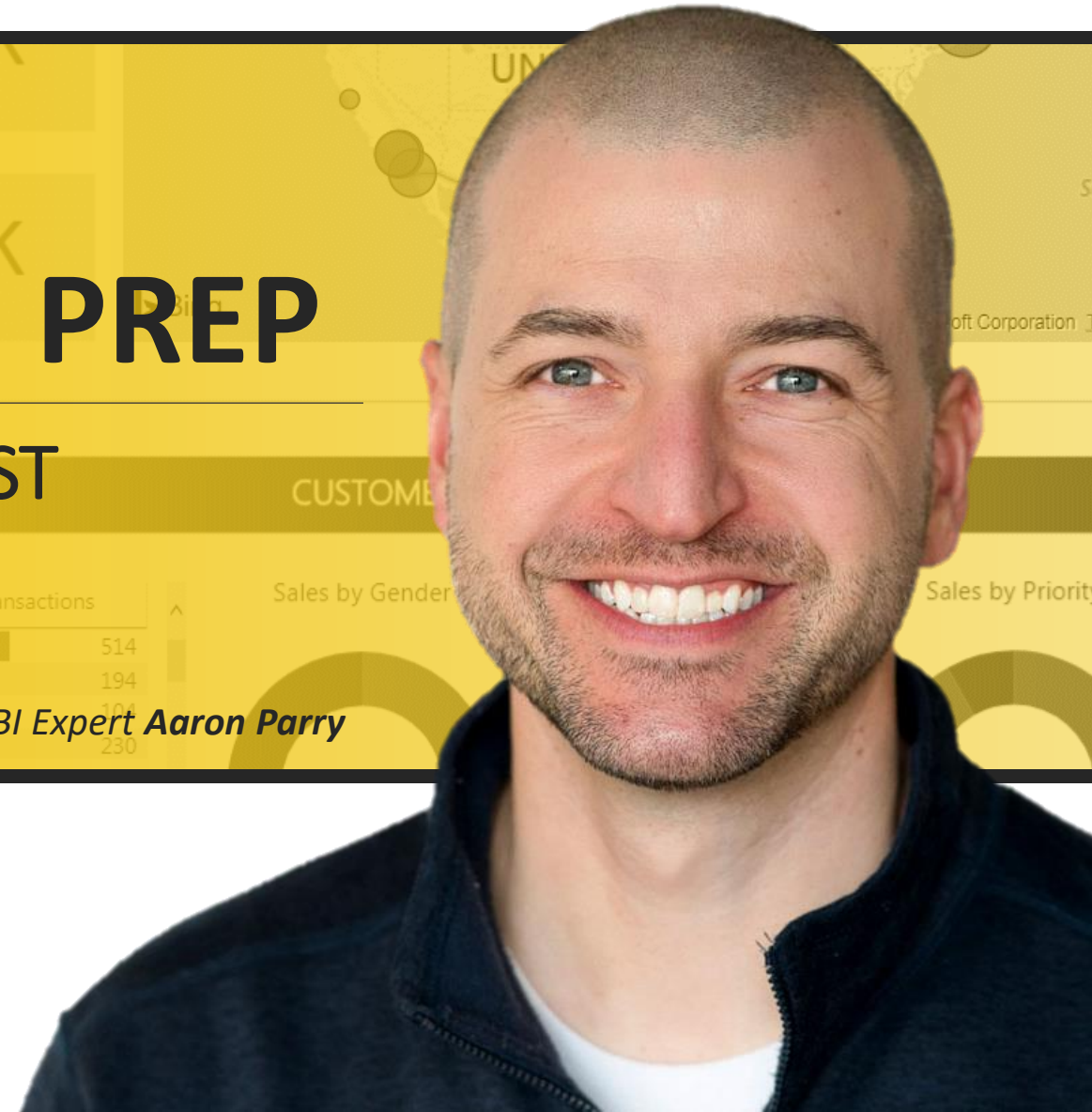
PL-300 EXAM PREP

POWER BI DATA ANALYST

★★★★★ *With Microsoft Certified Power BI Expert* ***Aaron Parry***



MAVEN ANALYTICS

COURSE STRUCTURE

This is a project-based course, designed to help you build the *exact* skills you need to pass the **Microsoft PL-300** exam *(formerly DA-100)*

Course resources include:

- **Downloadable PDF eBook** to serve as a helpful reference when you're offline or on the go (*or just need a refresher!*)
- **Quizzes** and **Hands-On Demos** to test and reinforce key concepts throughout the course
- Full-length **Practice Test** designed to replicate the PL-300 exam experience, with detailed solution walkthroughs

COURSE OUTLINE

1 Preparing for the PL-300 Exam

Review the skills measured in the PL-300 exam (formerly DA-100) and get familiar with the test structure and environment

2 Setting Up Desktop & Service

Install Power BI Desktop, set up a Power BI Service account, and activate your Pro/PPU trial

3 Measured Skills Review

- Prepare the Data
- Model the Data
- Visualize & Analyze the Data
- Deploy & Maintain Assets

Practice the core Power BI skills required for certification:

- *Extract, profile, clean, transform, and load data from different sources*
- *Design a data model, create DAX measures, and optimize performance*
- *Create reports & dashboards to identify patterns and trends*
- *Create and manage files, datasets & workspaces*

4 Practice Exam

Simulate the real testing experience with a full-scale practice exam and comprehensive solution walkthrough

INTRODUCING THE COURSE PROJECT

THE SITUATION

Congratulations! You've just been hired as the lead Business Intelligence Analyst for **Maven Cycles***, a boutique bicycle equipment shop

THE BRIEF

You've been asked to build an end-to-end business intelligence solution from the ground up, using raw data containing information about sales, products, customers, and store locations.

Your goal is to use the **entire Microsoft Power BI ecosystem** to design, build and deploy reports and dashboards to share across the organization.

THE OBJECTIVES

- Prepare the Data
- Model the Data
- Visualize & Analyze the Data
- Deploy & Maintain Assets

EST. 2021
MAVEN
CYCLES

*This data is provided by Microsoft and Maven Analytics for informational purposes only as an aid to illustrate a concept. These samples are provided "as is" without warranty of any kind. The example companies, organizations, products, domain names, e-mail addresses, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, person, place, or event is intended or should be inferred.

THE MAVEN CYCLES DATA MODEL

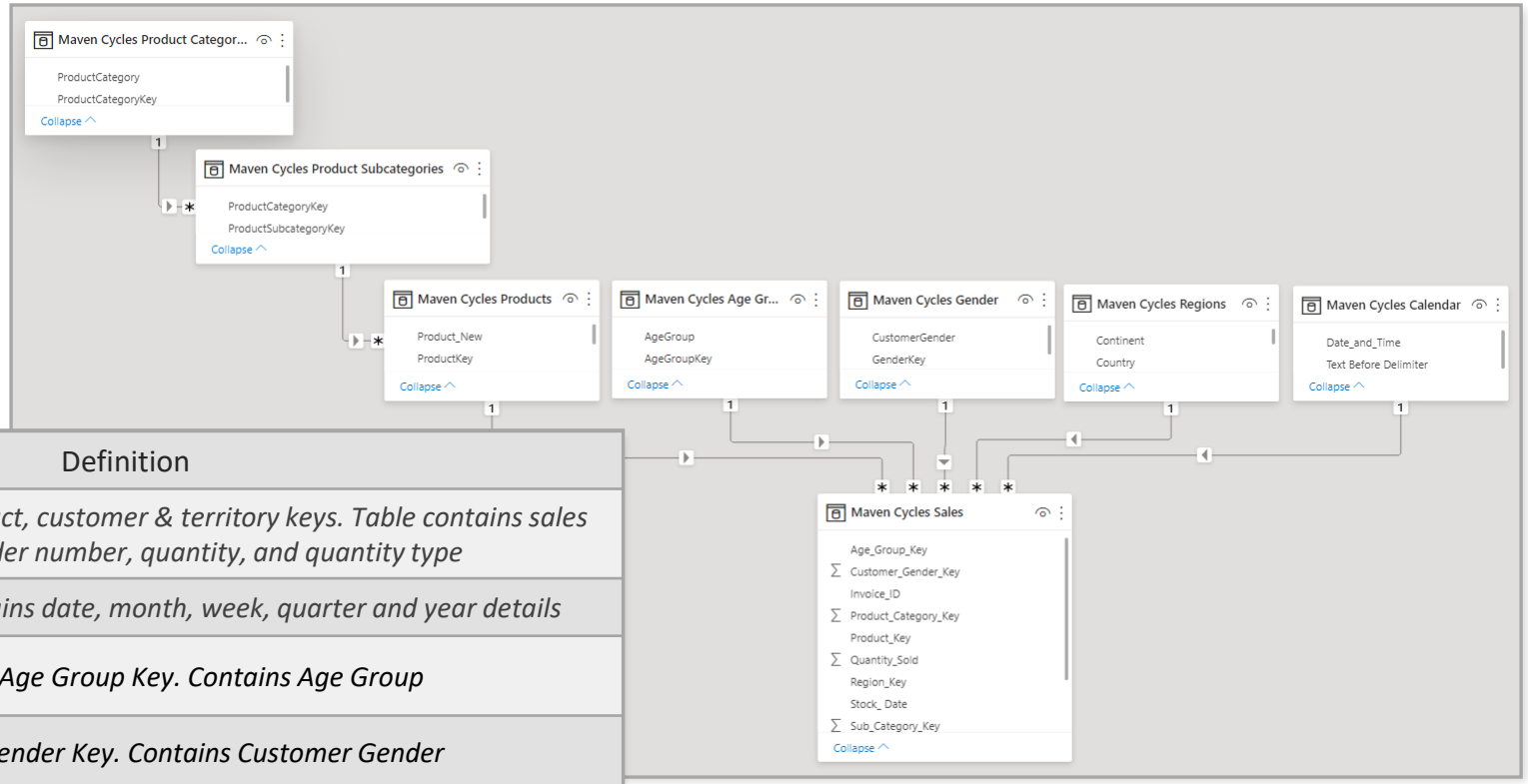


Table Name	Table Type	Definition
Sales	Fact	*Foreign keys are date, product, customer & territory keys. Table contains sales data including order number, quantity, and quantity type*
Calendar	Date	*Primary key is Date. Contains date, month, week, quarter and year details*
Age Group	Lookup	*Primary key is Age Group Key. Contains Age Group*
Gender	Lookup	*Primary key is Gender Key. Contains Customer Gender*
Product Category	Lookup	*Primary key is Product Category Key. Contains Product Category*
Product Subcategory	Lookup	*Primary key is Product Subcategory Key. Contains Subcategory and product category key*
Products	Lookup	*Primary key is Product Key. Contains subcategory key, product, cost, and unit price details*
Regions	Lookup	*Primary key is Sales Territory Key. Contains region, country, continent, territory manager, and manager email details*

SETTING EXPECTATIONS

This course is for users who already have a **basic level of Power BI proficiency**

- *We'll review fundamentals, but you should have a basic understanding of how to import data, create a data model, use DAX formulas, build visuals, and distribute reports through Power BI Service*

Our goal is to help you **ace the Microsoft PL-300 exam**

- *This is not a deep dive into ALL of Power BI's capabilities, but an effective and comprehensive guide to the specific topics covered in the PL-300 exam (formerly DA-100)*

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

- *Power BI features are updated frequently, so tools and interface options may look different or change over time (I'll be using a **Chrome** browser on a **PC/Windows** machine)*

You'll need a **compatible email address** to create a Power BI Service account

- ***NOTE:** You must use a **work or school email address** to access Power BI Service, and some features are only accessible with a PRO or Premium Per User license (60-day free trial)*

INTRO TO THE PL-300 EXAM

EXAM STRUCTURE

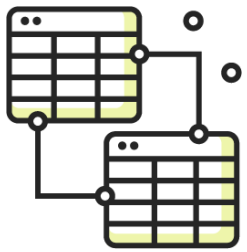
Platform	Pearson VUE (remote connection)
Question Format	Active Screen, Build List, Drag & Drop, Hot Area, Multiple Choice, Case Studies
Number of Questions	~40 - 60
Time Limit	100 minutes (120 minutes seat time)
Passing Score	70% (700/1000)
Expiration	1 Year* **Annual renewal required*
Cost	~$165 USD* **Price is based on the country in which the exam is proctored*

Exam duration & question types: ***https://docs.microsoft.com/en-us/learn/certifications/exam-duration-question-types***

SKILLS MEASURED

Prepare the Data

- Get data from different sources
- Clean, transform, and load data



Model the Data

- Design a data model
- Develop a data model
- Create model calculations with DAX
- Optimize model performance

Visualize & **Analyze** the Data

- Create reports
- Create dashboards
- Enrich reports for usability
- Enhance reports for usability & storytelling
- Identify patterns & trends

Deploy & **Maintain** Assets

- Manage files & datasets
- Manage workspaces

EXAM QUESTION TYPES

Knowledge-Based Questions

Evaluate Power BI product knowledge related to process, formulas, functions, modeling, analysis, etc.

Question types include:

- Active screen
- Build list
- Drag and drop (from a list of options)
- Hot area (hot spot)
- Multiple choice

HEY THIS IS IMPORTANT!

The exam doesn't include any questions where you use or interact with Power BI

Case Study Questions

Multiple choice questions which require you to derive the correct answer based on a provided case

Each case is divided into the following parts:

- Introduction & how to start
- Overview
- Existing environment
  - Data available
  - Data structure & format
  - Data concerns
- Reporting requirements
- Question

EXAMPLE: KNOWLEDGE-BASED QUESTION

Multiple Choice

From the Power Query editor, what data profiling tool (Data Preview) allows you to only see the **percentage of empty values** in each column?

a. Monospaced

b. Column quality

c. Column distribution

d. Column profile

Build List

You are building a report from a SQL Server database and need to import database tables into Power BI Desktop. What are the **first three actions** you should take?

Actions

- Set Data Connectivity mode to Import
- Get data -> SQL Server
- Verify credentials & connect
- Select tables and press Load
- Enter Server & Database names

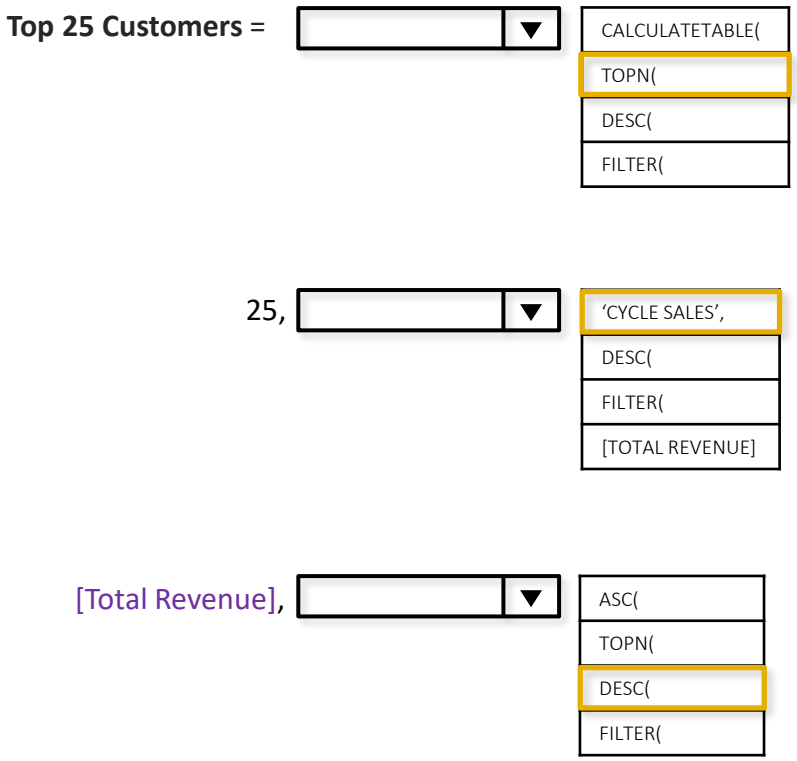
Answer Area

1. Get data -> SQL Server
2. Enter Server & Database names
3. Set Data Connectivity mode to Import

EXAMPLE: KNOWLEDGE-BASED QUESTION

Hot Area (Hot Spot)

How would you complete the following DAX calculated table expression, so it returns a table of the top 25 customers based on Total Revenue?



EXAMPLE: CASE STUDY

Overview

You've just been hired as a Web Analytics Manager for Maven Toy Emporium, an ecommerce toy shop, to help build a faster and more efficient reporting process. Currently, it takes multiple analysts' weeks to create quarterly business reports. You've been asked to help build an end-to-end Power BI solution to help the company streamline reporting and grow the business.

Sales Data

You have access to Maven Toy Emporium's entire online data repository, which includes information about products, stores, sales, customers, inventory, and projections →

Source	Table Name	Column Name	Data Type	Example
Azure SQL Database	Sales	Sale_ID	INT	15
		Date	Date	2021-07-28
		Product_ID	INT	31
		Product_Name	String	Teenie Genies
		Unit_Sales	INT	1
		Region_ID	INT	3
		Customer_ID	INT	101
		Stock_Date	Date	2020-09-06
	Inventory	Store_ID	INT	5
		Product_ID	INT	18
		Stock_Date	Date	2018-02-10
		Stock_On_Hand	INT	175
Dataverse	Customers	Customer_ID	INT	101
		First_Name	String	Adam
		Last_Name	String	Juan
		Email_Address	String	ajuan@mavencustomer.toy
		Billing_Address	String	555 Main St, Somewhereville
		Purchased_Last_30_Days	Boolean	Yes
	Products	Date_Time_Created	Date/Time	2015-05-31 11:23:00
		Product_ID	INT	38
		Product_Name	String	Action Figure
		Product_Category	String	Toys
		Cost	Currency	$9.99
		Price	Currency	$15.99
SharePoint Online	Projections	Product_ID	INT	19
		Forecast_Sales	INT	300,000
		Month_Start_Date	Date	2022-03-01
	Regions	Region_ID	INT	4
		Region_Name	String	Central

EXAMPLE: CASE STUDY

Concerns

You have no concerns about the structure or quality of the data

Reporting Requirements

Maven Analytics has identified the following technical requirements:

- Leadership team wants to see sales by region
- Regional sales managers need to see sales and returns at different date granularities
- Regional sales managers must be able to see the data for their assigned region only
- Leadership and regional sales managers need to analyze sales performance against projections
- Sales is requesting reports showing the stock on hand

Question #1

What is the best way to distribute the reports to the Leadership team?

A. Share the reports with each team member

B. Add the leadership team as members to the Workspace

C. Create and publish the reports via an App

D. Publish the reports to Web

EXAM RULES (REMOTE)

Power BI certification exams have **specific rules** which are actively enforced by proctors; any violation can result in disqualification, so be mindful!

