OK Cancel

HEY THIS IS IMPORTANT!

Connections to local files reference the **exact file path**, so if the file name or location changes you will need to update your data source settings



PRO TIP: DATA SOURCE PARAMETERS

Use **parameters** to dynamically manage and update connection paths in the Power Query editor

The diagram illustrates the process of creating and using parameters in the Power Query Editor.

Power Query Editor: Shows the ribbon with "Data source settings" selected. A callout points to the "Manage Parameters" button in the ribbon, which is highlighted with a yellow box.

Manage Parameters Dialog: Shows a list of parameters. A new parameter named "Parameter1" is being created, with its name set to "Database (Fuzzy Factory)". The "Type" is set to "Text". The "Suggested Values" dropdown shows "List of values". The value list contains two items: "1 mavenfuzzyfactory_development" and "2 mavenfuzzyfactory_production". The "Default Value" is set to "mavenfuzzyfactory_development" and the "Current Value" is set to "mavenfuzzyfactory production".

Data Source Settings Dialog: Shows the "Data Sources" tab. A callout points to the "Data source settings" button in the ribbon, which is highlighted with a yellow box. This dialog shows a "MySQL database" configuration with a "Server" dropdown set to "Server (Fuzzy Factory)" and a "Parameter" dropdown set to "Parameter (Fuzzy Factory)". A callout from the "Parameter" dropdown points to the "Parameter1" entry in the "Data Sources" list.

Annotations:

- Parameter name** (Name of the query/table)
- Parameter type** (Any value, text, date, etc.)
- Parameter value** (Any value, list, query)
- Parameter type** (Default & current)

Update Server & Database connection text values with parameters



REFRESHING QUERIES

The screenshot shows the Power BI desktop ribbon with the Home tab selected. In the External Tools section, the 'Refresh' button is highlighted with a yellow box and a yellow arrow pointing towards the text below.

By default, **all queries** will refresh when you use the **Refresh** command from the **Home** tab

The screenshot shows the Power BI desktop ribbon with the Transform tab selected. In the Tools section, the 'Refresh' button is highlighted with a yellow box and a yellow arrow pointing towards the text below.

Queries [13]

Transform File from Sales Data [2]

Other Queries [9]

Calendar Lookup

Customer Lookup

Product Categories

Product Lookup

Product Subcategory

Returns Data

Sales Data

Territory Lookup

Rolling Calendar

- Copy
- Paste
- Delete
- Rename
- Enable load
- Include in report refresh**
- Duplicate
- Reference
- Move To Group
- Move Up
- Move Down
- Create Function...
- Convert To Parameter
- Advanced Editor
- Properties...

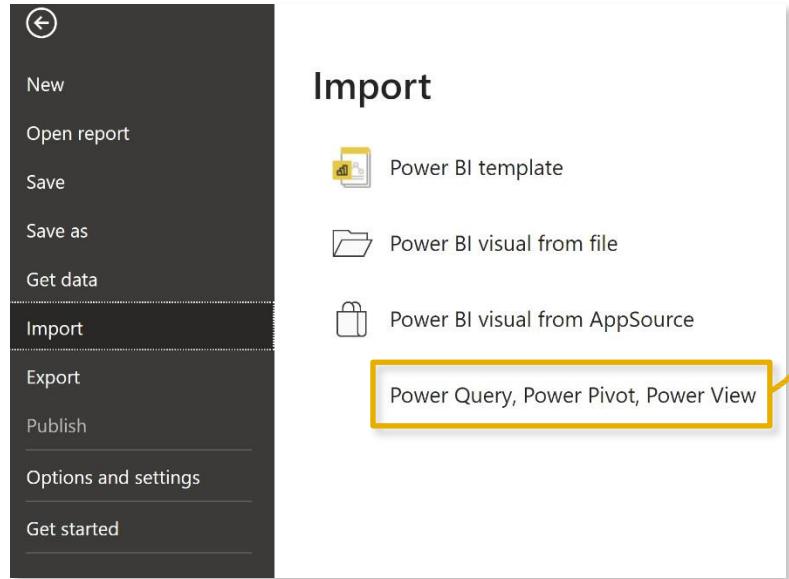
From the Query Editor, uncheck **Include in report refresh** to exclude individual queries from the refresh

PRO TIP: Exclude queries from refresh that don't change often (like lookups or static data tables)





PRO TIP: IMPORTING EXCEL MODELS



Already have a fully-built model in Excel?

You can import models built in Excel directly into Power BI Desktop using: ***Import > Power Query, Power Pivot, Power View***

Imported models retain the following:

- Data source **connections** and **queries**
- Query editing procedures and **applied steps**
- Table **relationships**, **hierarchies**, **field settings**, etc.
- All **calculated columns** and **DAX measures**



PRO TIP: If you are more comfortable working in Excel, build your models there first then import to Power BI!

POWER QUERY BEST PRACTICES



★ Get organized before connecting and loading data

- *Define clear and intuitive table/query names from the start, and establish an organized file/folder structure if you are working with local flat files to avoid changes to file names or paths*

★ Disable report refresh for any static data sources

- *There's no need to constantly refresh data sources that don't change, like lookups or static data tables*

★ When working with large tables, only load the data you need

- *Don't include hourly data when you only need daily, or transaction-level data when only need a product-level summary (extra data will only slow your report down!)*

CREATING A DATA MODEL



CREATING A DATA MODEL



In this section we'll cover **foundational data modeling topics** like normalization, fact and dimension tables, primary and foreign keys, relationship cardinality and filter flow

TOPICS WE'LL COVER:

Data Modeling 101

Normalization

Facts & Dimensions

Primary & Foreign Keys

Cardinality

Filter Flow

Common Schemas

Hierarchies

GOALS FOR THIS SECTION:

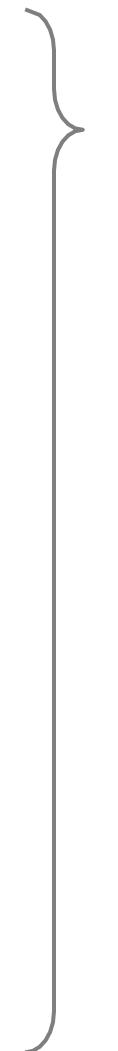
- Understand the basic principles of data modeling, including normalization, fact & dimension tables and common schemas
- Create table relationships using primary and foreign keys, and discuss different types of relationship cardinality
- Configure report filters and trace filter context as it flows between related tables in the model
- Explore data modeling options like hierarchies, data categories and hidden fields



WHAT IS A DATA MODEL?

The screenshot shows a data modeling interface with three tables:

- Product Lookup**: Contains columns: Model Name, Product Color, Product Cost, Product Description, Product Key, and Product Name. A "Collapse ^" button is at the bottom.
- Sales Data**: Contains columns: Customer Key, Index, Order Date, Order Line Item, Order Number, Order Quantity, Product Key, Stock Date, Territory Key. A "Collapse ^" button is at the bottom.
- Returns Data**: Contains columns: Product Key, Return Date, Return Quantity, Territory Key. A "Collapse ^" button is at the bottom.



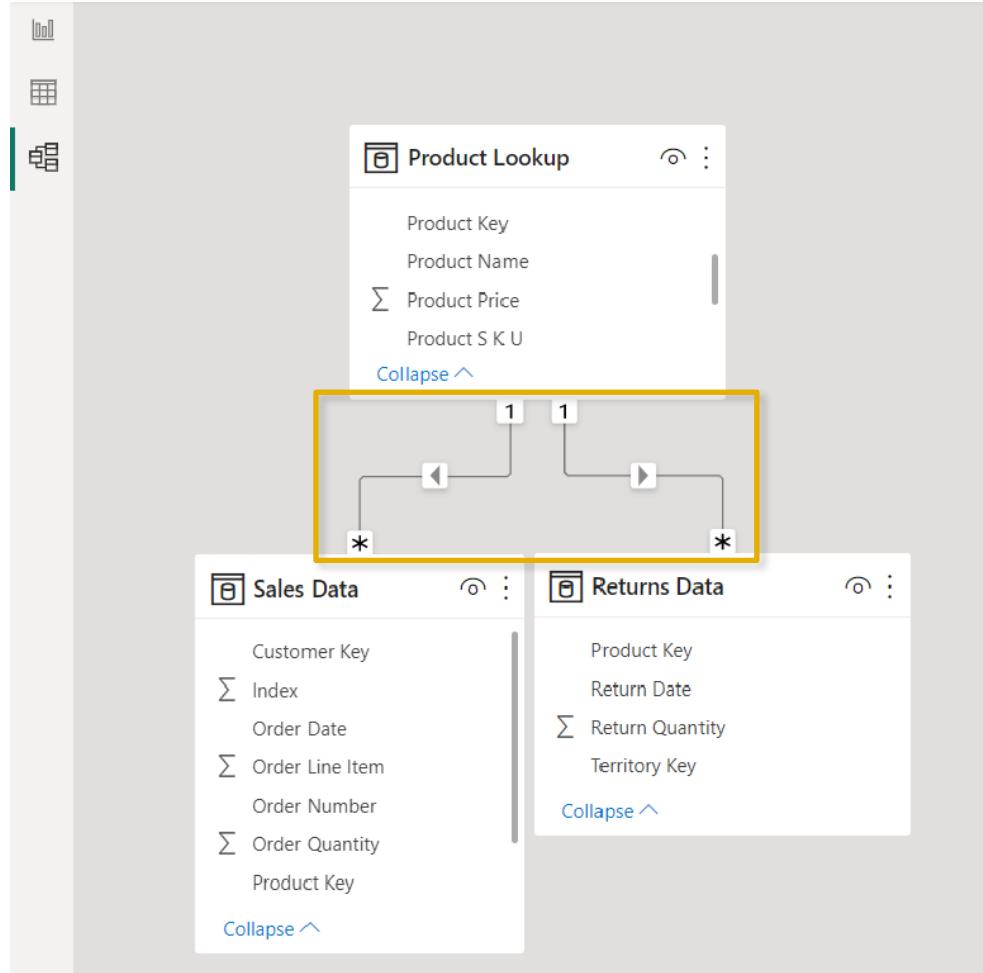
This **IS NOT** a data model 😞

- This is a collection of independent tables, which share no connections or relationships
- If you tried to visualize **Orders** and **Returns** by **Product**, this is what you'd get

| ProductName | OrderQuantity | ReturnQuantity |
|------------------------|---------------|----------------|
| All-Purpose Bike Stand | 84,174 | 1,828 |
| AWC Logo Cap | 84,174 | 1,828 |
| Bike Wash - Dissolver | 84,174 | 1,828 |
| Cable Lock | 84,174 | 1,828 |
| Chain | 84,174 | 1,828 |
| Classic Vest, L | 84,174 | 1,828 |
| Classic Vest, M | 84,174 | 1,828 |
| Classic Vest, S | 84,174 | 1,828 |
| Fender Set - Mountain | 84,174 | 1,828 |
| Total | 84,174 | 1,828 |



WHAT IS A DATA MODEL?



This **IS** a data model! 😊

- The tables are connected via relationships, based on a common field (Product Key)
- Now **Sales** and **Returns** data can be filtered using fields from the **Product Lookup** table!

| ProductName | OrderQuantity | ReturnQuantity |
|------------------------|---------------|----------------|
| All-Purpose Bike Stand | 234 | 8 |
| AWC Logo Cap | 4,151 | 46 |
| Bike Wash - Dissolver | 1,706 | 25 |
| Classic Vest, L | 182 | 4 |
| Classic Vest, M | 182 | 7 |
| Classic Vest, S | 157 | 8 |
| Fender Set - Mountain | 3,960 | 54 |
| Half-Finger Gloves, L | 840 | 18 |
| Half-Finger Gloves, M | 918 | 16 |
| Total | 84,174 | 1,828 |



DATABASE NORMALIZATION

Normalization is the process of organizing the tables and columns in a relational database to reduce redundancy and preserve data integrity. It's commonly used to:

- **Eliminate redundant data** to decrease table sizes and improve processing speed & efficiency
- **Minimize errors and anomalies** from data modifications (inserting, updating or deleting records)
- **Simplify queries** and structure the database for meaningful analysis

 In a normalized database, each table should serve a **distinct** and **specific** purpose
(i.e. product information, transaction records, customer attributes, store details, etc.)

| date | product_id | quantity | product_brand | product_name | product_sku | product_weight |
|----------|------------|----------|---------------|-----------------------------|-------------|----------------|
| 1/1/1997 | 869 | 5 | Nationeel | Nationeel Grape Fruit Roll | 52382137179 | 17 |
| 1/7/1997 | 869 | 2 | Nationeel | Nationeel Grape Fruit Roll | 52382137179 | 17 |
| 1/3/1997 | 1 | 4 | Washington | Washington Berry Juice | 90748583674 | 8.39 |
| 1/1/1997 | 1472 | 3 | Fort West | Fort West Fudge Cookies | 37276054024 | 8.28 |
| 1/6/1997 | 1472 | 2 | Fort West | Fort West Fudge Cookies | 37276054024 | 8.28 |
| 1/5/1997 | 2 | 4 | Washington | Washington Mango Drink | 96516502499 | 7.42 |
| 1/1/1997 | 76 | 4 | Red Spade | Red Spade Sliced Chicken | 62054644227 | 18.1 |
| 1/1/1997 | 76 | 2 | Red Spade | Red Spade Sliced Chicken | 62054644227 | 18.1 |
| 1/5/1997 | 3 | 2 | Washington | Washington Strawberry Drink | 58427771925 | 13.1 |
| 1/7/1997 | 3 | 2 | Washington | Washington Strawberry Drink | 58427771925 | 13.1 |
| 1/1/1997 | 320 | 3 | Excellent | Excellent Cranberry Juice | 36570182442 | 16.4 |

Models that aren't normalized contain **redundant**, **duplicate data**. In this case, all of the product-specific fields could be stored in a separate table containing a unique record for each **product id**

This may not seem critical now, but minor inefficiencies can become major problems at scale!



FACT & DIMENSION TABLES

Data models generally contain two types of tables: **fact** (“data”) tables, and **dimension** (“lookup”) tables:

- **Fact tables** contain **numerical values** or metrics used for summarization (*sales, orders, transactions, pageviews, etc.*)
- **Dimension tables** contain **descriptive attributes** used for filtering or grouping (*products, customers, dates, stores, etc.*)

| date | product_id | quantity |
|----------|------------|----------|
| 1/1/1997 | 869 | 5 |
| 1/1/1997 | 1472 | 3 |
| 1/1/1997 | 76 | 4 |
| 1/1/1997 | 320 | 3 |
| 1/1/1997 | 4 | 4 |
| 1/1/1997 | 952 | 4 |
| 1/1/1997 | 1222 | 4 |
| 1/1/1997 | 517 | 4 |
| 1/1/1997 | 1359 | 4 |
| 1/1/1997 | 357 | 4 |
| 1/1/1997 | 1426 | 5 |
| 1/1/1997 | 190 | 4 |
| 1/1/1997 | 367 | 4 |
| 1/1/1997 | 250 | 5 |
| 1/1/1997 | 600 | 4 |
| 1/1/1997 | 702 | 5 |

This **Fact** table contains **quantity** values, along with **date** and **product_id** fields

| date | day_of_month | month | year | weekday | week_of_year | week_ending | month_name | quarter |
|----------|--------------|-------|------|-----------|--------------|-------------|------------|---------|
| 1/1/1997 | 1 | 1 | 1997 | Wednesday | 1 | 1/5/1997 | January | Q1 |
| 1/2/1997 | 2 | 1 | 1997 | Thursday | 1 | 1/5/1997 | January | Q1 |
| 1/3/1997 | 3 | 1 | 1997 | Friday | 1 | 1/5/1997 | January | Q1 |
| 1/4/1997 | 4 | 1 | 1997 | Saturday | 1 | 1/5/1997 | January | Q1 |
| 1/5/1997 | 5 | 1 | 1997 | Sunday | 2 | 1/5/1997 | January | Q1 |
| 1/6/1997 | 6 | 1 | 1997 | Monday | 2 | 1/12/1997 | January | Q1 |

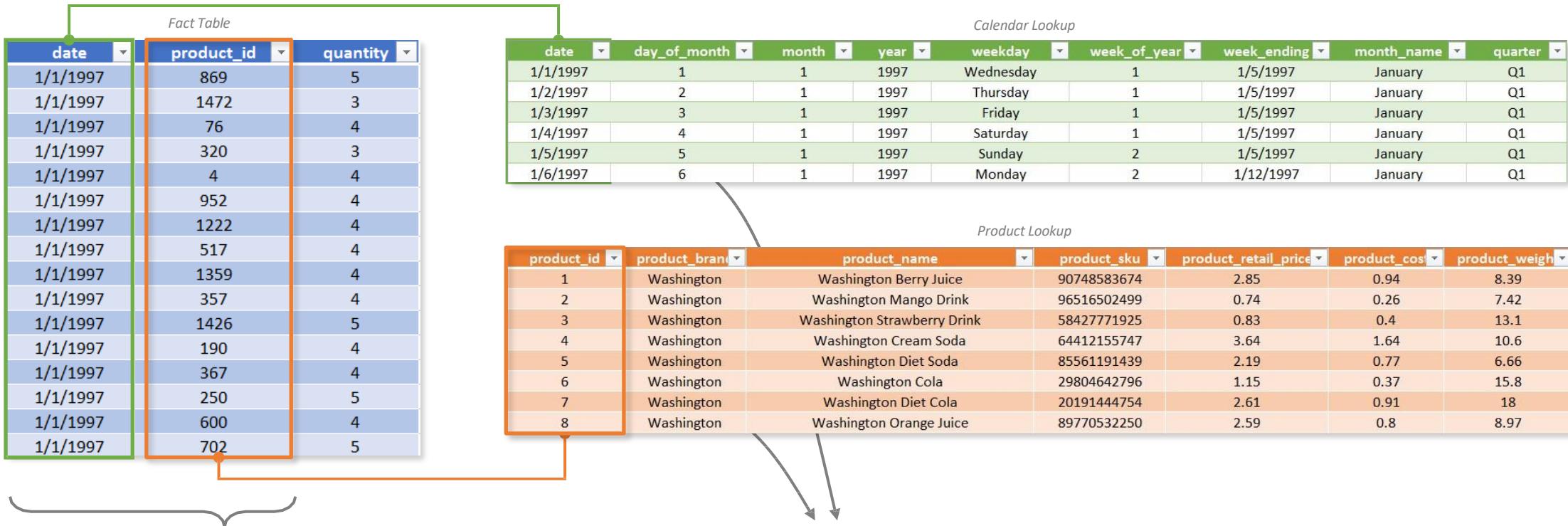
This **Calendar Lookup** table contains attributes about each **date** (month, year, quarter, etc.)

| product_id | product_brand | product_name | product_sku | product_retail_price | product_cost | product_weight |
|------------|---------------|-----------------------------|-------------|----------------------|--------------|----------------|
| 1 | Washington | Washington Berry Juice | 90748583674 | 2.85 | 0.94 | 8.39 |
| 2 | Washington | Washington Mango Drink | 96516502499 | 0.74 | 0.26 | 7.42 |
| 3 | Washington | Washington Strawberry Drink | 58427771925 | 0.83 | 0.4 | 13.1 |
| 4 | Washington | Washington Cream Soda | 64412155747 | 3.64 | 1.64 | 10.6 |
| 5 | Washington | Washington Diet Soda | 85561191439 | 2.19 | 0.77 | 6.66 |
| 6 | Washington | Washington Cola | 29804642796 | 1.15 | 0.37 | 15.8 |
| 7 | Washington | Washington Diet Cola | 20191444754 | 2.61 | 0.91 | 18 |
| 8 | Washington | Washington Orange Juice | 89770532250 | 2.59 | 0.8 | 8.97 |

This **Product Lookup** table contains attributes about each **product_id** (brand, SKU, price, etc.)



PRIMARY & FOREIGN KEYS



These are **foreign keys (FK)**

*They contain multiple instances of each value, and relate to **primary keys** in dimension tables*

These are **primary keys (PK)**

*They uniquely identify each row of the table, and relate to **foreign keys** in fact tables*



RELATIONSHIPS VS. MERGED TABLES



*Can't I just merge queries or use lookup functions to **pull everything into one single table**?*

- Anonymous confused man

| Original Fact Table fields | | | | | | | | | | Attributes from Calendar Lookup table | | | Attributes from Product Lookup table | | |
|-----------------------------------|------------|----------|--------------|-------|------|-----------|------------|---------|---------------|--|-------------|----------------|---|--|--|
| date | product_id | quantity | day_of_month | month | year | weekday | month_name | quarter | product_brand | product_name | product_sku | product_weight | | | |
| 1/1/1997 | 869 | 5 | 1 | 1 | 1997 | Wednesday | January | Q1 | Nationaleel | Nationaleel Grape Fruit Roll | 52382137179 | 17 | | | |
| 1/7/1997 | 869 | 2 | 7 | 1 | 1997 | Tuesday | January | Q1 | Nationaleel | Nationaleel Grape Fruit Roll | 52382137179 | 17 | | | |
| 1/3/1997 | 1 | 4 | 3 | 1 | 1997 | Friday | January | Q1 | Washington | Washington Berry Juice | 90748583674 | 8.39 | | | |
| 1/1/1997 | 1472 | 3 | 1 | 1 | 1997 | Wednesday | January | Q1 | Fort West | Fort West Fudge Cookies | 37276054024 | 8.28 | | | |
| 1/6/1997 | 1472 | 2 | 6 | 1 | 1997 | Monday | January | Q1 | Fort West | Fort West Fudge Cookies | 37276054024 | 8.28 | | | |
| 1/5/1997 | 2 | 4 | 5 | 1 | 1997 | Sunday | January | Q1 | Washington | Washington Mango Drink | 96516502499 | 7.42 | | | |
| 1/1/1997 | 76 | 4 | 1 | 1 | 1997 | Wednesday | January | Q1 | Red Spade | Red Spade Sliced Chicken | 62054644227 | 18.1 | | | |
| 1/1/1997 | 76 | 2 | 1 | 1 | 1997 | Wednesday | January | Q1 | Red Spade | Red Spade Sliced Chicken | 62054644227 | 18.1 | | | |
| 1/5/1997 | 3 | 2 | 5 | 1 | 1997 | Sunday | January | Q1 | Washington | Washington Strawberry Drink | 58427771925 | 13.1 | | | |
| 1/7/1997 | 3 | 2 | 7 | 1 | 1997 | Tuesday | January | Q1 | Washington | Washington Strawberry Drink | 58427771925 | 13.1 | | | |
| 1/1/1997 | 320 | 3 | 1 | 1 | 1997 | Wednesday | January | Q1 | Excellent | Excellent Cranberry Juice | 36570182442 | 16.4 | | | |

You can, **but it's extremely inefficient!**

- Merging tables creates **redundancy** and often requires **significantly more memory and processing power** to analyze compared to a relational model with multiple small tables



THE MODEL VIEW

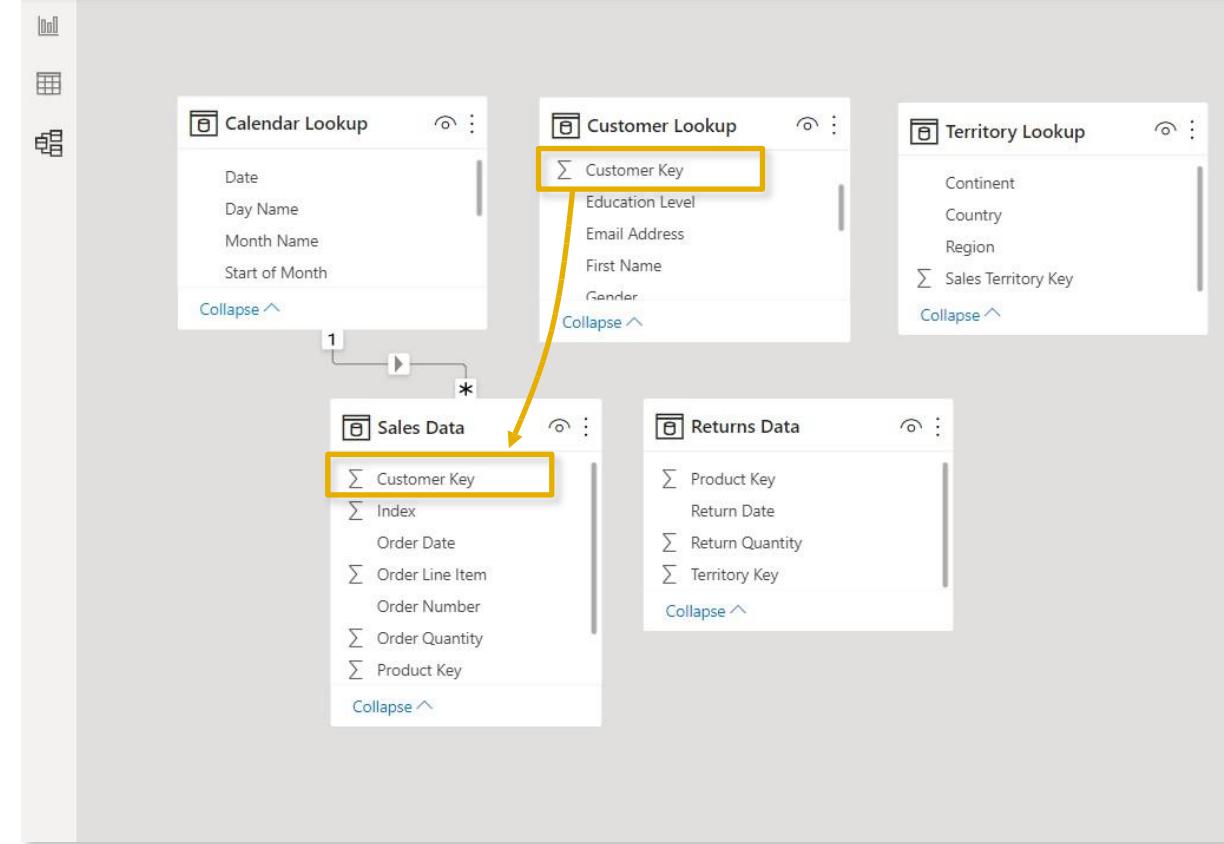
The screenshot illustrates the Microsoft Power BI Model View interface, which is used for managing data relationships and metadata. The interface includes the following components:

- Menu Ribbon (Home, Help):** Located at the top left, it contains standard file operations like Paste, Cut, Copy, and various data import options.
- Model canvas:** The central workspace where data tables and their relationships are visualized. It shows entities like Customer Lookup, Sales Data, Returns Data, and Product Lookup, connected by lines indicating their relationships.
- Properties pane:** A panel on the right side for managing table properties. It includes fields for Name, Description, Synonyms, Row label, Key column, and Is hidden.
- Data / Field List:** Another panel on the right side listing all data fields and columns available in the model.
- Model layout tabs:** Located at the bottom left of the canvas area, these tabs include "All tables" and a "+" button for adding new tables.
- View Options:** Located at the bottom right, these options allow for zooming and adjusting the layout of the model view.

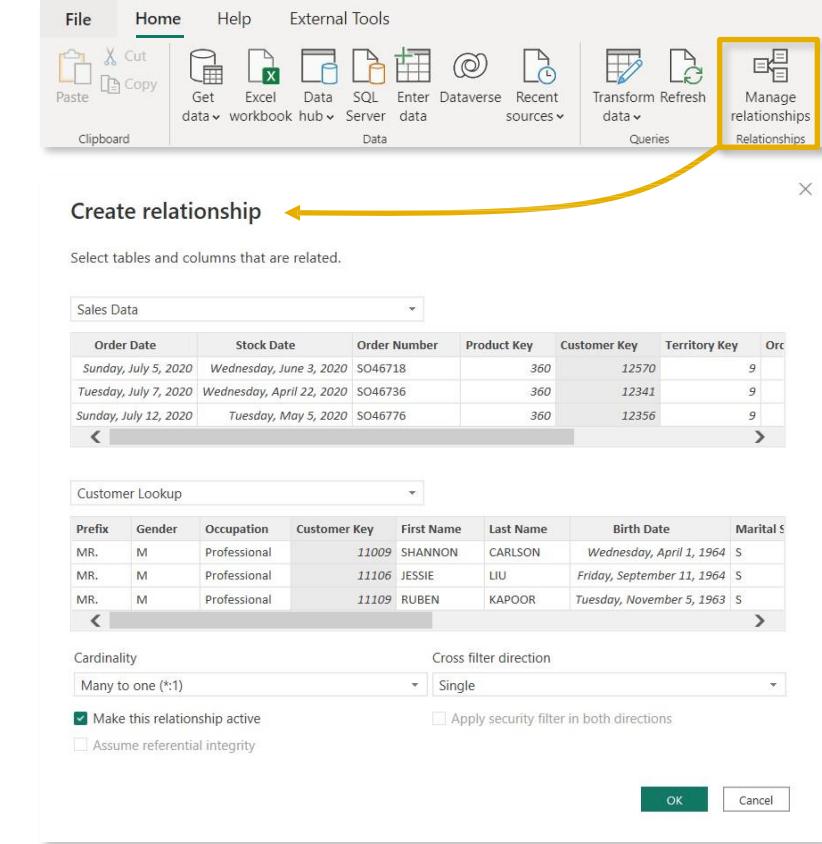


CREATING TABLE RELATIONSHIPS

OPTION 1: Click and drag to connect primary and foreign keys within the **Model** view



OPTION 2: Add or detect relationships using the **Manage Relationships** dialog box





MANAGING & EDITING RELATIONSHIPS

The screenshot shows the Power BI ribbon with the 'Home' tab selected. In the 'Column tools' section, the 'Relationships' icon is highlighted with a yellow box. A large yellow arrow points from this icon down to the 'Manage relationships' dialog box.

Manage relationships

| Active | From: Table (Column) | To: Table (Column) |
|-------------------------------------|---|--|
| <input checked="" type="checkbox"/> | Product Lookup (Product Subcategory Key) | Product Subcategories Lookup (Product Subcategory Key) |
| <input checked="" type="checkbox"/> | Product Subcategories Lookup (Product Category Key) | Product Categories Lookup (Product Category Key) |
| <input checked="" type="checkbox"/> | Sales Data (Customer Key) | Customer Lookup (Customer Key) |
| <input checked="" type="checkbox"/> | Sales Data (Order Date) | Calendar Lookup (Date) |
| <input checked="" type="checkbox"/> | Sales Data (Product Key) | Product Lookup (Product Key) |

New... Autodetect... Edit... Delete Close

Launch the **Manage Relationships** dialog box or double-click a relationship to modify it

Edit relationship

Select tables and columns that are related.

Sales Data

| Order Date | Stock Date | Order Number | Product Key | Customer Key | Territory Key | Or |
|-----------------------|---------------------------|--------------|-------------|--------------|---------------|----|
| Sunday, July 5, 2020 | Wednesday, June 3, 2020 | SO46718 | 360 | 12570 | 9 | |
| Tuesday, July 7, 2020 | Wednesday, April 22, 2020 | SO46736 | 360 | 12341 | 9 | |
| Sunday, July 12, 2020 | Tuesday, May 5, 2020 | SO46776 | 360 | 12356 | 9 | |

Customer Lookup

| Prefix | Gender | Occupation | Customer Key | First Name | Last Name | Birth Date | Marital S |
|--------|--------|--------------|--------------|------------|-----------|----------------------------|-----------|
| MR. | M | Professional | 11009 | SHANNON | CARLSON | Wednesday, April 1, 1964 | S |
| MR. | M | Professional | 11106 | JESSIE | LIU | Friday, September 11, 1964 | S |
| MR. | M | Professional | 11109 | RUBEN | KAPOOR | Tuesday, November 5, 1963 | S |

Cardinality Cross filter direction

Many to one (*:1) Single

Make this relationship active Apply security filter in both directions

Assume referential integrity

OK Cancel

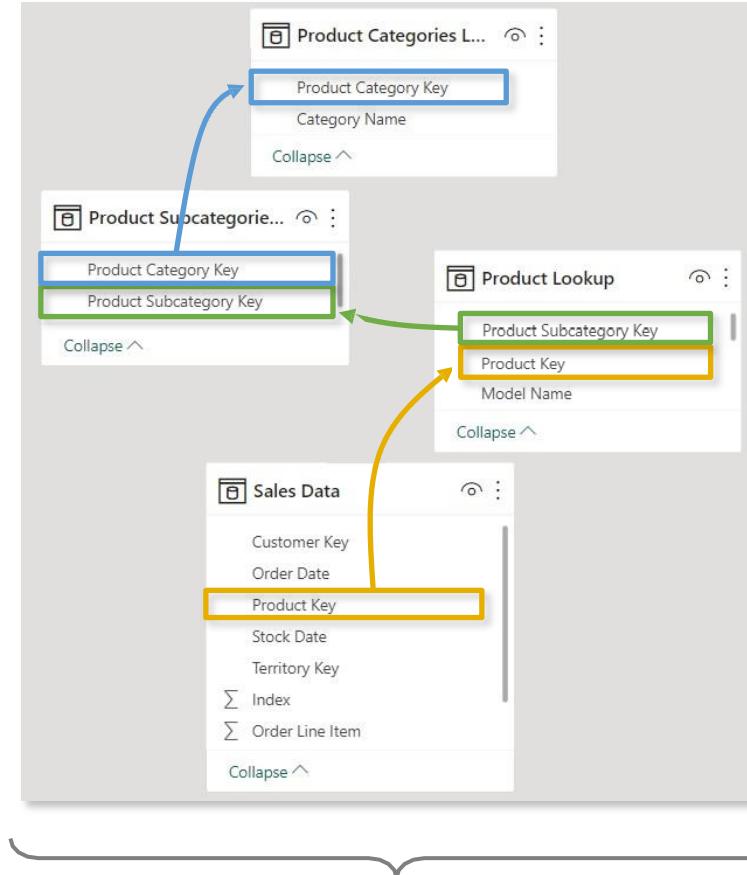
Editing tools allow you to **activate or deactivate** relationships and manage **cardinality** and **filter direction** – more on that soon!