Photo ID
(GOVT issued)

Clear Desk

Pass Tech Checks

No phones, headphones, external monitors

Close all programs

No Talking
(except proctor questions)

Browser Open

Alone in room

EXAM RULES (IN-PERSON)

Power BI certification exams have **specific rules** which are actively enforced by proctors; any violation can result in disqualification, so be mindful!

Two Forms of ID Required
(GOVT issued + 1 additional)

No Personal Items
(phones, bags, notes, watches, wallets, etc.)

No Talking
(except proctor questions)

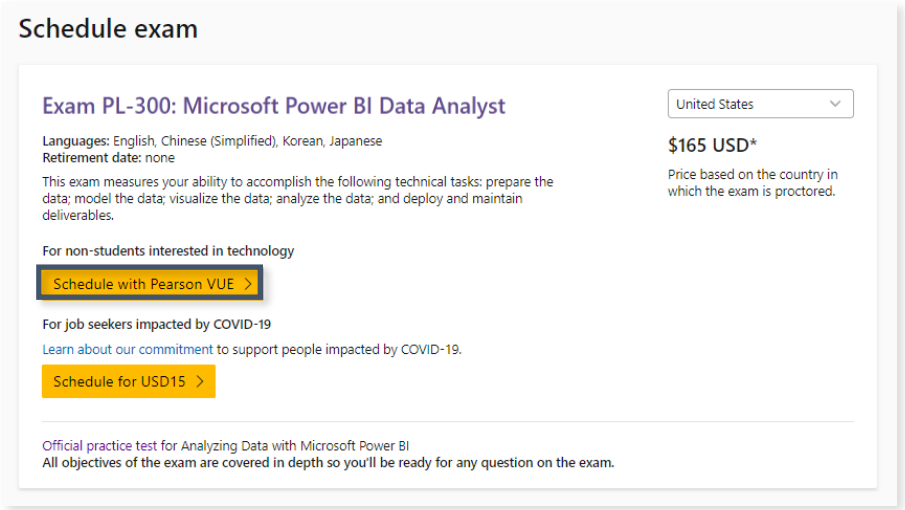
Enhanced security protocol
(digital photo & digital signature)

No Hats or hooded shirts

Arrive 15 Minutes **Early**
(more than 15 min late not admitted)

SCHEDULING THE EXAM

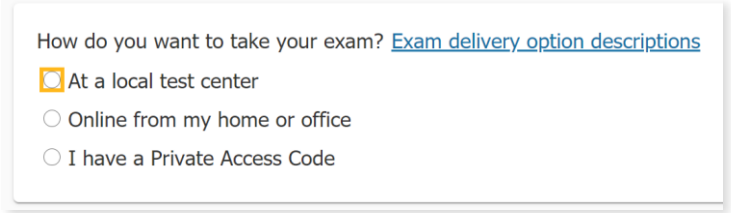
1 Head to the **PL-300 site*** and create an account



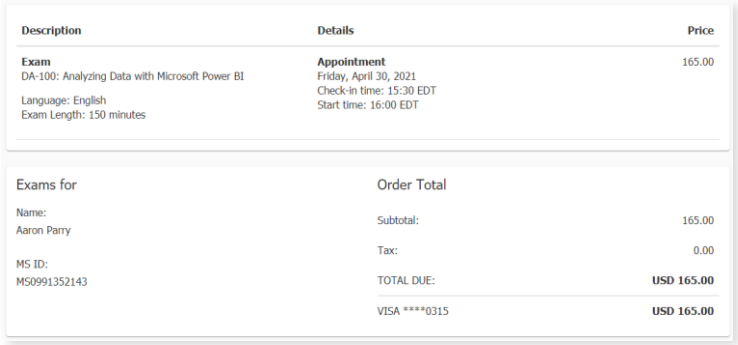
**Make sure account name matches your ID*

2 Select the **exam language & details**

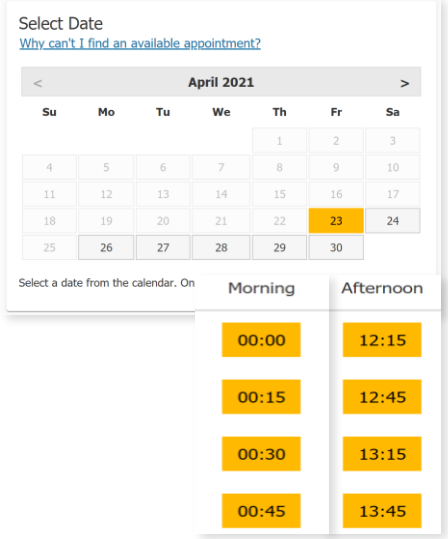
In-person or remote



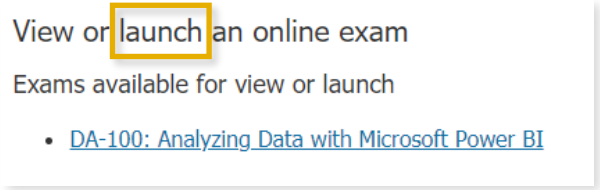
Payment



Date & time



3 **Launch the exam** on testing day!



PRO TIP: To avoid surprises, take 5 minutes to **run a system test** before your remote exam day to make sure your computer and internet speed is sufficient

** https://docs.microsoft.com/en-us/learn/certifications/exams/pl-300*

EXAM RESULTS

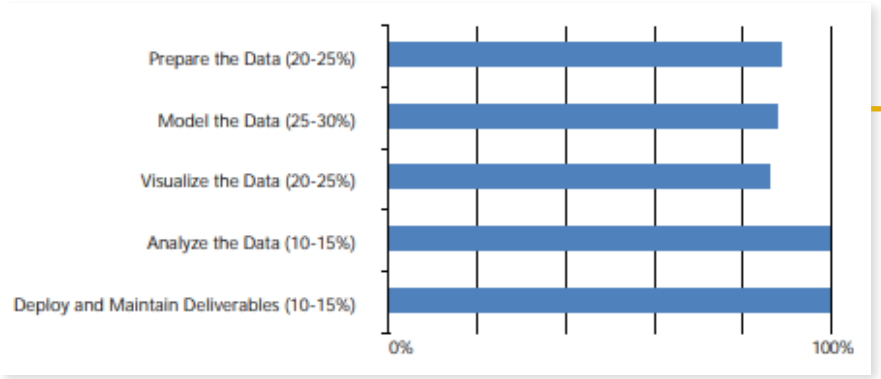
DA-100
Analyzing Data with Microsoft Power BI

Candidate: Aaron Parry
Candidate ID: MS0991137167
Result: PASS
Passing Score: 700
Your Score: 922

Date: Friday, April 16, 2021
Site Number: 82028
Registration: 394861362

BASIC EXAM INFO
Exam details including name, id, result, and score

EXAM DETAILS
Exam type, location, and registration information



PERFORMANCE BY EXAM SECTION
Bar chart that represents your section-level performance
(Bar length cannot be used to calculate number of questions answered correctly)

PERFORMANCE COMPARISON
How your performance compares to others who have taken the PL-300 exam

Content Area	Opportunity for Improvement	Comparable	Strength
Prepare the Data (20-25%)			✓
Model the Data (25-30%)			✓
Visualize the Data (20-25%)		✓	
Analyze the Data (10-15%)			✓
Deploy and Maintain Deliverables (10-15%)			✓

*Detailed score report is available from the **Pearson Vue** site*

RENEWAL PROCESS

The PL-300 certification must be **renewed annually** in order to remain active

- This is designed to promote continuous learning and help keep you up to date with technology

Renewal Information

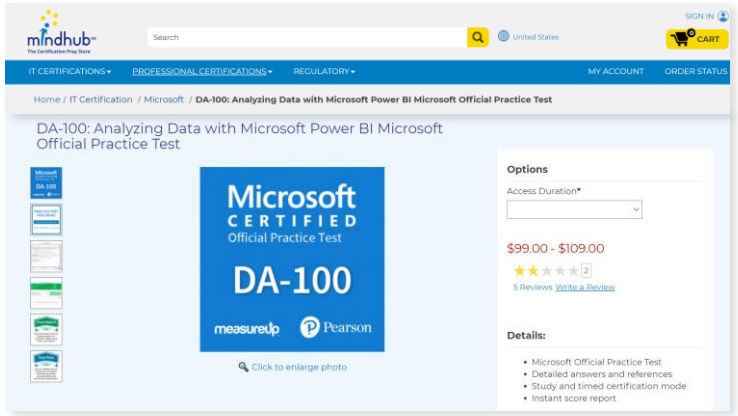
- Role-based certifications **expire every year**
- **Online renewal** process via Microsoft Learn
- Renewal window opens **6 months prior to expiration**
- Renewal covers **technology updates** only
- **No cost** *(free annual renewal!)*
- Renewal exam is **not proctored**
- **Scenario-based** multiple-choice questions

Renewal Process

1. **Connect** Microsoft Learn profile with certification profile
2. **Study** with the available learning modules *(Microsoft Learn)*
3. **Take** renewal assessment and pass before certification expires

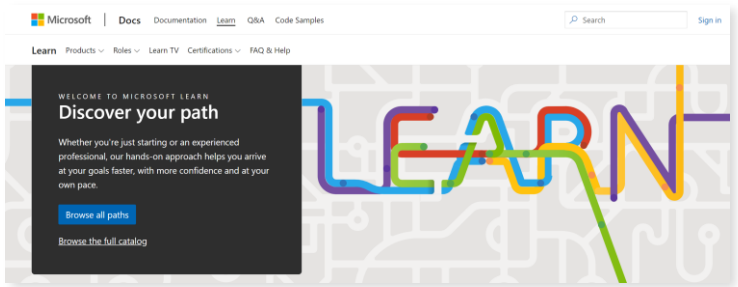
Certification Renewal: ***https://docs.microsoft.com/en-us/learn/certifications/renew-your-microsoft-certification***

HELPFUL RESOURCES

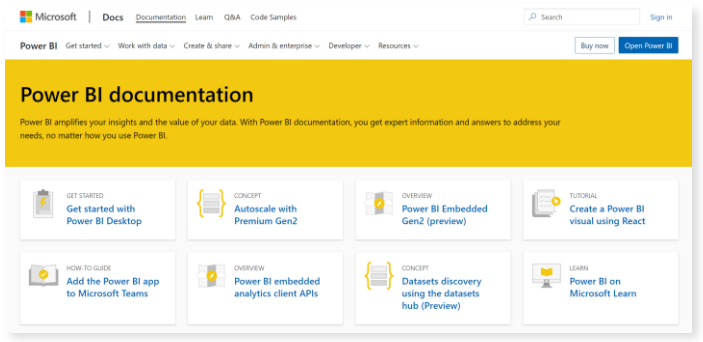


MeasureUp offers the official Microsoft Power BI practice test, which includes practice questions, detailed answers & references, study & certification testing modes, and an instant score report

(mindhub.com/da-100-analyzing-data-with-microsoft-power-bi-microsoft-official-practice-test/p/MU-DA-100)



Microsoft Learn offers on-demand or instructor-led learning paths tailored to the PL-300 (*docs.Microsoft.com/learn*)



Power BI Documentation provides detailed information for entire ecosystem (*powerbi.microsoft.com/blog*)

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

TIPS FOR SUCCESS

Learn by doing – use Power BI every day

- *Nothing beats on-the-job experience and hands-on practice with Power BI, so use it often!*

Use your resources, and study how you learn best

- *Leverage resources available to you, including the official practice test & Microsoft Learn*

Remember the rules of the exam *(don't fail on a technicality!)*

- *Be sure you follow all rules based on your exam setting (ID, no talking, clean environment, no phones)*

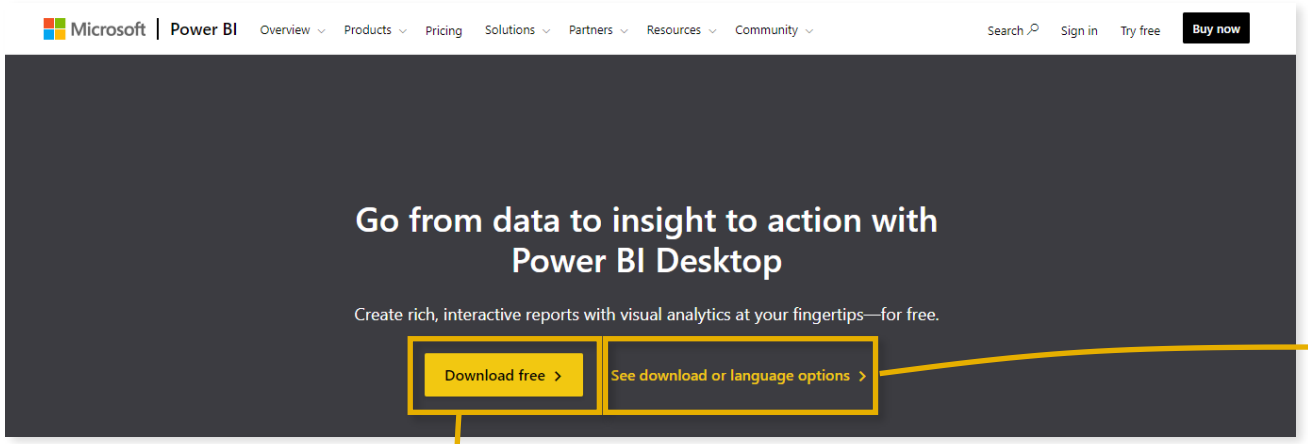
Skip questions if you get stuck, and revisit them later

- *When allowed, move on to familiar questions before returning to work through harder ones*

SETTING UP DESKTOP & SERVICE

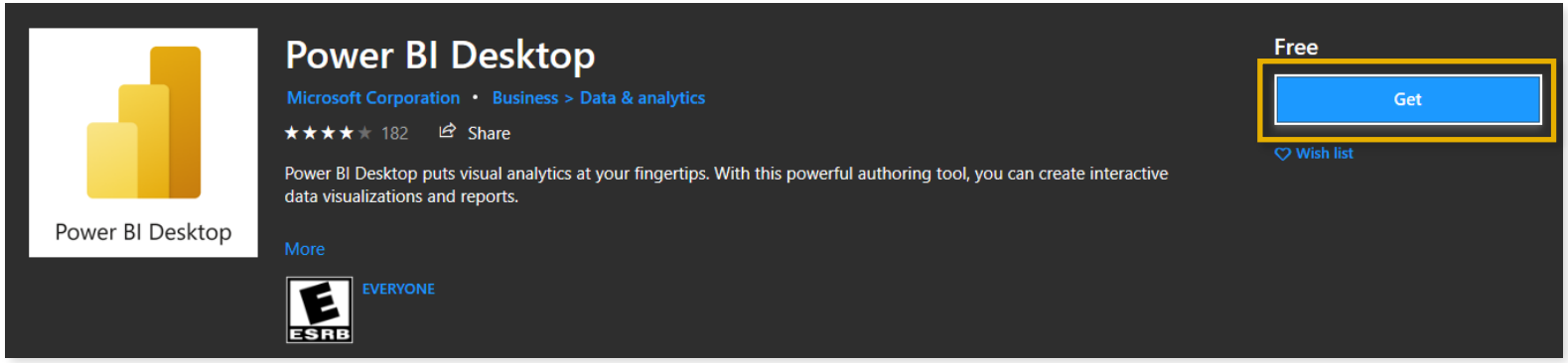
INSTALLING POWER BI DESKTOP

1 Head to **powerbi.microsoft.com/desktop** and click "**Download free**"



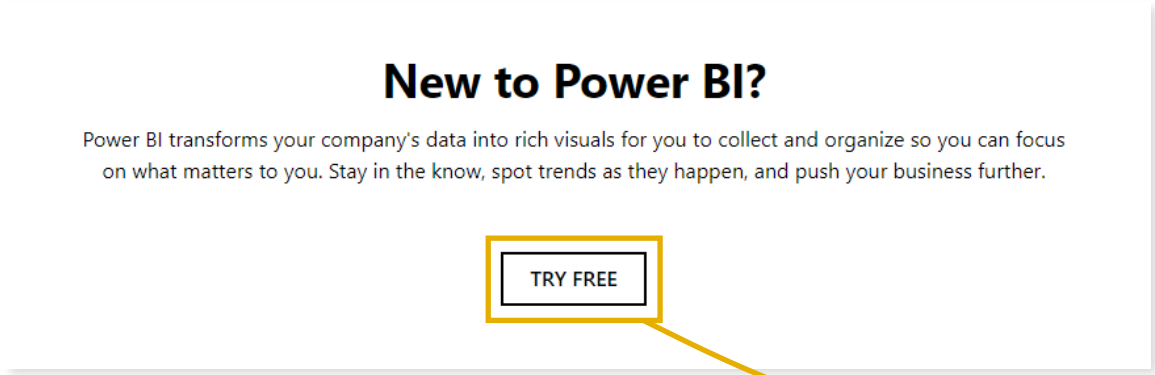
*Use **See download or language options** to update Power BI Desktop*

2 Click on "**Get**" to download Power BI Desktop from the Microsoft Store

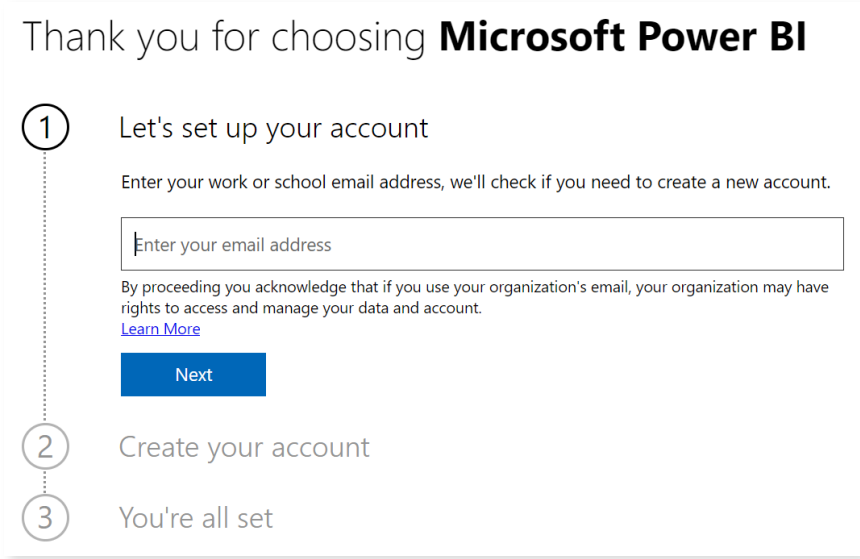


CREATING A POWER BI SERVICE ACCOUNT

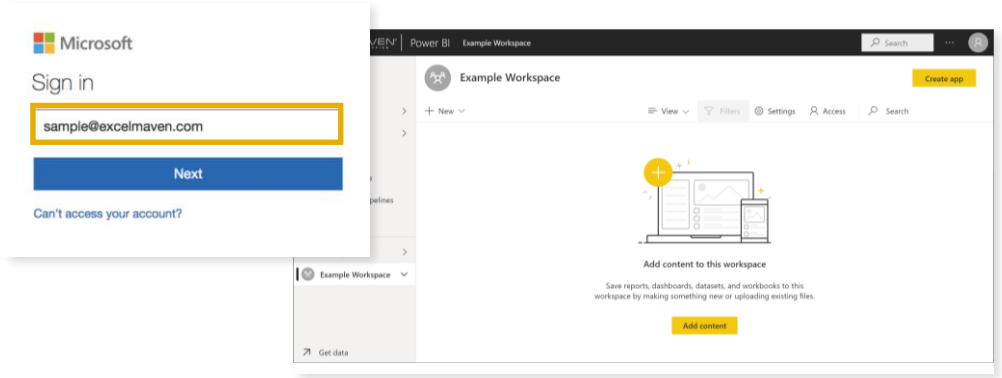
1 Head to **powerbi.microsoft.com/landing/signin** and click "**TRY FREE**"



2 **Create** & **verify** your account



3 Navigate directly to **app.powerbi.com** to log in

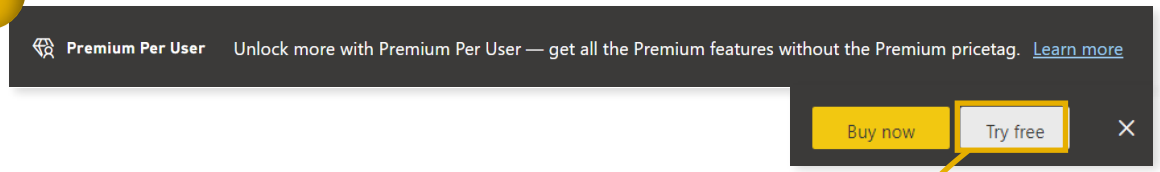


HEY THIS IS IMPORTANT!

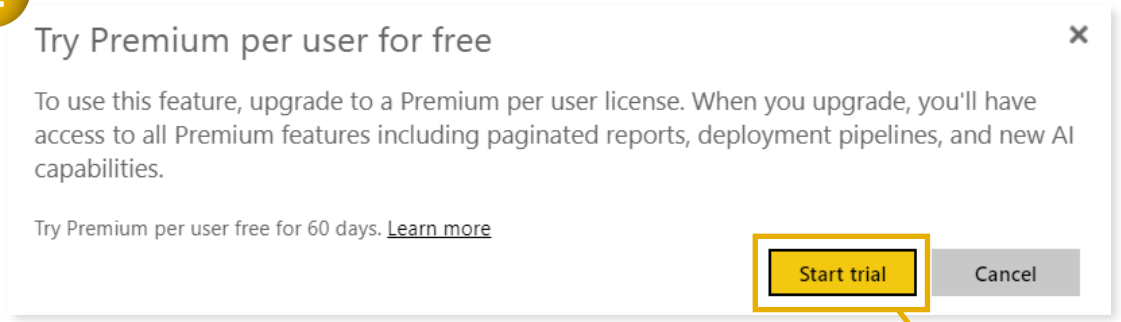
Microsoft requires a **work** or **school** email address, so domains like @gmail, @yahoo or @outlook will not be accepted

ACTIVATING A PREMIUM PER USER TRIAL

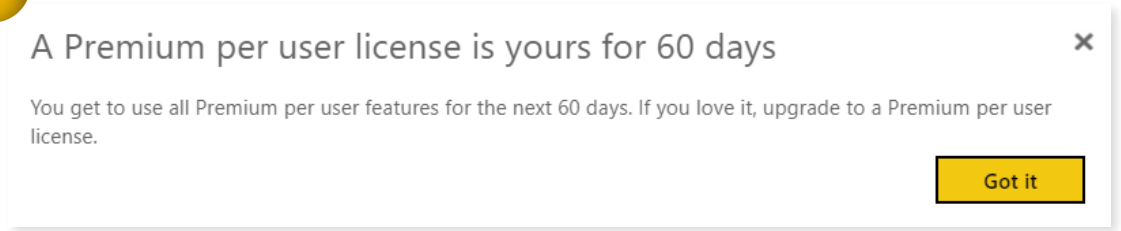
1



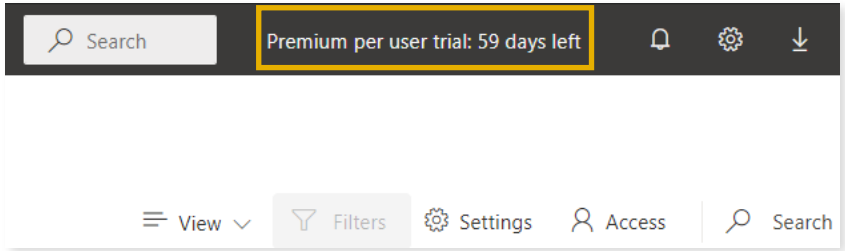
2



3



Your **trial period** lasts for **60 days**, and you'll see your remaining time in the header bar each time you log in



HEY THIS IS IMPORTANT!

You are activating a **Premium per User (PPU) trial**

All Pro features are included with a PPU trial, and we'll focus on additional features included with PPU later in the course

Refer to Microsoft's site for post-trial pricing: ***powerbi.microsoft.com***

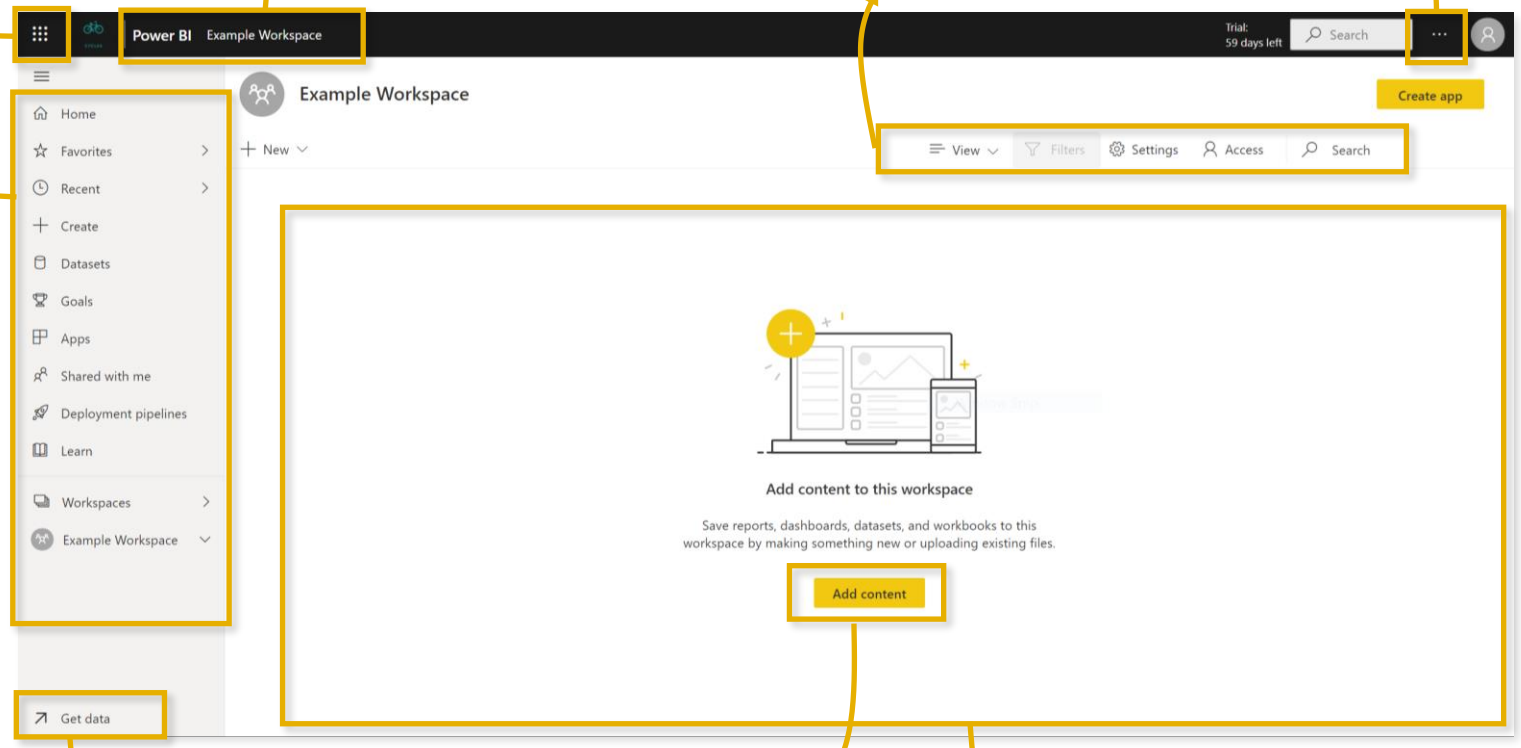
THE POWER BI SERVICE (APP) INTERFACE

OFFICE 365 APP LAUNCHER
Email, OneDrive, Calendar, etc.

NAVIGATION PATH
Highlights the active workspace, app, report or dashboard

WORKSPACE OPTIONS
Change view, update settings, manage access & search

SETTINGS MENU
Quick links for settings, help, feedback, downloads etc.



NAVIGATION PANE
Key Power BI views, including Favorites, Apps, Workspaces, etc.

GET DATA
Import, create or discover new data connections & content

CONTENT CANVAS
Displays all your datasets, visuals, reports, dashboards, and workbooks

WORKSPACES

There are *two types* of workspaces in Power BI Service: **My Workspace** & **Workspaces**

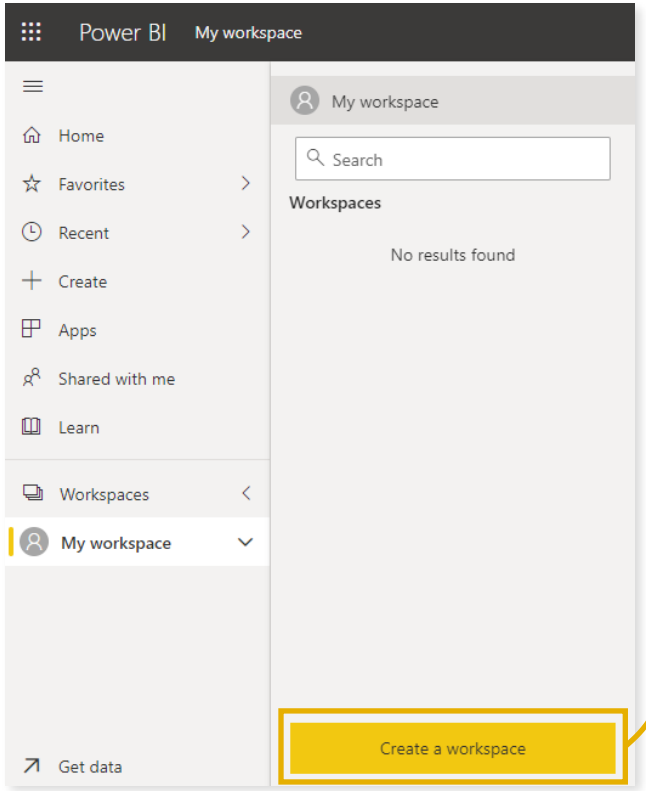
My Workspace

- **Personal workspace** for a single user (*Free or paid license*)
- **Only you** can access its content (*can't collaborate with others*)
- Content can be shared with **individual Pro/PPU users** *(shared with me)*
- Contains core **building blocks** (*datasets, workbooks, reports, and dashboards*)

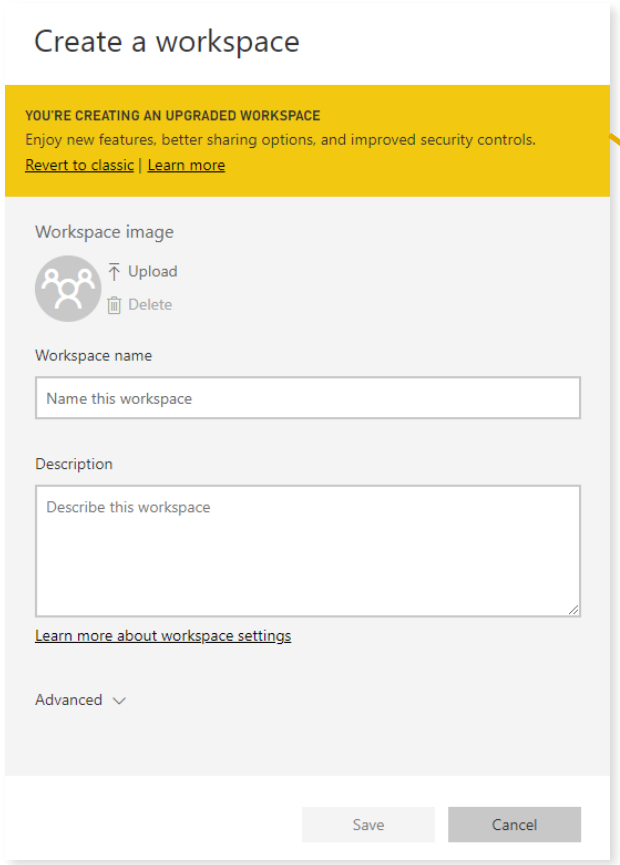
Workspaces

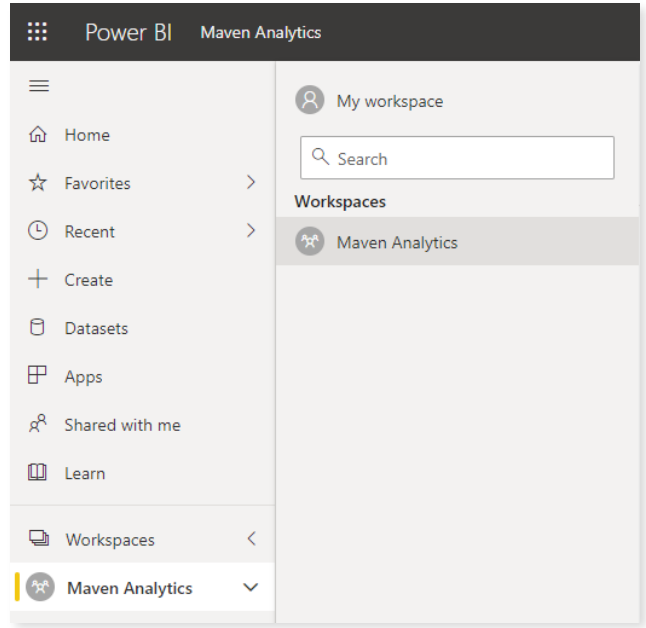
- **Shared workspace** for many users (*with paid licenses*)
- **Multiple users** can access & collaborate on content
- Content can be created & shared **across your organization**
- Contains core **building blocks** (*datasets, workbooks, reports, and dashboards*) plus **dataflows**

CREATING A WORKSPACE



*Add a **Name** (required) and optional **Description** for your workspace*





HEY THIS IS IMPORTANT!

We'll use this workspace later in the course when we publish a report to Power BI Service

MEASURED SKILLS REVIEW

PREPARING THE DATA

Prepare the Data

- Get data from different sources
- Clean, transform, and load data

Model the Data

- Design a data model
- Develop a data model
- Create model calculations with DAX
- Optimize model performance

Visualize & **Analyze** the Data

- Create reports
- Create dashboards
- Enrich reports for usability
- Enhance reports for usability & storytelling
- Identify patterns & trends

Deploy & **Maintain** Assets

- Manage files & datasets
- Manage workspaces

GETTING DATA FROM DIFFERENT SOURCES

In this section we'll cover **getting data from different sources**, including changing data source settings, selecting storage modes, identifying query issues, and using parameters

TOPICS WE'LL COVER:

- **Connecting to Data**
- **Data Sources**
- **Storage Modes**
- **Parameters**

COMMON QUESTIONS:

- *You need to provide users with a summary version of a sales dataset, which of the following actions should you take and in what order?*
- *You have a large SQL database table and only want to import a sample. Which of the following techniques will help you import a sample of data?*
- *What query editor tool can be used to dynamically change a data source between different servers?*

CONNECTING TO DATA

Connecting to Data

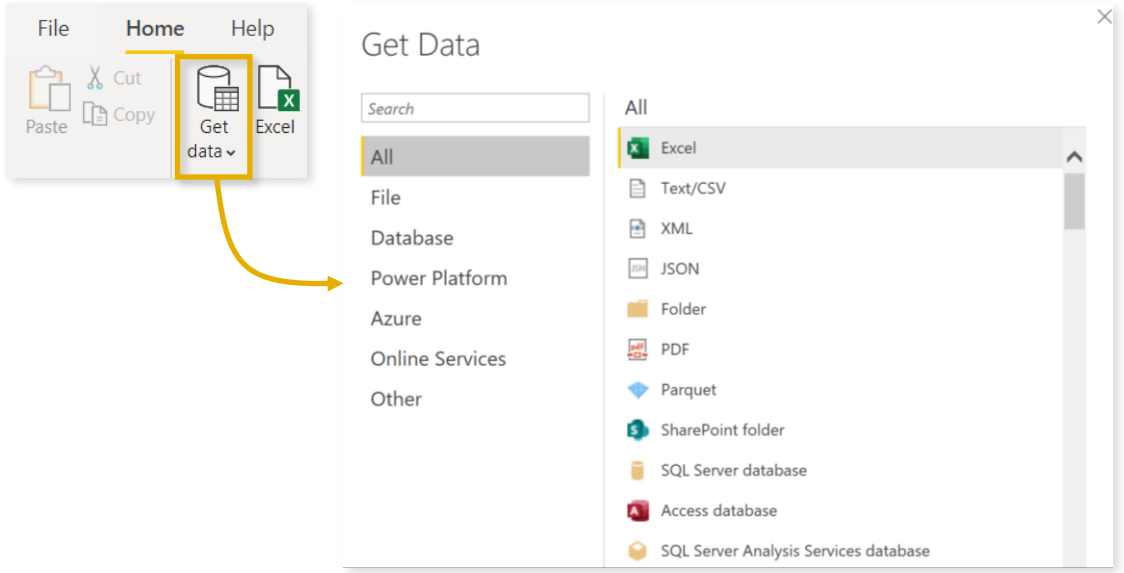
Data Sources

Storage Modes

Parameters

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

- Flat files & Folders (*csv, text, Excel, JSON etc.*)
- Databases (*SQL, Access, Oracle, IBM, MySQL, etc.*)
- Power Platform (Dataflows, Power BI datasets, Dataverse, etc.)
- Azure (SQL Database, Blob Storage, Cosmos DB, Data Lake Storage, etc.)
- Online Services (*SharePoint, GitHub, Dynamics 365, Google Analytics, Salesforce, etc.*)
- Others (*Web feeds, R scripts, Spark, Hadoop, etc.*)



THE QUERY EDITOR (INTRODUCTION)

Once you connect to data, the Query Editor lets you **shape** & **transform** the data to meet your needs, then **load** that model into Power BI Desktop

- **Connecting to Data**
- Data Sources
- Storage Modes
- Parameters

QUERY EDITING TOOLS
Table transformations, calculated columns, etc.