STAR & SNOWFLAKE SCHEMAS



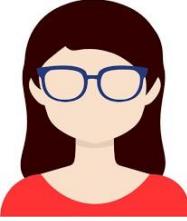
A **star schema** is the simplest and most common type of data model, characterized by a single fact table surrounded by related dimension tables



A **snowflake schema** is an extension of a star, and includes relationships between dimension tables and related sub-dimension tables



ASSIGNMENT: TABLE RELATIONSHIPS



NEW MESSAGE

From: **Dana Modelle (Analyst)**

Subject: **Need a favor...**

Hey there,
Ethan shared the data model you've been working on, and we
might have an issue...

Last night I left my laptop open, and my cat Dennis somehow got
his paws on our model. Now all the relationships are gone!

Could you please rebuild the model, including all three product
tables? I owe you one!

-Dana

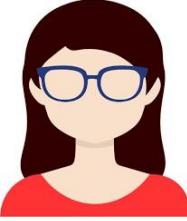
Reply **Forward**

Key Objectives

1. Delete all existing table relationships
2. Create a star schema by creating relationships between the Sales, Calendar, Customer, Product and Territories tables
3. Connect all three product tables (Product, Subcategory, Category) in a snowflake schema
4. Use the matrix visual to confirm that you can filter Order Quantity values using fields from each dimension table



SOLUTION: TABLE RELATIONSHIPS

  NEW MESSAGE

From: **Dana Modelle (Analyst)**

Subject: **Need a favor...**

Hey there,

Ethan shared the data model you've been working on, and we might have an issue...

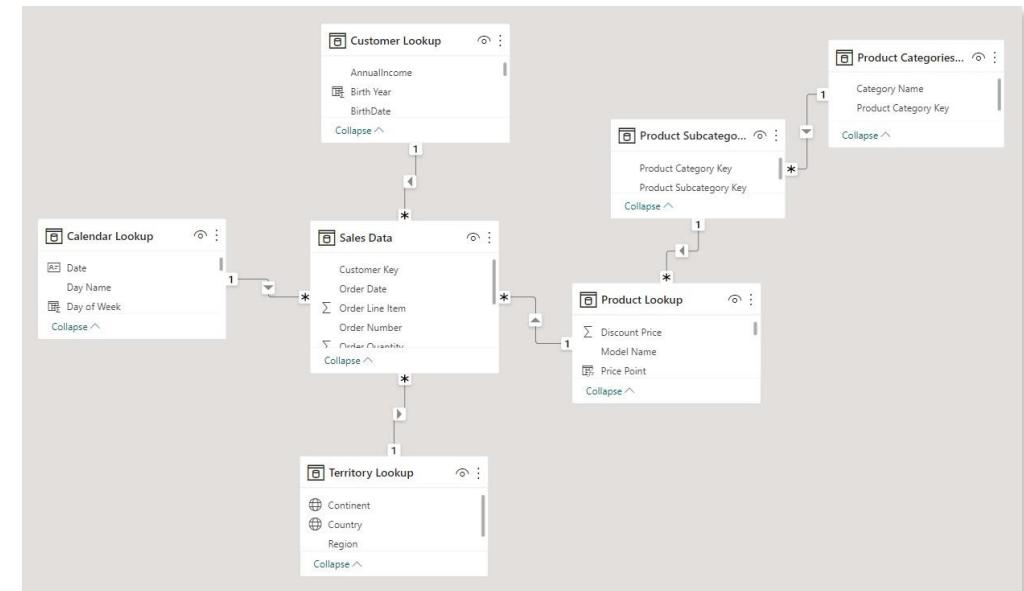
Last night I left my laptop open, and my cat Dennis somehow got his paws on our model. Now all the relationships are gone!

Could you please rebuild the model, including all three product tables? I owe you one!

-Dana

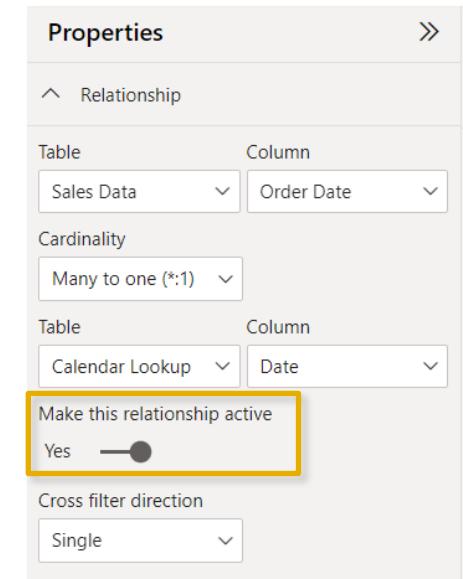
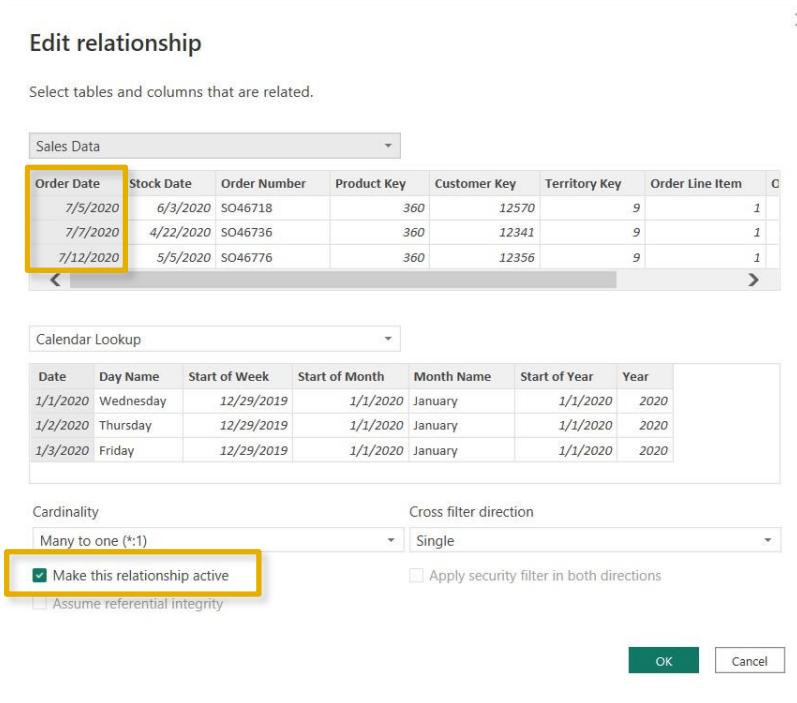
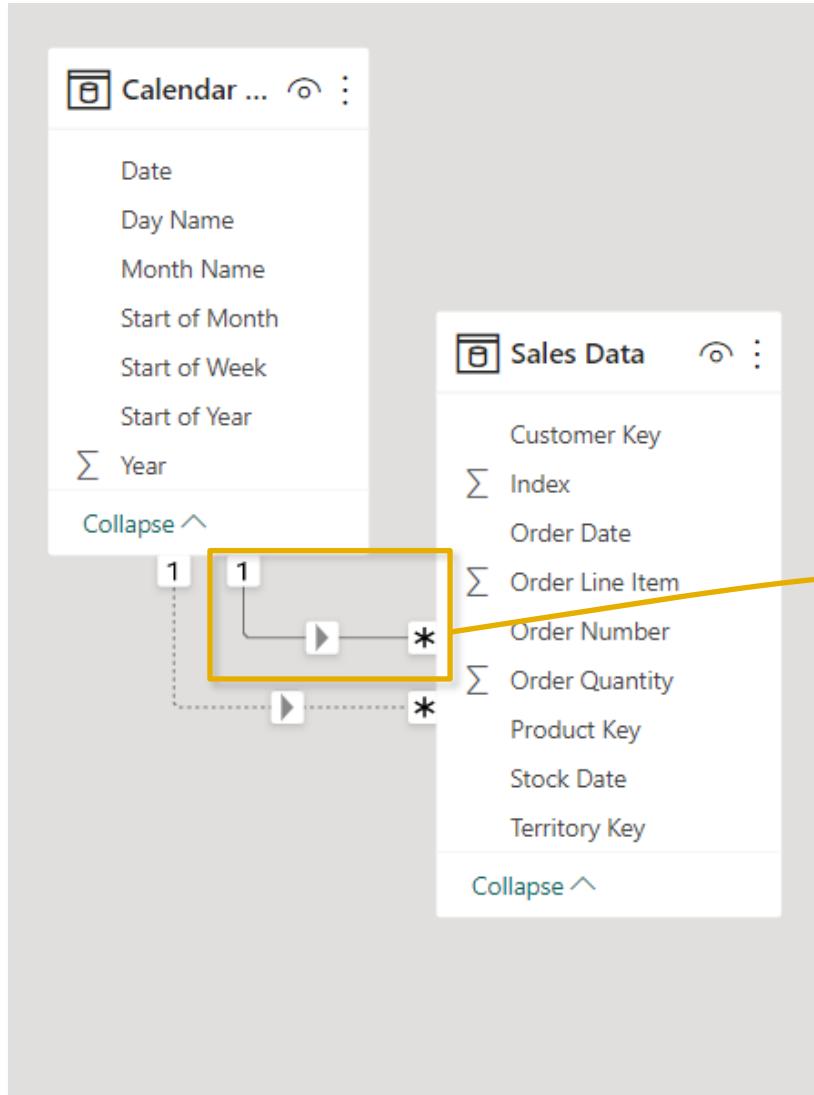
[Reply](#) [Forward](#)

Solution Preview





PRO TIP: ACTIVE & INACTIVE RELATIONSHIPS

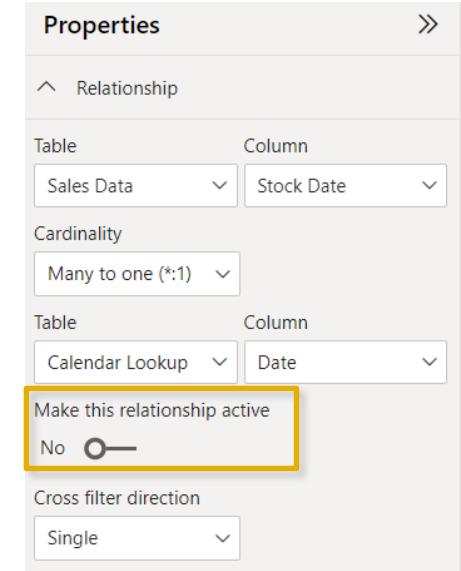
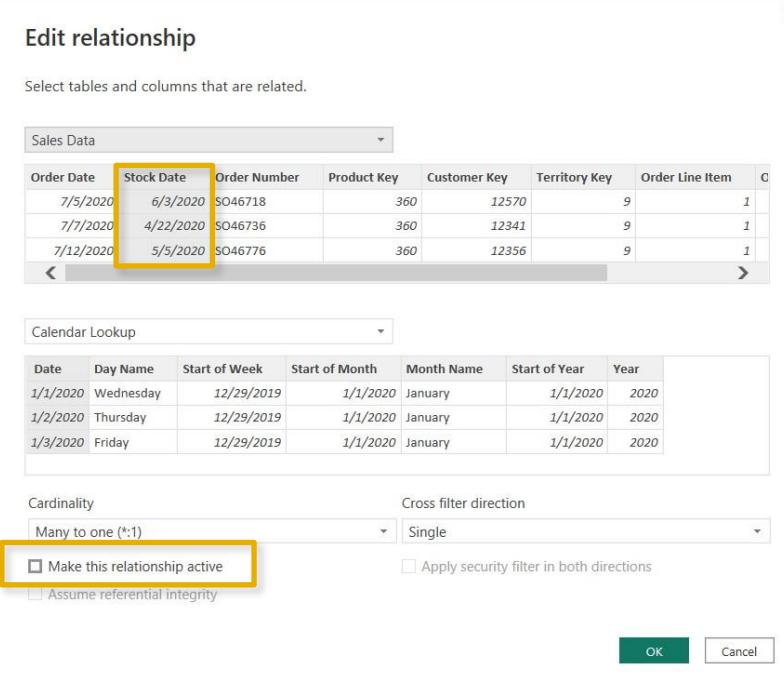
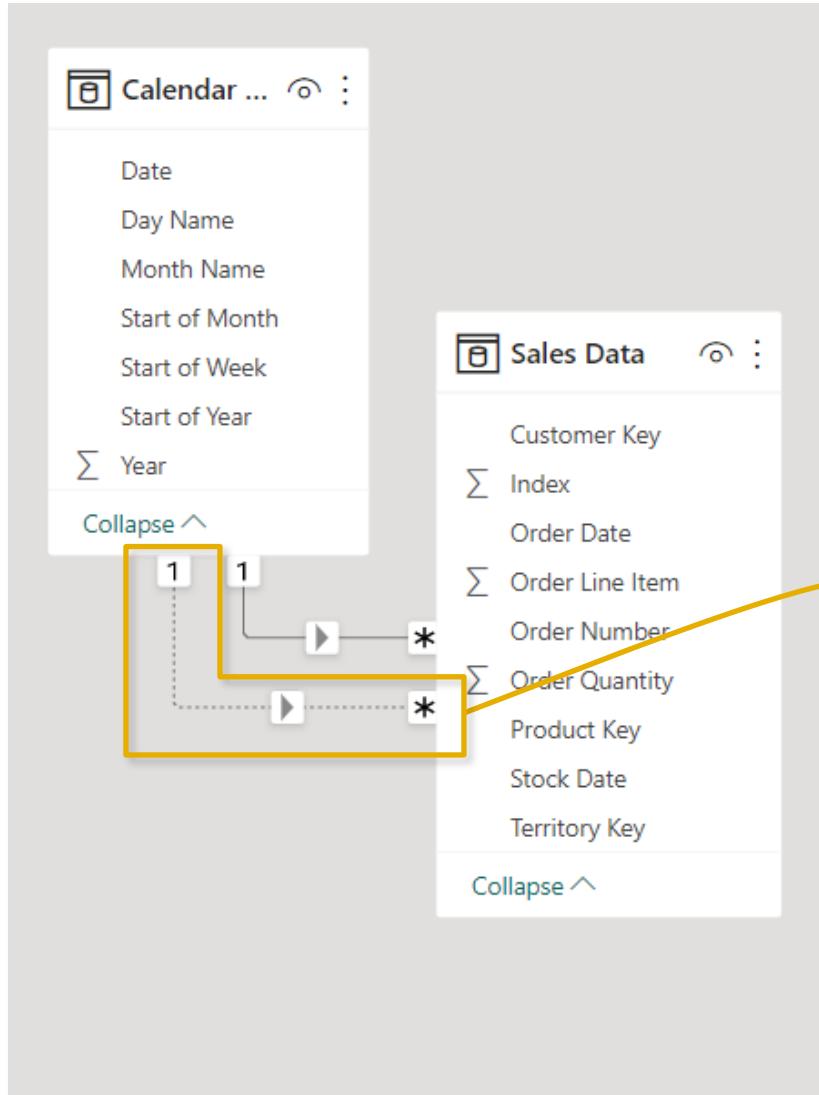


The **Sales Data** table contains two date fields (**Order Date** & **Stock Date**), but there can only be **one active relationship** to the Date key in the Calendar table

You can set relationships to active or inactive from either the **Edit Relationships** dialog box or the **Properties** (you must deactivate one before activating another)



PRO TIP: ACTIVE & INACTIVE RELATIONSHIPS

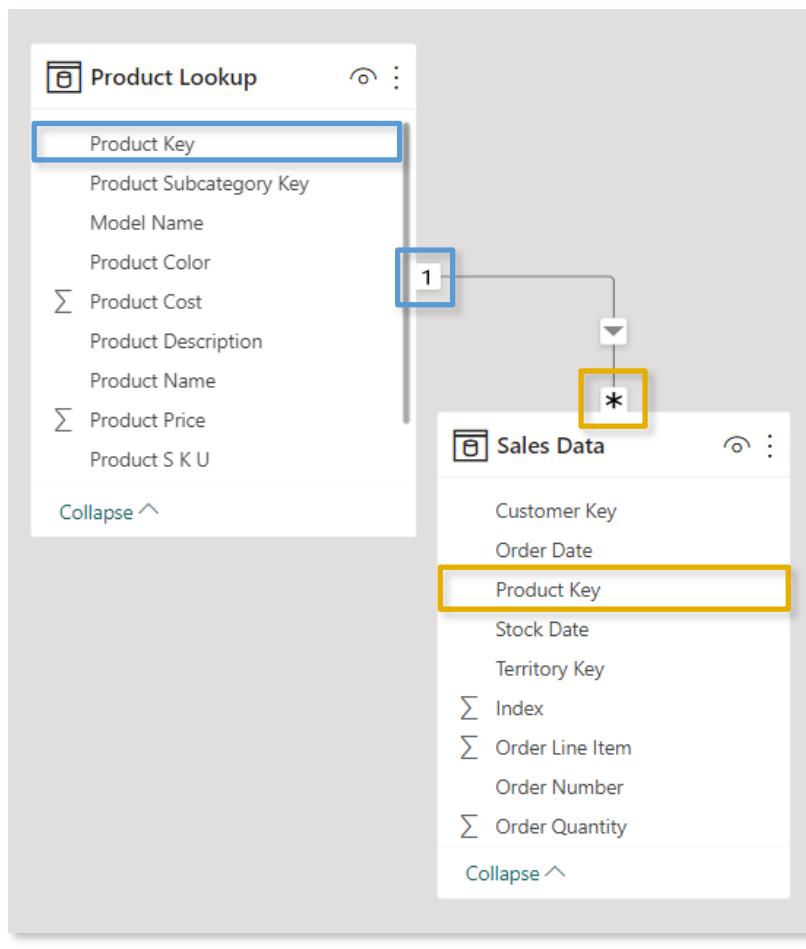


The **Sales Data** table contains two date fields (**Order Date** & **Stock Date**), but there can only be **one active relationship** to the Date key in the Calendar table

You can set relationships to active or inactive from either the **Edit Relationships** dialog box or the **Properties** (you must deactivate one before activating another)



RELATIONSHIP CARDINALITY



Cardinality refers to the uniqueness of values in a column

- Ideally, all relationships in the data model should follow a **one-to-many** cardinality: **one** instance of each primary key, and **many** instances of each foreign key

*In this example there is only **ONE instance of each Product Key** in the Product table (noted by a “1”), since each row contains **attributes of a single product** (name, SKU, description, price, etc.)*

*There are **MANY instances of each Product Key** in the Sales table (noted by an asterisk *), since there are **multiple sales for each product***



EXAMPLE: ONE-TO-ONE CARDINALITY

Product Lookup

| product_id | product_name | product_sku |
|------------|-------------------------|-------------|
| 4 | Washington Cream Soda | 64412155747 |
| 5 | Washington Diet Soda | 85561191439 |
| 7 | Washington Diet Cola | 20191444754 |
| 8 | Washington Orange Juice | 89770532250 |

Price Lookup

| product_id | product_price |
|------------|---------------|
| 4 | \$3.64 |
| 5 | \$2.19 |
| 7 | \$2.61 |
| 8 | \$2.59 |

- Connecting the two tables above using **product_id** creates a **one-to-one relationship**, since each product ID only appears once in each table
- This isn't necessarily a "bad" relationship, but you can simplify the model by merging the tables into a single, valid dimension table

| product_id | product_name | product_sku | product_price |
|------------|-------------------------|-------------|---------------|
| 4 | Washington Cream Soda | 64412155747 | \$3.64 |
| 5 | Washington Diet Soda | 85561191439 | \$2.19 |
| 7 | Washington Diet Cola | 20191444754 | \$2.61 |
| 8 | Washington Orange Juice | 89770532250 | \$2.59 |

NOTE: this still respects the rules of normalization, since all rows are unique and capture product-specific attributes



EXAMPLE: MANY-TO-MANY CARDINALITY

Product Lookup

| product_id | product_name | product_sku |
|------------|----------------------------|-------------|
| 4 | Washington Cream Soda | 64412155747 |
| 4 | Washington Diet Cream Soda | 81727382373 |
| 5 | Washington Diet Soda | 85561191439 |
| 7 | Washington Diet Cola | 20191444754 |
| 8 | Washington Orange Juice | 89770532250 |

Sales

| date | product_id | transactions |
|----------|------------|--------------|
| 1/1/2017 | 4 | 12 |
| 1/2/2017 | 4 | 9 |
| 1/3/2017 | 4 | 11 |
| 1/1/2017 | 5 | 16 |
| 1/2/2017 | 5 | 19 |
| 1/1/2017 | 7 | 11 |

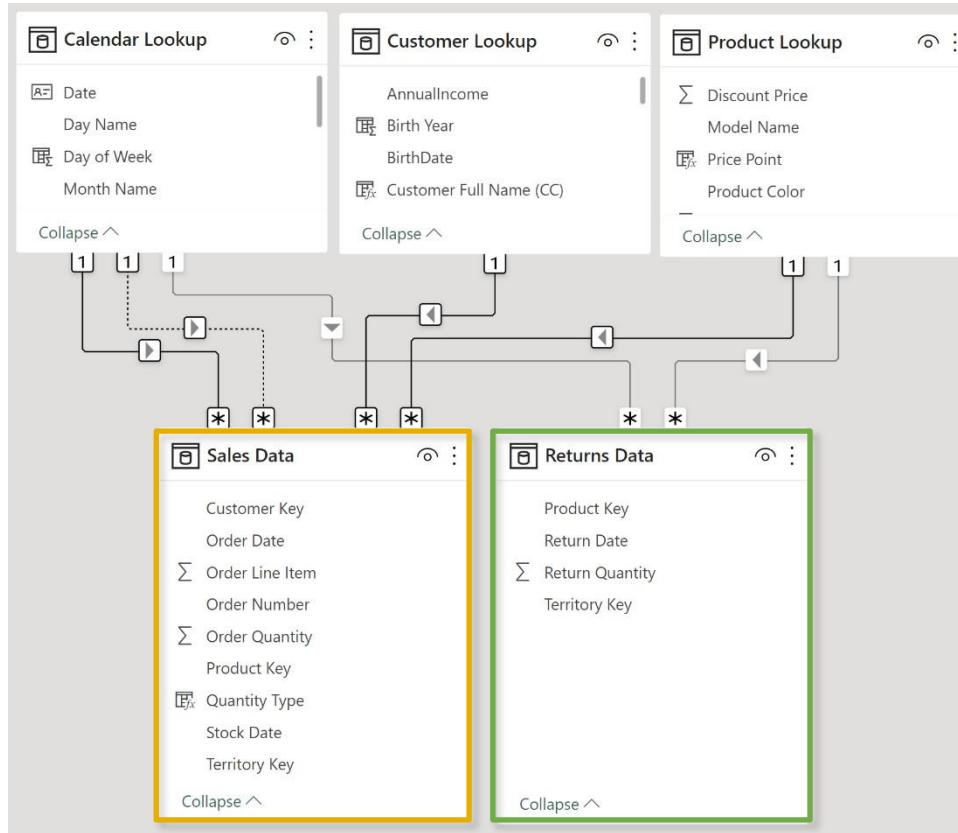


! This relationship has cardinality Many-Many. This should only be used if it is expected that neither column (product_id and product_id) contains unique values, and that the significantly different behavior of Many-many relationships is understood. [Learn more](#)

- If we try to connect the tables above using **product_id**, we'll get a **many-to-many relationship** warning since there are multiple instances of product_id in both tables
- Even if we force this relationship, how would we know which product was actually sold on each date – **Cream Soda** or **Diet Cream Soda**?



CONNECTING MULTIPLE FACT TABLES



This model contains two fact tables: **Sales Data** and **Returns Data**

- Since there is no primary/foreign key relationship, we can't connect them directly to each other
- But we *can* connect each fact table to related lookups, which allows us to filter both sales and returns data **using fields from any shared lookup tables**
- We can view orders and returns by product since both tables relate to Product Lookup, but we can't view returns by customer since no relationship exists

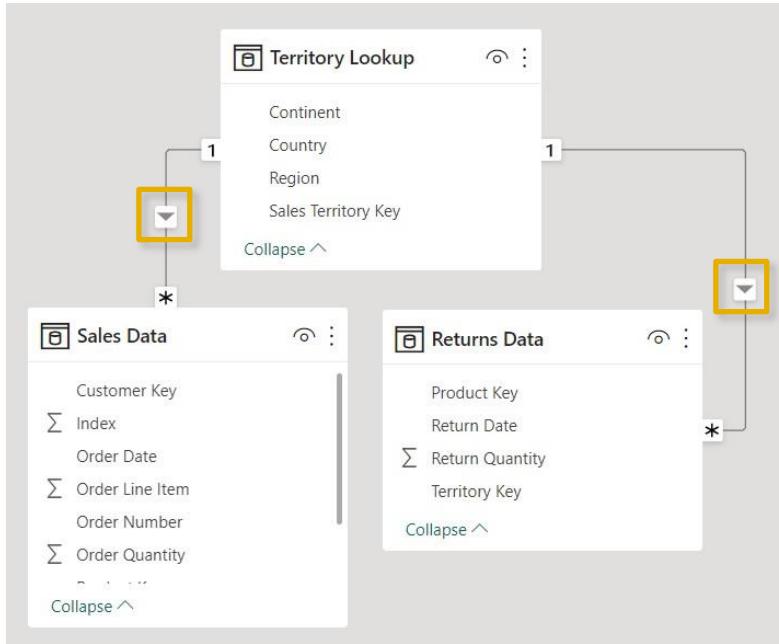


HEY THIS IS IMPORTANT!

Generally speaking, fact tables should **connect through shared dimension tables, not directly to each other**



FILTER CONTEXT & FLOW



Here we have two data tables (**Sales Data** and **Returns Data**), connected to **Territory Lookup**

The arrows show the **filter direction**, and point from the one (1) side of the relationship to the many (*) side

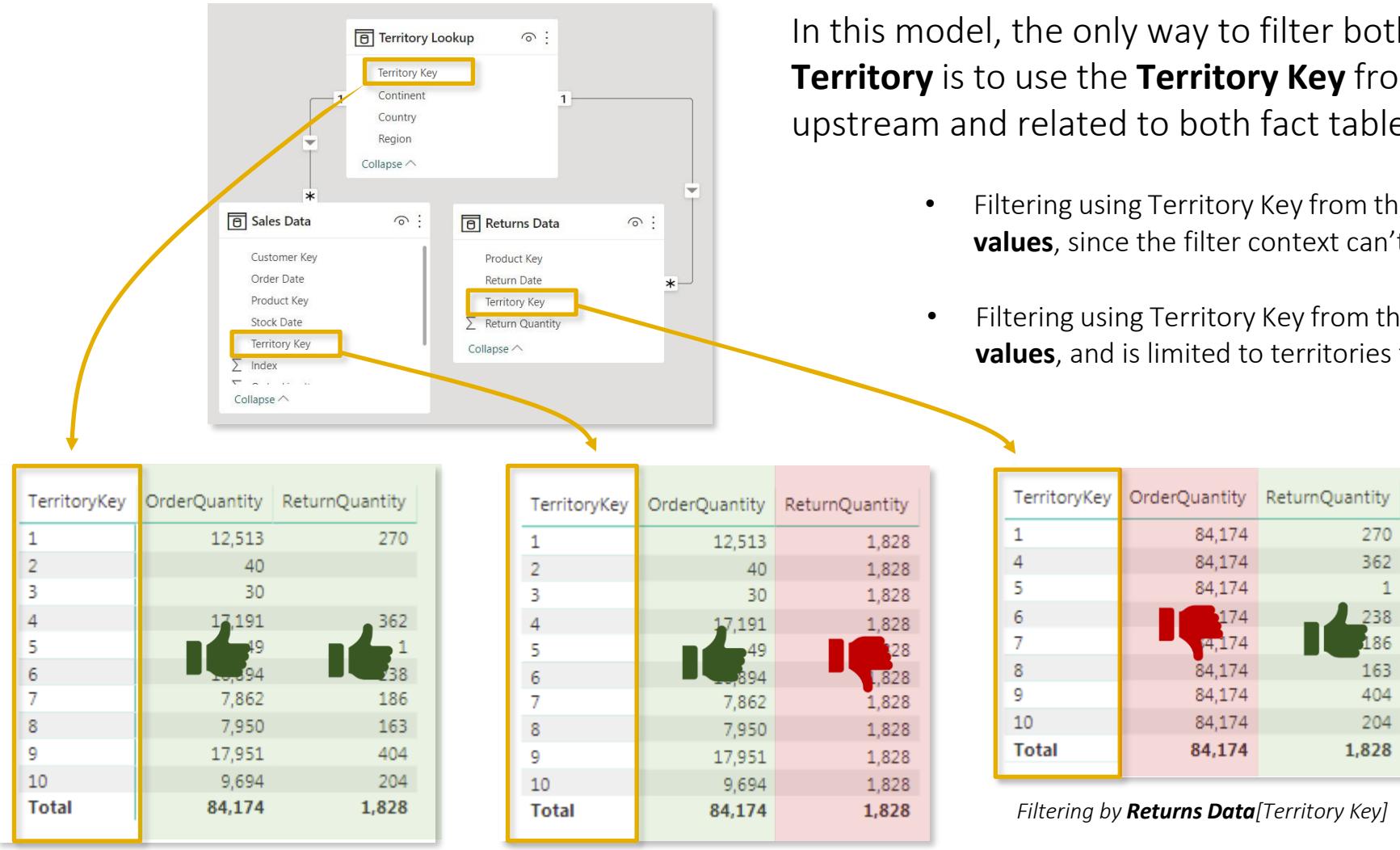
- When you filter a table, that **filter context** is passed to any related “downstream” tables, following the arrow’s direction
- Filter context CANNOT flow “upstream”



PRO TIP: Arrange lookup tables above fact tables in your model as a visual reminder that **filters always flow downstream**



EXAMPLE: FILTER FLOW



In this model, the only way to filter both **Sales** and **Returns** data by **Territory** is to use the **Territory Key** from the lookup table, which is upstream and related to both fact tables

- Filtering using Territory Key from the **Sales** table yields **incorrect Returns values**, since the filter context can't flow to any other table
- Filtering using Territory Key from the **Returns** table yields **incorrect Sales values**, and is limited to territories that exist in the returns table

Filtering by **Returns Data**[**Territory Key**]

Filtering by **Territory Lookup**[**Territory Key**]

Filtering by **Sales Data**[**Territory Key**]



BI-DIRECTIONAL FILTERS

Edit relationship

Select tables and columns that are related.

Sales Data

| Order Date | Stock Date | Order Number | Product Key | Customer Key | Territory Key | Or |
|-----------------------|---------------------------|--------------|-------------|--------------|---------------|----|
| Sunday, July 5, 2020 | Wednesday, June 3, 2020 | SO46718 | 360 | 12570 | 9 | |
| Tuesday, July 7, 2020 | Wednesday, April 22, 2020 | SO46736 | 360 | 12341 | 9 | |
| Sunday, July 12, 2020 | Tuesday, May 5, 2020 | SO46776 | 360 | 12356 | 9 | |

Territory Lookup

| Region | Country | Continent | Sales Territory Key |
|-----------|---------------|---------------|---------------------|
| Northwest | United States | North America | 1 |
| Northeast | United States | North America | 2 |
| Central | United States | North America | 3 |

Cardinality

Many to one (*:1)

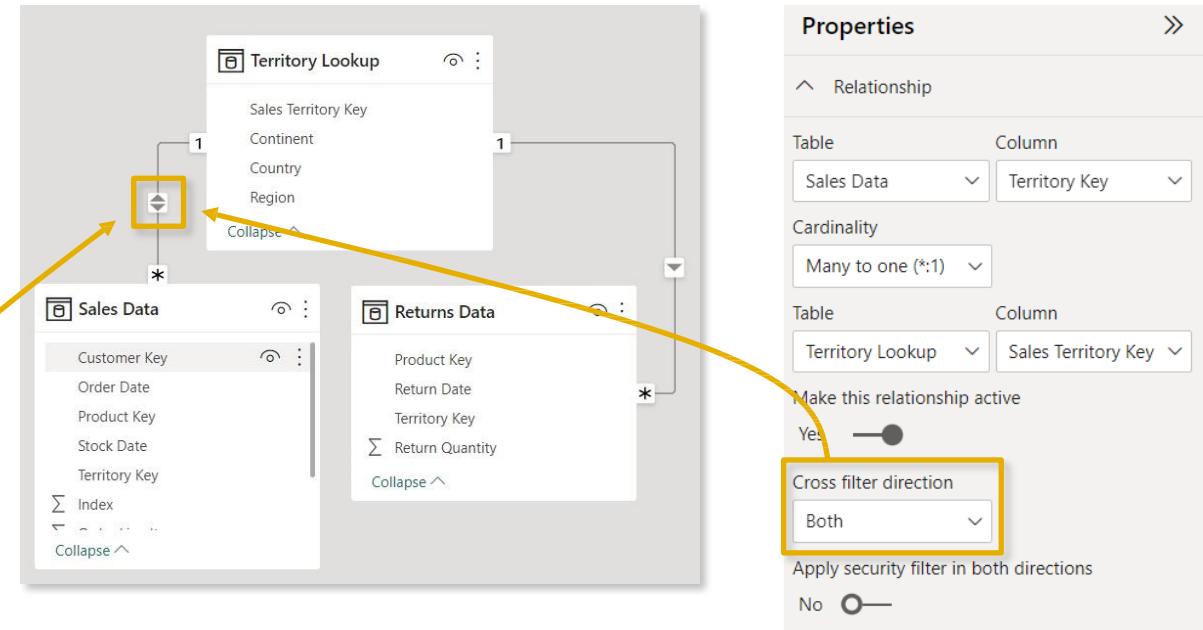
Make this relationship active

Assume referential integrity

Cross filter direction

Both

OK Cancel

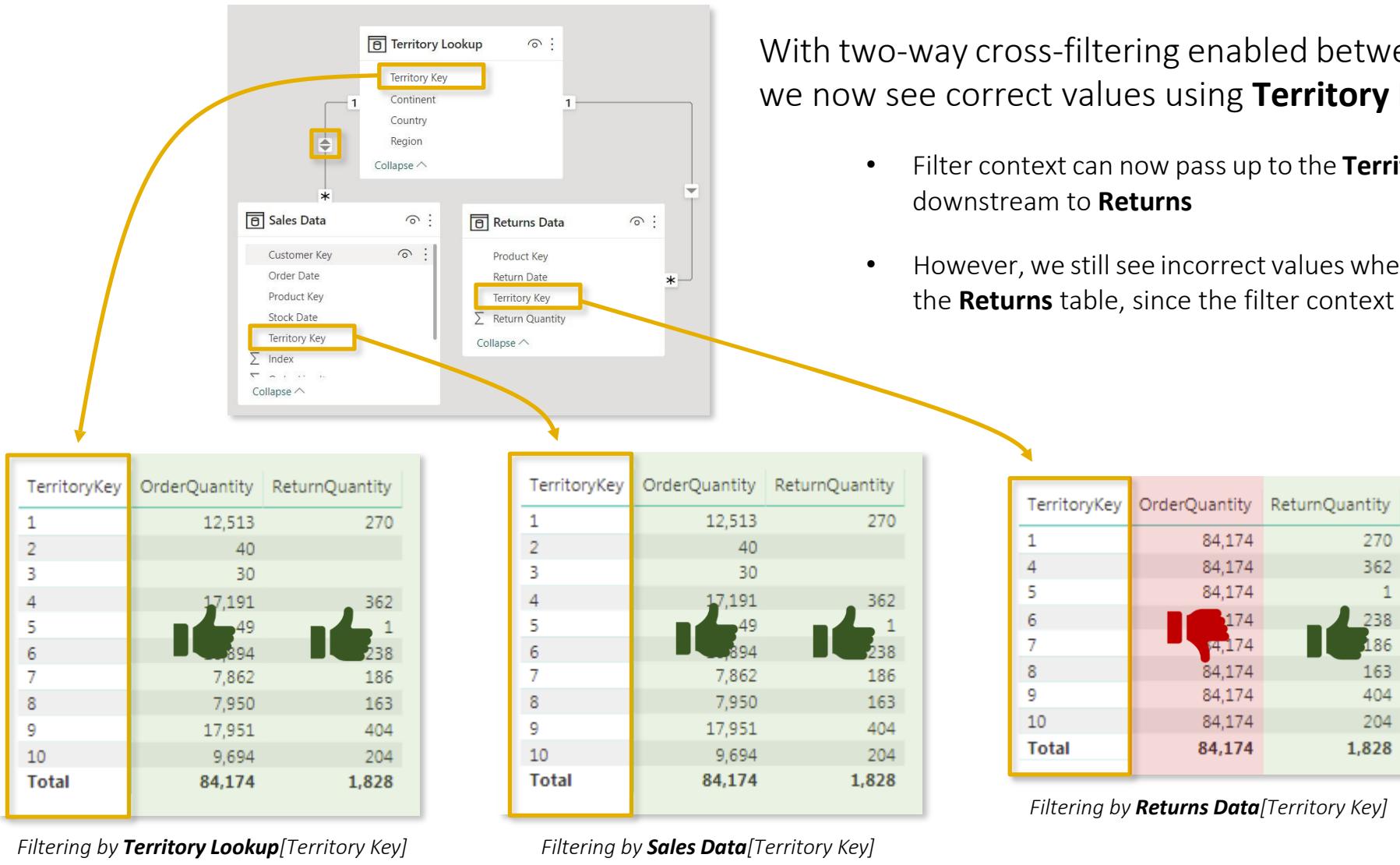


Updating the **cross-filter direction** from **Single** to **Both** allows filter context to flow in either direction

- In this example, filters applied to the **Sales** table can pass up to the **Territory Lookup** table, then down to **Returns**



EXAMPLE: BI-DIRECTIONAL FILTERS

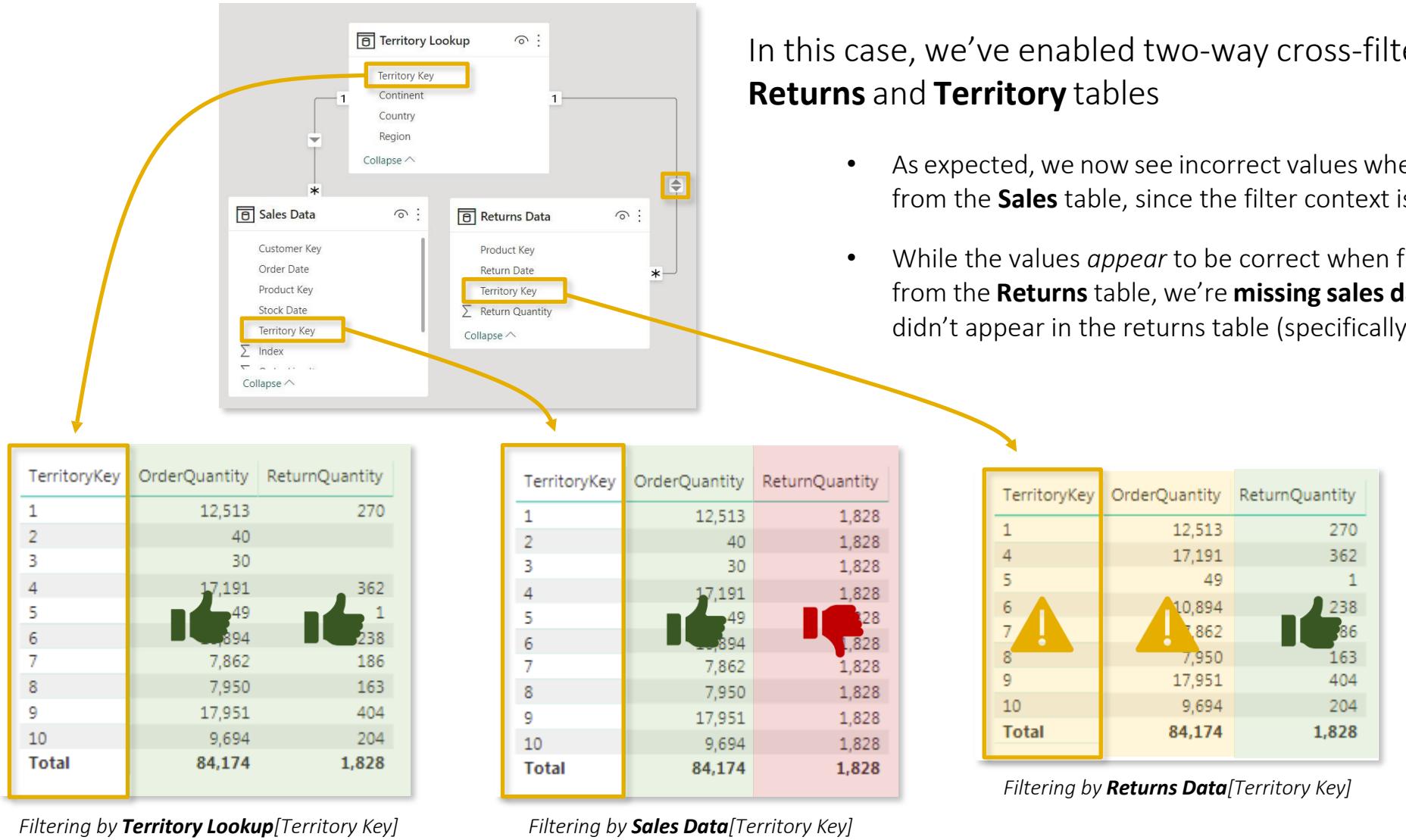


With two-way cross-filtering enabled between **Sales** and **Territory**, we now see correct values using **Territory Key** from *either* table

- Filter context can now pass up to the **Territory Lookup** table, then downstream to **Returns**
- However, we still see incorrect values when filtering using Territory Key from the **Returns** table, since the filter context is isolated to that single table



EXAMPLE: BI-DIRECTIONAL FILTERS

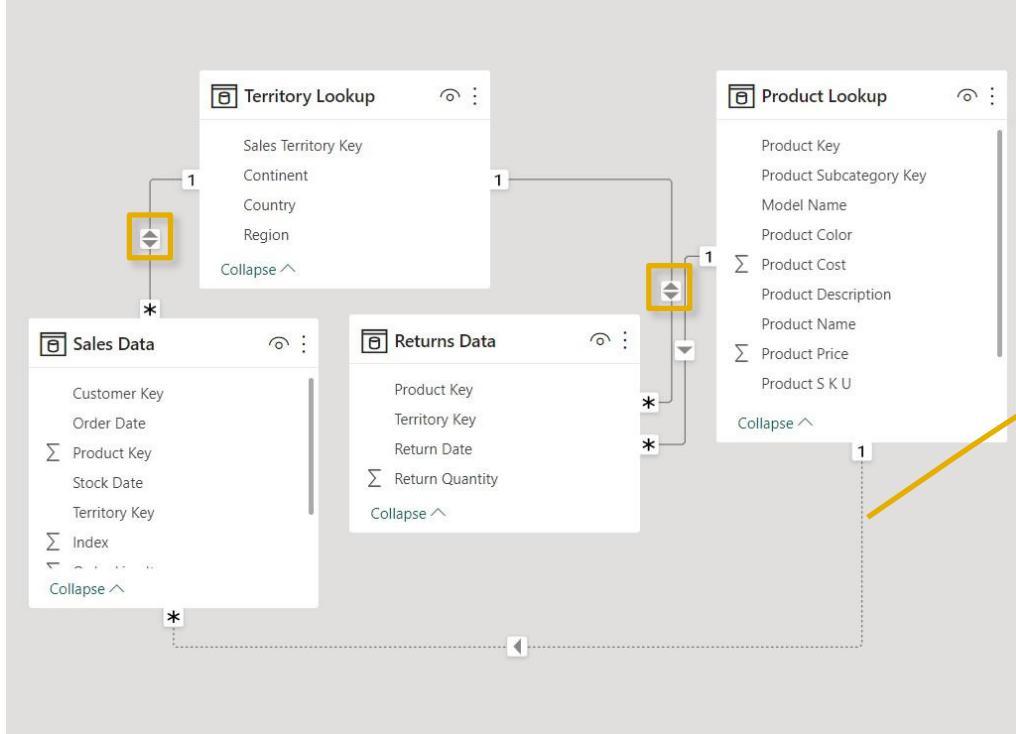


In this case, we've enabled two-way cross-filtering between the **Returns** and **Territory** tables

- As expected, we now see incorrect values when filtering using Territory Key from the **Sales** table, since the filter context is isolated to that single table
- While the values *appear* to be correct when filtering using Territory Key from the **Returns** table, we're **missing sales data** from any territories that didn't appear in the returns table (specifically Territories **2 & 3**)



AMBIGUITY



Use two-way filters carefully, and **only when necessary**

- Using multiple two-way filters can cause **ambiguity** by introducing multiple filter paths between tables

! You can't create a direct active relationship between **Sales_Data** and **Product_Lookup** because that would introduce ambiguity between tables **Product_Lookup** and **Territory_Lookup**. To make this relationship active, deactivate or delete one of the relationships between **Product_Lookup** and **Territory_Lookup** first.

In this example, filter context from the **Product** table can pass down to **Returns** and up to **Territory Lookup**, which would be filtered based on the Territory Keys passed from the Returns table

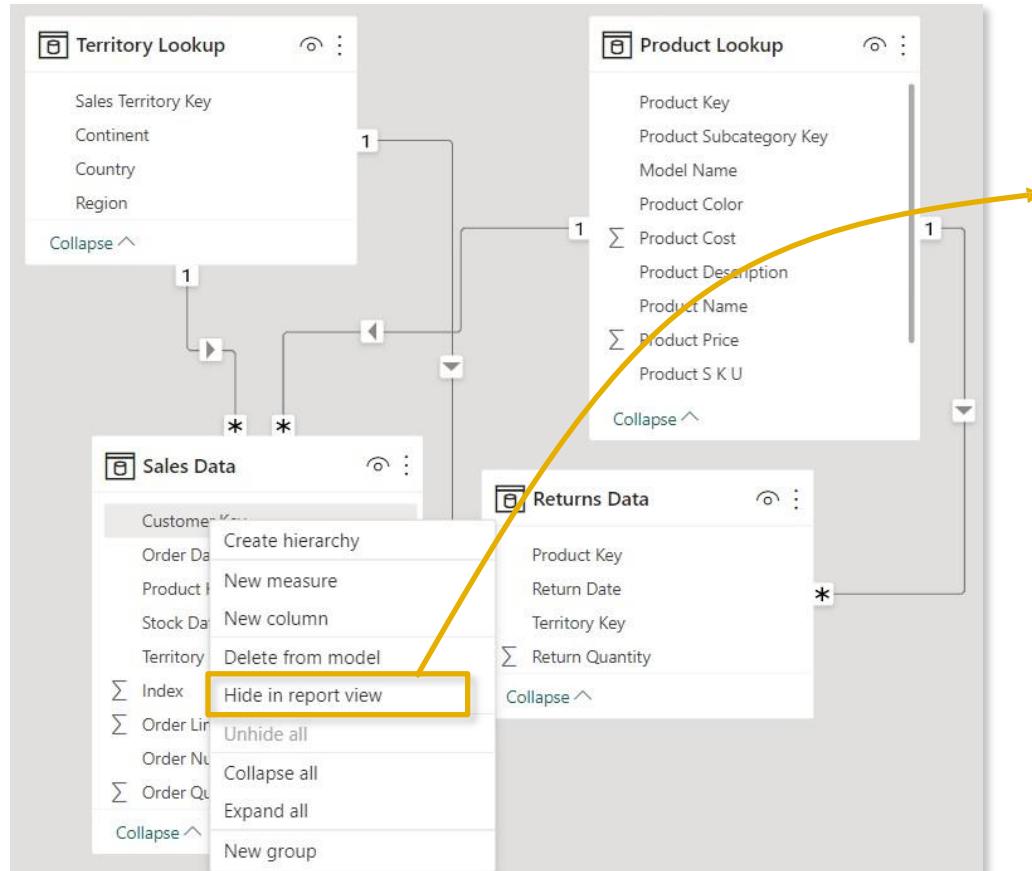
With an active relationship between **Product** and **Sales** as well, filter context could pass through **either the Sales or Returns table to reach the Territory Lookup table**, which could yield conflicting filter context



PRO TIP: Design your models with **one-way filters** and **1:many cardinality** unless more complex relationships are absolutely necessary



HIDING FIELDS



Hide in Report View makes fields inaccessible from the Report tab, but still available in **Data** and **Model** views

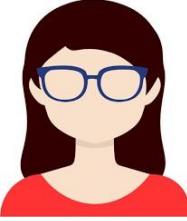
- This can be controlled by right-clicking a field in the Data or Model view, or by selecting “**Is hidden**” in the Properties pane
- This is commonly used to prevent users from filtering using invalid fields, reduce clutter, or to hide irrelevant metrics from view



PRO TIP: Hide the **foreign keys** in fact tables to force users to filter using **primary keys** in dimension tables



ASSIGNMENT: FILTER FLOW

 NEW MESSAGE

From: **Dana Modelle (Analyst)**

Subject: **Larry's gone rogue!**

Hey there, we've got another problem.

Larry from Sales just sent me this screenshot. I think he must have downloaded our Power BI model and messed with some relationships, because I KNOW we had sales for product 338.

Can you help diagnose what's going on, and prevent him from doing this again?

-Dana

P.S. Kevin says hi 

 Reply  Forward

Key Objectives

1. Replicate Larry's matrix below to diagnose what he must have done to the model*

| Product Key | Sum of Order Quantity | Sum of Return Quantity |
|-------------|-----------------------|------------------------|
| 322 | 55 | 2 |
| 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 |

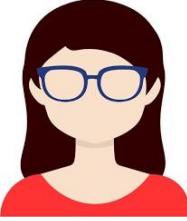
No sales for 338!?!?

- Which product is #338?
 - Why didn't Larry's matrix show any orders?
2. Hide any remaining foreign keys to prevent other users from making the same mistake

*Hint: you may need to temporarily change a relationship to bi-directional



SOLUTION: FILTER FLOW

 NEW MESSAGE

From: **Dana Modelle (Analyst)**

Subject: **Larry's gone rogue!**

Hey there, we've got another problem.

Larry from Sales just sent me this screenshot. I think he must have downloaded our Power BI model and messed with some relationships, because I KNOW we had sales for product 338.

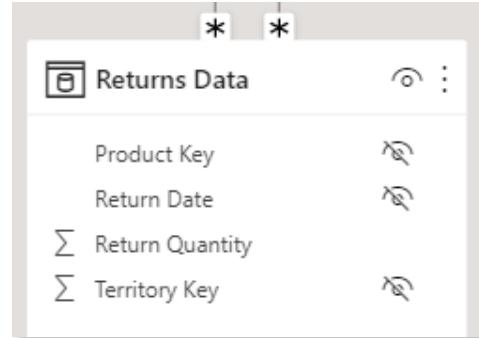
Can you help diagnose what's going on, and prevent him from doing this again?

-Dana

P.S. Kevin says hi 

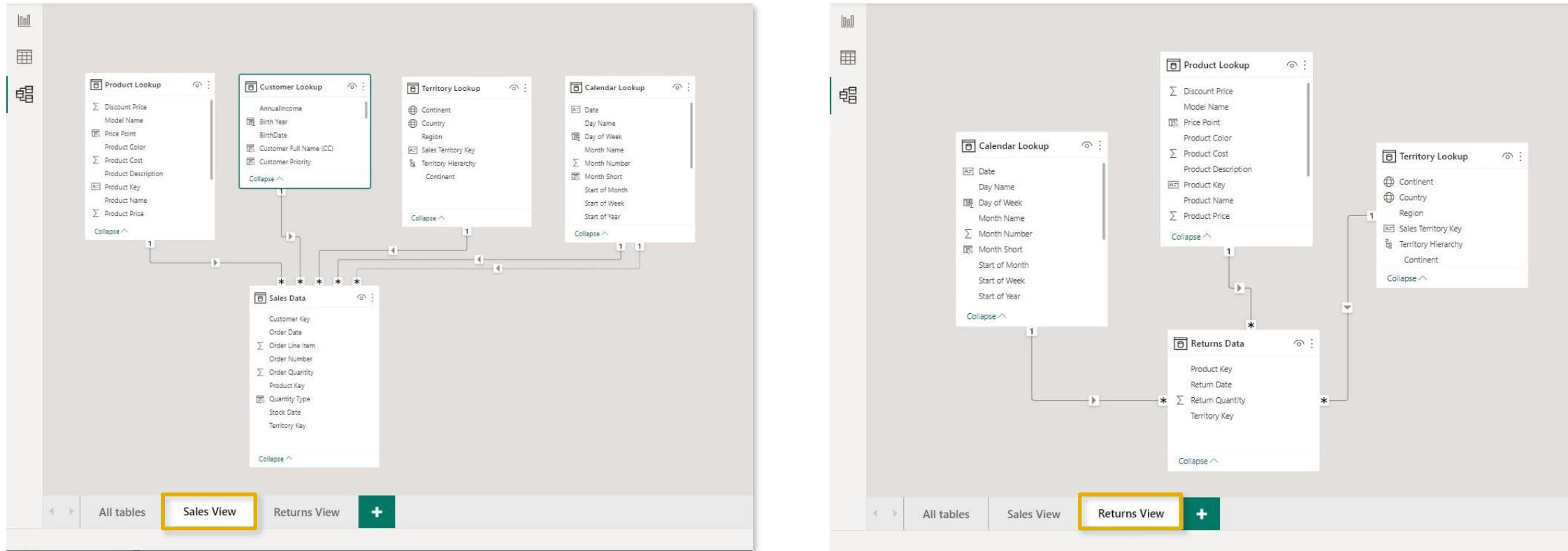
 

Solution Preview

1. Larry must have changed the relationship between **Returns Data** and **Product Lookup** to **bi-directional**, and filtered his matrix using product_id from the Returns table
 - Road bike (Road-650 Black, 44)
 - Product 338 doesn't exist in the Returns table, so it was excluded when that filter context passed to the Sales table
2. 



PRO TIP: 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

File Home Help External Tools **Table tools** **Column tools**

Name: Country Data type: Text

Structure:

| Region | Country | Continent | Sales Territory Key | |
|----------------|----------------|---------------|---------------------|----|
| Northwest | United States | North America | | 1 |
| Northeast | United States | North America | | 2 |
| Central | United States | North America | | 3 |
| Southwest | United States | North America | | 4 |
| Southeast | United States | North America | | 5 |
| Canada | Canada | North America | | 6 |
| France | France | Europe | | 7 |
| Germany | Germany | Europe | | 8 |
| Australia | Australia | Pacific | | 9 |
| United Kingdom | United Kingdom | Europe | | 10 |

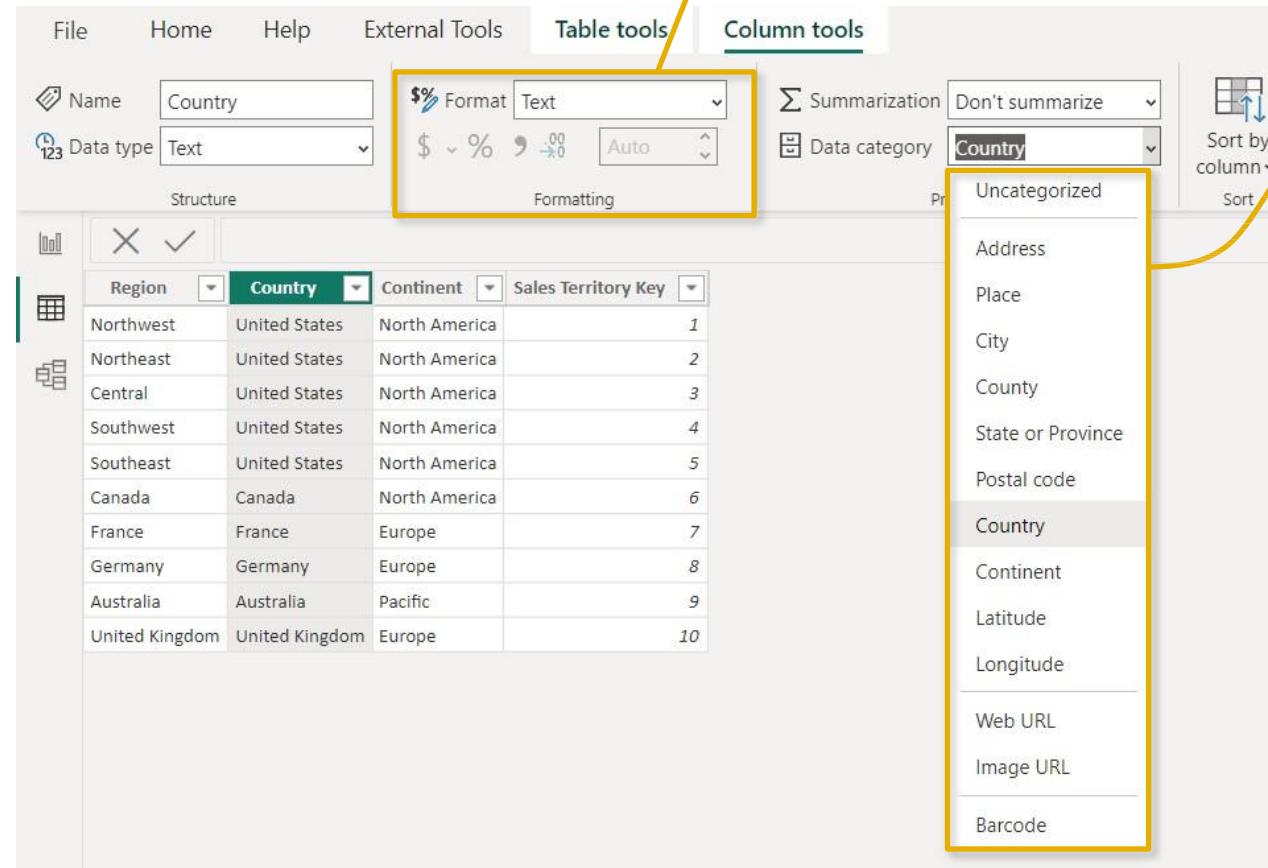
\$% Format Text **Formatting**

Σ Summarization Don't summarize

Data category: **Country**

Sort by column Sort

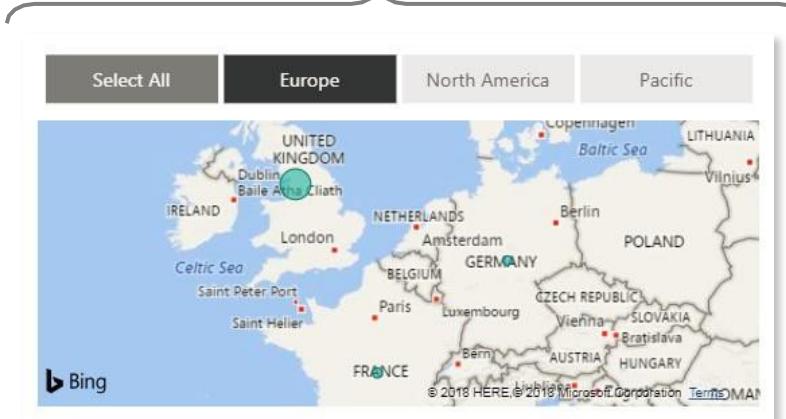
Uncategorized
Address
Place
City
County
State or Province
Postal code
Country
Continent
Latitude
Longitude
Web URL
Image URL
Barcode



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

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.





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 side-by-side screenshots of the Microsoft Power BI Data pane:

- Screenshot 1:** Shows the context menu for the "Product Price" column. The "Create hierarchy" option is highlighted with a yellow box and a callout arrow pointing to the second screenshot.
- Screenshot 2:** Shows the "Territory Hierarchy" node in the hierarchy tree under the "Territory Lookup" folder. The "Continent" node is also highlighted with a yellow box and a callout arrow pointing to the third screenshot.
- Screenshot 3:** Shows the context menu for the "Country" column. The "Add to hierarchy" option is highlighted with a yellow box and a callout arrow pointing to the first screenshot.

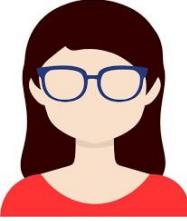
In the **Data** pane, right-click a field
and select **Create hierarchy**

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

 NEW MESSAGE

From: **Dana Modelle (Analyst)**

Subject: **Adding a date hierarchy**

Good morning!

Hoping you can help with a quick request.

Since we'll be doing a lot of time-series analysis, Ethan asked us to add a date hierarchy to the model so that users can quickly view trends at any level of granularity (year, month, day, etc.)

Please get that added before our afternoon call. Thanks!

-Dana

Reply Forward

Key Objectives

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



SOLUTION: HIERARCHIES

  NEW MESSAGE

From: **Dana Modelle (Analyst)**
Subject: **Adding a date hierarchy**

Good morning!

Hoping you can help with a quick request.

Since we'll be doing a lot of time-series analysis, Ethan asked us to add a date hierarchy to the model so that users can quickly view trends at any level of granularity (year, month, day, etc.)

Please get that added before our afternoon call. Thanks!

-Dana

[Reply](#) [Forward](#)

Solution Preview

Data >

Search

>  Measure Table

✓  Calendar Lookup

-  Average Revenue per Customer
-  Date
- ✓  Date Hierarchy
 - Start of Year
 - Start of Month
 - Start of Week
-  Date



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

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:

DAX 101

Columns & Measures

Row & Filter Context

DAX Syntax

Common Functions

Calculate

Iterators

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



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

| Customer Data | | | | | |
|----------------|------------------------------|---------------|----------------|-----------------|--------|
| 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 | martha13@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 | alexa8@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 |

Measures

The screenshot shows the Power BI Query Editor with a context menu open. The menu items are: Returns Data, Rolling Calendar, New measure (highlighted), New column, New quick me..., Refresh data, Edit query. Below the menu, three DAX measures are listed:

- Total Orders = DISTINCTCOUNT(Sales_Data[OrderNumber])
- Total Revenue = SUMX(Sales_Data, Sales_Data[OrderQuantity] * RELATED(Product_Lookup[ProductPrice]))
- Quantity Ordered = SUM(Sales_Data[OrderQuantity])



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

Query Editor:

Properties pane:
Name: Territory Lookup
All Properties

Applied Steps pane:
Source
Promoted Headers
Changed Type

```
#"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 |

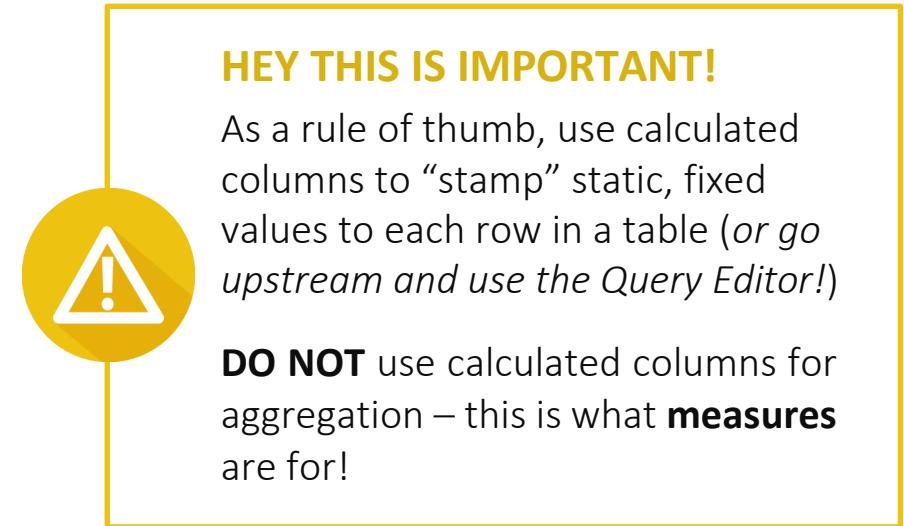
```
1 Bike Returns =
2 CALCULATE(
3     [Total Returns],                                // Counting total returns
4     'Product Categories Lookup'[Category Name] = "Bikes" // filtered for bikes only
5 )
```



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!



PRO TIP:

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



EXAMPLE: CALCULATED COLUMNS

The screenshot shows a Power BI Data View window. A calculated column named "Parent" is being defined. The formula is `1 Parent = IF('Customer Lookup'[Total Children]>0,"Yes","No")`. The resulting column "Parent" contains the value "Yes" for all rows. The "Fields" pane on the right lists various columns from the "Customer Lookup" table, including Annual Income, Birth Date, Customer Key, Education Level, Email Address, and First Name.

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 **TotalQuantity**

- 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.

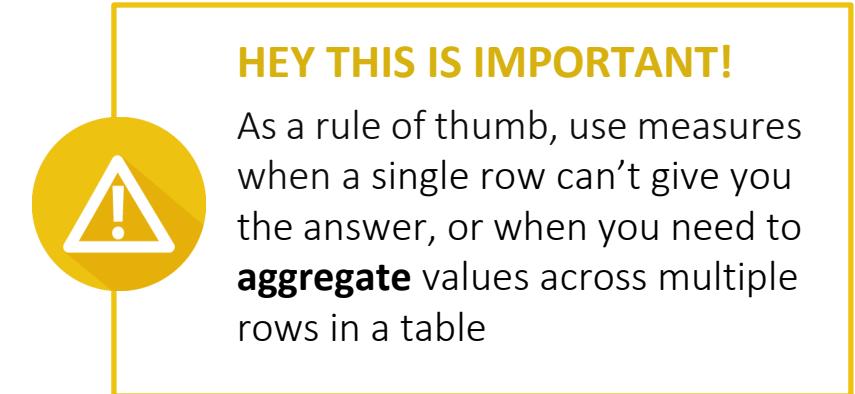
The screenshot shows a Power BI Data View window. A calculated column named "TotalQuantity" is being defined. The formula is `1 TotalQuantity = SUM('Sales Data'[Order Quantity])`. The resulting column "TotalQuantity" contains the value 360 for all rows. The "Fields" pane on the right lists various columns from the "Sales Data" table, including Customer Key, Territory Key, Order Line Item, Order Quantity, Index, and Order Date.



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



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

The screenshot shows the 'Build a visual' interface in Power BI. On the left, under 'Y-axis', there is a box containing 'Sum of Order ...'. This box is highlighted with a yellow border. Below it is a button labeled '+Add data'. A yellow arrow points from this button to the 'Order Quantity' item in the 'Select data' dialog box on the right. The 'Select data' dialog lists various data items, with 'Order Quantity' being the one selected.

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”



QUICK MEASURES

Quick measures automatically create formulas based on pre-built templates or natural language prompts

Quick measure

Select a calculation to create a measure or describe the measure you need and we'll generate suggestions in DAX, which you can customize later.

Calculations Suggestions

Weighted average per category

Calculate a weighted average of the base value for each category. Multiply the value by weight for each category, sum total, and then divide by the sum total of the weight. [Learn more](#)

Base value ?

Sum of Order Quantity

X | >

Weight ?

Add data

Category ?

Add data

Quick measure **calculations** can be used to build measures using **predefined templates** (*weighted averages, percent difference, time intelligence, etc.*)

Quick measure

Select a calculation to create a measure or describe the measure you need and we'll generate suggestions in DAX, which you can customize later.

Calculations Suggestions

Sum of quantity sold by calendar lookup year

X

Generate

Suggested measures

Total quantity sold per year

Quick measure **suggestions** can be used to find suggested measures based on **natural language queries** (*i.e. “sum of quantity sold by calendar year”*)

PRO TIP:



Quick measures can be a great learning tool for beginners or for building more complex formulas but use them with caution; **mastering DAX requires a deep understanding of the underlying theory!**



RECAP: CALCULATED COLUMNS VS. MEASURES

CALCULATED COLUMNS

- Values are calculated based on information from each row of a table (**row context**)
- Appends static values to each row in a table and stores them in the model (*which increases file size*)
- Recalculate on data source refresh or when changes are made to component columns
- Primarily used for **filtering** data in reports

| Birth Date | Marital Status | Email Address | Annual Income | Total Children | Education Level | Parent |
|------------|----------------|-------------------------------|---------------|----------------|-----------------|--------|
| 9/3/1943 | M | emma32@adventure-works.com | 70000 | 5 | Bachelors | Yes |
| 9/14/1967 | M | barry20@adventure-works.com | 40000 | 5 | High School | Yes |
| 8/5/1945 | M | martha13@adventure-works.com | 70000 | 5 | High School | Yes |
| 6/4/1946 | S | tamara16@adventure-works.com | 40000 | 5 | High School | Yes |
| 10/16/1970 | S | gerald21@adventure-works.com | 130000 | 5 | Bachelors | Yes |
| 5/10/1945 | M | alex8@adventure-works.com | 40000 | 5 | High School | Yes |
| 9/24/1938 | M | jack53@adventure-works.com | 70000 | 5 | Graduate Degree | Yes |
| 7/21/1959 | S | ricky1@adventure-works.com | 100000 | 5 | Bachelors | Yes |
| 1/6/1962 | M | keith4@adventure-works.com | 70000 | 5 | Partial College | Yes |
| 8/15/1962 | M | latoya19@adventure-works.com | 70000 | 5 | Bachelors | Yes |
| 1/26/1967 | S | micalah11@adventure-works.com | 70000 | 5 | Bachelors | Yes |
| 3/8/1946 | M | mindy22@adventure-works.com | 80000 | 5 | Partial College | Yes |
| 6/11/1960 | M | teresa8@adventure-works.com | 70000 | 5 | Partial College | Yes |

Calculated columns “live” in **tables**

MEASURES

- Values are calculated based on information from any filters in the report (**filter context**)
- Does not create new data in the tables themselves (*doesn’t increase file size*)
- Recalculate in response to any change to filters within the report
- Primarily used for **aggregating values** in report visuals



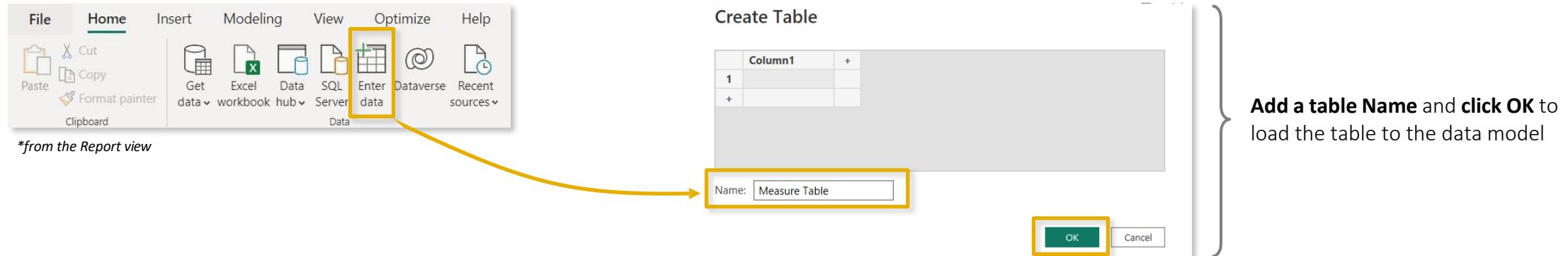
Measures “live” in **visuals**



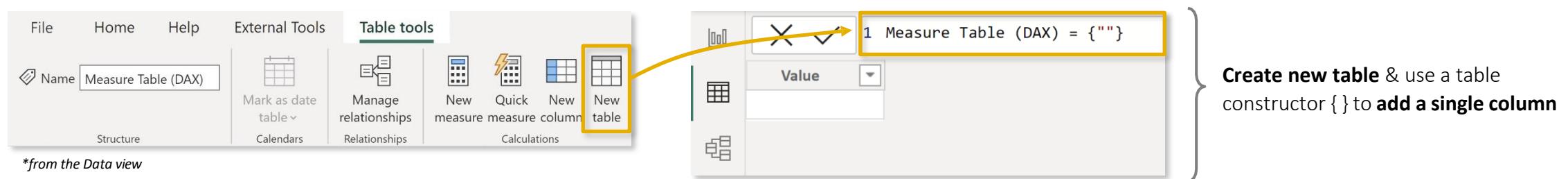
PRO TIP: MEASURE TABLES

It's a common best practice to **create a dedicated table to store your measures**; this will help you stay organized, find measures quickly, and allow you to group related measures into folders

Option 1: Enter Data into Power Query (loads the table to the data model – table is visible in Power Query)



Option 2: Create a calculated table using DAX directly in the model (table is not visible in Power Query)





FILTER CONTEXT

Measures are evaluated based on **filter context**, which means that they recalculate whenever the fields or filters around them change

| Top 10 Products | Orders | Revenue | Return % |
|-------------------------|---------------|------------------|--------------|
| Water Bottle - 30 oz. | 3,983 | \$39,755 | 1.95% |
| Patch Kit/8 Patches | 2,952 | \$13,506 | 1.61% |
| Mountain Tire Tube | 2,846 | \$28,333 | 1.64% |
| Road Tire Tube | 2,173 | \$17,265 | 1.55% |
| Sport-100 Helmet, Red | 2,099 | \$73,444 | 3.33% |
| AWC Logo Cap | 2,062 | \$35,865 | 1.11% |
| Sport-100 Helmet, Blue | 1,995 | \$67,112 | 3.31% |
| Fender Set - Mountain | 1,975 | \$87,041 | 1.36% |
| Sport-100 Helmet, Black | 1,940 | \$65,262 | 2.68% |
| Mountain Bottle Cage | 1,896 | \$38,062 | 2.02% |
| Total | 15,587 | \$465,644 | 1.85% |

For this value in the matrix (2,846), the **Orders** measure is calculated based on the following filter context: *Products[Product Name] = “Mountain Tire Tube”*

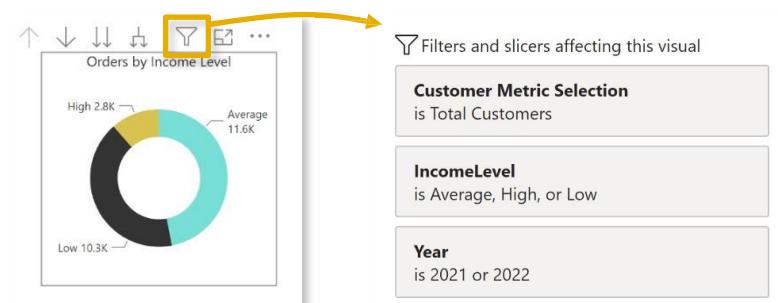
- This allows the measure to return the total order quantity for each product specifically (or whatever context the row and column labels dictate – years, countries, categories, customer names, etc.)

This total (15,587) does **NOT** calculate by summing the values above; it evaluates as an independent measure with **no filter context** applied

- IMPORTANT:** Every measure value in a report evaluates **independently** (like an island) and calculates based on its own filter context



PRO TIP: Clicking the **filter icon** will show you the filters currently applied to a selected visual





EXAMPLE: FILTER CONTEXT

MEASURE: Revenue Per Customer

FILTER CONTEXT:

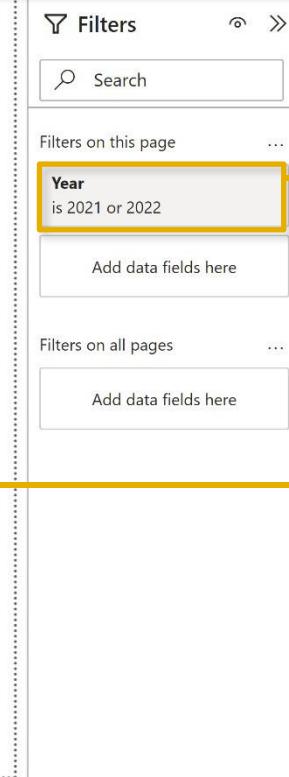
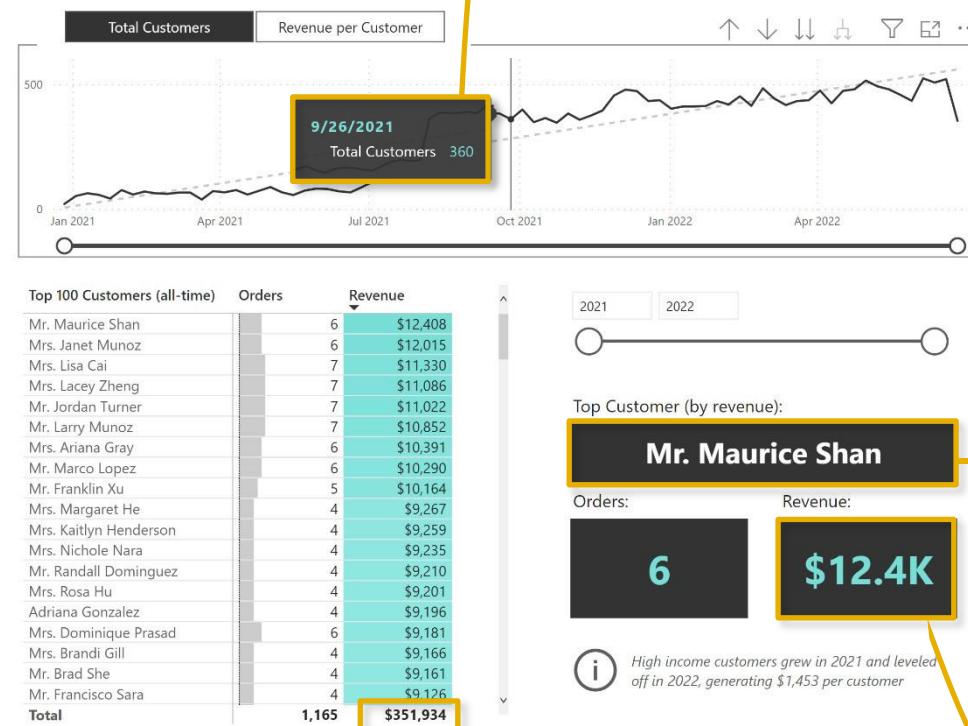
- Calendar[Year] = 2021 or 2022



MEASURE: Total Customers

FILTER CONTEXT:

- Calendar[Date] = September 26, 2021



This is a **page-level filter**, which impacts **ALL** visuals on this report page (more on this later!)

MEASURE: Total Orders

FILTER CONTEXT:

- Calendar[Year] = 2021 or 2022
- Customers[Occupation] = Skilled Manual

MEASURE: Total Revenue

FILTER CONTEXT:

- Calendar[Year] = 2021 or 2022
- Customer[Full Name] = Top 100 by Total Orders

MEASURE: Total Revenue

FILTER CONTEXT:

- Calendar[Year] = 2021 or 2022
- Customer[Full Name] = Mr. Maurice Shan



STEP-BY-STEP MEASURE CALCULATION

| Product Color | Quantity Sold |
|---------------|---------------|
| Black | 10,590 |
| Red | 4,011 |
| Yellow | 4,638 |

How exactly is this measure value calculated?

- **NOTE:** This all happens *instantly* behind the scenes, every time the filter context changes

STEP 1

Filter context is detected & applied



| Product Color | Quantity Sold |
|---------------|---------------|
| Black | 10,590 |
| Red | 4,011 |
| Yellow | 4,638 |

'Product Lookup'[Product Color] = "Black"

STEP 2

Filters flow “downstream” to related tables



The screenshot shows the 'Product Lookup Table' with columns: Product ID, Product Name, Model Name, Product Description, Product Color, Product Type, and Product Key. The 'Product Color' column highlights the row for 'Black'.

Black

The screenshot shows the 'Sales Data' table with columns: Order Date, Stock Date, Order Number, Product Key, and Returns Data. The 'Product Color' column highlights the row for 'Black'.

Black

STEP 3

Measure evaluates against the filtered table



```

1 Quantity Sold =
2 SUM(
3   |
4     'Sales Data'[Order Quantity]
5   )

```

Sum of values in the **Order Quantity** column of the **Sales Data** table, filtered to rows where the product color is "**Black**"

= 10,590

The screenshot shows the 'Returns Data' table with columns: Return Date, Territory Key, Product Key, and Return Quantity. The 'Product Color' column highlights the row for 'Black'.

Black



DAX SYNTAX

MEASURE NAME

- Measures are always surrounded by brackets (i.e. **[Total Quantity]**) when referenced in formulas, so spaces are OK

Total Quantity: = **SUM(Transactions[quantity])**

FUNCTION NAME

- Calculated columns don't always use functions, but measures do:
 - In a **Calculated Column**, **=Transactions[quantity]** returns the value from the quantity column in each row (*since it evaluates one row at a time*)
 - In a **Measure**, **=Transactions[quantity]** will return an **error** since Power BI doesn't know how to translate that as a single value – you need some sort of aggregation

Referenced
TABLE NAME

Referenced
COLUMN NAME

This is a “**fully qualified**” column, since it’s preceded by the table name.

NOTE: Table names with spaces must be surrounded by **single quotes**:

- Without a space: **Transactions[quantity]**
- With a space: **'Transactions Table'[quantity]**



PRO TIP:

Column references use fully qualified names (i.e. **'Table'[Column]**)

Measure references just use the measure name (i.e. **[Measure]**) and can be called by typing an open square bracket “ [“



DAX OPERATORS

| Arithmetic Operator | Meaning | Example |
|---------------------|----------------|--------------|
| + | Addition | $2 + 7$ |
| - | Subtraction | $5 - 3$ |
| * | Multiplication | $2 * 6$ |
| / | Division | $4 / 2$ |
| \wedge | Exponent | $2 \wedge 5$ |

Pay attention to these!

| Comparison Operator | Meaning | Example |
|---------------------|--------------------------|---------------------------|
| = | Equal to | [City] = "Boston" |
| > | Greater than | [Quantity] > 10 |
| < | Less than | [Quantity] < 10 |
| \geq | Greater than or equal to | [Unit Price] \geq 2.5 |
| \leq | Less than or equal to | [Unit Price] \leq 2.5 |
| \neq | Not equal to | [Country] \neq "Mexico" |

| Text/Logical Operator | Meaning | Example |
|-----------------------|---|---|
| & | Concatenates two values to produce one text string | [City] & " " & [State] |
| && | Create an AND condition between two logical expressions | ([State] = "MA") && ([Quantity] > 10) |
| (double pipe) | Create an OR condition between two logical expressions | ([State] = "MA") ([State] = "CT") |
| IN | Creates a logical OR condition based on a given list (using curly brackets) | 'Store Lookup'[State] IN { "MA", "CT", "NY" } |

*Head to <https://learn.microsoft.com> for more information about DAX syntax, operators, troubleshooting, etc.



COMMON FUNCTION CATEGORIES

| MATH & STATS Functions | LOGICAL Functions | TEXT Functions | FILTER Functions | TABLE Functions | DATE & TIME Functions | RELATIONSHIP Functions |
|--|---|---|--|--|---|--|
| <p><i>Functions used for aggregation or iterative, row-level calculations</i></p> <p>Common Examples:</p> <ul style="list-style-type: none">• SUM• AVERAGE• MAX/MIN• DIVIDE• COUNT/COUNTA• COUNTROWS• DISTINCTCOUNT <p>Iterator Functions:</p> <ul style="list-style-type: none">• SUMX• AVERAGEX• MAXX/MINX• RANKX• COUNTX | <p><i>Functions that use conditional expressions (IF/THEN statements)</i></p> <p>Common Examples:</p> <ul style="list-style-type: none">• IF• IFERROR• AND• OR• NOT• SWITCH• TRUE• FALSE | <p><i>Functions used to manipulate text strings or value formats</i></p> <p>Common Examples:</p> <ul style="list-style-type: none">• CONCATENATE• COMBINEVALUES• FORMAT• LEFT/MID/RIGHT• UPPER/LOWER• LEN• SEARCH/FIND• REPLACE• SUBSTITUTE• TRIM | <p><i>Functions used to manipulate table and filter contexts</i></p> <p>Common Examples:</p> <ul style="list-style-type: none">• CALCULATE• FILTER• ALL• ALLEXCEPT• ALLSELECTED• KEEPFILTERS• REMOVEFILTERS• SELECTEDVALUE | <p><i>Functions that create or manipulate tables and output tables vs. scalar values</i></p> <p>Common Examples:</p> <ul style="list-style-type: none">• SUMMARIZE• ADDCOLUMNS• GENERATESERIES• DISTINCT• VALUES• UNION• INTERSECT• TOPN | <p><i>Functions used to manipulate date & time values or handle time intelligence calculations</i></p> <p>Common Examples:</p> <ul style="list-style-type: none">• DATE• DATEDIFF• YEARFRAC• YEAR/MONTH• DAY/HOUR• TODAY/NOW• WEEKDAY• WEEKNUM• NETWORKDAYS <p>Time Intelligence:</p> <ul style="list-style-type: none">• DATESYTD• DATESMTD• DATEADD• DATESBETWEEN | <p><i>Functions used to manage & modify table relationships</i></p> <p>Common Examples:</p> <ul style="list-style-type: none">• RELATED• RELATEDTABLE• CROSSFILTER• USERELATIONSHIP |

***Note:** This is NOT a comprehensive list. DAX contains more than 250 different functions!



BASIC MATH & STATS FUNCTIONS

SUM

Evaluates the sum of a column

=SUM(Column**Name**)

AVERAGE

Returns the average (arithmetic mean) of all the numbers in a column

=AVERAGE(Column**Name**)

MAX

Returns the largest value in a column or between two scalar expressions

=MAX(Column**NameOrScalar1**, [Scalar2])

MIN

Returns the smallest value in a column or between two scalar expressions

=MIN(Column**NameOrScalar1**, [Scalar2])

DIVIDE

Performs division and returns the alternate result (or blank) if DIV/0

=DIVIDE(Numerator, Denominator, [AlternateResult])



COUNTING FUNCTIONS

COUNT

Counts the number of non-empty cells in a column
(excluding Boolean values)

=**COUNT**(ColumnName)

COUNTA

Counts the number of non-empty cells in a column
(including Boolean values)

=**COUNTA**(ColumnName)

DISTINCTCOUNT

Counts the number of distinct values in a column

=**DISTINCTCOUNT**(ColumnName)

COUNTROWS

Counts the number of rows in the specified table,
or a table defined by an expression

=**COUNTROWS**([Table])



ASSIGNMENT: MATH & STATS



NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Help with a few measures**

Hey there, excited to start working with you!

I'll need to pull some high-level metrics from our model to share with leadership, and I could use some help with the calculations.

For now, could you please create one measure to calculate the total number of distinct customers, and a second measure that we can use to calculate return rate (quantity returned / quantity sold)? Thank you!

-Dianne

Reply **Forward**

Key Objectives

1. Create a measure named **Total Customers**, to calculate the number of distinct AdventureWorks customers who made a transaction
2. Create a measure named **Return Rate**, defined as quantity returned divided by quantity sold



SOLUTION: MATH & STATS

 NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Help with a few measures**

Hey there, excited to start working with you!

I'll need to pull some high-level metrics from our model to share with leadership, and I could use some help with the calculations.

For now, could you please create one measure to calculate the total number of distinct customers, and a second measure that we can use to calculate return rate (quantity returned / quantity sold)? Thank you!

-Dianne

Reply Forward

Solution Preview

```
1 Total Customers =  
2 DISTINCTCOUNT(  
3     'Sales Data'[Customer Key]  
4 )
```

```
1 Return Rate =  
2 DIVIDE(  
3     [Quantity Returned],  
4     [Quantity Sold],  
5     "No Sales"  
6 )
```



BASIC LOGICAL FUNCTIONS

IF

Checks if a given condition is met and returns one value if the condition is TRUE, and another if the condition is FALSE

=**IF**(LogicalTest, ResultIfTrue, [ResultIfFalse])

IFERROR

Evaluates an expression and returns a specified value if it returns an error, otherwise returns the expression itself

=**IFERROR**(Value, ValueIfError)

SWITCH

Evaluates an expression against a list of values and returns one of multiple possible expressions

=**SWITCH**(Expression, Value1, Result1, ..., [Else])

AND

Checks whether both arguments are TRUE to return TRUE, otherwise returns FALSE

=**AND**(Logical1, Logical2)

*Note: Use the **&&** and **||** operators to include more than two conditions*

OR

Checks whether any argument is TRUE to return TRUE, otherwise returns FALSE

=**OR**(Logical1, Logical2)



SWITCH

SWITCH

Evaluates an expression against a list of values and returns one of multiple possible expressions

=**SWITCH**(Expression, Value1, Result1, ..., [Else])

Any **DAX expression** that returns a single scalar value, evaluated multiples times

Examples:

- *Calendar[Month ID]*
- *'Product Lookup'[category]*

List of **values** produced by the expression, each paired with a result to return for rows/cases that match

Examples:

```
=SWITCH( Calendar[Month ID],  
        1, "January",  
        2, "February" )
```

Value returned if the expression doesn't match any value argument



PRO TIP

SWITCH(TRUE) is a common DAX pattern to replace multiple nested IF statements

ASSIGNMENT: LOGICAL FUNCTIONS





NEW MESSAGE

From: **Dianne A. Xu (Senior Analyst)**

Subject: **Customer segmentation fields**

Hey there!

Ethan has been working with the DS team on a customer segmentation analysis, and came back to us with a few requests.

Could you please add some new columns in our customer table to identify “priority” customers, segment customers based on income level, and group some of the education categories?

I've attached the logic to use, but reach out with any questions!

-Dianne

Reply **Forward**

Key Objectives

1. Create a calculated column in the Customer Lookup table named **Customer Priority**:
 - If the customer is a parent and has an annual income > \$100,000, Customer Priority = **Priority**
 - Otherwise, Customer Priority = **Standard**

2. Create a calculated column in the Customer Lookup table named **Income Level**:
 - If annual income is >= \$150,000, **Very High**
 - If annual income is >= \$100,000, **High**
 - If annual income is >= \$50,000, **Average**
 - Otherwise, Income Level = **Low**

ASSIGNMENT: LOGICAL FUNCTIONS



 NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Customer segmentation fields**

Hey there!

Ethan has been working with the DS team on a customer segmentation analysis, and came back to us with a few requests.

Could you please add some new columns in our customer table to identify “priority” customers, segment customers based on income level, and group some of the education categories?

I've attached the logic to use, but reach out with any questions!

-Dianne

Reply Forward

Key Objectives

BONUS: Use a SWITCH function* to create another column named **Education Category**:

- If EducationLevel is High School or Partial High School, Education Category = **High School**
- If EducationLevel is Bachelors or Partial College, Education Category = **Undergrad**
- If EducationLevel is Graduate Degree, Education Category = **Graduate**

*You can use the “data groups” tool to do this too!



SOLUTION: LOGICAL FUNCTIONS

 NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Customer segmentation fields**

Hey there!

Ethan has been working with the DS team on a customer segmentation analysis, and came back to us with a few requests.

Could you please add some new columns in our customer table to identify “priority” customers, segment customers based on income level, and group some of the education categories?

I've attached the logic to use, but reach out with any questions!

-Dianne

Reply Forward

Solution Preview

```
1 Customer Priority =  
2 IF(  
3     'Customer Lookup'[AnnualIncome] > 100000 &&  
4     'Customer Lookup'[Is Parent?] = "Yes",  
5     "Priority",  
6     "Standard"  
7 )
```

```
1 Income Level =  
2 IF('Customer Lookup'[AnnualIncome] >= 150000, "Very High",  
3 IF('Customer Lookup'[AnnualIncome] >= 100000, "High",  
4 IF('Customer Lookup'[AnnualIncome] >= 50000, "Average",  
5 "Low")))
```

```
1 Education Category =  
2 SWITCH('Customer Lookup'[EducationLevel],  
3 "High School", "High School",  
4 "Partial High School", "High School",  
5 "Bachelors", "Undergrad",  
6 "Partial College", "Undergrad",  
7 "Graduate Degree", "Graduate")
```



TEXT FUNCTIONS

LEN

Returns the number of characters in a string

=**LEN**(Text)

Note: Use the & operator as a shortcut, or to combine more than two strings

CONCATENATE

Joins two text strings into one

=**CONCATENATE**(Text1, Text2)

UPPER/LOWER

Converts a string to upper or lower case

=**UPPER/LOWER** (Text)

LEFT/RIGHT/MID

Returns a number of characters from the start/middle/end of a text string

=**LEFT/RIGHT**(Text, [NumChars])
=**MID**(Text, StartPosition, NumChars)

SUBSTITUTE

Replaces an instance of existing text with new text in a string

=**SUBSTITUTE**(Text, OldText, NewText, [InstanceNumber])

SEARCH

Returns the position where a specified string or character is found, reading left to right

=**SEARCH**(FindText, WithinText, [StartPosition], [NotFoundValue])



ASSIGNMENT: TEXT





NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Couple random requests**

Good morning!

Hoping you can help with a couple quick updates to the model:

- 1) Ethan wants to make the month abbreviations ALL CAPS to make them more readable in our reports.
- 2) The product team asked us to break out the SKU category into its own field, which we can define as any characters before the first hyphen (“-”) in the ProductSKU column.

Thanks, reach out with any questions!

Reply Forward

Key Objectives

1. Update the **Month Short** column in the Calendar Lookup table to extract and capitalize the first 3 characters of the month name
2. Create a new column in the Product Lookup table named **SKU Category**, to return any number of characters before the first hyphen in the ProductSKU column

*Copyright Maven Analytics, LLC



SOLUTION: TEXT

  NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)
Subject: **Couple random requests**

Good morning!

Hoping you can help with a couple quick updates to the model:

- 1) Ethan wants to make the month abbreviations ALL CAPS to make them more readable in our reports.
- 2) The product team asked us to break out the SKU category into its own field, which we can define as any characters before the first hyphen (“-”) in the ProductSKU column.

Thanks, reach out with any questions!

Reply Forward

Solution Preview

```
1 Month Short =  
2 UPPER(  
3   LEFT(  
4     'Calendar Lookup'[Month Name],  
5     3  
6   )  
7 )
```

```
1 SKU Category =  
2 LEFT(  
3   'Product Lookup'[Product SKU],  
4   SEARCH(  
5     "-",  
6     'Product Lookup'[Product SKU]  
7   )  
8   -1  
9 )
```



BASIC DATE & TIME FUNCTIONS

TODAY/NOW

Returns the current date or exact time

=**TODAY/NOW()**

DAY/MONTH/YEAR

Returns the day of the month (1-31), month of the year (1-12), or year of a given date

=**DAY/MONTH/YEAR(Date)**

HOUR/MINUTE/SECOND

Returns the hour (0-23), minute (0-59), or second (0-59) of a given datetime value

=**HOUR/MINUTE/SECOND(Datetime)**

WEEKDAY/WEEKNUM

Returns a weekday number from 1 (Sunday) to 7 (Saturday), or the week # of the year

=**WEEKDAY/WEEKNUM(Date, [ReturnType])**

EOMONTH

Returns the date of the last day of the month, +/- a specified number of months

=**EOMONTH(StartDate, Months)**

DATEDIFF

Returns the difference between two dates, based on a given interval (day, hour, year, etc.)

=**DATEDIFF(Date1, Date2, Interval)**



ASSIGNMENT: DATE & TIME



NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Customer birth years**

Hey there, super easy one for you.

The customer segmentation project got me wondering if there are any interesting patterns or insights based on customer age.

Could you please add a field in our customer table to extract only the year from the birthdate field?

Thanks!
-Dianne

Reply **Forward**

Key Objectives

1. Create a new column in the Customer Lookup table named **Birth Year**, to extract only the year from the BirthDate column



SOLUTION: DATE & TIME

  NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)
Subject: **Customer birth years**

Hey there, super easy one for you.

The customer segmentation project got me wondering if there are any interesting patterns or insights based on customer age.

Could you please add a field in our customer table to extract only the year from the birthdate field?

Thanks!
-Dianne

Reply Forward

Solution Preview