FORMULA BAR
This is "M" code

QUERY PANE

COLUMN PROFILE
Based on first 1,000 rows

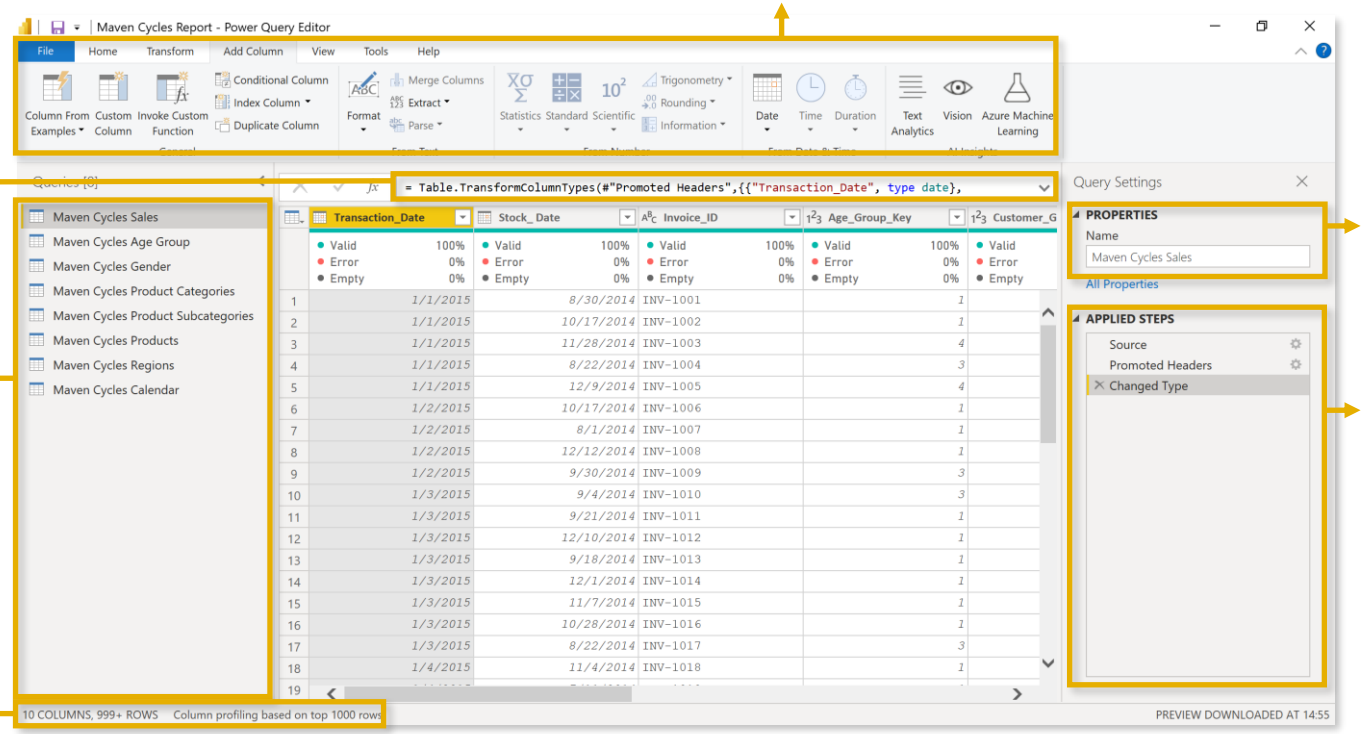


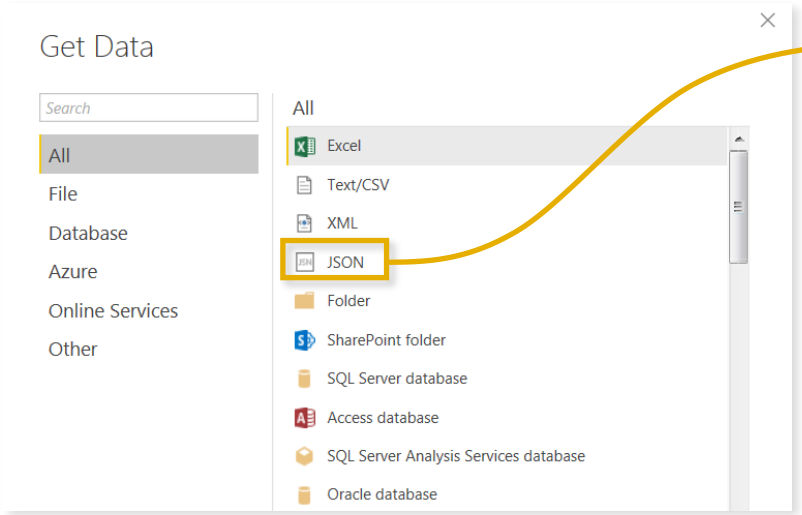
TABLE NAME & PROPERTIES

APPLIED STEPS
Like a macro!

DATA SOURCE: JSON FILE

- Connecting to Data
- **Data Sources**
- Storage Modes
- Parameters

To connect to a **JSON file** or transform a field within a flat file that contains JSON:



1 **Convert** the JSON list to a table

```
= Table.FromRecords({Source})
```

	people
1	List

2 **Expand** the attributes in the list to columns

```
= Table.ExpandRecordColumn(#"Expanded people", "people", {"firstName",
```

	people.firstName	people.lastName	people.gender	people.age
1	Joe	Jackson	male	28
2	James	Smith	male	32

3 **Change** the data type for each column

```
= Table.TransformColumnTypes(#"Expanded people1",{{"people.firstName", type
```

	people.firstName	people.lastName	people.gender	people.age
1	Joe	Jackson	male	28
2	James	Smith	male	32

For more detail visit: ***https://docs.microsoft.com/en-us/power-query/connectors/json***

DATA SOURCE: POWER BI DATA SOURCE FILES (PBIDS)

Power BI Data Source files (PBIDS) contain a single set of pre-wired data source connection settings (no data)

Key Benefits:

- ✓ They make data sources easier to share with other users
- ✓ They are useful for new report designers who are not familiar with available data sources
- ✓ They speed up the "Get Data" process
- ✓ They only support a single data source in one file
- ✓ They can be autogenerated within Power BI Desktop (preferred method) or directly from a text editor

HEY THIS IS IMPORTANT!

Power BI Data Source files do not store user authentication credentials like username and password

DATA SOURCE: POWER BI DATA SOURCE FILES (PBIDS)

Connecting to Data

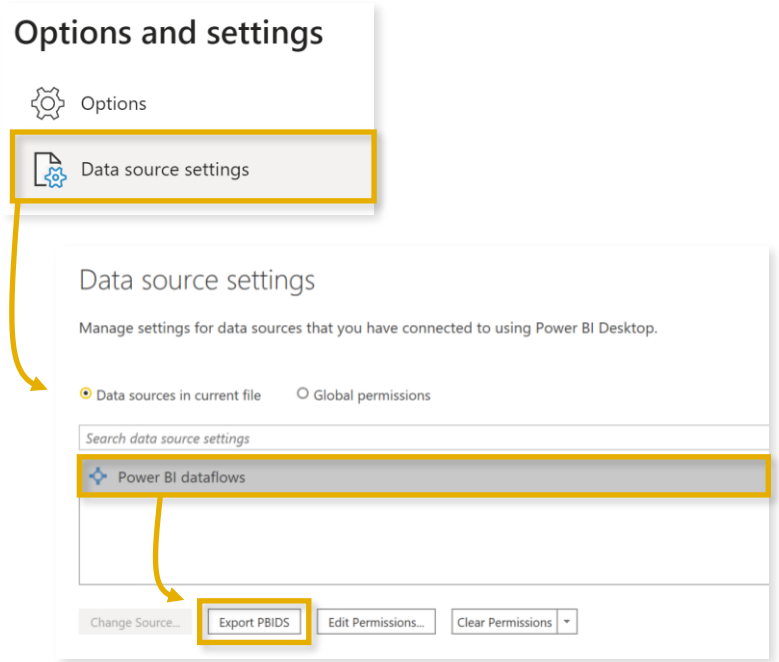
Data Sources

Storage Modes

Parameters

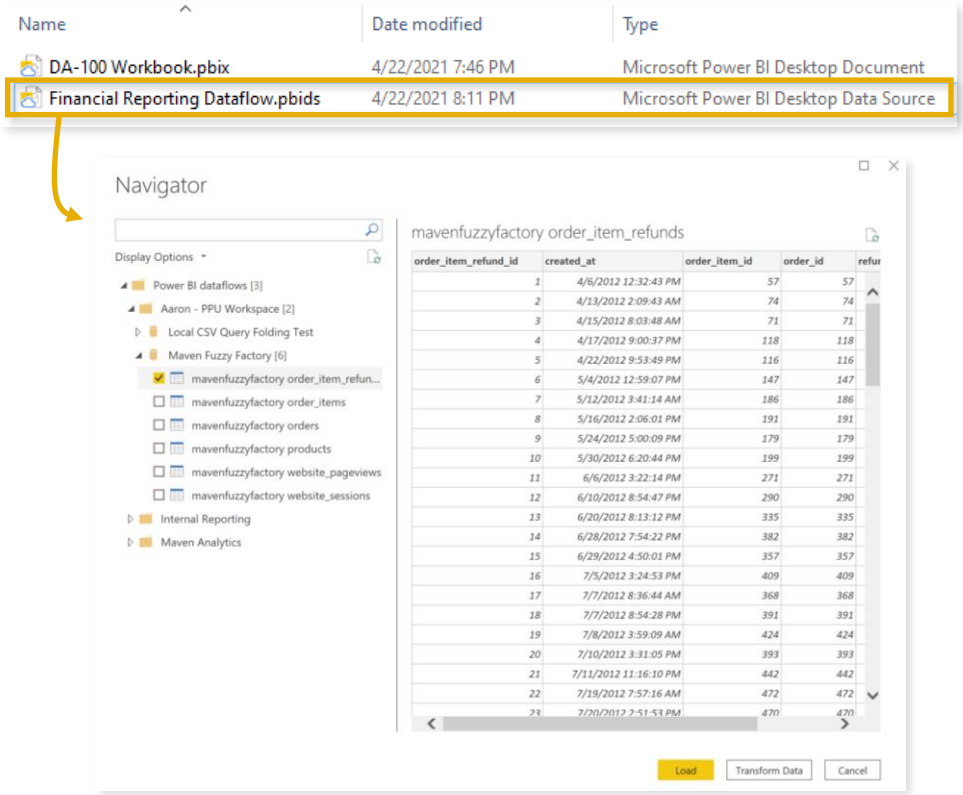
To **create** a PBIDS file:

1. **Open** "Data source settings"
2. **Select** the data source to convert
3. **Export** & **save** as a PBIDS file



To **connect to** a PBIDS file:

1. **Open** the PBIDS file, enter credentials, launch "Data Source Navigator" window



DATA SOURCE: MICROSOFT DATAVERSE

Connecting to Data

Data Sources

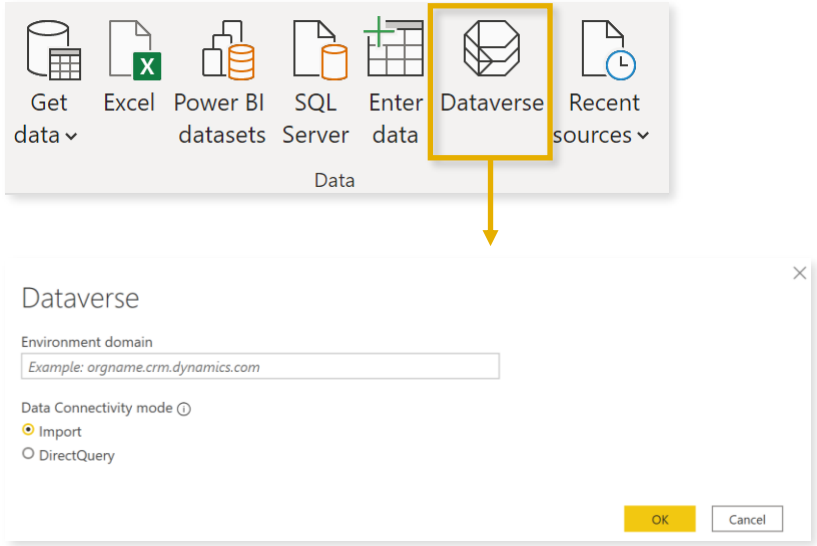
Storage Modes

Parameters

The **Microsoft Dataverse** (Common Data Service) is a cloud-based storage option for your organization's data that you can connect to business applications

Key Benefits:

- ✓ **Easy to manage** – Both the metadata and data are stored in the cloud
- ✓ **Easy to secure** – Users can only see data if they are granted access
- ✓ **Easy to access** – Connect to Power BI, Power Apps, Power Automate, and Power Virtual Agents



HEY THIS IS IMPORTANT!

To connect to Microsoft Dataverse, you'll need to know the server address, which typically has the following format: *https://company.crm.dynamics.com.* And maker permissions to access the portal and read permissions to access data within tables

For more detail visit: ***https://docs.microsoft.com/en-us/powerapps/maker/data-platform/data-platform-powerbi-connector***

DATA SOURCE: SQL SERVER ANALYSIS SERVICES TABULAR (SSAS)

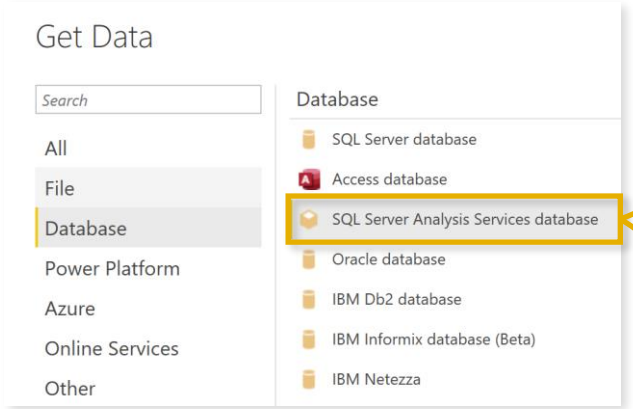
Connecting to Data

Data Sources

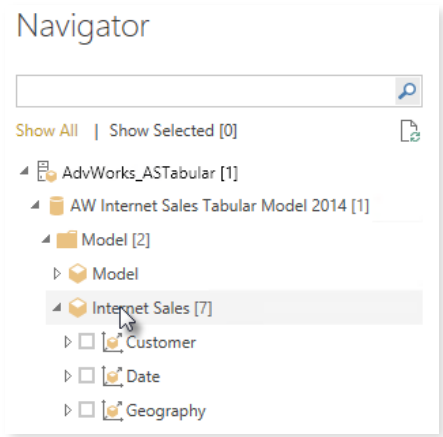
Storage Modes

Parameters

There are two methods to connect SSAS tabular models: **Import** & **Live Connection**

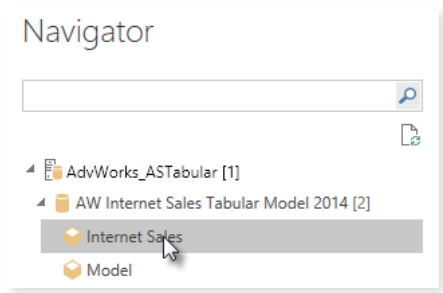


Import



- ✓ Select model or perspective, and then specific table or column
- ✓ Import data into memory
- ✓ Shape data with Query Editor
- ✓ Create & enhance data model

Live Connection



- ✓ Live connection to tabular model
- ✓ Select model or perspective
- ✓ Looking at most recent data

HEY THIS IS IMPORTANT!