```
1 Birth Year =  
2 YEAR(  
3     'Customer Lookup'[BirthDate]  
4 )
```



RELATED

RELATED()

Returns related values in each row of a table based on relationships with other tables

=RELATED(Column**Name**)

The **column** from a related table containing the values you want to retrieve

Examples:

- 'Product Lookup'[Product Name]
- 'Territory Lookup'[Country]



HEY THIS IS IMPORTANT!

RELATED works like a **VLOOKUP** function in Excel – it uses the relationship between tables (*defined by primary and foreign keys*) to pull values from one table into a new column of another.

Since this function requires row context, it can only be used as a **calculated column** or as part of an **iterator function** that cycles through all rows in a table (*FILTER, SUMX, MAXX, etc.*)



PRO TIP:

Instead of using RELATED to create extra columns (which increases file size), **nest it within measures like FILTER or SUMX**



CALCULATE

CALCULATE()

Evaluates an expression in a context that is modified by filters

=CALCULATE(Expression, [Filter1], [Filter2],...)

Name of an **existing measure** or a **DAX formula** for a valid measure

Examples:

- *[Total Orders]*
- *SUM('Returns Data'[Return Quantity])*

A Boolean (True/False) expression or a table expression that defines a filter

Note: these require fixed values or aggregation functions that return a scalar value (you cannot create filters based on measures)

Examples:

- *'Territory Lookup'[Country] = "USA"*
- *Calendar[Year] <> MAX(Calendar[Year])*



PRO TIP:

Think of CALCULATE as a **filter modifier**; it allows you to overrule existing report filters and “force” new filter context



EXAMPLE: CALCULATE

X ✓ 1 Red Sales = CALCULATE([Quantity Sold], 'Product Lookup'[Product Color] = "Red")

Here we've defined a new measure named **Red Sales**, which evaluates the **Quantity Sold** measure under a filter context where the product color is "**Red**"

| Product Color | Quantity Sold | Red Sales |
|---------------|---------------|--------------|
| Black | 10,590 | 4,011 |
| Multi | 5,756 | 4,011 |
| Red | 4,011 | 4,011 |
| Silver | 3,257 | 4,011 |
| Total | 23,614 | 4,011 |

Note how we see the **the same repeated values** for each product color, and even the total!



HEY THIS IS IMPORTANT!

The **CALCULATE** function modifies and overrules any competing filter context!

In this matrix, the "Black" row has competing filter context: Product Color = **Black** (from the row label) and Product Color= "**Red**" (from the CALCULATE function)

Both can't be true at the same time, so the "**Red**" filter from CALCULATE takes priority



EXAMPLE: CALCULATE

CALCULATE

Filters are modified by CALCULATE

[Product Color] = "Red"

If the measure being evaluated contains a **CALCULATE** function,
filter context is *overwritten* between **Step 1 & Step 2**

STEP 1

Filter context is detected
& applied

| Product Color | Quantity Sold | Red Sales |
|---------------|---------------|-----------|
| Black | 10,590 | 4,011 |
| Red | 4,011 | 4,011 |
| Silver | 3,257 | 4,011 |

'Product Lookup'[Product Color] = "Black"

| Model Name | Product Description | Product Color |
|---------------|---|---------------|
| Mountain-100 | Top-of-the-line competition mountain bike. Performance-enhancing options include the innovative Ht Frame, in Black. | Black |
| Mountain-150 | Top-of-the-line competition mountain bike. Performance-enhancing options include the innovative Ht Frame, in Black. | Black |
| Mountain-200 | Top-of-the-line competition mountain bike. Performance-enhancing options include the innovative Ht Frame, in Black. | Black |
| Mountain-250 | Top-of-the-line competition mountain bike. Performance-enhancing options include the innovative Ht Frame, in Black. | Black |
| Road-100 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-150 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-200 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-250 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-300 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-350 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-400 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-450 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Road-500 | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Ht Road Frame | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Ht Road Frame | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Ht Road Frame | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Ht Road Frame | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |



STEP 2

Filters flow "downstream"
to related tables

| Model Name | Product Description | Product Color |
|---------------|---|---------------|
| Mountain-100 | This bike is ridden by race winners. Developed with the adventure Works Cycles professional race team, it has a Red aluminum alloy frame provides a light, stiff ride, whether you are racing in the velodrome or on a demanding c... Red | Red |
| Mountain-150 | This bike is ridden by race winners. Developed with the adventure Works Cycles professional race team, it has a Red aluminum alloy frame provides a light, stiff ride, whether you are racing in the velodrome or on a demanding c... Red | Red |
| Mountain-200 | This bike is ridden by race winners. Developed with the adventure Works Cycles professional race team, it has a Red aluminum alloy frame provides a light, stiff ride, whether you are racing in the velodrome or on a demanding c... Red | Red |
| Mountain-250 | This bike is ridden by race winners. Developed with the adventure Works Cycles professional race team, it has a Red aluminum alloy frame provides a light, stiff ride, whether you are racing in the velodrome or on a demanding c... Red | Red |
| Mountain-300 | This bike is ridden by race winners. Developed with the adventure Works Cycles professional race team, it has a Red aluminum alloy frame provides a light, stiff ride, whether you are racing in the velodrome or on a demanding c... Red | Red |
| Road-100 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-150 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-200 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-250 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-300 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-350 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-400 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-450 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Road-500 | A true multi-sport bike that often兼顾了操控和 a 高强度设计. Aerodynamic design lets you in Red | Red |
| Ht Road Frame | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Ht Road Frame | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |
| Ht Road Frame | Our lightest and best quality aluminum frame made from the newest alloy; it is welded and heat-treated for one. Red | Red |



STEP 3

Measure evaluates against the
filtered table

| Order Date | Stock Date | * Order Number | Product Key | Return Quantity |
|----------------------------|-------------------------|----------------|-------------|-----------------|
| Sunday, July 5, 2020 | Wednesday, July 8, 2020 | 5046718 | 360 | 1 |
| Tuesday, July 7, 2020 | Wednesday, July 8, 2020 | 5046736 | 360 | 1 |
| Wednesday, July 8, 2020 | Wednesday, July 8, 2020 | 5046778 | 360 | 1 |
| Thursday, July 9, 2020 | Wednesday, July 8, 2020 | 5046808 | 360 | 1 |
| Friday, July 10, 2020 | Wednesday, July 8, 2020 | 5046826 | 360 | 1 |
| Saturday, July 11, 2020 | Wednesday, July 8, 2020 | 5046975 | 360 | 1 |
| Sunday, July 12, 2020 | Wednesday, July 8, 2020 | 5047098 | 360 | 1 |
| Monday, July 13, 2020 | Wednesday, July 8, 2020 | 5047149 | 360 | 1 |
| Tuesday, July 14, 2020 | Wednesday, July 8, 2020 | 5047212 | 360 | 1 |
| Wednesday, July 15, 2020 | Wednesday, July 8, 2020 | 5047302 | 360 | 1 |
| Thursday, July 16, 2020 | Wednesday, July 8, 2020 | 5047382 | 360 | 1 |
| Friday, July 17, 2020 | Wednesday, July 8, 2020 | 5047446 | 360 | 1 |
| Saturday, July 18, 2020 | Wednesday, July 8, 2020 | 5047508 | 360 | 1 |
| Sunday, July 19, 2020 | Wednesday, July 8, 2020 | 5047578 | 360 | 1 |
| Monday, July 20, 2020 | Wednesday, July 8, 2020 | 5047638 | 360 | 1 |
| Tuesday, July 21, 2020 | Wednesday, July 8, 2020 | 5047708 | 360 | 1 |
| Wednesday, July 22, 2020 | Wednesday, July 8, 2020 | 5047778 | 360 | 1 |
| Thursday, July 23, 2020 | Wednesday, July 8, 2020 | 5047848 | 360 | 1 |
| Friday, July 24, 2020 | Wednesday, July 8, 2020 | 5047918 | 360 | 1 |
| Saturday, July 25, 2020 | Wednesday, July 8, 2020 | 5047988 | 360 | 1 |
| Sunday, July 26, 2020 | Wednesday, July 8, 2020 | 5048058 | 360 | 1 |
| Monday, July 27, 2020 | Wednesday, July 8, 2020 | 5048128 | 360 | 1 |
| Tuesday, July 28, 2020 | Wednesday, July 8, 2020 | 5048198 | 360 | 1 |
| Wednesday, July 29, 2020 | Wednesday, July 8, 2020 | 5048268 | 360 | 1 |
| Thursday, July 30, 2020 | Wednesday, July 8, 2020 | 5048338 | 360 | 1 |
| Friday, July 31, 2020 | Wednesday, July 8, 2020 | 5048408 | 360 | 1 |
| Saturday, August 1, 2020 | Wednesday, July 8, 2020 | 5048478 | 360 | 1 |
| Sunday, August 2, 2020 | Wednesday, July 8, 2020 | 5048548 | 360 | 1 |
| Monday, August 3, 2020 | Wednesday, July 8, 2020 | 5048618 | 360 | 1 |
| Tuesday, August 4, 2020 | Wednesday, July 8, 2020 | 5048688 | 360 | 1 |
| Wednesday, August 5, 2020 | Wednesday, July 8, 2020 | 5048758 | 360 | 1 |
| Thursday, August 6, 2020 | Wednesday, July 8, 2020 | 5048828 | 360 | 1 |
| Friday, August 7, 2020 | Wednesday, July 8, 2020 | 5048898 | 360 | 1 |
| Saturday, August 8, 2020 | Wednesday, July 8, 2020 | 5048968 | 360 | 1 |
| Sunday, August 9, 2020 | Wednesday, July 8, 2020 | 5049038 | 360 | 1 |
| Monday, August 10, 2020 | Wednesday, July 8, 2020 | 5049108 | 360 | 1 |
| Tuesday, August 11, 2020 | Wednesday, July 8, 2020 | 5049178 | 360 | 1 |
| Wednesday, August 12, 2020 | Wednesday, July 8, 2020 | 5049248 | 360 | 1 |
| Thursday, August 13, 2020 | Wednesday, July 8, 2020 | 5049318 | 360 | 1 |



Sum of the Order **Quantity** column in the

Sales Data table, filtered to rows where

the product color is "Red"

= 4,011



DAX MEASURE TOTALS

Measure totals may seem incorrect or inconsistent depending on how they are calculated, because they **don't simply add up the visible values in the report**



*Total Returns look right, but
shouldn't Total Orders be 37,888??*
-Anonymous confused man

| Category Name | Total Returns | Total Orders |
|---------------|---------------|---------------|
| Accessories | 1,115 | 16,983 |
| Bikes | 427 | 13,929 |
| Clothing | 267 | 6,976 |
| Total | 1,809 | 25,164 |

PRO TIP:

Understand EXACTLY how your measures calculate and **what they are designed to measure**

```
1 Total Orders =  
2 DISTINCTCOUNT(  
3 | 'Sales Data'[Order Number]  
4 )
```

[Total Orders] counts **distinct orders** in the Sales Data table

| Order Date | Stock Date | Order Number | Product Key |
|-------------------------|--------------------------|--------------|-------------|
| Thursday, June 30, 2022 | Thursday, April 07, 2022 | SO74140 | 568 |
| Thursday, June 30, 2022 | Friday, March 04, 2022 | SO74140 | 477 |
| Thursday, June 30, 2022 | Monday, May 30, 2022 | SO74140 | 223 |
| Thursday, June 30, 2022 | Friday, April 29, 2022 | SO74141 | 604 |
| Thursday, June 30, 2022 | Wednesday, May 04, 2022 | SO74141 | 471 |
| Thursday, June 30, 2022 | Monday, May 30, 2022 | SO74142 | 383 |
| Thursday, June 30, 2022 | Friday, March 18, 2022 | SO74142 | 490 |
| Thursday, June 30, 2022 | Tuesday, March 15, 2022 | SO74143 | 479 |
| Thursday, June 30, 2022 | Friday, April 08, 2022 | SO74143 | 606 |
| Thursday, June 30, 2022 | Tuesday, March 22, 2022 | SO74143 | 477 |
| Thursday, June 30, 2022 | Thursday, June 02, 2022 | SO74143 | 462 |
| Thursday, June 30, 2022 | Monday, April 25, 2022 | SO74144 | 574 |
| Thursday, June 30, 2022 | Sunday, April 24, 2022 | SO74144 | 220 |
| Thursday, June 30, 2022 | Monday, March 14, 2022 | SO74145 | 561 |
| Thursday, June 30, 2022 | Tuesday, June 14, 2022 | SO74146 | 584 |
| Thursday, June 30, 2022 | Friday, March 18, 2022 | SO74147 | 605 |
| Thursday, June 30, 2022 | Sunday, May 29, 2022 | SO74147 | 538 |
| Thursday, June 30, 2022 | Thursday, March 24, 2022 | SO74147 | 490 |

Table: Sales Data (56,046 rows) Column: Order Number (25,164 distinct values)

Order **SO74144** included **two products**: a bike and a helmet.

That counts as **1** distinct order for the Total and **1** distinct order for BOTH **Accessories & Bikes**

With no filter context, there are **25,164** total distinct orders



ASSIGNMENT: CALCULATE



NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **URGENT: Bike returns**

Hey there,

Apparently George (our Product VP) has been speaking with some of the store managers, and they've raised concerns about the number of bike returns they are seeing recently.

Can you please create a measure to calculate total returns for bikes specifically, and let me know what you see? Volume alone won't tell the full story, so let's calculate the return *rate* for bikes as well, and see how it's trending before responding to George.

Need this ASAP – thank you!

Reply **Forward**

Key Objectives

1. Create a new measure named **Bike Returns** to calculate the total quantity of bikes returned
2. Create a matrix to show **Bike Returns** (values) by **Start of Month** (rows). What do you notice about the volume of bike returns over time?
3. Create a new measure named **Bike Sales** to calculate the total quantity of bikes sold, and add it to the matrix. What do you notice?
4. Create a new measure named **Bike Return Rate** using either CALCULATE or DIVIDE, and add it to the matrix
5. How would you respond to the Product VP's concerns about rising bike returns?



SOLUTION: CALCULATE

 NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **URGENT: Bike returns**

Hey there,

Apparently George (our Product VP) has been speaking with some of the store managers, and they've raised concerns about the number of bike returns they are seeing recently.

Can you please create a measure to calculate total returns for bikes specifically, and let me know what you see? Volume alone won't tell the full story, so let's calculate the return *rate* for bikes as well, and see how it's trending before responding to George.

Need this ASAP – thank you!

Reply Forward

Solution Preview

```
1 Bike Returns =  
2 CALCULATE(  
3     [Total Returns],  
4     'Product Categories Lookup'[Category Name] = "Bikes"  
5 )
```

```
1 Bike Sales =  
2 CALCULATE(  
3     [Quantity Sold],  
4     'Product Categories Lookup'[Category Name] = "Bikes"  
5 )
```

```
1 Bike Return Rate =  
2 CALCULATE(  
3     [Return Rate],  
4     'Product Categories Lookup'[Category Name] = "Bikes"  
5 )
```

(Solution continued on next slide)



SOLUTION: CALCULATE

  NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **URGENT: Bike returns**

Hey there,

Apparently George (our Product VP) has been speaking with some of the store managers, and they've raised concerns about the number of bike returns they are seeing recently.

Can you please create a measure to calculate total returns for bikes specifically, and let me know what you see? Volume alone won't tell the full story, so let's calculate the return *rate* for bikes as well, and see how it's trending before responding to George.

Need this ASAP – thank you!

Reply Forward

Solution Preview

| | | | |
|--------------|------------|--------------|---------------|
| 6/1/2021 | 8 | 312 | 2.564% |
| 7/1/2021 | 12 | 506 | 2.372% |
| 8/1/2021 | 14 | 485 | 2.887% |
| 9/1/2021 | 22 | 575 | 3.826% |
| 10/1/2021 | 26 | 612 | 4.248% |
| 11/1/2021 | 25 | 688 | 3.634% |
| 12/1/2021 | 26 | 1038 | 2.505% |
| 1/1/2022 | 14 | 766 | 1.828% |
| 2/1/2022 | 22 | 806 | 2.730% |
| 3/1/2022 | 27 | 888 | 3.041% |
| 4/1/2022 | 38 | 956 | 3.975% |
| 5/1/2022 | 36 | 1116 | 3.226% |
| 6/1/2022 | 34 | 1157 | 2.939% |
| Total | 429 | 13929 | 3.080% |

The volume of bike returns has risen over time, but so has the number of bikes being sold.

When we look at the rate of returns as a percent of sales, we don't see a concerning trend.



ALL

ALL

Returns all rows in a table, or all values in a column, ignoring any filters that have been applied

=ALL(Table or Column, [Column2], [Column3],...)

The **table** or **column** that you want to clear filters on

Examples:

- *Transactions*
- *Products[Category]*

Additional columns that you want to clear filters on (optional)

- Cannot specify columns if your first parameter is a **table**
- All columns must include the **table name** and come from the **same table**

Examples:

- *'Customer Lookup'[City], 'Customer Lookup'[Country]*
- *Products[Product Name]*

PRO TIP:



Instead of adding filter context, **the ALL function removes it**. This is often used in “**% of Total**” calculations, when the denominator needs to remain fixed regardless of filter context.



ASSIGNMENT: CALCULATE & ALL



NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Return analysis follow-up**

Hey again,

Thanks for the quick turnaround on that bike return analysis – crisis averted!

That got me thinking about how we could start analyzing the return data in our reports. Could you please help me create two new measures, one to calculate ALL returns (regardless of filter context), and another that divides Total Returns by All Returns?

That should allow us to see the % of returns by different products and product categories.

Reply **Forward**

Key Objectives

1. Create a new measure named **All Returns** to calculate the total number of returns, regardless of filter context
2. Create a new measure named **% of All Returns** that divides Total Returns by All Returns
3. Create a matrix to show % of All Returns (values) by product Category Name (rows). Which category accounts for the largest percentage of returns? The smallest?



SOLUTION: CALCULATE & ALL

 NEW MESSAGE

From: Dianne A. Xu (Senior Analyst)

Subject: Return analysis follow-up

Hey again,

Thanks for the quick turnaround on that bike return analysis – crisis averted!

That got me thinking about how we could start analyzing the return data in our reports. Could you please help me create two new measures, one to calculate ALL returns (regardless of filter context), and another that divides Total Returns by All Returns?

That should allow us to see the % of returns by different products and product categories.

Reply Forward

Solution Preview

```
1 All Returns =  
2 CALCULATE(  
3     [Total Returns],  
4     ALL(  
5         'Returns Data'  
6     ))  
7 )
```

```
1 % of All Returns =  
2 DIVIDE(  
3     [Total Returns],  
4     [All Returns]  
5 )
```

| Category Name | % of All Returns |
|---------------|------------------|
| Bikes | 23.60% |
| Clothing | 14.76% |
| Accessories | 61.64% |
| Total | 100.00% |



FILTER

FILTER

Returns a table that represents a subset of another table or expression

=**FILTER**(Table, FilterExpression)

Table to be filtered

Examples:

- Territory Lookup
- Customer Lookup

A Boolean (True/False) filter expression to be evaluated for each row of the table

Examples:

- 'Territory Lookup'[Country] = "USA"
- Calendar[Year] = 1998
- Products[Price] > [Overall Avg Price]

HEY THIS IS IMPORTANT!

FILTER is used to add new filter context, and can handle **more complex filter expressions** than CALCULATE (by referencing measures, for example)

Since FILTER returns an entire table, it's often **nested within other functions**, like CALCULATE or SUMX



PRO TIP:

Since FILTER **iterates through each row in a table**, it can be slow and computationally expensive; only use FILTER if a simple CALCULATE function won't get the job done!





ITERATOR FUNCTIONS

Iterator (or “X”) **functions** allow you to loop through the same expression on each row of a table, then apply some sort of aggregation to the results (SUM, MAX, etc.)

=**SUMX**(Table, Expression)

Aggregation to apply to calculated rows*

Table in which the expression will be evaluated

Expression to be evaluated for each row of the given table

Examples:

- SUMX
- COUNTX
- AVERAGEX
- RANKX
- MAXX/MINX

Examples:

- Sales
- FILTER(Sales,
RELATED(Products[Category])="Clothing")

Examples:

- [Total Orders]
- Sales[Retail Price] * Sales[Quantity]

PRO TIP:



Imagine that iterator functions **add a temporary new column** to a table, calculate a value in each row based on the given expression, then aggregate the values within that temporary column (similar to **SUMPRODUCT** in Excel)

In this example we're looking at **SUMX, but other iterator functions follow a similar syntax*



ASSIGNMENT: ITERATORS



NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Profit calculation – HELP!**

Hey,
Ethan asked for a quick analysis of company profit over the past few years, but I'm struggling with the calculation.
We need a measure that multiplies order quantity by product cost, but I'd like to do it without adding redundant columns to our Sales table.
Could you take a stab at this please?
-Dianne

Reply **Forward**

Key Objectives

1. Create a new measure named **Total Cost** that multiplies the order quantities in the Sales Data table by the product cost in the Product Lookup table, then calculates the sum
2. Create a new measure named **Total Profit** (revenue minus cost)
3. Create a matrix to show Total Profit (values) by Year (rows). How much profit has AdventureWorks earned so far in 2022?



SOLUTION: ITERATORS

 NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Profit calculation – HELP!**

Hey,

Ethan asked for a quick analysis of company profit over the past few years, but I'm struggling with the calculation.

We need a measure that multiplies order quantity by product cost, but I'd like to do it without adding redundant columns to our Sales table.

Could you take a stab at this please?

-Dianne

Reply Forward

Solution Preview

```
1 Total Cost =  
2 SUMX(  
3     'Sales Data',  
4     'Sales Data'[Order Quantity]  
5     *  
6     RELATED(  
7         'Product Lookup'[Product Cost]  
8     )  
9 )
```

```
1 Total Profit =  
2 [Total Revenue] - [Total Cost]
```

| Year | Total Profit |
|--------------|---------------------|
| 2020 | \$2,601,606 |
| 2021 | \$3,967,023 |
| 2022 | \$3,888,952 |
| Total | \$10,457,581 |



TIME INTELLIGENCE

Time Intelligence patterns are used to calculate common date-based comparisons

Performance
To-Date

=**CALCULATE**(Measure, **DATESYTD**(Calendar[Date]))

→ Use **DATESYTD** for Years, **DATESQTD** for Quarters, **DATESMTD** for Months

Previous
Period

=**CALCULATE**(Measure, **DATEADD**(Calendar[Date], -1, **MONTH**))

Select an interval (**DAY**, **MONTH**, **QUARTER**, or **YEAR**) and the
of intervals to compare (e.g. previous month, rolling 10-day)

Running
Total

=**CALCULATE**(Measure,

DATESINPERIOD(Calendar[Date], **MAX**(Calendar[Date]), -10, **DAY**))

{ }



PRO TIP:

To calculate a **moving average**, use the running total calculation above and **divide by the number of intervals**

ASSIGNMENT: TIME INTELLIGENCE







NEW MESSAGE

From: Dianne A. Xu (Senior Analyst)

Subject: Time Intelligence Measures

Hey there, need a big favor!
The leadership team has been asking a lot of questions about month-over-month and year-over-year comparisons, and I've been pulling the numbers pretty manually.
Could you please add the following list of measures, to make these metrics easier to track and share with stakeholders?
Thank you!
-Dianne

[Reply](#) [Forward](#)

Key Objectives

Add the following measures to the model:

1. **Previous Month Returns**
2. **Previous Month Orders**
3. **Previous Month Profit**
4. **Order Target** (10% increase over previous month)
5. **Profit Target** (10% increase over previous month)
6. **90-day Rolling Profit**



SOLUTION: TIME INTELLIGENCE

 NEW MESSAGE

From: **Dianne A. Xu** (Senior Analyst)

Subject: **Time Intelligence Measures**

Hey there, need a big favor!

The leadership team has been asking a lot of questions about month-over-month and year-over-year comparisons, and I've been pulling the numbers pretty manually.

Could you please add the following list of measures, to make these metrics easier to track and share with stakeholders?

Thank you!

-Dianne

Reply **Forward**

Solution Preview

```
1 Previous Month Orders =  
2 CALCULATE(  
3     [Total Orders],  
4     DATEADD(  
5         'Calendar Lookup'[Date],  
6         -1,  
7         MONTH  
8     )  
9 )
```

```
1 90-day Rolling Profit =  
2 CALCULATE(  
3     [Total Profit],  
4     DATESINPERIOD(  
5         'Calendar Lookup'[Date],  
6         LASTDATE(  
7             'Calendar Lookup'[Date]  
8         ),  
9         -90,  
10        DAY  
11    )  
12 )
```

```
1 Order Target =  
2 [Previous Month Orders] * 1.1
```



DAX BEST PRACTICES



Know when to use calculated columns vs. measures

- *Use calculated columns for filtering, and measures for aggregating values*



Use explicit measures, even for simple calculations

- *Explicit measures can be referenced anywhere, and nested within other measures*



Use fully-qualified column references in measures

- *This makes your DAX more readable, and differentiates column references from measure references*



Move column calculations “upstream” when possible

- *Adding calculated columns at the source or in Power Query improves report speed and efficiency*



Minimize the use of “expensive” iterator functions

- *Use iterators with caution, especially if you are working with large tables or complex models*

VISUALIZING DATA

VISUALIZING DATA



In this section we'll **build dynamic interactive reports**, introduce visualization best practices, and explore features like bookmarks, drillthrough filters, parameters, tooltips, and more

TOPICS WE'LL COVER:

Data Viz Best Practices

Formatting & Filtering

Bookmarks

Report Interactions

User Roles

Parameters

Custom Tooltips

Mobile Layouts

GOALS FOR THIS SECTION:

- Review frameworks and best practices for visualizing data and designing effective reports and dashboards
- Explore tools and techniques for inserting, formatting and filtering visuals in the Power BI Report view
- Add interactivity using tools like bookmarks, slicer panels, parameters, tooltips, and report navigation
- Learn how to configure row-level security with user roles
- Optimize reports for mobile viewing using custom layouts



THREE KEY QUESTIONS

1

What **TYPE OF DATA** are you working with?

- Geospatial? Time-series? Hierarchical? Financial?
-

2

What do you want to **COMMUNICATE**?

- Comparison? Composition? Relationship? Distribution?
-

3

Who is the **END USER** and what do they need?

- Analyst? Manager? Executive? General public?



THREE KEY QUESTIONS

1 What **TYPE OF DATA** are you working with?

 Time-series

 Financial

 Geospatial

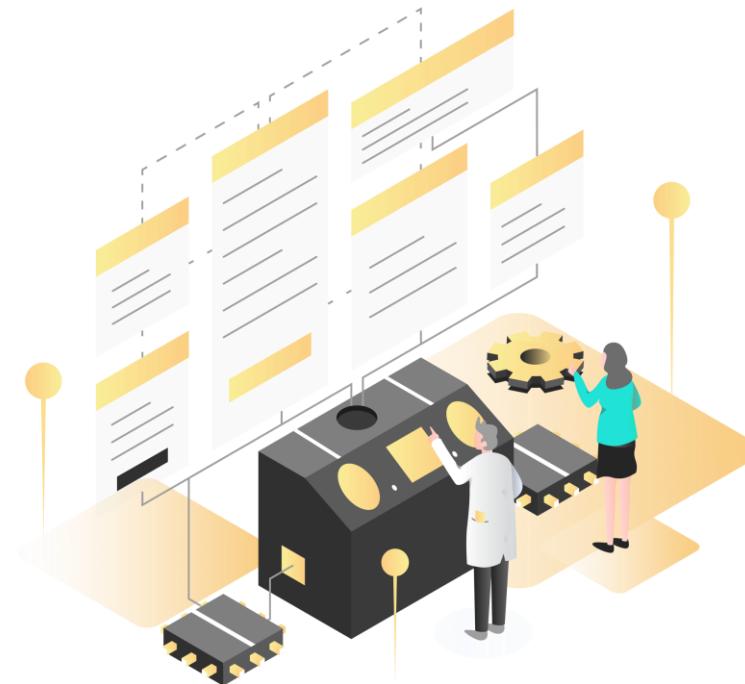
 Textual

 Categorical

 Funnel

 Hierarchical

 Survey



The type of data you're working with often determines **which type of visual will best represent it**; for example, using maps to represent geospatial data, line charts for time-series data, or tree maps for hierarchical data

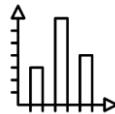


THREE KEY QUESTIONS

2

What do you want to **COMMUNICATE?**

COMPARISON



Used to **compare values over time or across categories**

Common visuals:

- Column/Bar Chart
- Clustered Column/Bar
- Data Table/Heat Map
- Radar Chart
- Line Chart (*time series*)
- Area Chart (*time series*)

COMPOSITION



Used to **break down the component parts of a whole**

Common visuals:

- Stacked Bar/Column Chart
- Pie/Donut Chart
- Stacked Area (*time series*)
- Waterfall Chart (*gains/losses*)
- Funnel Chart (*stages*)
- Tree Map/sunburst (*hierarchies*)

DISTRIBUTION

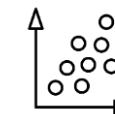


Used to **show the frequency of values within a series**

Common visuals:

- Histogram
- Density Plot
- Box & Whisker
- Scatter Plot
- Data Table/Heat Map
- Map/Choropleth (*geospatial*)

RELATIONSHIP



Used to **show correlation between multiple variables**

Common visuals:

- Scatter Plot
- Bubble Chart
- Data Table/Heat Map
- Correlation Matrix

Keep it simple! While there are *hundreds* of charts to choose from, basic options like bars and columns, line charts, histograms and scatterplots often tell the simplest and clearest story



THREE KEY QUESTIONS

3 Who is the **END USER** and what do they need?

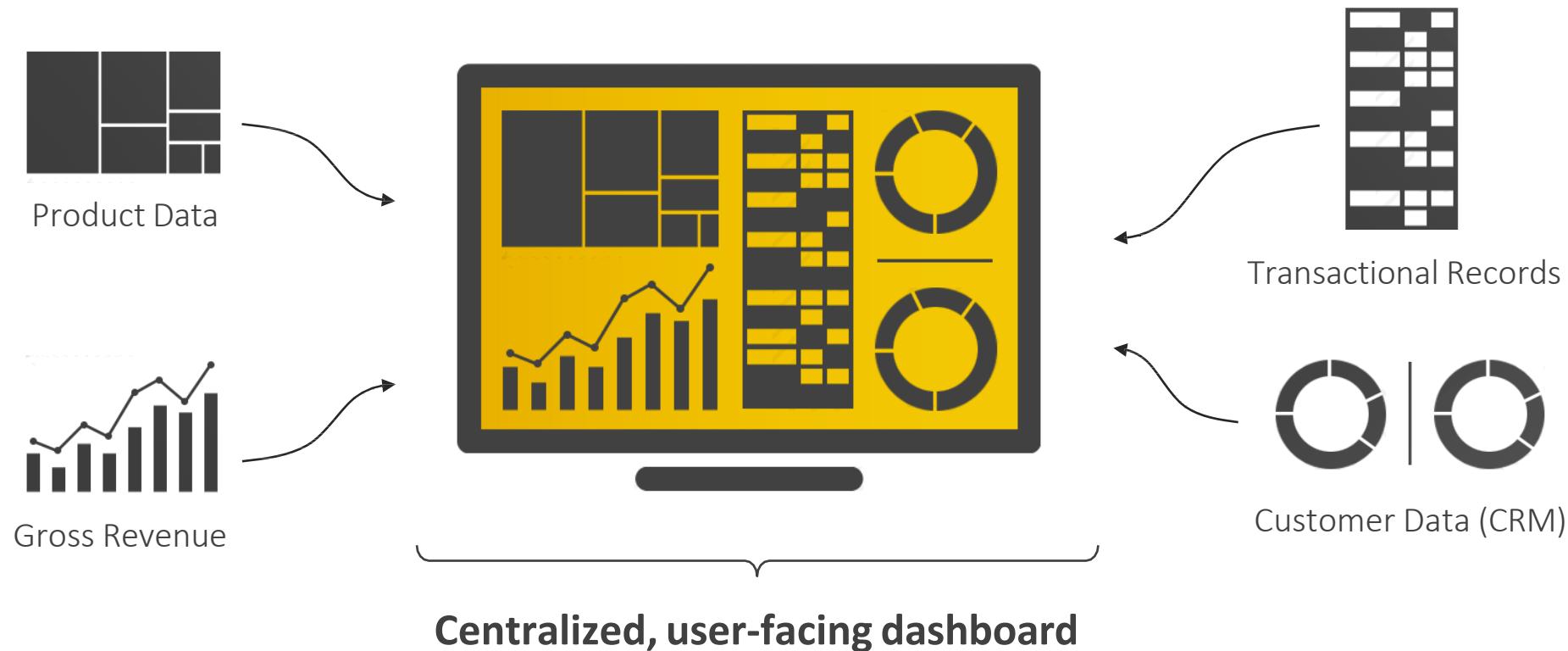


How you visualize and present your data is a function of **who will be consuming it**; a fellow analyst may want to see granular details, while managers and executives often prefer topline KPIs and clear, data-driven insights



ANALYTICS DASHBOARDS

Dashboards are analytics tools designed to consolidate data from multiple sources, track key metrics at a glance, and facilitate data-driven storytelling and decision making



DASHBOARD DESIGN FRAMEWORK



1

Define the purpose

2

Choose the right metrics

3

Present the data effectively

4

Eliminate clutter & noise

5

Use layout to focus attention

6

Tell a clear story

A well-designed dashboard should **serve a distinct purpose for a distinct audience**, use **clear and effective metrics and visuals**, and **provide a simple, intuitive user experience**.



Key questions to consider:

- Who are the **end-users** of your dashboard?
- What are their **key business goals** and objectives?
- What are the **most important questions** they need answers to?
- How can I present information **as clearly as possible**?

“Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away”

Antoine de Saint-Exupery



THE REPORT VIEW

Report View

Report Canvas

Insert Menu (Add pages, visuals, buttons, shapes, images, etc.)

Report Pages (each tab is a blank report canvas)

Panes (Data, Format, Bookmarks, Selection)

Filter Pane (Page-level, report-level, visual-level filters)

View Options (Zoom, fit to page)

The screenshot shows the Microsoft Power BI Report View interface. At the top is the ribbon with the 'Insert' tab selected. Below the ribbon is the 'Report View' pane on the left, containing navigation icons and a 'Report Canvas' area displaying various visualizations like a line chart, bar charts, and summary cards. To the right of the canvas is the 'Panes' pane, which includes the 'Filters' pane for page-level filters and the 'Filter Pane' for more granular filters. At the bottom is the 'Report Pages' tab bar and the 'View Options' pane for zooming.

Report View

Report Canvas

Insert Menu (Add pages, visuals, buttons, shapes, images, etc.)

Report Pages (each tab is a blank report canvas)

Panes (Data, Format, Bookmarks, Selection)

Filter Pane (Page-level, report-level, visual-level filters)

View Options (Zoom, fit to page)



ASSIGNMENT: CARDS



NEW MESSAGE

From: **Victor Ignatius Zabel** (BI Analyst)

Subject: **Let's get visual!**

Hey there!

We've kicked off the visualization work for our Power BI dashboard, and I'm hoping you can help.

For now I'd love for you to focus on building out the **Customer Detail** report. Can you start by adding some KPIs to show total customers and revenue per customer?

-Vic

Reply **Forward**

Key Objectives

1. Insert a **card** in the **Customer Detail** report page to show **Total Customers**, and rename the field "UNIQUE CUSTOMERS"
2. Add a background shape and match the formatting of the cards in the **Exec Dashboard** tab
3. Copy and paste to create a second card showing **Average Revenue per Customer**, and rename the field "REVENUE PER CUSTOMER"



SOLUTION: CARDS

  NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**
Subject: **Let's get visual!**

Hey there!
We've kicked off the visualization work for our Power BI dashboard, and I'm hoping you can help.
For now I'd love for you to focus on building out the **Customer Detail** report. Can you start by adding some KPIs to show total customers and revenue per customer?

-Vic

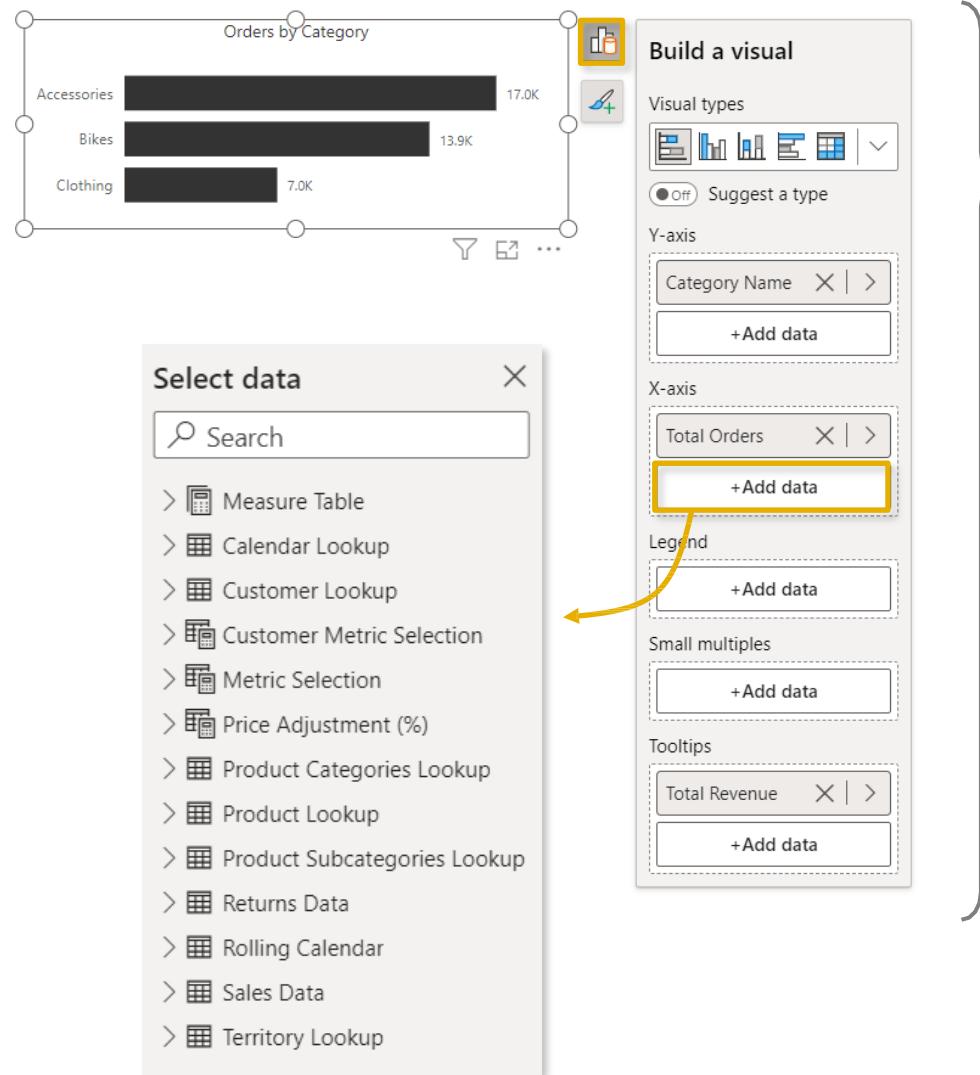
Reply Forward

Solution Preview





BUILDING & FORMATTING CHARTS

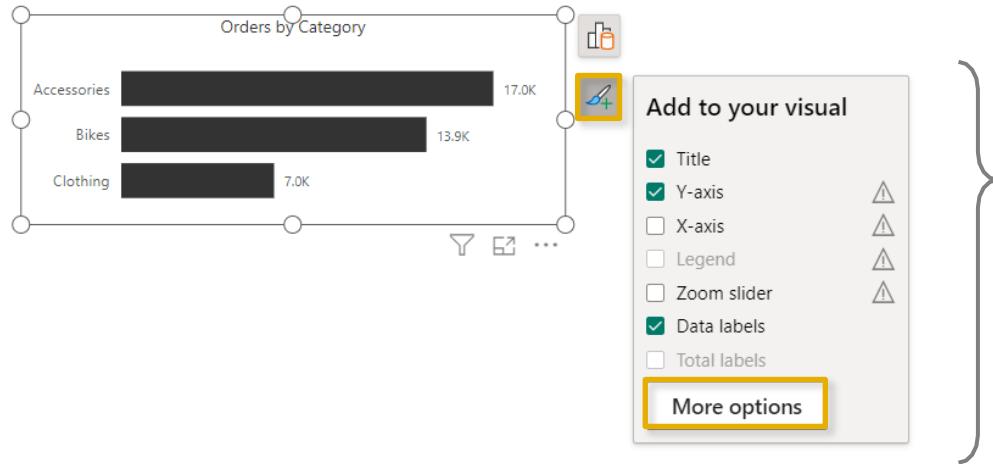


The **Build** menu allows you to change the visual type, auto-suggest visuals, and add data to customize chart components (*x-axis, y-axis, legend, tooltips, etc.*)

- This is a **contextual menu**, so you will only see options which are relevant to the selected visual
- You can build visuals by either inserting a specific chart type and adding data, or by dragging a field from the Data pane onto the canvas

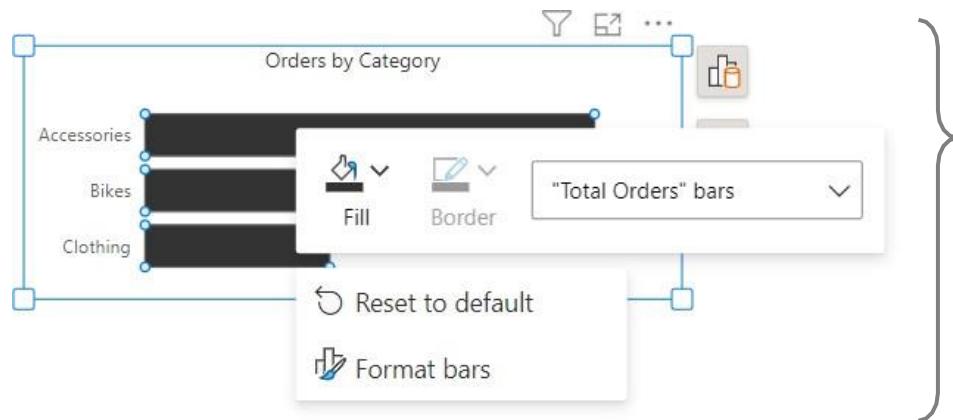


BUILDING & FORMATTING CHARTS



The **Format** menu allows you to quickly add common chart elements (*title, axis labels, data labels, legends, etc.*) and access additional options and properties in the Format pane

- This is a **contextual menu**, so you will only see options which are relevant to the selected visual



Enable **on-object formatting** by double-clicking the chart object (*or right-click > format*), which allows you to select and edit individual chart elements

- On-object formatting is only available for certain visuals (bar, column, line, area, combo & scatter)



ASSIGNMENT: LINE CHARTS



NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Customer count by week**

Nice work on those cards!

Next up let's add a weekly line chart to show how our customer base is trending over time.

Please add a zoom bar to make it interactive, and format the tooltips to match the line chart in the Exec Dashboard.

Thanks!

-Vic

Reply **Forward**

Key Objectives

1. Add a **line chart** to the **Customer Detail** report showing **Total Customers** by week
2. Add a **trend line** and a **zoom slider** to the x-axis
3. Enable **tooltips**, and format to match line chart in the **Exec Dashboard** tab



SOLUTION: LINE CHARTS

  NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**
Subject: **Customer count by week**

Nice work on those cards!

Next up let's add a weekly line chart to show how our customer base is trending over time.

Please add a zoom bar to make it interactive, and format the tooltips to match the line chart in the Exec Dashboard.

Thanks!

-Vic

[Reply](#) [Forward](#)

Solution Preview





FILTERING OPTIONS

The Filters pane in Power BI displays three levels of filters:

- Filters on this visual**: Contains a search bar and a single filter: "Sum of Order Quantity is (All)".
- Filters on this page**: Contains a search bar and a button "Add data fields here".
- Filters on all pages**: Contains a search bar and a button "Add data fields here".

There are **3 types of filters** accessible from the **Filters** pane*:

- 1. Visual-level** filters apply to specific visuals
- 2. Page-level** filters apply to all visuals on the report page
- 3. Report-level** filters apply to all visuals across all report pages

**Drillthrough filters can be configured in the page formatting pane – more on that later!*

Four examples of filter configurations:

- Basic Options**: Shows a list of categories: Select all, Accessories (1), Bikes (1), Clothing (1), Components (1).
- Top N Options**: Set to "Top N" with value "2".
 - Show items: Top 2
 - By value: Total Orders
- Advanced (Values)**: Shows logical operators:
 - Show items when the value is greater than
 - is less than
 - is less than or equal to
 - is greater than** (selected)
 - is greater than or equal to
 - is
 - is not
- Advanced (Text)**: Shows text-based operators:
 - Show items when the value contains
 - contains (selected)
 - does not contain
 - starts with
 - does not start with
 - is

Filters can be configured using basic **selections**, **logical operators**, or **Top N** conditions



ASSIGNMENT: DONUT CHARTS



NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Customer demographics**

Good morning!

Just got a note from Ethan to see if we can build some demographic info into the customer report.

Let's add a couple donut charts to show the composition of customers by income level and occupation. We'll want to limit to just a few segments (maybe 3?) and do some formatting to match the rest of the dashboard.

Thanks, you rock!

-Vic

Reply **Forward**

Key Objectives

1. Add a **donut chart** to the **Customer Detail** report showing **Total Orders** by **Income Level**
2. Add a **chart title**, turn off the **legend**, and update the **data labels** to show the category and value (font size 8, 1 decimal place)
3. Update the colors of the slices to match the screenshot in the solution preview
4. Add a **visual-level filter** to exclude customers with a “Very High” income level
5. Copy the chart to show **Total Orders** by **Occupation**, and add a **visual-level filter** to display the three occupations with the most orders (*bonus points if you use a Top N filter!*)



SOLUTION: DONUT CHARTS

  NEW MESSAGE

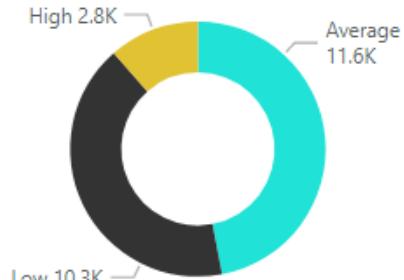
From: **Victor Ignatius Zabel (BI Analyst)**
Subject: **Customer demographics**

Good morning!
Just got a note from Ethan to see if we can build some demographic info into the customer report.
Let's add a couple donut charts to show the composition of customers by income level and occupation. We'll want to limit to just a few segments (maybe 3?) and do some formatting to match the rest of the dashboard.
Thanks, you rock!
-Vic

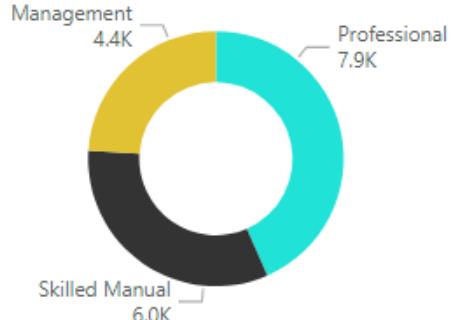
[Reply](#) [Forward](#)

Solution Preview

Orders by Income Level



Orders by Occupation





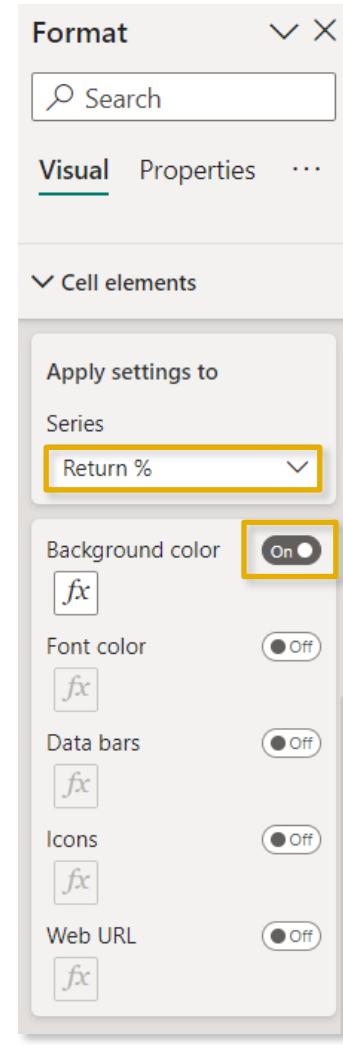
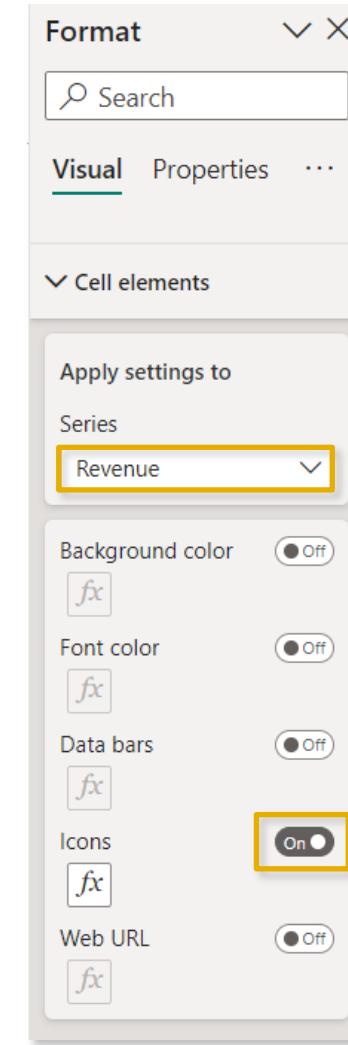
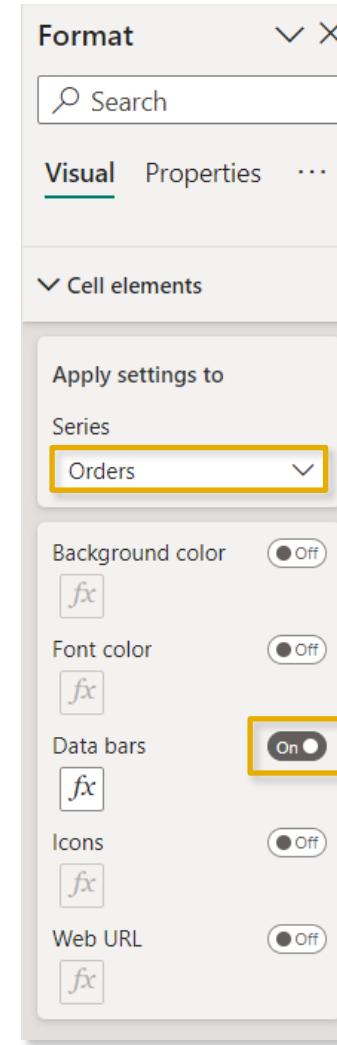
CONDITIONAL FORMATTING

| Top 10 Products | Orders | Revenue | Return % |
|-------------------------|--------|----------|----------|
| Water Bottle - 30 oz. | 3,983 | \$39,755 | 1.95% |
| Patch Kit/8 Patches | 2,952 | \$13,506 | 1.61% |
| Mountain Tire Tube | 2,846 | \$28,333 | 1.64% |
| Road Tire Tube | 2,173 | \$17,265 | 1.55% |
| Sport-100 Helmet, Red | 2,099 | \$73,444 | 3.33% |
| AWC Logo Cap | 2,062 | \$35,865 | 1.11% |
| Sport-100 Helmet, Blue | 1,995 | \$67,112 | 3.31% |
| Fender Set - Mountain | 1,975 | \$87,041 | 1.36% |
| Sport-100 Helmet, Black | 1,940 | \$65,262 | 2.68% |
| Mountain Bottle Cage | 1,896 | \$38,062 | 2.02% |



Conditional formatting allows you to dynamically format Table or Matrix visuals based on cell values

- Conditionally formatting options can be found in the **Format** pane, under **Cell elements**
- Options include background color, font color, data bars, icons, or Web URL





ASSIGNMENT: TABLES



NEW MESSAGE

From: **Victor Ignatius Zabel** (BI Analyst)

Subject: **Top customer table**

Hey there, this customer report is really coming together! Since the management team needs a way to identify high-value customers, let's add a table to our report showing customer keys, full names, orders, and revenue. Probably makes sense to add some conditional formatting and limit to the top 100 customers for now.

Thanks!

-Vic

Reply **Forward**

Key Objectives

1. Add a **table** to the **Customer Detail** report to show **Customer Key**, **Full Name**, **Total Orders** (as “Orders”) and **Total Revenue** (as “Revenue”)
2. Use conditional formatting to add light gray **data bars** to the orders column and a white > blue **color scale** to the revenue column
3. Add a **visual-level filter (Top N)** to show the 100 customers with the most orders, and add a **chart title** (“Top 100 Customers”)
4. **Sort** the table descending by orders



SOLUTION: TABLES

  NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Top customer table**

Hey there, this customer report is really coming together! Since the management team needs a way to identify high-value customers, let's add a table to our report showing customer keys, full names, orders, and revenue.

Probably makes sense to add some conditional formatting and limit to the top 100 customers for now.

Thanks!

-Vic

Reply Forward

Solution Preview

| Top 100 Customers | | | |
|-------------------|-----------------------|--------|-----------|
| Customer Key | Full Name | Orders | Revenue |
| 11091 | Mr. Dalton Perez | 26 | \$1,513 |
| 11223 | Mrs. Hailey Patterson | 26 | \$1,616 |
| 11300 | Mr. Fernando Barnes | 26 | \$1,839 |
| 11330 | Mr. Ryan Thompson | 26 | \$1,597 |
| 11331 | Mrs. Samantha Jenkins | 26 | \$1,740 |
| 11185 | Mrs. Ashley Henderson | 25 | \$1,717 |