If you plan to publish your tabular model to Power BI Service and it's built from a **Live connection**, you must have a **data gateway** installed

For more detail visit: ***https://docs.microsoft.com/en-us/power-bi/connect-data/desktop-analysis-services-tabular-data***

DATA SOURCE: SHAREPOINT ONLINE

Connecting to Data

Data Sources

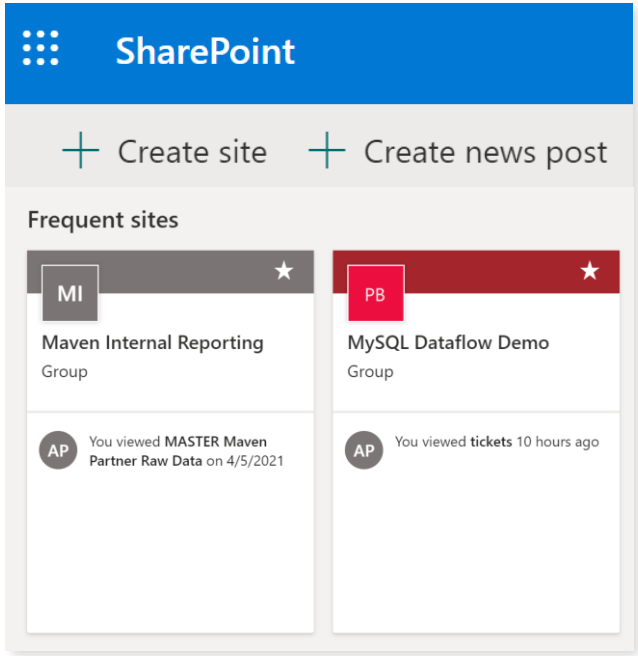
Storage Modes

Parameters

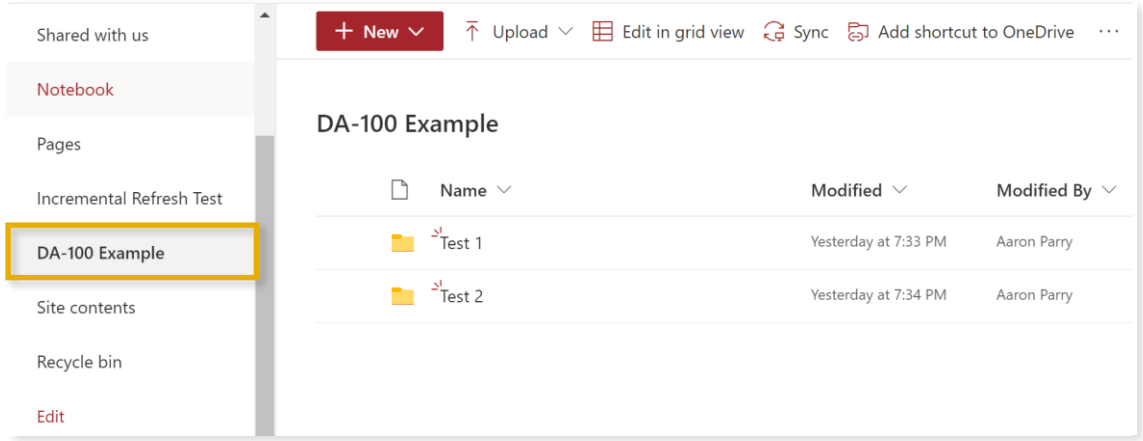
SharePoint Online lets companies organize, share, and access information via **sites**

Sites contain **document libraries**, a special type of folder, that store folders and files

SharePoint Sites



Document Libraries



DATA SOURCE: SHAREPOINT ONLINE

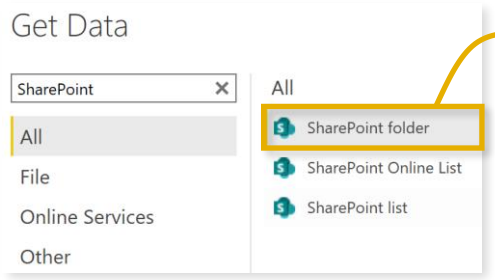
Connecting to Data

Data Sources

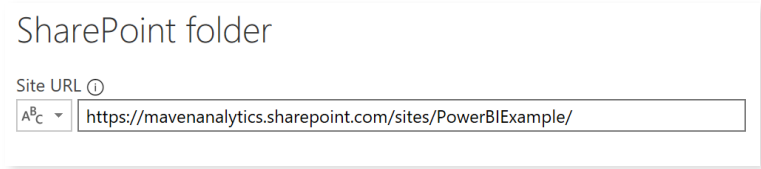
Storage Modes

Parameters

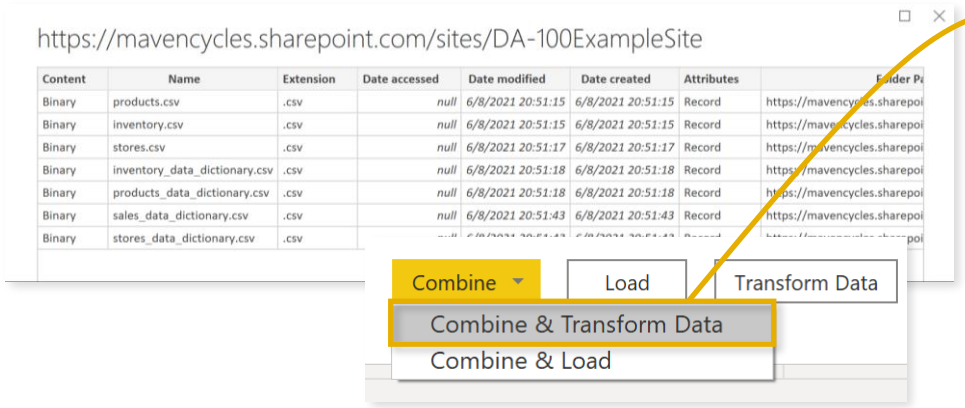
You can get data from a SharePoint Online site by connecting to a **SharePoint folder**



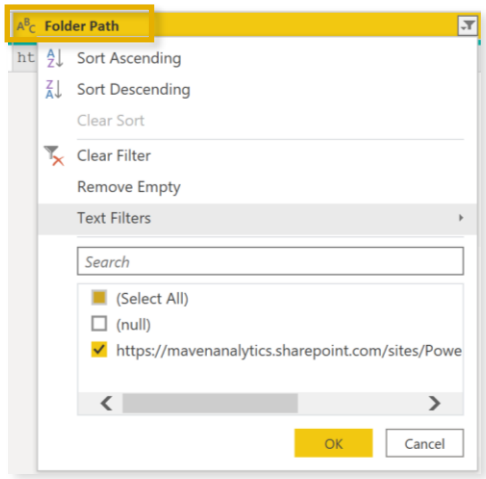
1 **Enter** the site root URL



2 **Combine** & **Transform** the data



3 **Filter** folder path to correct document library



STORAGE MODES

Connecting to Data

Data Sources

Storage Modes

Parameters

Power BI lets you choose between these types of **storage modes** for your data sources:

- **Import:** Tables stored in-memory within Power BI and queries are fulfilled by cached data
- **DirectQuery:** Tables connected directly to the source & queries executed on-demand at the source
- **Dual:** Tables come from in-memory data or by an on-demand query to the data source

Use **DirectQuery** when:

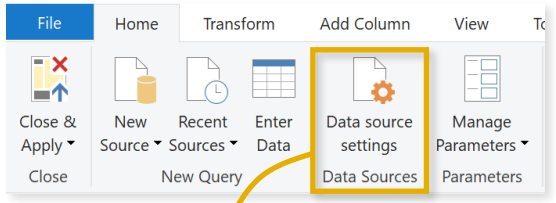
- ✓ Dataset is too large to be stored in-memory
- ✓ Source data changes frequently & reports must show the most recent data
- ✓ Company policy states data can only be accessed from the original source

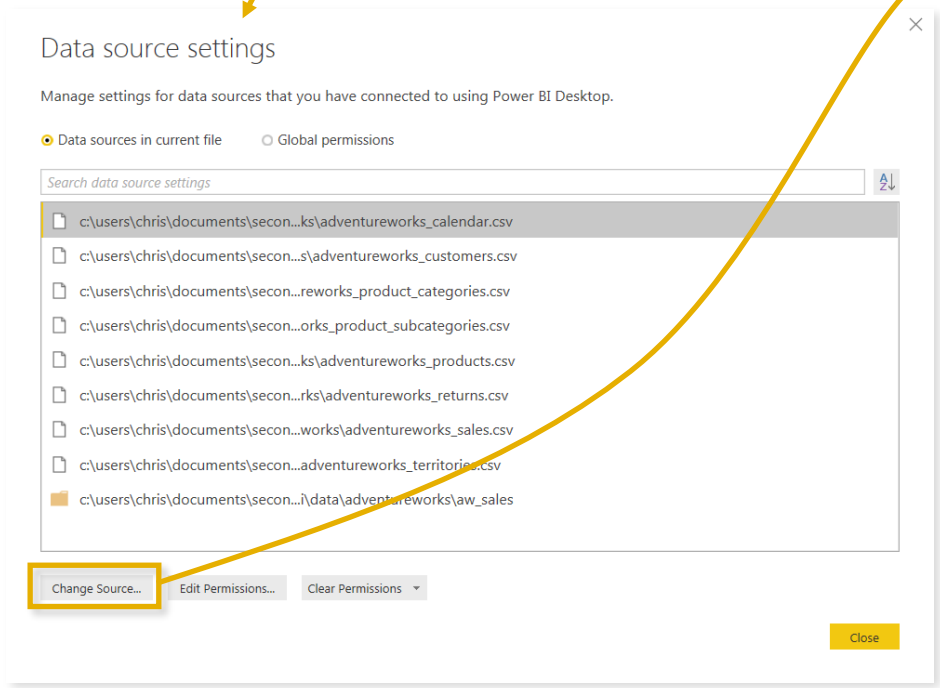
	Imported Data	DirectQuery
Performance	-Best	-Depends on the data source. Generally, slower compared to imported data
Number of Data Sources	-Unlimited	-Unlimited
Data Transformations	-No M transform restrictions	-Limited M transforms -Transforms based on data source language
Data Modeling	-No restrictions on data modeling	-Very restricted -Limited DAX & no calculated tables -Quick Insights not supported
Data Model Size	-Loaded in-memory (increases model size) -Pro: 1GB per dataset -Premium: Capacity based	-Large/frequent volume of data -Does not increase model size -Limited by data source hardware
Data Refresh	-Data only current to last refresh -Pro: 8x per day 30-min intervals -Full refreshes are “expensive”	-Near real-time -Report always shows latest data available
Row-level Security	-User-level role definitions	-User-level role definitions only available for some data sources

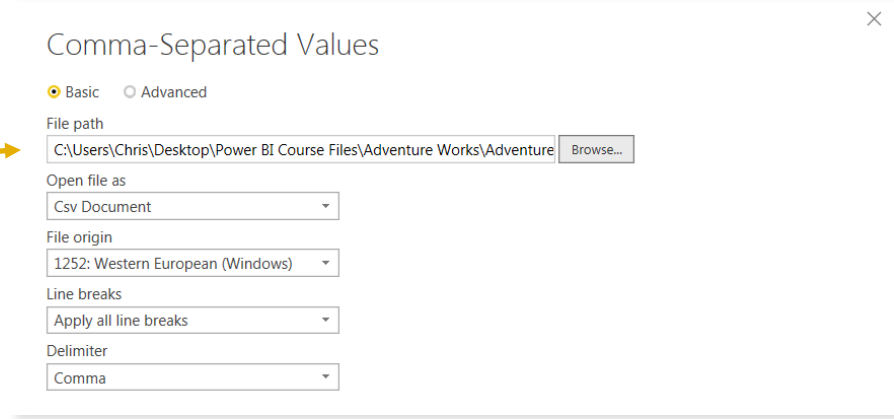
DATA SOURCE SETTINGS

The **Data Source Settings** allow you to manage data connections and permissions

- Connecting to Data
- Data Sources
- Storage Modes
- Parameters







HEY THIS IS IMPORTANT!

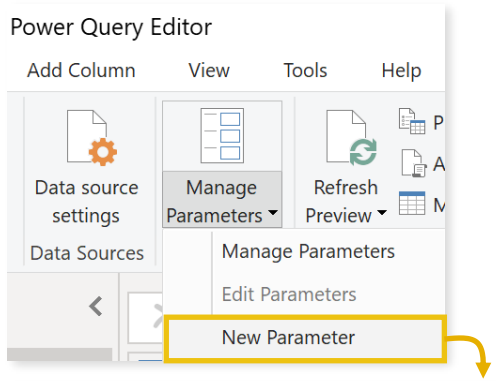
Connections to local files reference the ***exact*** path

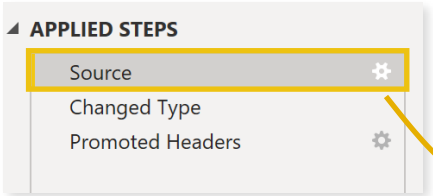
If the file name or location changes, you will need to **change the source** and browse to the current version

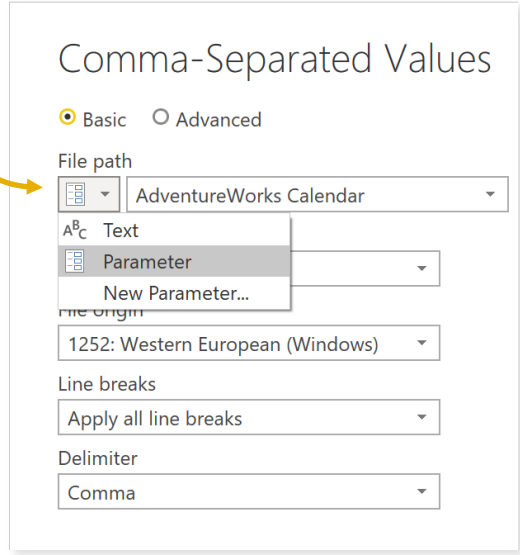
PRO TIP: DYNAMIC SOURCES WITH PARAMETERS

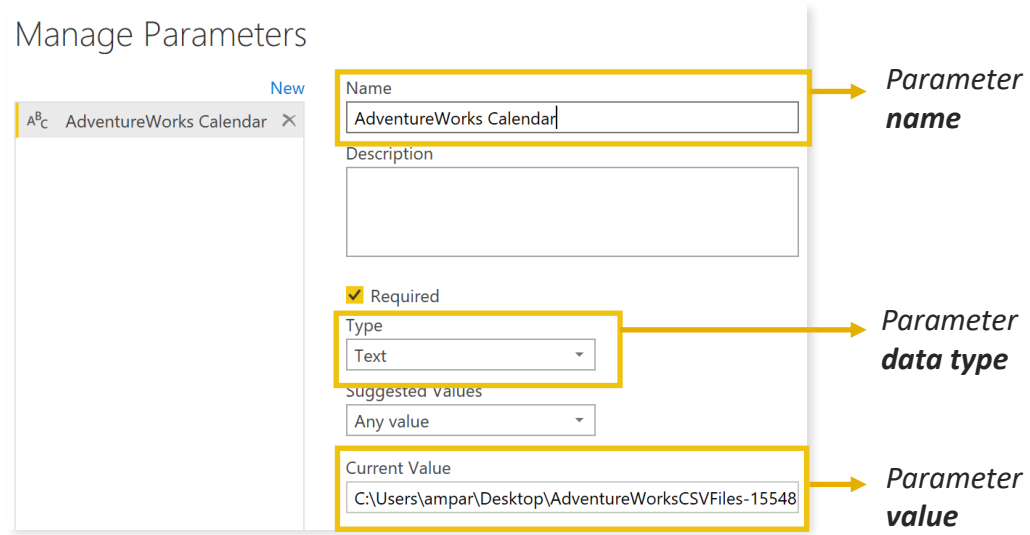
- Connecting to Data
- Data Sources
- Storage Modes
- Parameters

Parameters are a useful way to change data source values dynamically in Power Query









HEY THIS IS IMPORTANT!

These parameters are different from the "What-If" parameters created in the Power BI Desktop front-end

DATA PROFILING

In this section we'll cover **data profiling** with the Query Editor, including identifying data anomalies, examining data structures, and interrogating column properties and statistics

TOPICS WE'LL COVER:

- View Menu
- Column Quality
- Column Distribution
- Column Profile

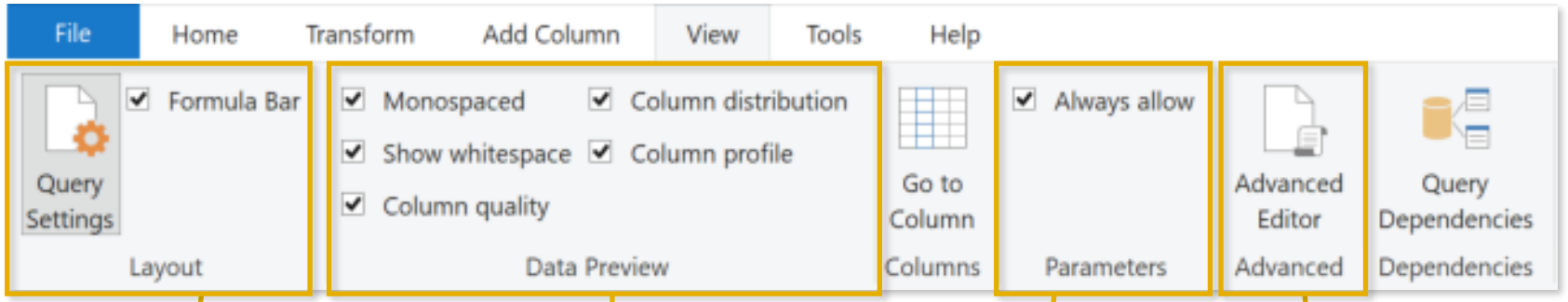
COMMON QUESTIONS:

- *What data profiling tool shows row values as either valid, error, or empty?*
- *As part of data QA, you need to determine if the 'state' column values are defined consistently, what should you do?*
- *From the query editor, you turn on column profile and column distribution, which of the following are true?*

VIEW MENU

- View Menu
- Column Quality
- Column Distribution
- Column Profile

*The **VIEW** tab includes **data preview** and other **advanced editing tools***



*Turn off or on the **Applied Steps** menu and **M code formula bar***

***Data profiling tools** like column quality, distribution, and profile*

*Enable **parameter creation***

*Open the **M code advanced editor***

DATA PROFILING TOOLS (COLUMN QUALITY)

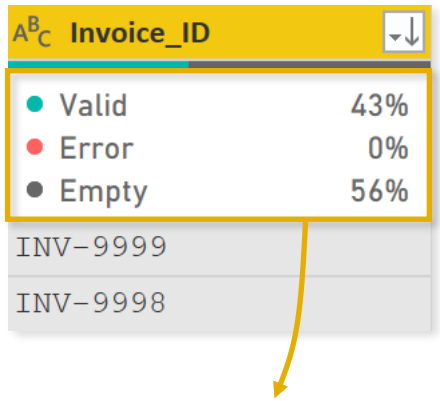
View Menu

Column Quality

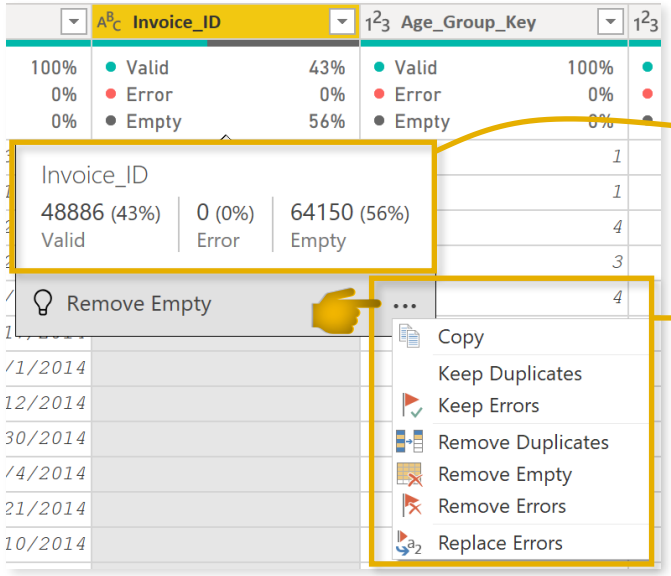
Column Distribution

Column Profile

Data profiling tools like *column quality*, *column distribution*, and *column profile* provide a visual way to **explore data** and get a sense of your **dataset composition**



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



Hover over the column quality box to reveal a ***contextual menu***

Use the ***options menu*** *to clean duplicates, errors & empty values*

PRO TIP: Use the contextual menu to solve **column quality issues** instead of having to right-click or find the right Power Query option

DATA PROFILING TOOLS (COLUMN DISTRIBUTION)

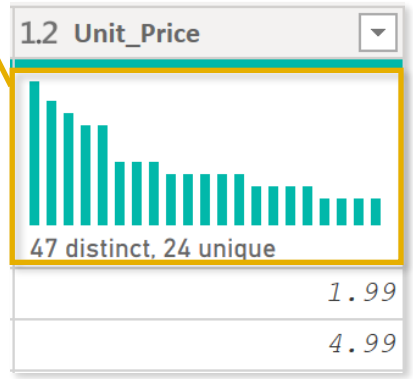
View Menu

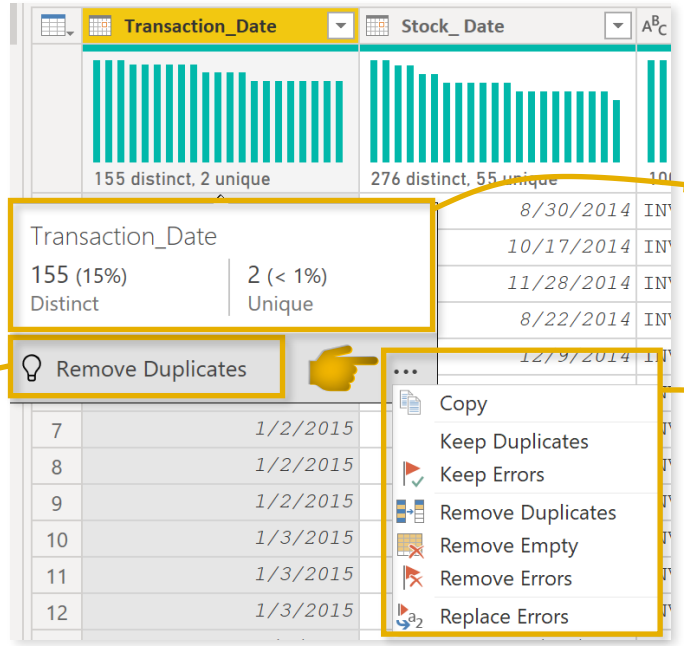
Column Quality

Column Distribution

Column Profile

Column distribution *provides a sample distribution of the data within the column*





Suggested action *based on column distribution results*

Hover over the ***column quality*** *box to reveal a contextual menu*

Use the ***options menu*** *to clean duplicates, errors & empty values*

PRO TIP: Use column distribution to identify **primary keys** within your lookup tables

DATA PROFILING TOOLS (COLUMN PROFILE)

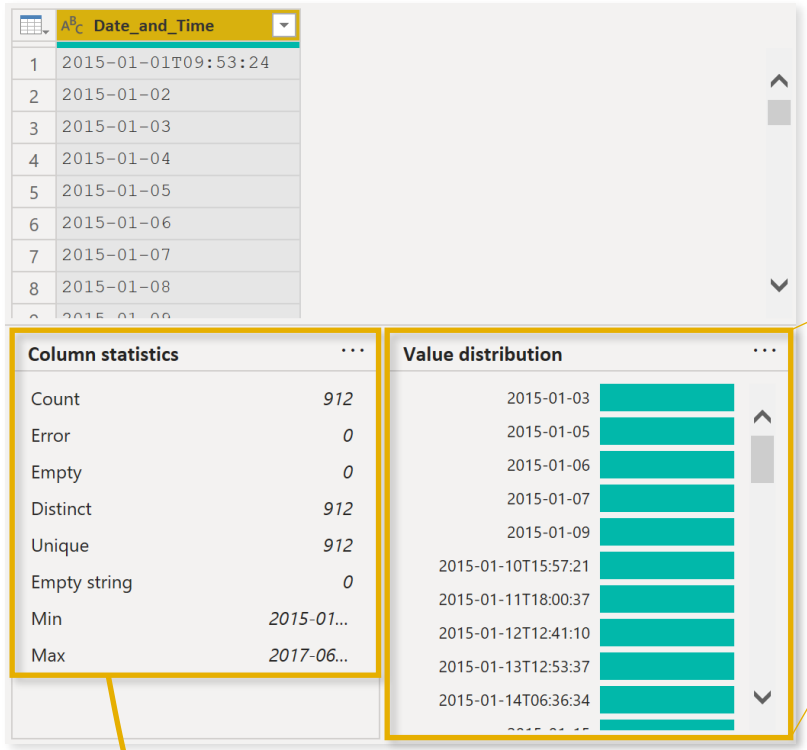
View Menu

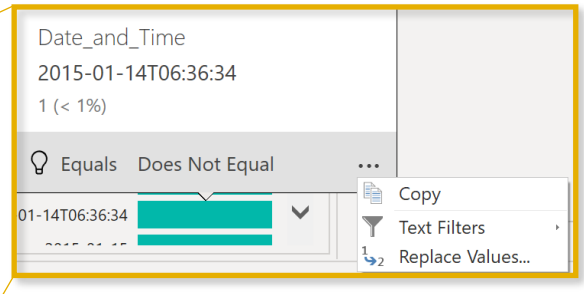
Column Quality

Column Distribution

Column Profile

Column profile *provides a more holistic view of data within a column providing sample distribution of the data and column statistics*





Hover over value distribution bar for ***suggested transformations*** *and more options*

Column statistics *provide a bit more detail than distribution view*

CLEAN, TRANSFORM & LOAD DATA

In this section we'll cover how to **clean, transform, and load data into Power BI**, including identifying keys for joins, shaping tables, combining queries, and modifying M code

TOPICS WE'LL COVER:

- **Cleaning Data**
- **Identifying Keys**
- **Shaping Tables**
- **Combining Queries**
- **Modifying M Code**

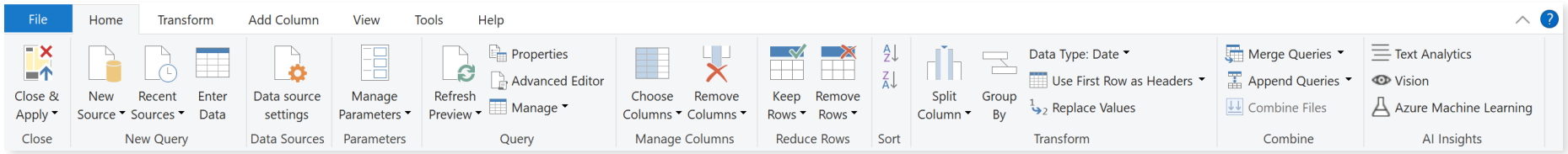
COMMON QUESTIONS:

- *Based on the following M code, what will be the output of the applied step?*
- *From the Query Editor, you want to remove a leading underscore from records in the 'source' column. How would you write the M code?*
- *You import the dataset pictured below. How should you transform the dataset to support period-over-period calculations?*

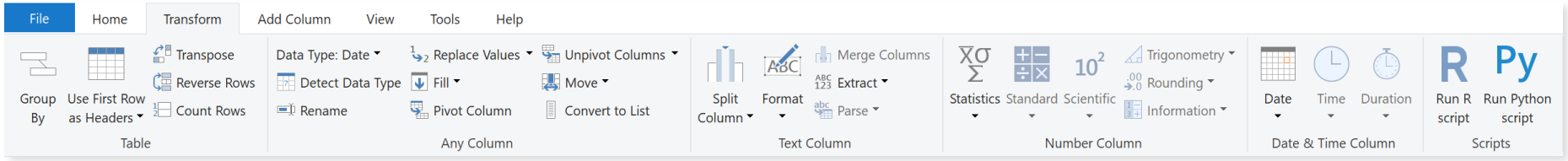
BASIC TABLE TRANSFORMATIONS

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

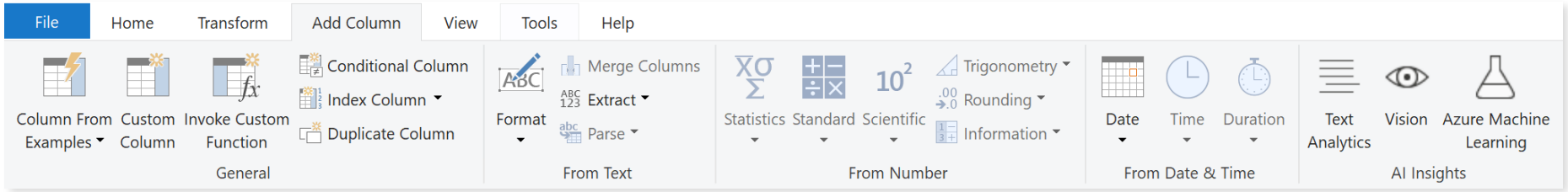
*The **HOME** tab includes **general settings** and **common table transformation tools***



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



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

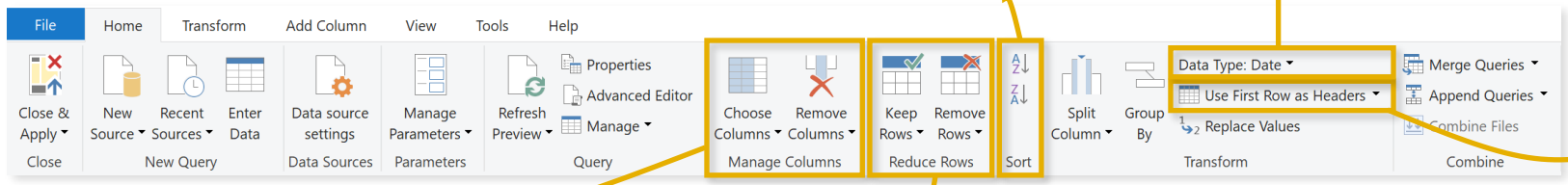


BASIC TABLE TRANSFORMATIONS

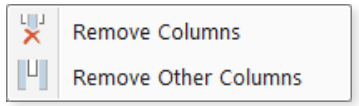
- **Cleaning Data**
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

Sort values (*A-Z, Low-High, etc.*)

Change data type (*date, $, %, text, etc.*)

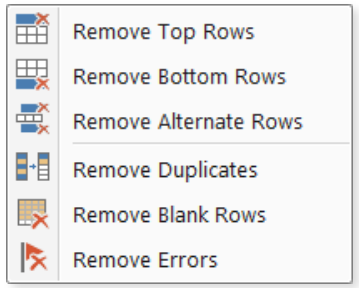


Promote header row



Choose or remove columns

Tip: *use the "Remove Other Columns" option if you always want a specific set*

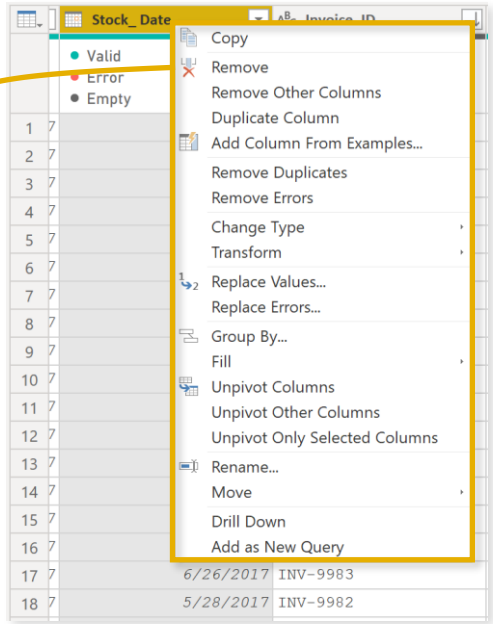


Keep or remove rows

Tip: *use the "Remove Duplicates" option to create a new lookup table from scratch*

Duplicate, move & rename columns

Tip: *right-click the column header to access common tools*

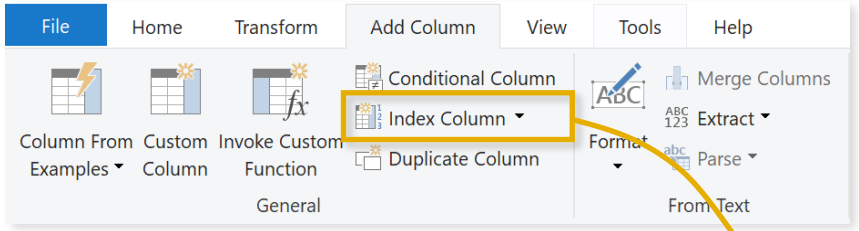


ADDING INDEX COLUMNS

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

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)*

- They are often used to create **unique IDs** and form relationships between tables *(more on that later!)*

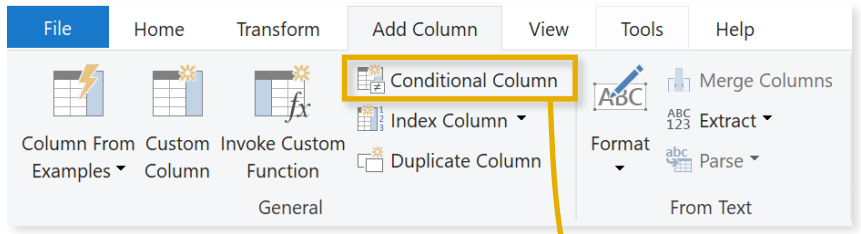


	Index	Transaction_Date	Stock_ Date	Invoice_ID
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 43% Error 0% Empty 56%
1	1	8/5/2017	7/7/2017	INV-9999
2	2	8/5/2017	7/10/2017	INV-9998
3	3	8/5/2017	7/31/2017	INV-9997
4	4	8/5/2017	7/9/2017	INV-9996
5	5	8/5/2017	6/11/2017	INV-9995
6	6	8/5/2017	2/13/2017	INV-9994
7	7	8/5/2017	6/23/2017	INV-9993
8	8	8/5/2017	3/12/2017	INV-9992
9	9	8/5/2017	3/19/2017	INV-9991
10	10	8/5/2017	6/16/2017	INV-9990
11	11	8/5/2017	7/12/2017	INV-9989
12	12	8/5/2017	7/25/2017	INV-9988
13	13	8/5/2017	7/29/2017	INV-9987
14	14	8/5/2017	2/26/2017	INV-9986
15	15	8/5/2017	4/8/2017	INV-9985
16	16	8/5/2017	6/13/2017	INV-9984

ADDING CONDITIONAL COLUMNS

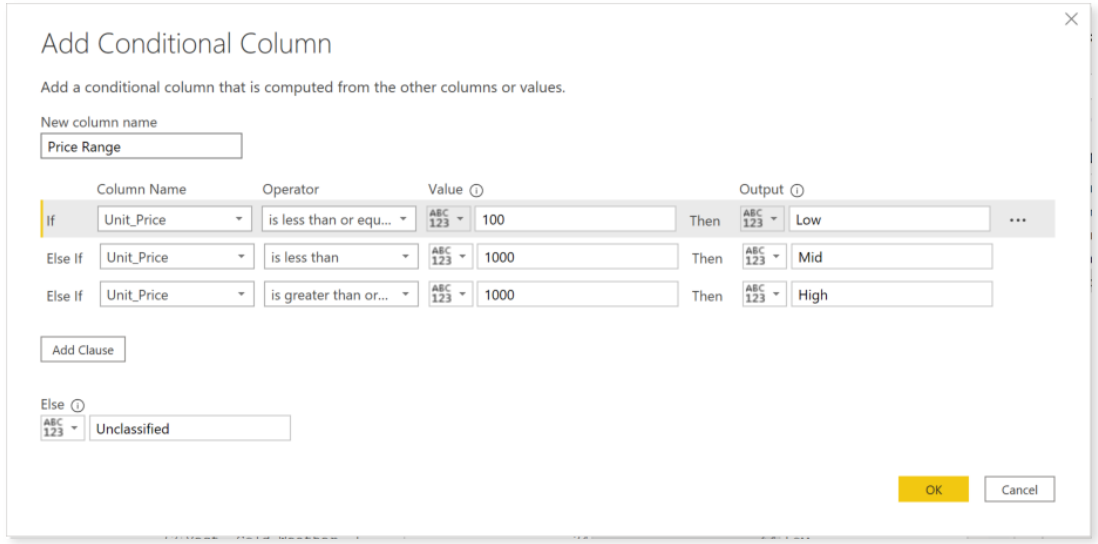
- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

Conditional columns allow you to define new fields based on logical rules *(IF statements)*



*In this case we're creating a new conditional column called "**QuantityType**", which depends on the values in the "**OrderQuantity**" column, as follows:*

- *If Unit Price **<= 100**, then Price Range = "**Low**"*
- *If Unit Price **< 1,000**, then Price Range = "**Mid**"*
- *If Unit Price **>=1,000**, then Price Range = "**High**"*
- *Otherwise, Price Range = "**Unclassified**"*

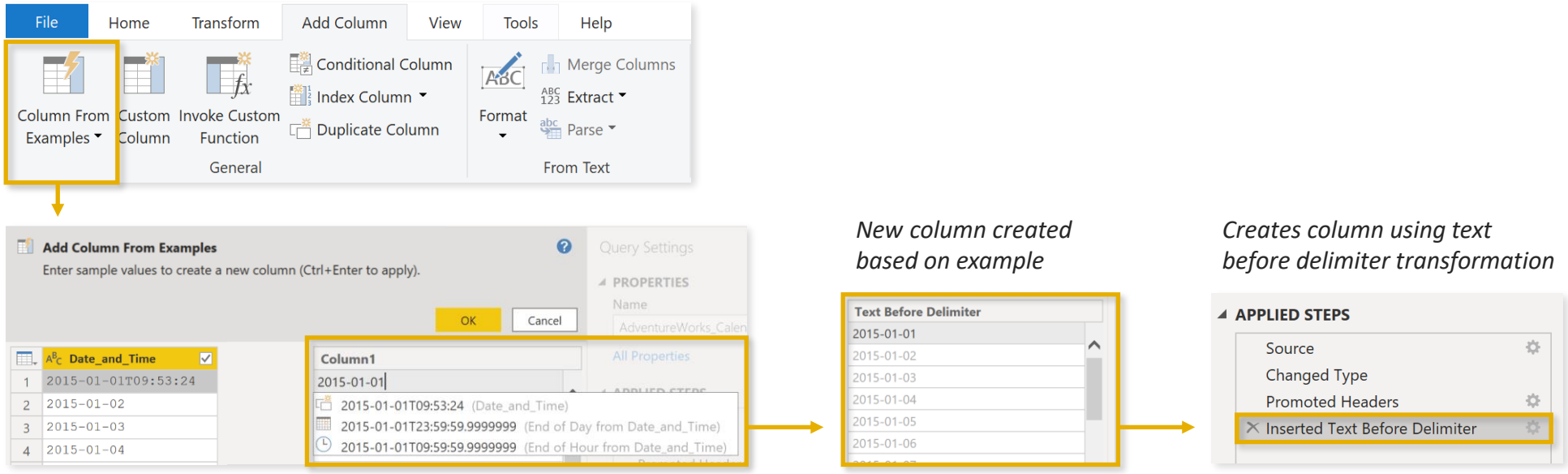


ADDING COLUMNS FROM EXAMPLE

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

Columns from examples allow you to add new columns by providing an example value

- Column examples can be created from individual or all existing columns in the table



PRO TIP: Use this when you know the outcome you want but don't know which transformation, or group of transformations, to use