| 11200 | Mr. Jason Griffin | 25 | \$1,614 |
| 11176 | Mr. Mason Roberts | 24 | \$1,526 |
| 11262 | Mrs. Jennifer Simmons | 24 | \$1,465 |
| 11277 | Mr. Charles Jackson | 24 | \$1,777 |
| 11287 | Mr. Henry Garcia | 24 | \$1,443 |
| 11566 | Ms. April Shan | 24 | \$1,424 |
| 11711 | Mr. Daniel Davis | 24 | \$1,404 |
| 11276 | Mrs. Nancy Chapman | 23 | \$1,111 |
| 11203 | Mr. Luis Diaz | 17 | \$1,002 |
| 11215 | Mrs. Ana Perry | 17 | \$1,336 |
| 11078 | Ms. Gina Martin | 16 | \$991 |
| Total | | 1,272 | \$615,328 |



ASSIGNMENT: TOP N TEXT CARDS



NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Top customers by revenue**

Hey,

Ethan is loving the customer report so far – great job!

He mentioned that he'd like to highlight top customers based on *revenue* as well, so I'm thinking we could add some text cards to show the top customer name, along with total revenue and the number of orders placed.

We'll be offering some coupons based on how much customers have spent in the past, so accuracy is critical here!

-Vic

Reply **Forward**

Key Objectives

1. Add a **card** to the **Customer Detail** report to show **Full Name**
2. Add a **visual-level filter (Top N)** to show the top customer (Full Name) in terms of **Total Revenue**
 - What do you notice when you filter the report for low income customers?
(Hint: check your value against the table)
 - How could you modify the Top N filter to correct this?
3. Copy and paste the card (x2) to show **Total Orders** and **Total Revenue** for the top customer
4. Add **text boxes** for titles and adjust formatting to match the solution preview



SOLUTION: TOP N TEXT CARDS

  NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**
Subject: **Top customers by revenue**

Hey,
Ethan is loving the customer report so far – great job!
He mentioned that he'd like to highlight top customers based on *revenue* as well, so I'm thinking we could add some text cards to show the top customer name, along with total revenue and the number of orders placed.
We'll be offering some coupons based on how much customers have spent in the past, so accuracy is critical here!
-Vic

[Reply](#) [Forward](#)

Solution Preview

Top Customer (by revenue):

Mr. Maurice Shan

Orders:

6

Revenue:

\$12.4K

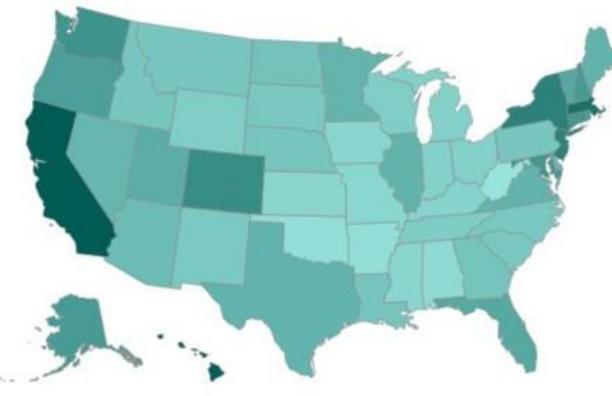


MAP VISUALS

Map



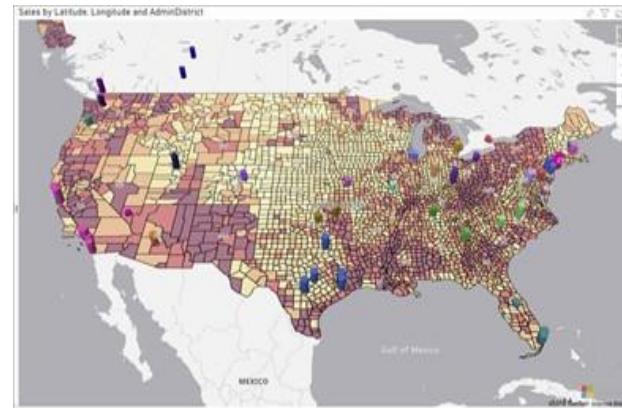
Shape map



Filled map



Azure map



Power BI includes several types of **map visuals** powered by Bing Maps

Tips for creating accurate maps:

1. Assign **categories** to geospatial fields
2. Add **multiple location** fields
3. Use **latitude/longitude** when possible



HEY THIS IS IMPORTANT!

An administrator may need to **enable maps in your tenant settings** in order to use them in Power BI Service



SLICERS

Slicers are visual filters which affect all other visuals on a report page (by default)

- Slicers can take many formats depending on the data type, including **lists**, **dropdowns**, **tiles**, **ranges**, and more

The diagram illustrates various slicer styles categorized into three groups:

- Categorical/Text options:** Vertical list, Tile, Dropdown.
- Date/Time options:** Between, Relative Date, After.
- Numeric Range options:** Single Value, Less than or equal to.

Categorical/Text options:

- IncomeLevel:
 - Average
 - High
 - Low
 - Very High
- Continent:
 - Europe
 - North America
 - Pacific

Date/Time options:

- Date range: 1/1/2020 - 6/30/2022
- Date range: Last 11 Months (3/18/2022 - 2/17/2023)
- Date range: 1/1/2020 - 6/30/2022

Numeric Range options:

- Price Adjustment (%): 0.20
- Price Adjustment (%): -1.00 to 0.50

PRO TIP:
Use **Apply/Clear All Slicers** buttons for more filtering control



ASSIGNMENT: SLICERS



NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Year slicer for customer report**

Hey there, quick request when you get a sec...
Could you please add a slicer to the customer report, so that users can filter the entire page by year?
No preference for which specific type of slicer you use, as long as managers can filter customers for a specific year or across multiple years.
Thanks!
-Vic

Reply **Forward**

Key Objectives

1. Add a **slicer** to filter the **Customer Detail** report page by **Year**
2. Add a **visual-level filter** to exclude blanks
3. Choose any **slicer style** that allows users to filter individual years or across multiple years



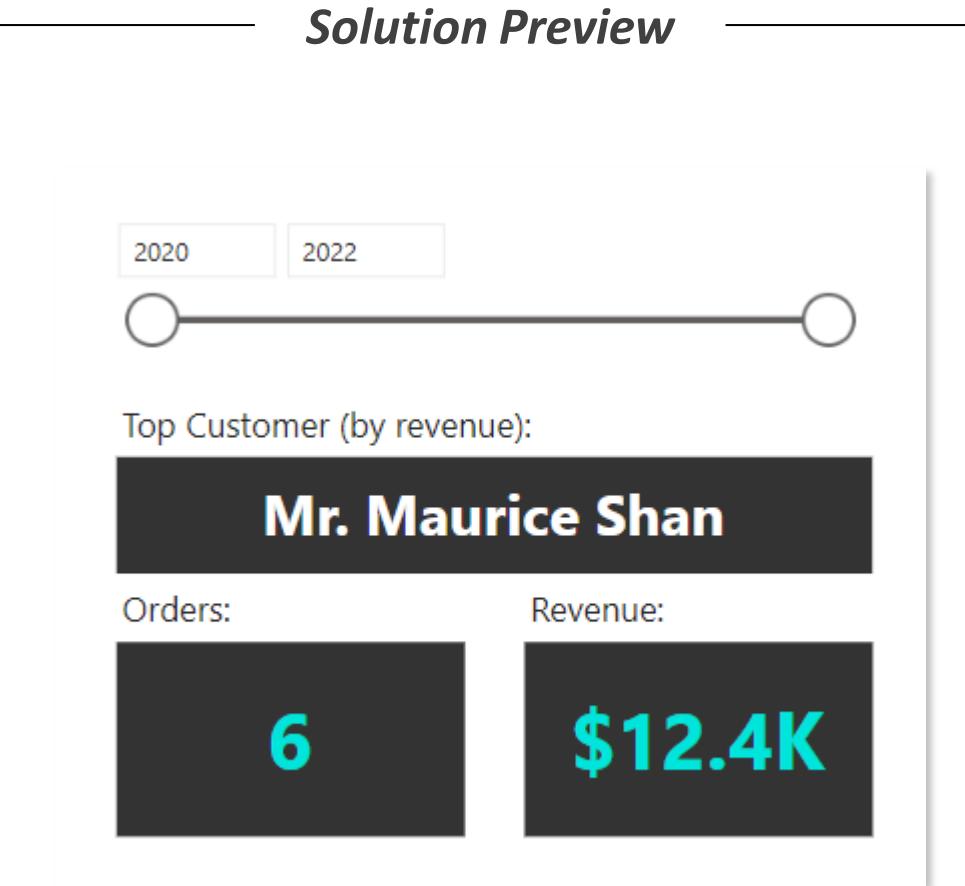
SOLUTION: SLICERS

  NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**
Subject: **Year slicer for customer report**

Hey there, quick request when you get a sec...
Could you please add a slicer to the customer report, so that users can filter the entire page by year?
No preference for which specific type of slicer you use, as long as managers can filter customers for a specific year or across multiple years.
Thanks!
-Vic

[Reply](#) [Forward](#)





DRILL UP & DRILL DOWN

Build a visual

Visual types

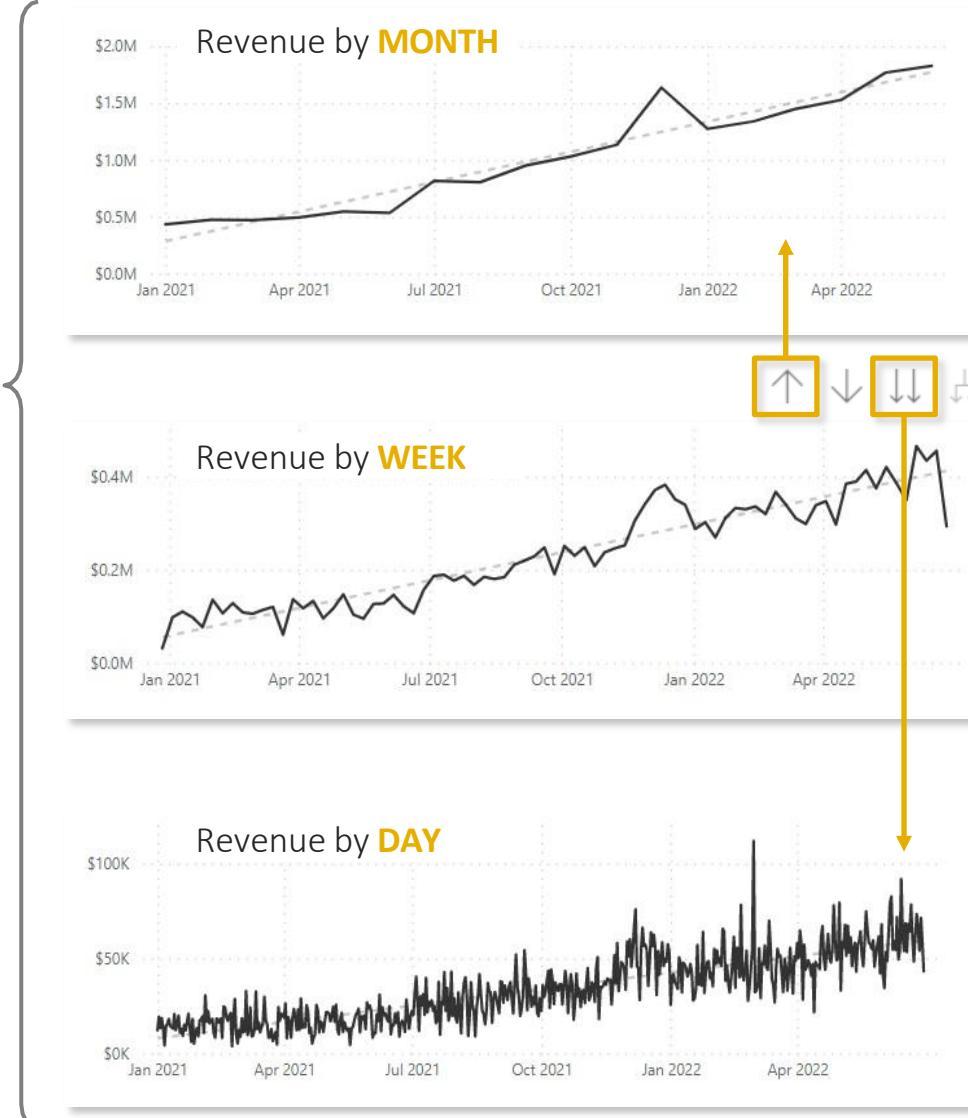
(• Off) Suggest a type

X-axis

- Start of Month X | >
- Start of Week X | >
- Date X | >
- +Add data

Y-axis

- Total Revenue X | >
- +Add data



Drill Up and **Drill Down** tools allow you to switch between different levels of granularity

- In this example users can “drill up” from **weekly** to **monthly**, or “drill down” to **daily**
- The single down arrow activates **drill mode**, allowing users to drill by clicking data points
- The forked down arrow **expands each level** of the hierarchy (used in matrix visuals)

PRO TIP:

Use **location hierarchies** and enable drill mode to create interactive map visuals



ASSIGNMENT: DRILL DOWN



NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Dynamic time periods**

Hey again, just got some feedback from the managers about our customer report.

Chad loves the weekly trending chart, but Thad wants to see the data by *day* and Vlad was hoping for an *annual* breakdown.

Instead of building multiple versions of the same line chart, could you please make it interactive so that Chad, Thad and Vlad get the views they want?

Thanks!

-Vic

Reply **Forward**

Key Objectives

1. In the **Customer Detail** report, update the X-axis of the line chart to pull in **Date Hierarchy**
2. Use the chart header to **drill up** and **drill down** to explore trends at each level of granularity
3. Test **drill mode** to change the granularity by selecting individual data points in the chart
 - Why do some weeks look very low?
4. Turn off drill mode and show the chart at a weekly level of granularity by default



SOLUTION: DRILL DOWN



NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Dynamic time periods**

Hey again, just got some feedback from the managers about our customer report.

Chad loves the weekly trending chart, but Thad wants to see the data by day and Vlad was hoping for an *annual* breakdown.

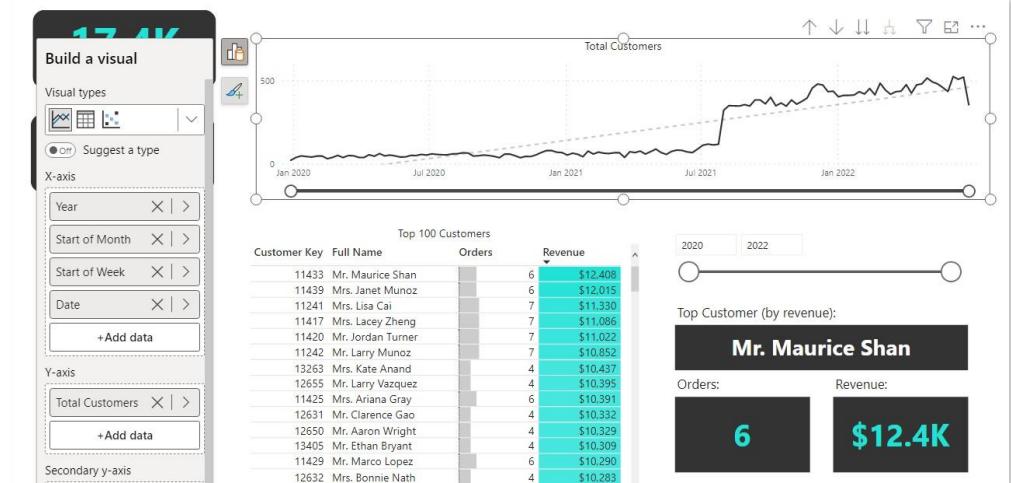
Instead of building multiple versions of the same line chart, could you please make it interactive so that Chad, Thad and Vlad get the views they want?

Thanks!

-Vic

Reply **Forward**

Solution Preview





DRILL THROUGH FILTERS

Drill through filters allow users to navigate to a specific report page, pre-filtered on the item selected

- Here we've created a **Product Detail** page, set the type to **Drillthrough**, and configured drill through from **Product Name**
- This means that users can right-click any instance of product name (i.e. in a matrix visual) and use the Drill through option to navigate straight to the Product Detail report filtered on that product (in this case "Mountain Tire Tube")

The screenshot illustrates the setup and execution of a drill-through filter in Power BI.

Left Side: A table titled "Top 10 Products" is displayed. The "Mountain Tire Tube" row is highlighted with a yellow box. A context menu is open at this row, with the "Drill through" option selected and highlighted by a yellow box. A yellow arrow points from the "Mountain Tire Tube" cell in the table to the "Drill through" menu item.

Middle Right: The "Product Detail" report page is shown. It features a title "Selected Product: Mountain Tire Tube". Below it are three donut charts: "Monthly Orders vs. Target" (value 275), "Monthly Revenue vs. Target" (\$2,735), and "Monthly Profit vs. Target" (\$1,710). A line chart tracks "Price Adjustment (%)" over time from July 2021 to May 2022. Another line chart shows "Total Orders" over the same period. A legend indicates the data series: Total Orders (black line), Total Revenue (light blue line), Total Profit (dark blue line), Total Returns (light grey line), and Return Rate (grey line).

Right Side: The "Format" pane is open, showing "Page type" set to "Drillthrough". Other settings include "Keep all filters" (On) and "Cross-report" (On). Under "Drill through from", "Product Name" is selected, and a button "+Add data" is visible.



REPORT INTERACTIONS

Edit **report interactions** to customize how filters applied to one visual impact other visuals on the page

- Cross-filter options include **filter** (), **highlight** () and **none** (), depending on the visual type

Format > **Edit Interactions**



In this example, selecting a product in the matrix visual:

- **Filters** the line chart & KPIs
- **Highlights** the bar chart
- **Doesn't impact** the text cards



ASSIGNMENT: REPORT INTERACTIONS



NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Weird report interactions**

Hey there,
I was playing with the customer report this morning and noticed some odd visual interactions. For example, selecting a specific customer shouldn't filter the line chart, and probably shouldn't filter the donut charts either.

Could you please take a pass through the report interactions and update any that seem off?

Thanks!
-Vic

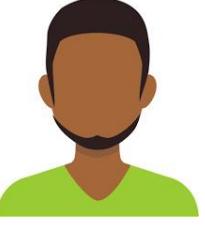
Reply **Forward**

Key Objectives

1. On the **Customer Detail** tab, edit the **report interactions** based on the following logic:
 - When a filter is applied to the line chart, the donut charts should **filter** (not highlight)
 - When a filter is applied to the table, the line chart and donuts should **not filter**
 - The slicer should **filter all visuals** on the report page



SOLUTION: REPORT INTERACTIONS



**NEW MESSAGE**

From: **Victor Ignatius Zabel (BI Analyst)**

Subject: **Weird report interactions**

Hey there,

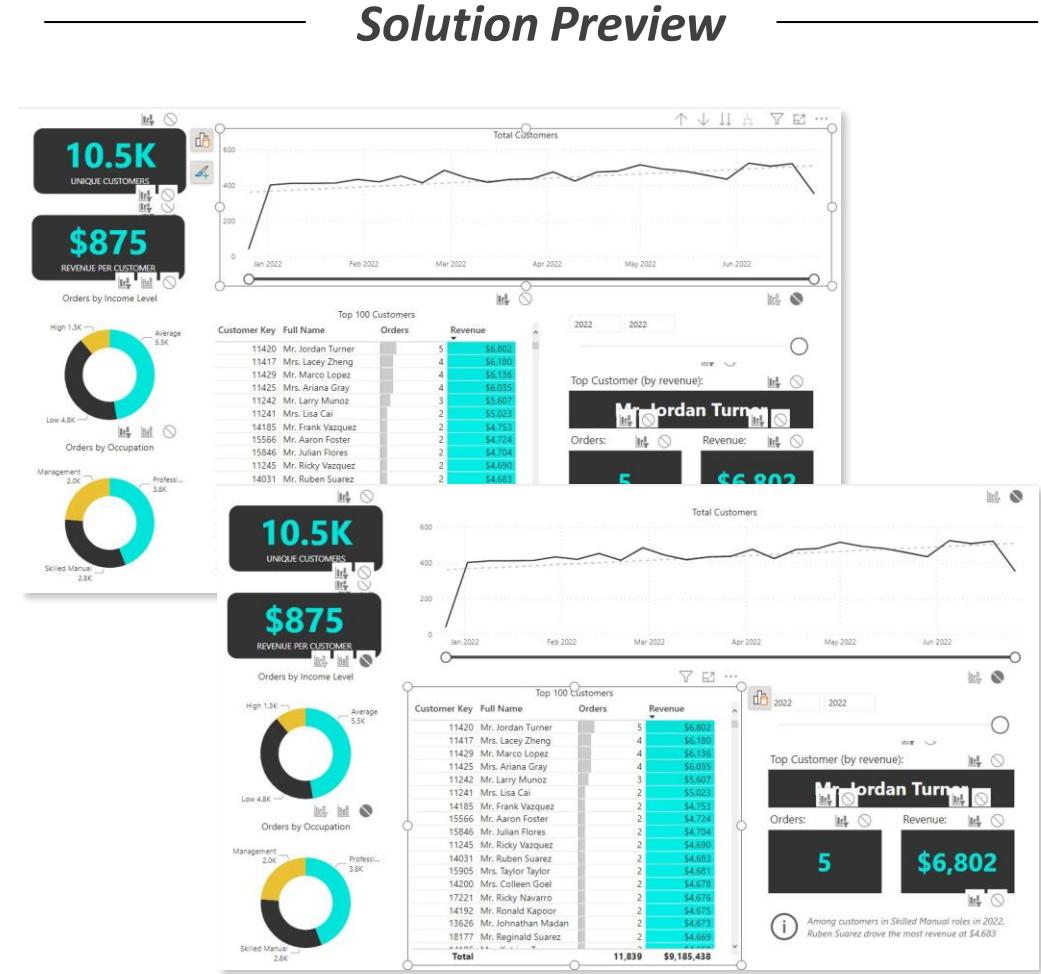
I was playing with the customer report this morning and noticed some odd visual interactions. For example, selecting a specific customer shouldn't filter the line chart, and probably shouldn't filter the donut charts either.

Could you please take a pass through the report interactions and update any that seem off?

Thanks!

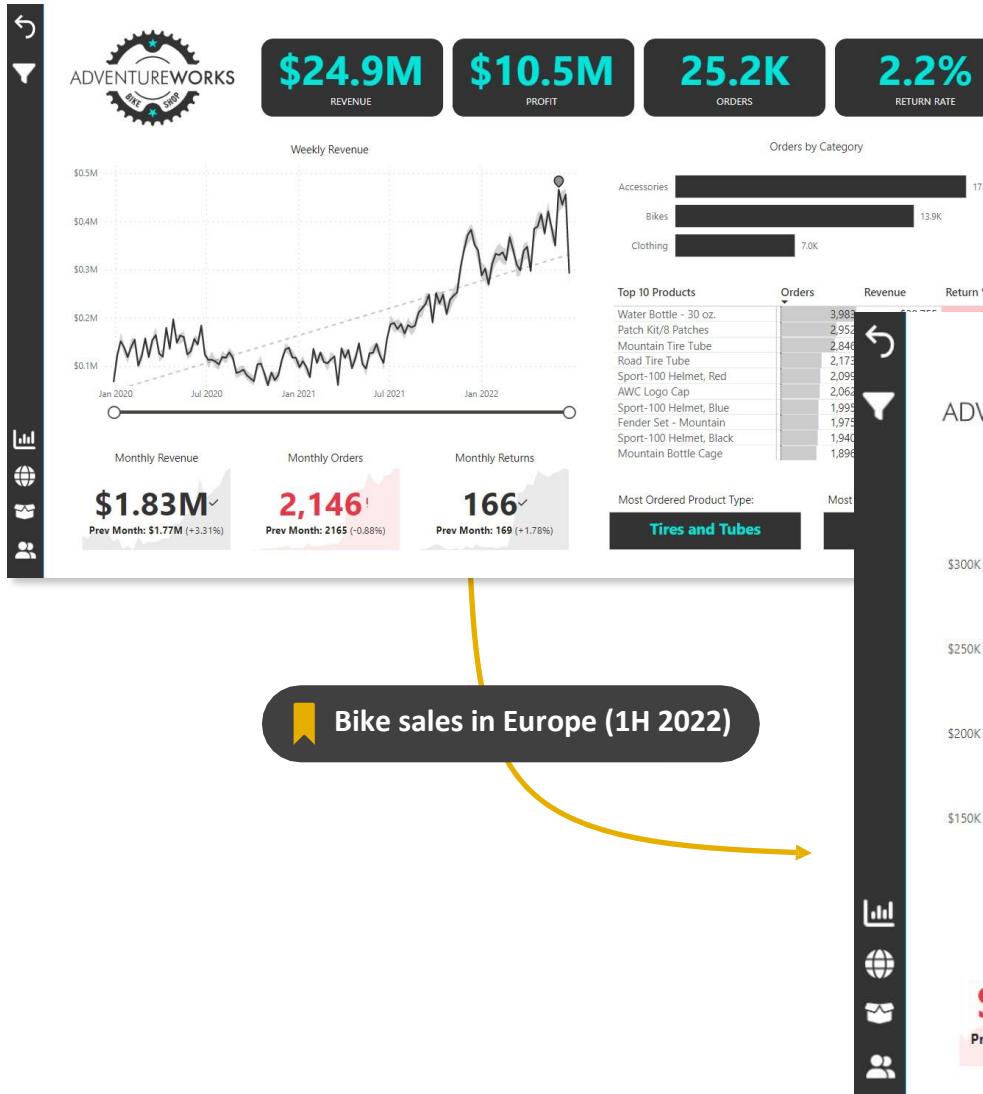
-Vic

Reply **Forward**





BOOKMARKS



Bookmarks capture the current state of a page, and allow users to return to that state using report actions

- Bookmarks are commonly used for clearing filters, highlighting specific insights, navigating reports, etc.





ASSIGNMENT: BOOKMARKS



NEW MESSAGE

From: **Victor Ignatius Zabel** (BI Analyst)

Subject: **Finding anything interesting?**

Hey,

Now that you're getting pretty familiar with our customer data, are you noticing any interesting insights or trends that might be worth explicitly calling out in the report?

This could be a great way for us to use bookmarks to draw attention to some specific stories in the dashboard. While we're at it, let's add another one to clear all filters from the page.

Let me know what you think!

-Vic

Reply **Forward**

Key Objectives

1. Explore the **Customer Detail** report by adjusting filters until you find an interesting insight or trend (*this can be anything you choose!*)
2. Add a new **bookmark** to capture the current state of the report, and name it “Customer Insight”
3. Insert an **Information button** and add text to the button style to summarize what you’ve found
4. Assign a **bookmark action** to the button, and link to the Customer Insight bookmark you created
5. Create a second bookmark named “Clear all Customer Filters” which returns the page to an unfiltered state, and link it to a **Reset button**
6. Test both bookmarks using **CTRL-click**



SOLUTION: BOOKMARKS



NEW MESSAGE

From: **Victor Ignatius Zabel** (BI Analyst)

Subject: **Finding anything interesting?**

Hey,

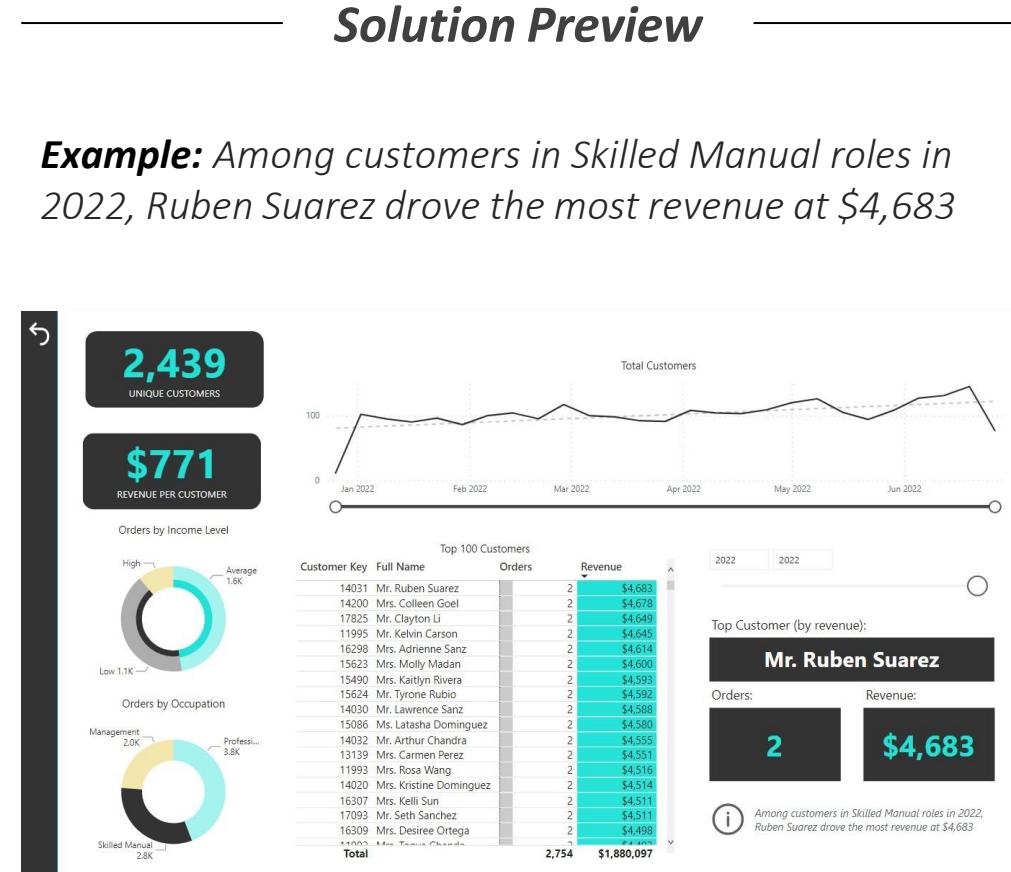
Now that you're getting pretty familiar with our customer data, are you noticing any interesting insights or trends that might be worth explicitly calling out in the report?

This could be a great way for us to use bookmarks to draw attention to some specific stories in the dashboard. While we're at it, let's add another one to clear all filters from the page.

Let me know what you think!

-Vic

Reply **Forward**





PARAMETERS

Parameters allow you to create variables which can be referenced in measures and controlled via slicers

Numeric range parameters

Typically used for scenario testing, where users adjust numerical inputs to see the impact on a given output

Fields parameters

Typically used to allow users to dynamically change the metrics or dimensions displayed in a report visual

The screenshot shows two side-by-side 'Parameters' dialog boxes from the Power BI interface.

Left Dialog (Numeric range parameter):

- Header: Modeling tab selected.
- Section: 'What will your variable adjust?' dropdown set to 'Numeric range'.
- Fields:
 - Name: Price Adjustment (%)
 - Data type: Decimal number
 - Minimum: -1
 - Maximum: 1
 - Increment: 0.1
 - Default: 0
- Buttons: 'Add slicer to this page' checked, 'Create' (green), 'Cancel'.

Right Dialog (Fields parameter):

- Header: Modeling tab selected.
- Section: 'What will your variable adjust?' dropdown set to 'Fields'.
- Fields:
 - Name: Y-Axis Dynamic Value
 - Add and reorder fields section:
 - Total Cost
 - Total Revenue
 - Total Profit
- Fields list:
 - Search bar: 'Search'
 - List:
 - > Measure Table
 - > Calendar Lookup
 - > Customer Lookup
 - > Price Adjustment (%)
 - > Product Categories Lookup
 - > Product Lookup
 - > Product Subcategories Lookup
 - > Returns Data
 - > Rolling Calendar
 - > Sales Data
 - > Territory Lookup
- Buttons: 'Add slicer to this page' checked, 'Create' (green), 'Cancel'.



EXAMPLE: NUMERIC RANGE PARAMETER

Parameters

Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. [Learn more](#)

What will your variable adjust?

Numeric range

Name: Price Adjustment (%)

Data type: Decimal number

Minimum: -1

Maximum: 1

Increment: 0.1

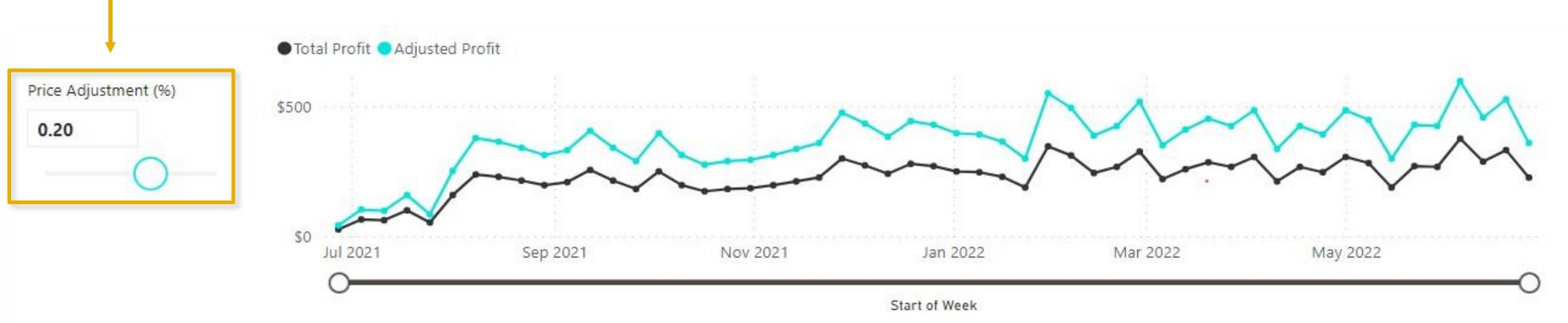
Default: 0

When you create a numeric parameter, Power BI generates **two new measures**: one to define the parameter and another to capture the selected value:

Parameter = `GENERATESERIES(-1, 1, 0.1)`

Parameter Value = `SELECTEDVALUE(Parameter[Parameter], 0)`

Here we've created a parameter named **Price Adjustment %**, added it as a slicer, and created measures to calculate **Adjusted Profit** based on the parameter value





EXAMPLE: FIELDS PARAMETER

Parameters

Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. [Learn more](#)

What will your variable adjust?

Fields

Name

Metric Selection

Add and reorder fields

- Total Orders
- Total Revenue
- Total Profit
- Total Returns
- Return Rate

Add slicer to this page

Create **Cancel**

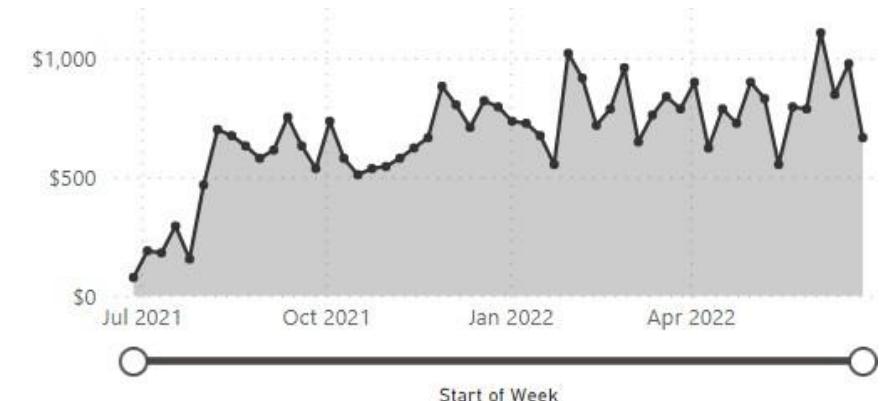
When you create a fields parameter, Power BI **adds a report slicer** and **generates a new measure** to capture the selected value:

```
1 Metric Selection = {  
2     ("Total Orders", NAMEOF('Measure Table'[Total Orders]), 0),  
3     ("Total Revenue", NAMEOF('Measure Table'[Total Revenue]), 1),  
4     ("Total Profit", NAMEOF('Measure Table'[Total Profit]), 2),  
5     ("Total Returns", NAMEOF('Measure Table'[Total Returns]), 3),  
6     ("Return Rate", NAMEOF('Measure Table'[Return Rate]), 4)  
7 }
```

Here we've created a parameter named **Metric Selection** and added it to the Y-axis to let users dynamically change the metric shown

Metric Selection

- Total Orders
- Total Revenue
- Total Profit
- Total Returns
- Return Rate



ASSIGNMENT: FIELDS PARAMETERS





NEW MESSAGE

From: **Victor Ignatius Zabel** (BI Analyst)

Subject: **More line chart updates**

Good news and bad news...

The good news is that Chad, Thad and Vlad LOVE the drill options in the line chart – nice work!

The bad news is that now they can't align on what's the best metric to show. Chad likes seeing total customers, but Vlad is pushing for revenue per customer.

What do you think we should do?

-Vic

Reply **Forward**

Key Objectives

1. Add a new **Fields parameter** named “Customer Metric Selection”, which includes **Total Customers** and **Average Revenue per Customer**
2. Add the parameter as a slicer to the **Customer Detail** report, change the slicer style to **Tile**, turn off the **header**, update to **single select**, and resize to create a horizontal layout
3. Select the DAX measure automatically created, and update the text from “Average Revenue per Customer” to “Revenue per Customer”
4. Update the line chart Y-Axis to use the **Customer Metric Selection** parameter, remove the chart title, and update the line colors to match the solution preview



SOLUTION: FIELDS PARAMETERS

  NEW MESSAGE

From: **Victor Ignatius Zabel (BI Analyst)**
Subject: **More line chart updates**

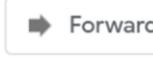
Good news and bad news...

The good news is that Chad, Thad and Vlad LOVE the drill options in the line chart – nice work!

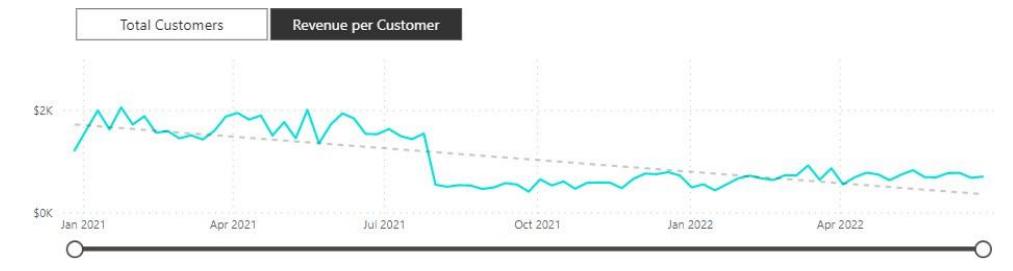
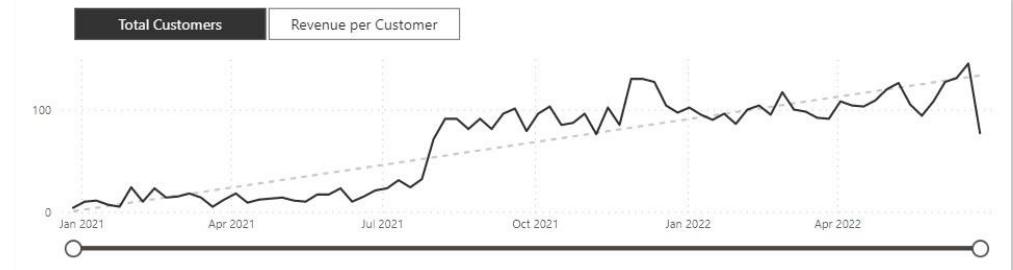
The bad news is that now they can't align on what's the best metric to show. Chad likes seeing total customers, but Vlad is pushing for revenue per customer.

What do you think we should do?

-Vic

Solution Preview





PRO TIP: CUSTOM TOOLTIPS

Create **custom tooltips** by designing a new report page, setting the page type to **Tooltip**, and configuring a visual to use the “Report page” tooltip type

Category Tooltip report page

The screenshot shows a 'Category Tooltip' report page. On the left, there's a chart titled 'Weekly Orders' with data for 2020, 2021, and 2022. To the right, there's a table titled 'Orders by Category' with a single row for 'Accessories'. A tooltip is overlaid on the chart, displaying detailed metrics: Total Revenue (\$24,914,567), Total Profit (\$10,457,581), Total Orders (25,164), Total Returns (1,809), and Return Rate (2.17%). The tooltip is configured in the 'Format' pane under 'Page information' with the name 'Category Tooltip' and 'Page type' set to 'Tooltip'. The 'Keep all filters' option is turned on. The 'Format' pane also includes 'Canvas settings' for height (225 px) and width (425 px). On the right, the 'Format' pane for the 'Orders by Category' table shows the 'Toolips' section is enabled, with the 'Type' set to 'Report page' and 'Page' set to 'Category Tooltip'. A callout bubble at the bottom right contains the text 'PRO TIP: Keep your published reports clean by hiding your tooltip pages'.

Category Tooltip report page

Weekly Orders

| | Revenue | Return % |
|-------|----------|----------|
| 2,067 | \$20,694 | 3.74% |
| 1,808 | \$17,849 | 2.60% |
| 1,527 | \$6,962 | 3.13% |
| 1,290 | \$56,533 | 2.10% |
| 1,096 | \$21,998 | 3.50% |
| 924 | \$31,083 | 7.14% |
| 920 | \$30,949 | 5.65% |
| 916 | \$32,051 | 7.64% |
| 869 | \$6,879 | 3.89% |
| 867 | \$30,345 | 5.65% |

Orders by Category

Accessories

Weekly Orders

| | Revenue | Return % |
|------|----------|----------|
| 5.3K | \$20,694 | 3.74% |

Monthly Returns

Sport-100 Helmet, Black

Sport-100 Helmet, Red

Road Tire Tube

HL Mountain Tire

Format

Search

Visual Properties

Header icons

On

Toolips

Type

Report page

Page

Category Tooltip

PRO TIP:
Keep your published reports clean by **hiding your tooltip pages**



IMPORTING CUSTOM VISUALS

Power BI offers a library of **custom visuals** (via **AppSource**) from Microsoft-certified partners and developers, which can be imported into the visualizations pane

The screenshot shows the Power BI desktop application. The top navigation bar includes 'File', 'Home', 'Insert' (which is underlined in green), 'Modeling', 'View', and 'Optimize'. Below the navigation bar, there are buttons for 'New page', 'New visual', and 'Pages'. The main area is titled 'Power BI visuals' and shows a grid of visual components. At the top of the grid, there are tabs: 'All visuals', 'Organizational visuals', and 'AppSource visuals' (which is currently selected). Below these tabs, there is a search bar and a filter dropdown set to 'All'. The grid contains several visual components, including a Gantt Chart, a Tachometer, Zebra BI Charts, a Supermetrics Chart (highlighted with a yellow box and arrow), and a Balance Sheet Visual. Each visual has a preview image, a title, and a developer name.

The screenshot shows the Microsoft AppSource website. The URL is 'AppSource | Apps for Power BI visuals'. The page displays a product card for 'Supermetrics Charts – Tile grid map' by Supermetrics. The card includes a preview image of a heatmap, an 'Add' button, a 'Download Sample' button, and a 'Sample Instructions' link. To the right of the card, there is a detailed description of the visual, including its purpose (Create (heat)maps with tiles of the same shape and size), features (Build your own tile grid map and compare your metrics across regions or countries. Use pre-defined map configurations or create a custom map. The tile grid map isn't applicable just to geographical data, it can also be used as a custom heatmap for many data sources.), and a sample image of the heatmap. Below the card, there is a section for 'Visual capabilities' with a note about external services.



HEY THIS IS IMPORTANT!

You need a **Power BI account** to browse or import custom visuals from the AppSource marketplace



MANAGING & VIEWING ROLES

Manage security roles

Create new security roles and use filters to define row-level data restrictions.

Roles

- + New
- Europe
- North America
- Pacific

Select tables

- Customer Met...
- Measure Table
- Price Adjustme...
- Product Categ...
- Product Lookup
- Product Metric...
- Product Subca...
- Returns Data
- Rolling Calendar
- Sales Data
- Territory Lookup

Filter data

- Select all
- + Add
- Delete
- Group
- Ungroup

Show data when...

All of these rules are true

Continent Equals North America

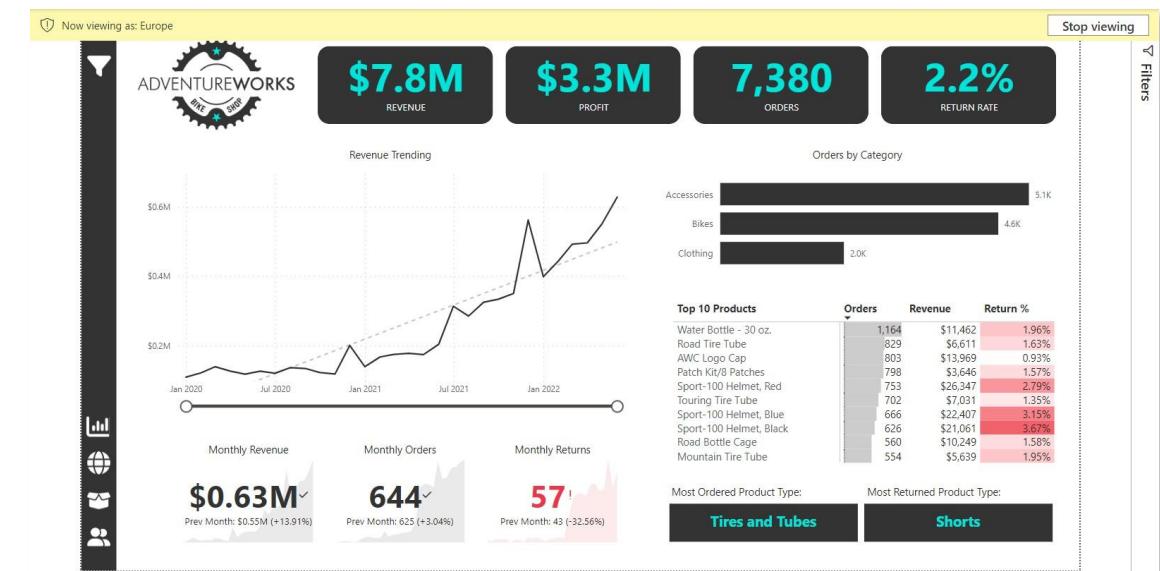
Switch to DAX editor

HEY THIS IS IMPORTANT!

Row-level security (RLS) rules are defined in Power BI Desktop, but actually applied in Power BI Service

Roles allow you to define row-level security rules, and create filtered views to restrict access for specific audiences

- Here we've created views for territory managers (**Europe**, **N. America**, **Pacific**), which filters records in the model





MOBILE LAYOUT

Mobile layout allows you to design mobile-specific versions of report pages by assembling visuals into new layouts

- **NOTE:** This is designed to optimize reports for viewing on the Power BI mobile app (after publishing to Power BI Service)

The screenshot illustrates the process of creating a mobile layout in Power BI. On the left, the desktop interface shows the ribbon with the 'View' tab selected. Below it, a preview of the report includes various cards and charts. An orange arrow points from the 'Mobile' button in the ribbon to the mobile layout preview on the right. The mobile layout preview shows a simplified version of the report, optimized for mobile devices, with large numbers and key metrics like '\$24.9M', '\$10.5M', '25.2K', and '2.2%'. It also includes a weekly revenue chart and a top 10 products table. A second orange arrow points from the 'Orders by Category' card in the desktop view to its corresponding card in the mobile layout preview.

Mobile layout options in the ribbon:

- Page view
- Mobile layout
- Mobile

Desktop View (Left):

- Themes
- Scale to fit
- Mobile

Mobile Layout Preview (Right):

- Monthly Revenue: \$1.83M (Prev Month: \$1.77M +3.31%)
- Monthly Returns: 166 (Prev Month: 169 +1.78%)
- Weekly Revenue: \$0.6M
- Orders by Category: Accessories (17.0K), Bikes (13.9K), Clothing (7.0K)
- Top 10 Products (Table):

| Product | Orders | Revenue | Return % |
|-------------------------|--------|----------|----------|
| Water Bottle - 30 oz. | 3,983 | \$39,755 | 1.95% |
| Patch Kit/B Patches | 2,952 | \$13,506 | 1.61% |
| Mountain Tire Tube | 2,846 | \$28,333 | 1.64% |
| Road Tire Tube | 2,173 | \$17,265 | 1.55% |
| Sport-100 Helmet, Red | 2,099 | \$73,444 | 3.33% |
| AWC Logo Cap | 2,062 | \$35,865 | 1.11% |
| Sport-100 Helmet, Blue | 1,995 | \$67,112 | 3.31% |
| Fender Set - Mountain | 1,975 | \$87,041 | 1.36% |
| Sport-100 Helmet, Black | 1,940 | \$65,262 | 2.68% |
| Mountain Bottle Cage | 1,896 | \$38,062 | 2.02% |

- Most Ordered Product: Tires and Tubes
- Most Returned Product Type: Shorts
- Monthly Revenue: \$1.83M (Prev Month: \$1.77M +3.31%)
- Monthly Orders: 2,146 (Prev Month: 2,165 -0.88%)
- Monthly Returns: 166 (Prev Month: 169 +1.78%)

Mobile Visuals Preview (Right):

- Orders by Category: Accessories (17.0K), Bikes (13.9K), Clothing (7.0K)
- Monthly Revenue: \$1.83M (Prev Month: \$1.77M +3.31%)
- Matrix: Top 10 Products (Table):

| Product | Count |
|-------------------------|-------|
| Water Bottle - 30 oz. | 3,983 |
| Patch Kit/B Patches | 2,952 |
| Mountain Tire Tube | 2,846 |
| Road Tire Tube | 2,173 |
| Sport-100 Helmet, Red | 2,099 |
| AWC Logo Cap | 2,062 |
| Sport-100 Helmet, Blue | 1,995 |
| Fender Set - Mountain | 1,975 |
| Sport-100 Helmet, Black | 1,940 |
| Mountain Bottle Cage | 1,896 |

- Card: 25.2K
- Card: \$24.9M
- Card: 2.2%
- Card: 166



DATA VISUALIZATION BEST PRACTICES



Always ask yourself the three key questions

- *What type of data are you visualizing, what are you communicating, and who is the end user?*



Strive for clarity and simplicity above all else

- *"Perfection is achieved not when there's nothing more to add, but when there's nothing left to take away"*



Focus on creating clear narratives and intuitive user experiences

- *Use bookmarks, drillthroughs, tooltips and navigation buttons to seamlessly guide users through reports*

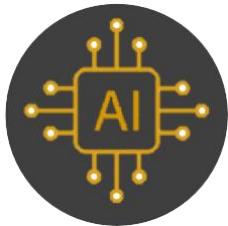


Create optimized layouts for mobile viewers

- *Create custom mobile layouts if you plan to publish reports to Power BI Service or use the Power BI app*

ARTIFICIAL INTELLIGENCE

ARTIFICIAL INTELLIGENCE



In this section we'll explore Power BI's artificial intelligence features, including anomaly detection, smart narratives, natural language Q&A, decomposition trees, and more

TOPICS WE'LL COVER:

Anomaly Detection

Smart Narrative

Q&A Visual

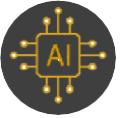
Decomposition Tree

Key Influencers

Top Segments

GOALS FOR THIS SECTION:

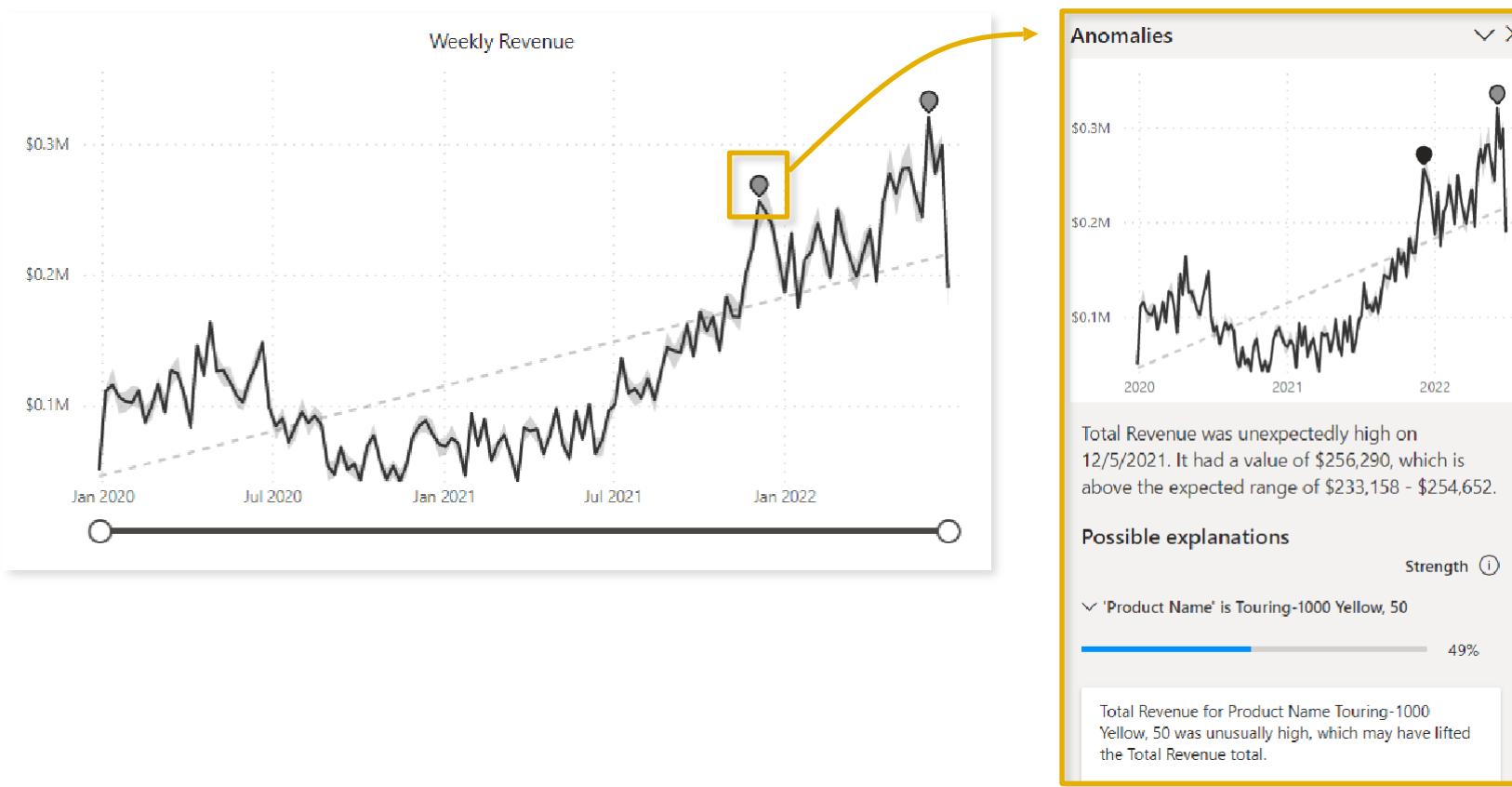
- Explore AI-generated insights using smart narratives and anomaly detection
- Build and train Q&A visuals to allow users to explore Power BI models using natural language queries
- Learn how to use decomposition trees for data exploration and root cause analysis
- Use key influencer visuals to identify the underlying factors that drive specific outcomes for the business



ANOMALY DETECTION

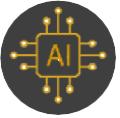
Anomaly detection is used to automatically detect and explain anomalies in time series data

- The anomaly detection feature adds “flags” to existing line charts, which link to AI-generated explanations and summaries



Limitations:

- Only supported for line charts with a time-series field on the X-axis
- Does not support charts with legends, multiple values, or a secondary axis
- Cannot be applied at the same time as forecasts
- Not compatible with drill up/drill down
- Requires at least four data points



SMART NARRATIVES

Smart narratives create customizable, AI-generated text summaries based on report pages or visuals

- Smart narratives react to report filters like any other visual, and can be updated with custom, dynamic values

Selected Product:
Patch Kit/8 Patches

Monthly Orders vs. Target: 265 / 319

Monthly Revenue vs. Target: \$1,225 / \$1.49K

Monthly Profit vs. Target: \$765 / \$1,530

Total Profit: ● Total Profit ○ Adjusted Profit

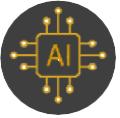
Price Adjustment (%): 0.00

Return Rate: 10% / 5% / 0%

Metric Selection: Total Orders, Total Revenue, Total Profit, Total Returns, Return Rate

Report Summary: Total orders for Patch Kit/8 Patches were 265 this month. All metrics trended up between Sunday, June 27, 2021 and Sunday, June 26, 2022, each increasing by 3.2000%. Return Rate had two high anomalies on Sunday, July 4, 2021 (8.00%) and Sunday, July 25, 2021 (9.52%).

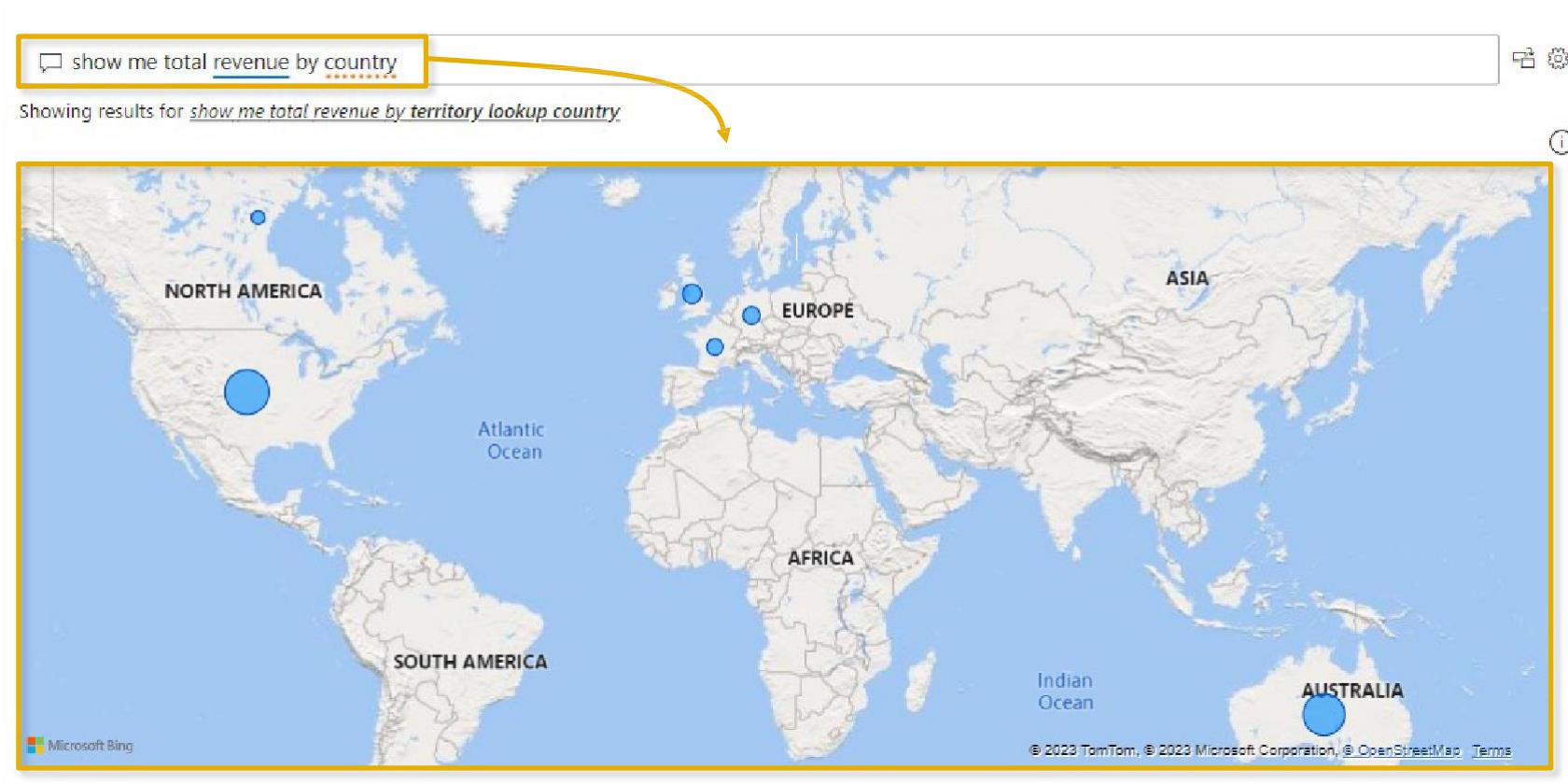
Create a dynamic value that updates with your data
How would you calculate this value
current product
Result
Patch Kit/8 Patches
\$%
Name your value
Product
Save Cancel

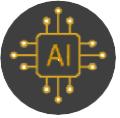


Q&A VISUALS

Q&A visuals allow users to explore and visualize data using intuitive, natural language prompts

- Q&A visuals are only as useful as the data model behind them, and typically require significant “training” to be effective

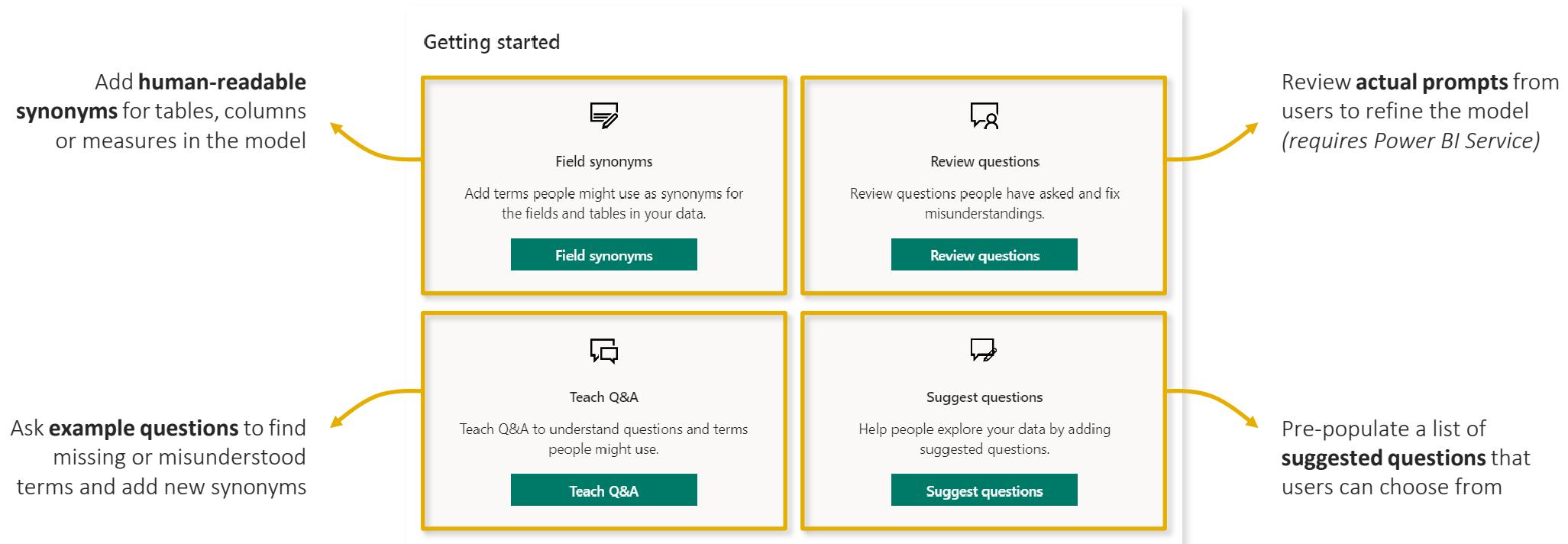


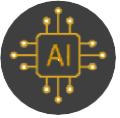


Q&A TRAINING

Q&A visuals allow users to explore and visualize data using intuitive, natural language prompts

- Q&A visuals are only as useful as the data model behind them, and typically require significant “training” to be effective

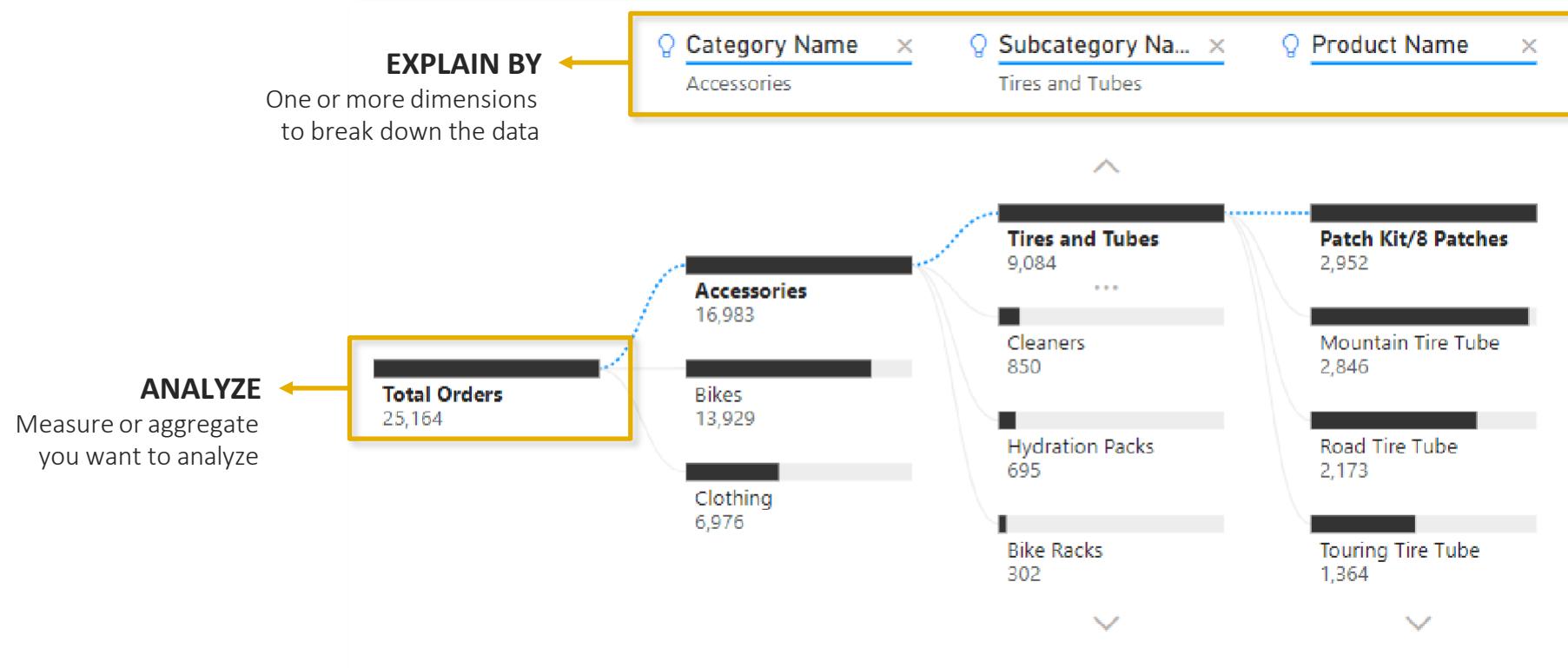


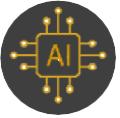


DECOMPOSITION TREES

Decomposition trees allow you to visualize how data is distributed across multiple dimensions

- Decomposition trees can be configured manually for data exploration, or leverage AI to support root cause analysis

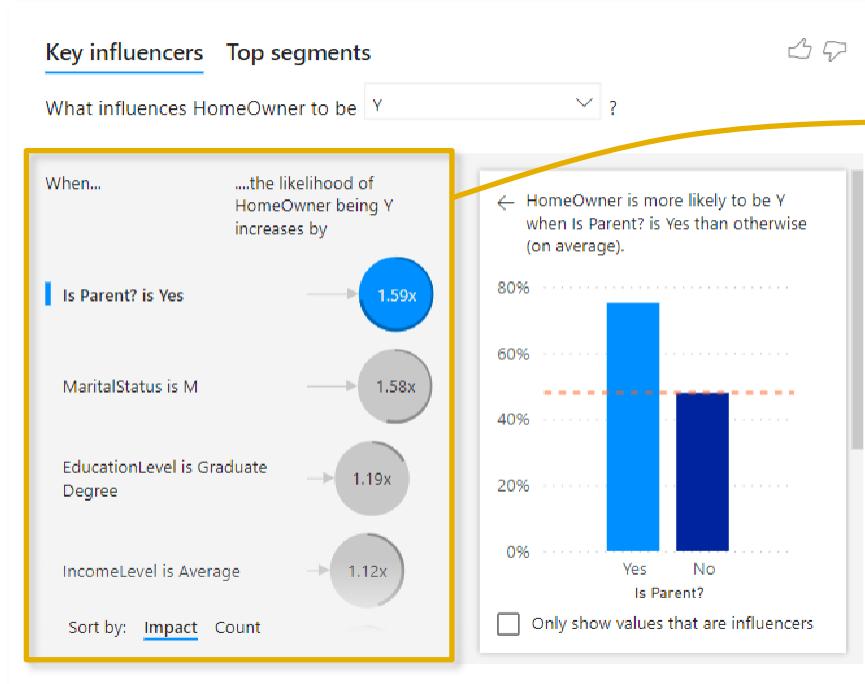




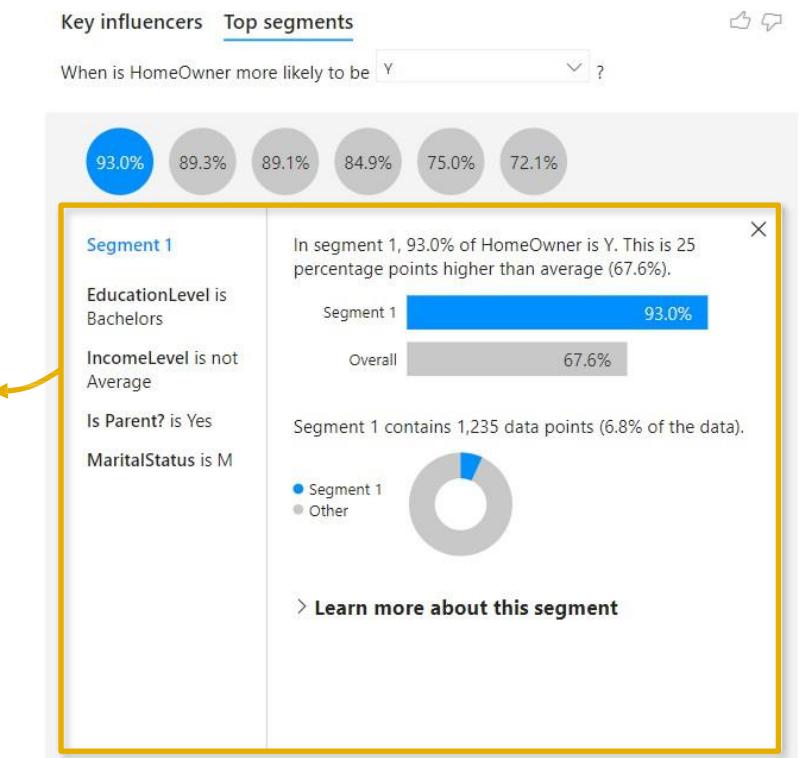
KEY INFLUENCERS

The **key influencer** visual helps you understand the factors that drive specific metrics or outcomes

- This can be used to analyze categorical or continuous outcomes, or identify top segments based on combinations of factors



Here we're identifying factors that are highly correlated with owning a home; for example, **parents are 1.59X more likely to be homeowners**, all else equal



We can also identify customer segments where this outcome is likely; for example, **93% of married customers with children and a Bachelors degree own a home** (vs. 67.6% overall)

OPTIMIZATION TOOLS

PREVIEW: POWER BI OPTIMIZATION



In this section, we'll investigate several native and external tools that can be used to optimize and enhance your Power BI reports

TOPICS WE'LL COVER:

Optimize Ribbon

Pause Visuals

Optimization Presets

Apply all Slicers

Performance Analyzer

External Tools

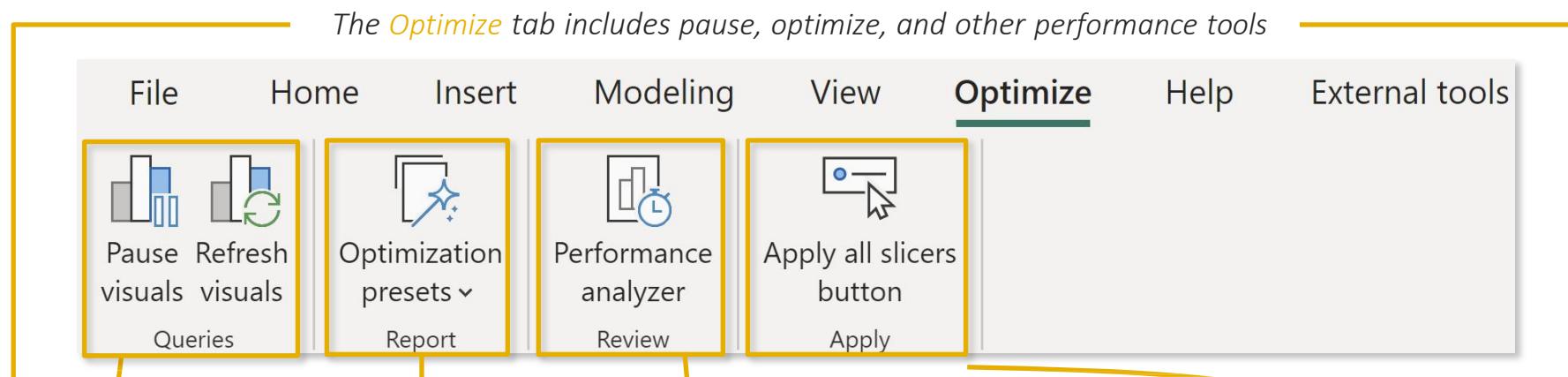
GOALS FOR THIS SECTION:

- Explore the optimize ribbon tools, features, and use cases
- Understand how and when pausing visuals can aid in report development and creation
- Use Performance Analyzer to measure and compare the impact of report elements on speed and performance
- Explore external tools that can aid in report development, learning, and optimization



OPTIMIZE RIBBON

The **Optimize ribbon** helps report authoring by allowing developers to pause data source queries, apply preset settings, and view logs that measure report element performance



Pause or **refresh queries** to make updates without processing changes

Show **record logs** that measure each elements performance within a report

Predefined optimization presents that can be applied based on your reporting scenario

Apply and **clear multiple slicer selections** on a report page at once



PAUSE VISUALS

Pause visuals stops queries from running and is used when you don't want to immediately apply additions or changes made to a report page or visual

The screenshot shows a Power BI dashboard with the following key elements:

- Ribbon:** File, Home, Insert, Modeling, View, **Optimize**, Help, External tools.
- Paused visuals button:** Located in the Queries group of the ribbon, highlighted with a yellow box.
- Banner:** "Visuals are paused. Some edits won't be applied until you refresh or resume visual queries. [Learn more](#)"
- Visuals:**
 - Adventureworks:** Logo with BI and SWIP icons.
 - Weekly Revenue:** Line chart showing revenue from Jan 2020 to Jan 2022.
 - Orders by Category:** Bar chart showing Accessories, Bikes, and Clothing.
 - Top 10 Products:** Table showing product details like Water Bottle - 30 oz., Patch Kit/8 Patches, Mountain Tire Tube, etc.
 - Product Details:** Includes Monthly Revenue (\$1.83M), Monthly Orders (2,146), and Monthly Returns (166).
- Bottom Navigation:** Exec Dashboard, Map, Product Detail, Customer Detail, Category Tooltip, AI: Q&A, AI: Decomposition Tree, AI: Key Influence, +

When paused, the report:

- Holds all changes & updates and sets them to a **"visual has pending changes"** state
- Shows a banner with **refresh & resume visual queries**
- Adds a **refresh button to individual visuals** allowing you to only refresh that visual
- Allows you to **add, move, and remove columns** and measures without having to wait for visuals to refresh
- **Blocks formatting actions**



OPTIMIZATION PRESETS

Optimization presets allow you to apply different predefined query optimization settings like query reduction, interactive, and custom

The screenshot shows the Power BI desktop interface with the 'Optimize' tab selected in the ribbon. A yellow box highlights the 'Optimization presets' dropdown menu, which is open to show 'Query reduction' (selected), 'Interactivity' (checked), and 'Customize'. A tooltip for 'Query reduction' states: 'This preset allows cross-highlighting and cross-filtering, as well as real-time slicer selections. Usually best for import mode.' Below the menu are several dashboard cards displaying financial metrics: '\$24.9M REVENUE', '\$10.5M PROFIT', '25.2K ORDERS', and '2.2% RETURN RATE'. To the left, there's a chart titled 'Weekly Revenue' showing a line graph from Jan 2020 to Jan 2022. Other cards include 'Monthly Revenue' (\$1.83M), 'Monthly Orders' (2,146), and 'Monthly Returns' (166). On the right, there are sections for 'Orders by Category' (Accessories: 17.0K, Bikes: 13.9K, Clothing: 7.0K) and 'Top 10 Products' (Water Bottle - 30 oz: 3,983 orders, \$39,755 revenue, 1.95% return). The bottom navigation bar includes tabs for 'Exec Dashboard', 'Map', 'Product Detail', 'Customer Detail', 'Category Tooltip', 'AI: Q&A', 'AI: Decomposition Tree', and 'AI: Key Influence'.

Query Reduction

- Is *best for DirectQuery connections* because it follows the best practices for DirectQuery optimization, turns off cross-highlighting, cross-filtering, and adds an Apply button to the filters pane

Interactivity

- Is the default setting and *best used for Import mode* because it allows cross-highlighting, cross-filtering, and real-time changes to slicers and filters

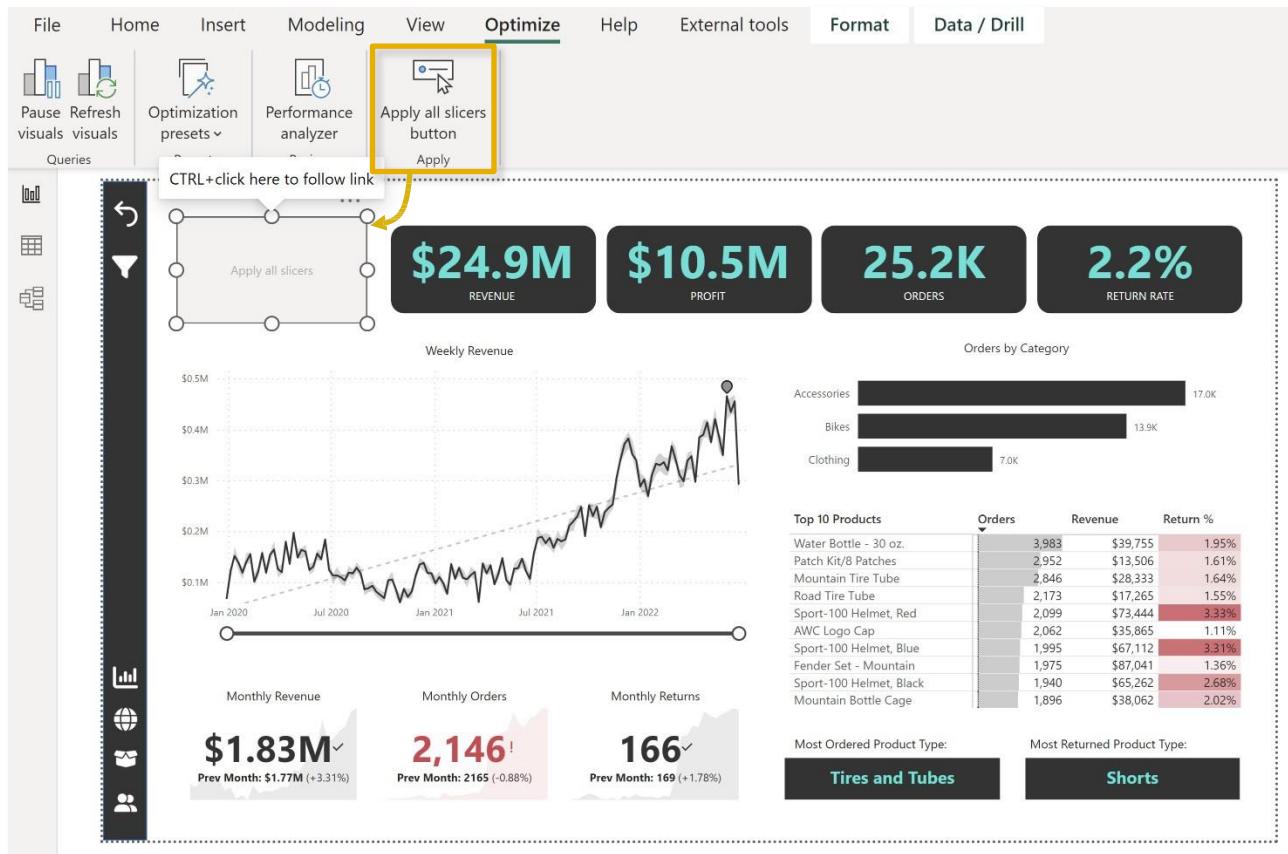
Customize

- Is best when you want to *choose which query reduction features to use*



APPLY ALL SLICERS BUTTON

Adding an **apply all slicers** button to your report page tracks all slicer selections and can be used to either apply or clear all slicers at once



Common scenarios & considerations:

- **Apply multiple slicers** on a report page at once
- **Clear all slicers** on a report page at once
- Apply & clear all slicer buttons **impact all slicers** on the report page (*you can't pick and choose!*)
- You can have as **many of these buttons** as you'd like
- Button can be **added and formatted** just like other buttons in Power BI



PERFORMANCE ANALYZER

Performance Analyzer records user actions (*like Excel's macro recorder*), and tracks the load time (*in milliseconds*) for each step in the process

The screenshot shows a Power BI dashboard for ADVENTUREWORKS. The ribbon is visible at the top with the 'Optimize' tab selected. On the left, there's a navigation bar with icons for File, Home, Insert, Modeling, View, Optimize, Help, and External tools. The 'Performance analyzer' button is highlighted with a yellow box and a callout arrow. The main area displays various visualizations including a line chart of Weekly Revenue from Jan 2020 to Jan 2022, and summary metrics: \$24.9M REVENUE, \$10.5M PROFIT, and 25.2K ORDERS. A 'Filters' pane on the right shows categories like Accessories, Bikes, and Clothing. A 'Performance analyzer' overlay window is open, listing the duration of various components in milliseconds. The 'DAX Query' section shows durations for Shape, Image, Dashboard Icon, Map Icon, Product Icon, Customer Icon, and Card components. The 'Visual display' section shows durations for DAX query, Visual display, and Other components. The 'Other' section shows durations for Card components.

| Name | Duration (ms) |
|----------------|---------------|
| Shape | 123 |
| Image | 124 |
| Dashboard Icon | 123 |
| Map Icon | 123 |
| Product Icon | 122 |
| Customer Icon | 122 |
| Card | 329 |
| DAX query | 5 |
| Visual display | 6 |
| Other | 318 |
| Card | 356 |
| Card | 462 |
| Card | 384 |

DAX Query

- Shows the amount of time it takes for the visual to send the query to the engines, and for the engines to return the result (**Note:** DAX Studio can only help optimize this)

Visual Display

- Shows the amount of time it takes for the visual to populate, or “draw”, on the screen. Includes time to retrieve web-based and geocoded images

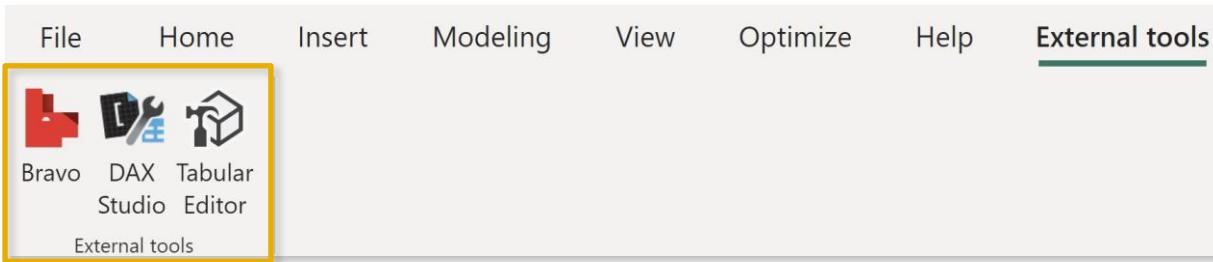
Other

- Shows the amount of time required by the visual to prepare the query, wait for other visuals to complete their queries and perform other processing tasks



EXTERNAL TOOLS

External tools allows quick access to third-party built tools that are *locally installed* on your computer and *registered* with Power BI Desktop



External tools generally fall into one of the following categories:

Semantic Modeling

These tools extend Power BI's functionality for specific data modeling scenarios like DAX optimization, ALM, and metadata translation

- *DAX Studio*
- *ALM Toolkit*
- *Tabular Editor*
- *Bravo*

Data Analysis

Includes tools for connecting a PBI data model to a client application, in read-only mode, to query data and perform analysis tasks

- *Python*
- *Excel*
- *Power BI Report Builder*

Miscellaneous

Some tools are used to make Power BI more useful and accessible but don't connect to the data model

- *PBI.tips tutorials*
- *DAX Guide*
- *PowerBI.tips*

Featured open-source tools: <https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-external-tools#featured-open-source-tools>