GROUPING & AGGREGATING DATA

Cleaning Data

Identifying Keys

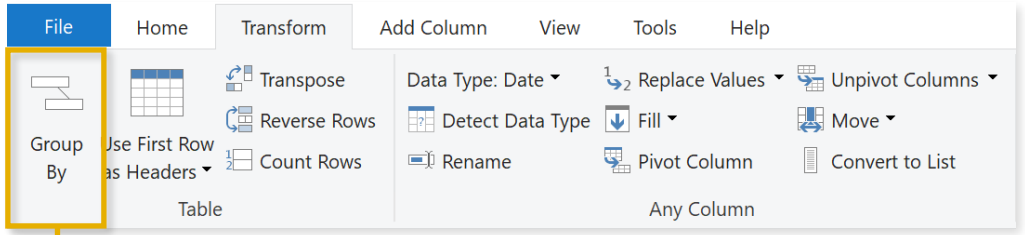
Shaping Tables

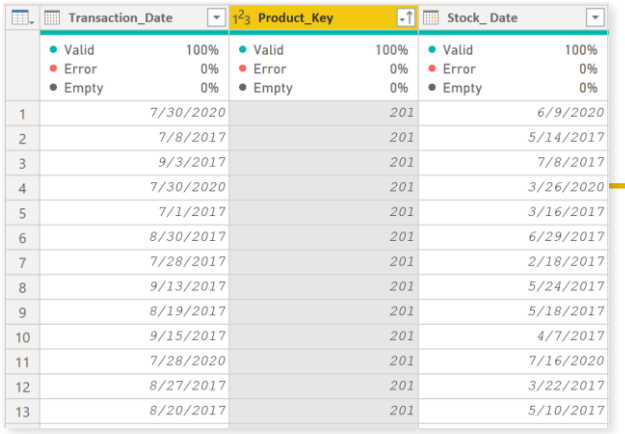
Combining Queries

Modifying M Code

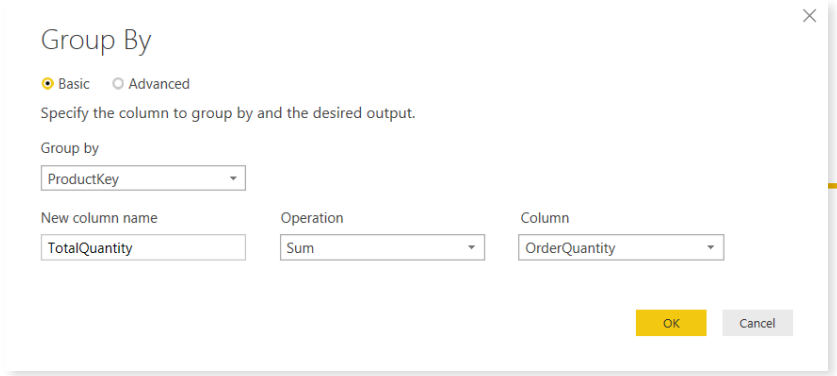
Group by allows you to aggregate your data at a different level

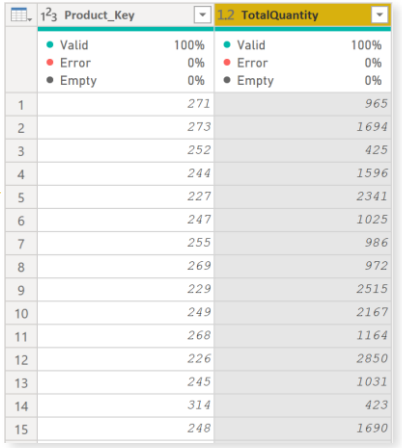
- **For example:** Transform daily data into monthly, roll up transaction-level data by store, etc.





	Transaction_Date	Product_Key	Stock_Date
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	7/30/2020	201	6/9/2020
2	7/8/2017	201	5/14/2017
3	9/3/2017	201	7/8/2017
4	7/30/2020	201	3/26/2020
5	7/1/2017	201	3/16/2017
6	8/30/2017	201	6/29/2017
7	7/28/2017	201	2/18/2017
8	9/13/2017	201	5/24/2017
9	8/19/2017	201	5/18/2017
10	9/15/2017	201	4/7/2017
11	7/28/2020	201	7/16/2020
12	8/27/2017	201	3/22/2017
13	8/20/2017	201	5/10/2017





	Product_Key	TotalQuantity
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	271	965
2	273	1694
3	252	425
4	244	1596
5	227	2341
6	247	1025
7	255	986
8	269	972
9	229	2515
10	249	2167
11	268	1164
12	226	2850
13	245	1031
14	314	423
15	248	1690

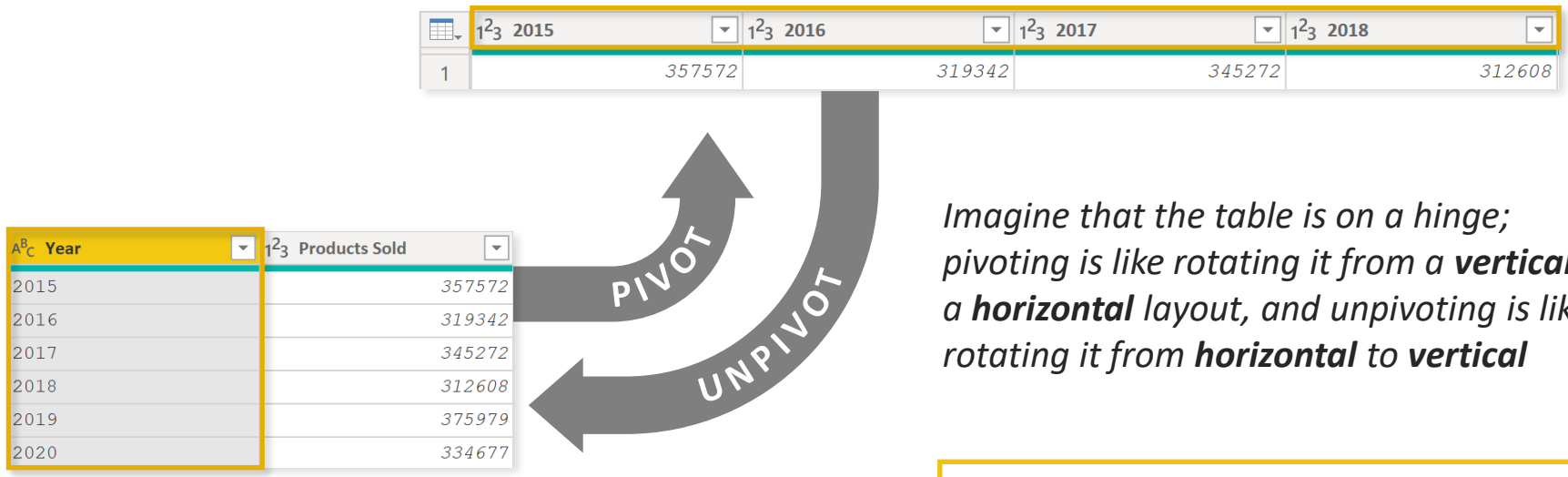
*In this case we're transforming a daily, transaction-level table into a summary of "**TotalQuantity**" rolled up by "**ProductKey**"*

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

PIVOTING & UNPIVOTING

- Cleaning Data
- Identifying Keys
- **Shaping Tables**
- Combining Queries
- Modifying M Code

"Pivoting" is a fancy way to describe the process of turning **distinct row values** into **columns** *("pivoting")* or turning **columns** into **rows** *("unpivoting")*



*Imagine that the table is on a hinge; pivoting is like rotating it from a **vertical** to a **horizontal** layout, and unpivoting is like rotating it from **horizontal** to **vertical***

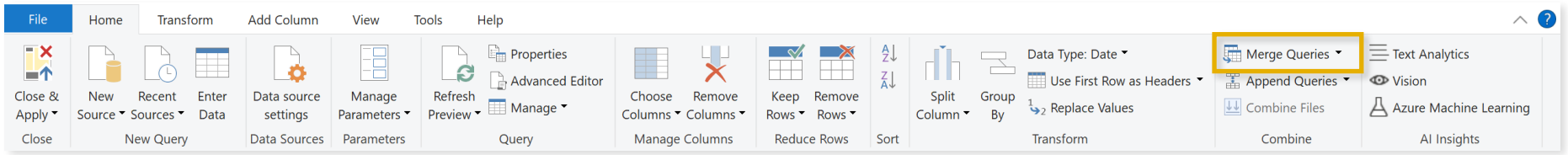
HEY THIS IS IMPORTANT!

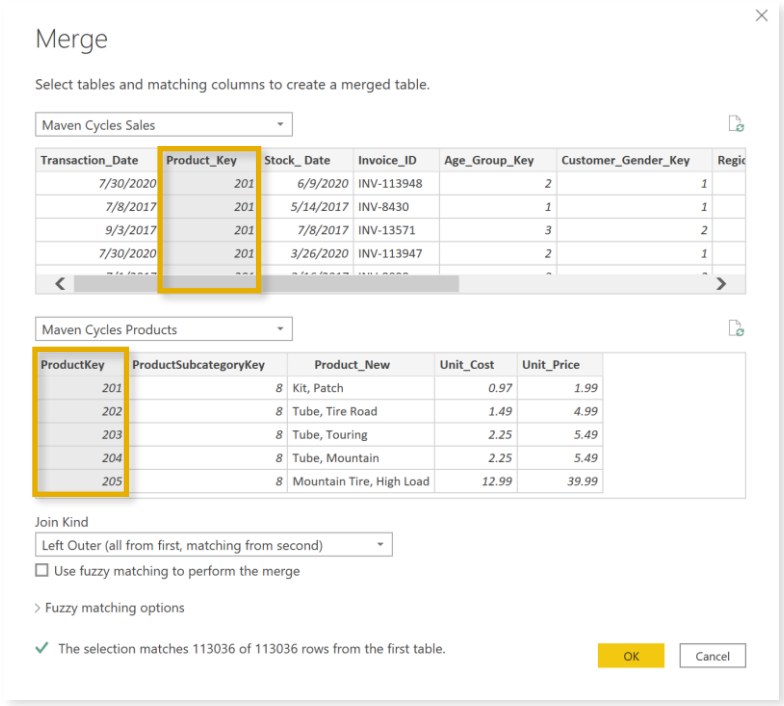
Transpose works very similarly, but doesn't recognize unique values; instead, the entire table is transformed so that each row becomes a column and vice versa

MERGING QUERIES

Merging queries allows you to join tables based on a common column

- Cleaning Data
- Identifying Keys
- Shaping Tables
- **Combining Queries**
- Modifying M Code





*In this case we're merging the **Maven Cycles Sales** table with the **Maven Cycles Products** table, which share a common "Product_Key" column*

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

HEY THIS IS IMPORTANT!

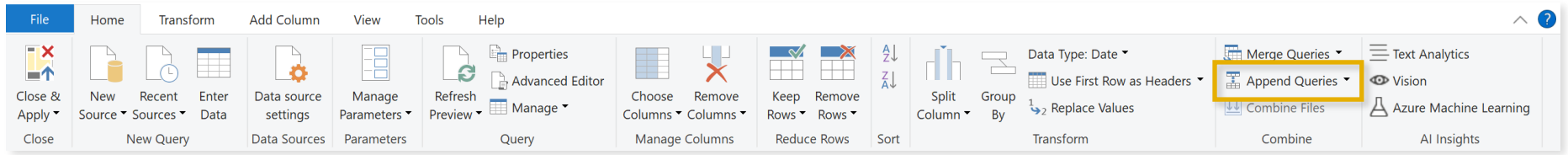
Just because you ***can*** merge tables, doesn't mean you ***should***.

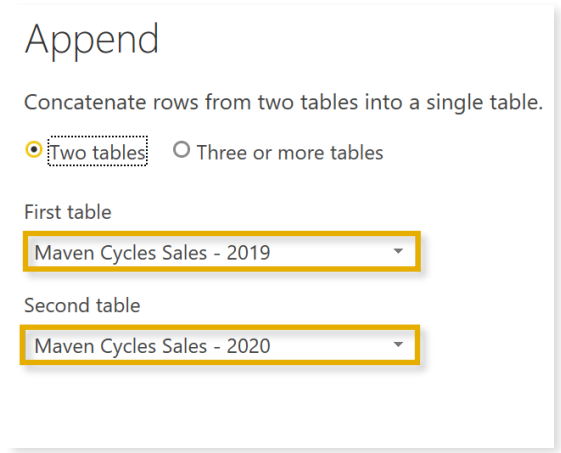
In general, it's better to keep tables separate and define **relationships** between them in your data model

APPENDING QUERIES

- Cleaning Data
- Identifying Keys
- Shaping Tables
- **Combining Queries**
- Modifying M Code

Appending queries allows you to combine *(or stack)* tables that share the exact same column structure and data types





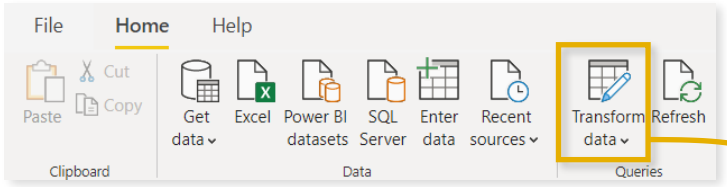
*In this case we're appending the **Maven Cycles Sales – 2019** table to the **Maven Cycles Sales – 2020** table, which is valid since they share identical table structures*

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

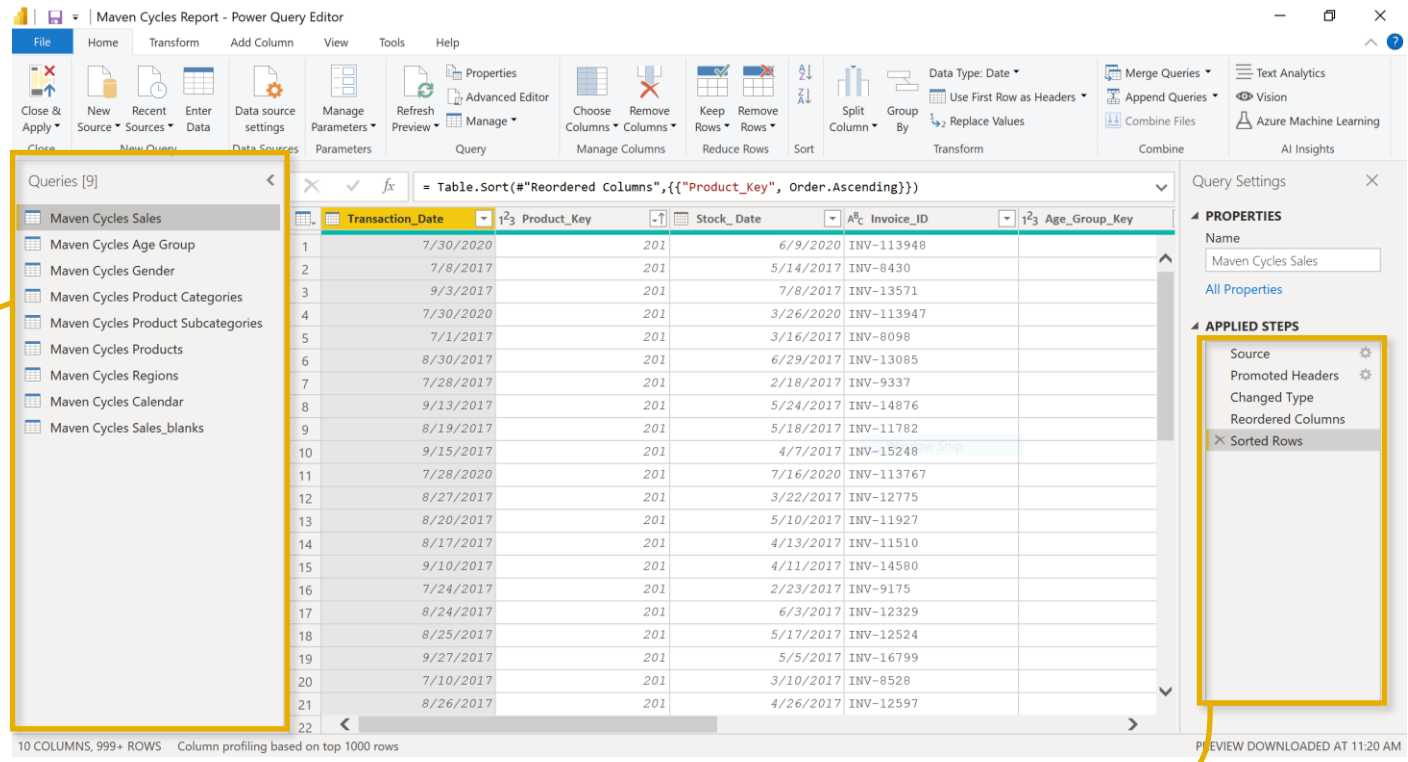
PRO TIP: Use the "Folder" option (Get Data > More > Folder) to append all files within a folder (assuming they share the same structure); as you add new files, simply refresh the query and they will automatically append!

MODIFYING QUERIES

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code



Select **Transform Data** from the Home tab to launch the Query Editor



View or modify existing queries in the "**Queries**" pane

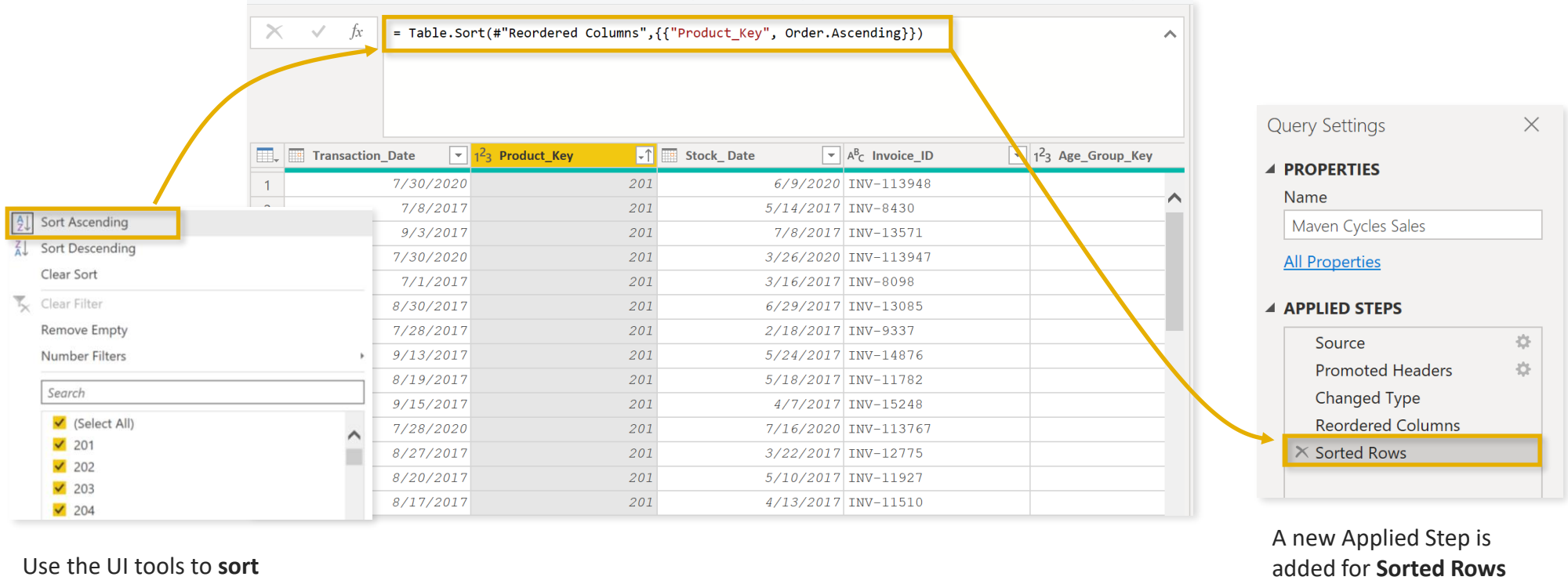
Within each query, you can click each item within the "**Applied Steps**" pane to view each stage of the transformation, add new steps, delete existing ones, or modify individual steps by clicking the gear icons

INTRO TO M CODE

Data Mashup, or **M code**, is the formula language that drives Power Query

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

The Query Editor writes the corresponding **M code**



Use the UI tools to **sort** *Product_Key* **ascending**

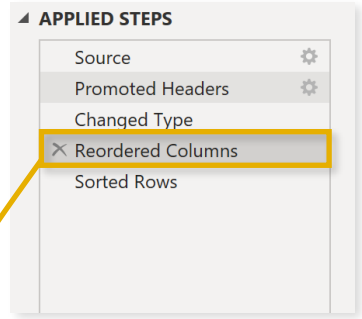
A new Applied Step is added for **Sorted Rows**

EDITING & ADDING APPLIED STEPS

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

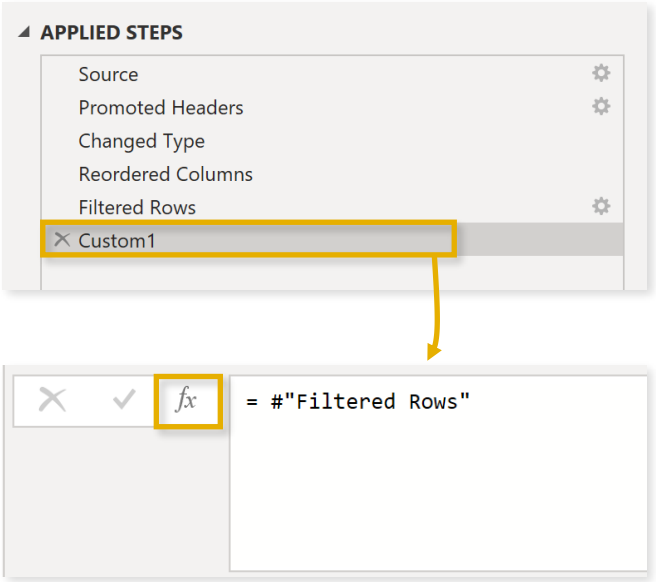
Although you can perform many transformations with the Power Query UI tools, you can do even more by directly **editing** or **writing new M code**

Click on an applied step to **edit** the M code from the formula bar




```
= Table.ReorderColumns(#"Changed Type",{"Transaction_Date",
    "Product_Key", "Stock_ Date", "Invoice_ID", "Age_Group_Key",
    "Customer_Gender_Key", "Region_Key", "Product_Category_Key",
    "Sub_Category_Key", "Quantity_Sold"})
```

Click the *fx* button to open a Blank Query and **write your own** custom applied step

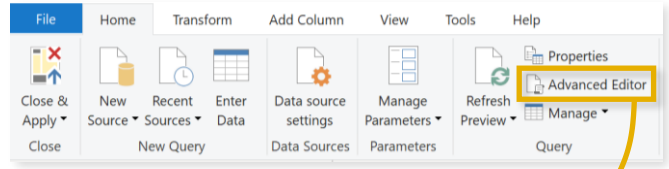


ADVANCED EDITOR (INTRO TO M CODE)

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

Opening the **advanced editor** allows you to see the M code that makes up your query, which consists of two blocks:

1. **let:** the definition of all variables
2. **in:** the output of your query



let block that defines all the variables

Variable names that match the individual steps in the Applied Steps pane

```
let
    Source = Csv.Document(File.Contents("G:\Shared drives\Author Drive (A. Parry)\Microsoft Power BI - DA-100\Raw Data\Maven Cycles
        Sales.csv"),[Delimiter=",", Columns=10, Encoding=65001, QuoteStyle=QuoteStyle.None]),
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"Transaction_Date", type date}, {"Stock_ Date", type date},
        {"Invoice_ID", type text}, {"Age_Group_Key", Int64.Type}, {"Customer_Gender_Key", Int64.Type}, {"Region_Key", Int64.Type},
        {"Product_Category_Key", Int64.Type}, {"Sub_Category_Key", Int64.Type}, {"Product_Key", Int64.Type}, {"Quantity_Sold", Int64.Type}}
        ),
    #"Reordered Columns" = Table.ReorderColumns(#"Changed Type",{"Transaction_Date", "Product_Key", "Stock_ Date", "Invoice_ID",
        "Age_Group_Key", "Customer_Gender_Key", "Region_Key", "Product_Category_Key", "Sub_Category_Key", "Quantity_Sold"}),
    #"Sorted Rows" = Table.Sort(#"Reordered Columns",{{"Product_Key", Order.Ascending}})
in
    #"Sorted Rows"
```

***in** block that is the final output of all variables in the query*

*Final **output** variable*

***Function** called that reorders columns within the table*

COMMON FUNCTION CATEGORIES

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- **Modifying M Code**

TABLE Functions

Functions to create and manipulate table values

Common Examples:

- Table.FromList
- Table.ToList
- Table.IsEmpty
- Table.FindText
- Table.RemoveColumns
- Table.Contains

Table function categories:

- Table construction
- Conversion
- Information
- Row operations
- Column operations
- Membership

LIST Functions

Functions to create and manipulate list values

Common Examples:

- List.Select
- List.Contains
- List.Union
- List.Median
- List.Numbers

List function categories:

- Selection
- Membership
- Set operations
- Ordering
- Generators

TEXT Functions

Functions to create and manipulate text values

Common Examples:

- Text.Length
- Text.From
- Text.Middle
- Text.Contains
- Text.Remove
- Text.BeforeDelimiter

Text function categories:

- Information
- Text comparisons
- Extraction
- Membership
- Modification
- Transformations

DATE Functions

Functions to create and manipulate date, datetime, and datetimezone values

Common Examples:

- Date.EndOfMonth
- Date.EndOfQuarter
- Date.Day
- Date.StartOfWeek
- Date.StartOfMonth

M CODE SYNTAX

- Cleaning Data
- Identifying Keys
- Shaping Tables
- Combining Queries
- Modifying M Code

Let's say we want to filter Maven Cycles Sales where the quantity sold equals 2

In order to accomplish this, we need to write the following code as an applied step:

```
= Table.SelectRows(#"Reordered Columns", each ([Quantity_Sold] = 2))
```

FUNCTION NAME
This example uses a table function to select certain rows based on filter criteria

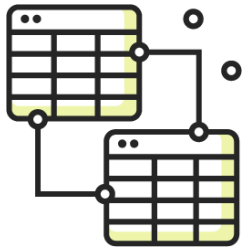
PREVIOUS STEP
This function will be applied after the Reordered Columns step

FUNCTION ARGUMENTS
The function in this example requires a filter condition (quantity sold = 2)

MODELING THE DATA

Prepare the Data

- Get data from different sources
- Clean, transform, and load data



Model the Data

- Design a data model
- Develop a data model
- Create model calculations with DAX
- Optimize model performance

Visualize & **Analyze** the Data

- Create reports
- Create dashboards
- Enrich reports for usability
- Enhance reports for usability & storytelling
- Identify patterns & trends

Deploy & **Maintain** Assets

- Manage files & datasets
- Manage workspaces

DATA MODELING

In this section we'll cover core **data modeling** topics, including designing a data model, developing a data model, and optimizing model performance

TOPICS WE'LL COVER:

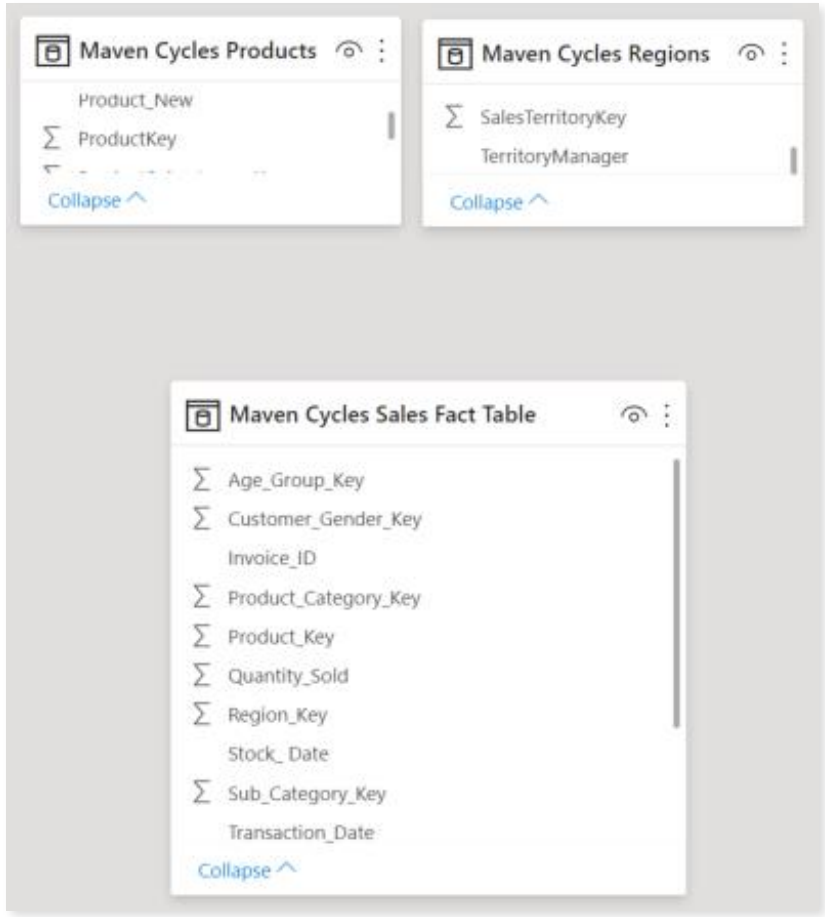
- Data Model 101
- Table Roles
- Table Relationships
- Data Model Types
- Date Tables
- Optimization Tips

COMMON QUESTIONS:

- *You have multiple tables in your data model and need to show MTD revenue compared to last year. Which of the following actions should you take?*
- *In order to reduce the size of your data model, what should you do with a DateTime column that contains unique values for each record in the table?*
- *Which of the following relationship types is the most efficient way to connect a fact and lookup table?*

WHAT'S A "DATA MODEL"?

- Data Model 101
- Table Roles
- Table Relationships
- Data Model Types
- Date Tables
- Optimization Tips



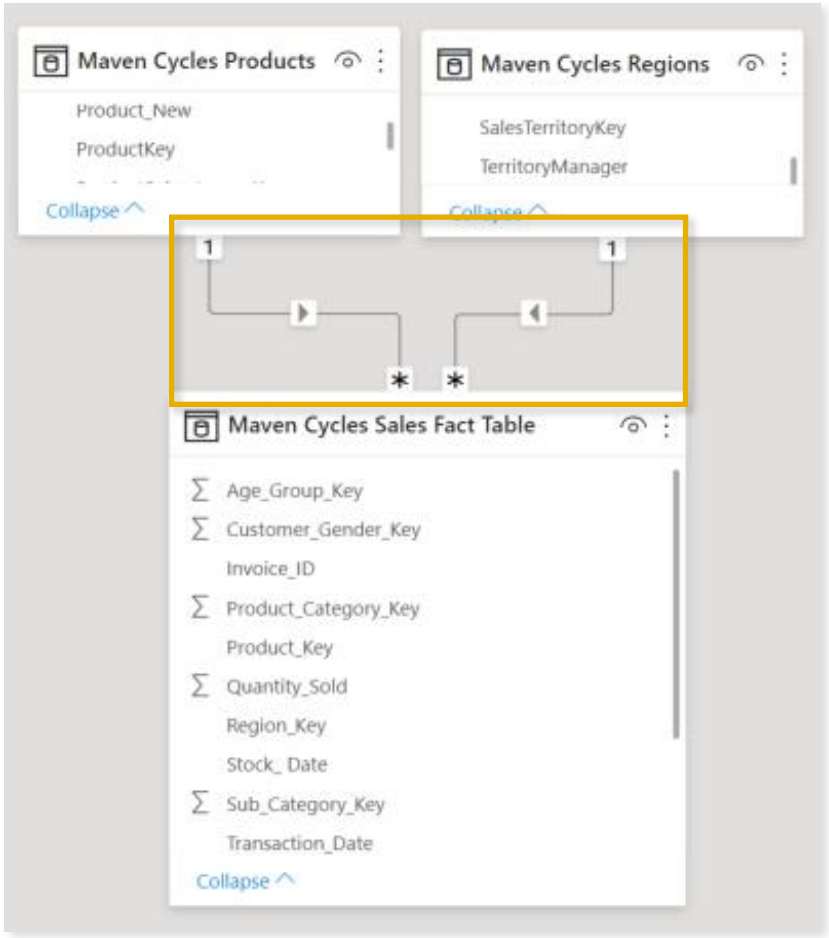
This **IS NOT** a data model

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

Country	Total Revenue	Total Profit
Australia	88,747.96	150,232.47
Canada	88,747.96	150,232.47
France	88,747.96	150,232.47
Germany	88,747.96	150,232.47
United Kingdom	88,747.96	150,232.47
United States	88,747.96	150,232.47
Total	**88,747.96**	**150,232.47**

WHAT'S A "DATA MODEL"?

- Data Model 101
- Table Roles
- Table Relationships
- Data Model Types
- Date Tables
- Optimization Tips



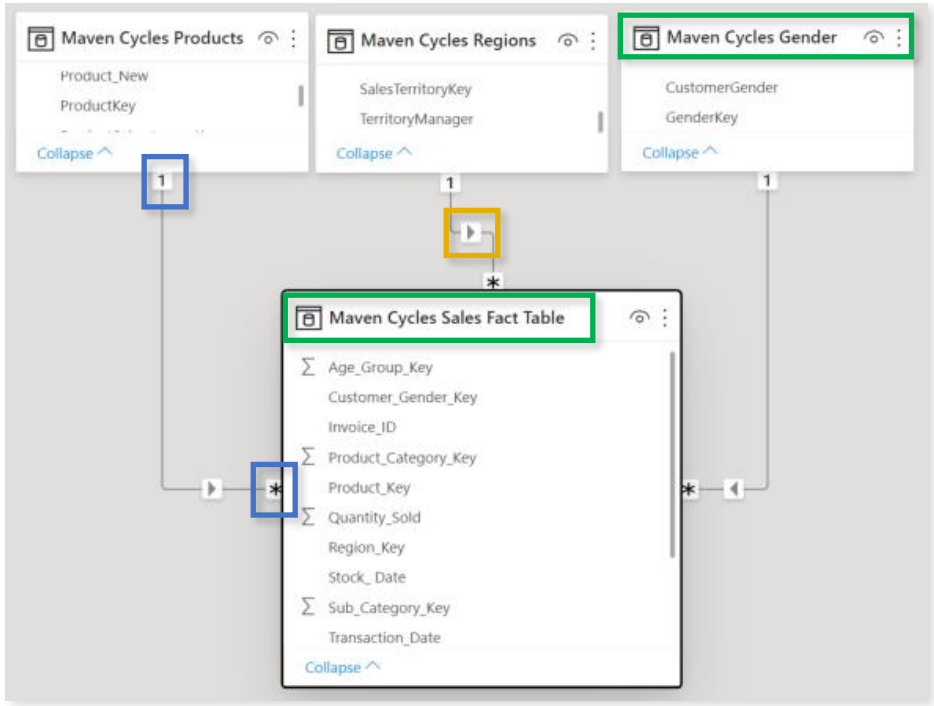
This **IS** a data model

- The tables are connected via relationships, based on their common fields
- Now the **Sales** table knows how to filter using fields from the **Products** & **Regions** tables!

Country	Total Revenue	Total Profit
United States	$62,726,170	$31,156,122
Australia	$49,811,320	$23,283,029
United Kingdom	$23,632,362	$11,623,161
Germany	$20,499,693	$9,896,988
France	$20,090,553	$9,691,311
Canada	$16,655,827	$8,750,364
Total	**$193,415,925**	**$94,400,977**

DATA MODEL BEST PRACTICES

- Data Model 101
- Table Roles
- Table Relationships
- Data Model Types
- Date Tables
- Optimization Tips



A well-designed model is **critical** and ideally should:

- ✓ Use a star schema with **one-to-many** *(1:*)* relationships
- ✓ Contain relationships with **one-way** filters *(vs. bidirectional)*
- ✓ Contain tables that each serve a *specific* purpose, including **data** *(fact)* tables and **lookup** *(dim)* tables
- ✓ Only include the data you need for analysis *(no redundant or unnecessary records or fields)*
- ✓ Split out individual **date** and **time** components from **DateTime** fields

DATA TABLES VS LOOKUP TABLES

Models generally contain two types of tables: **Data** tables & **Lookup** tables

- **Data** (or *"fact"*) tables contain measurable *metrics* about the business *(quantity, revenue, views, etc.)*
- **Lookup** (or *"dimension"*) tables provide descriptive *attributes* about each dimension in your model *(customers, products, 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 **Data Table** contains "**quantity**" values, and connects to lookup tables via the "**date**" and "**product_id**" columns*

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 provides additional 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 provides additional attributes about each **product** (brand, name, price, etc.)*

PRIMARY VS FOREIGN KEYS

- Data Model 101
- Table Roles
- Table Relationships
- Data Model Types
- Date Tables
- Optimization Tips

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

These columns are **foreign keys**; they contain *multiple* instances of each value, and are used to match the **primary keys** in related lookup tables

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

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

These columns are **primary keys**; they *uniquely* identify each row of a table, and match the **foreign keys** in related data tables

RELATIONSHIPS VS MERGED TABLES

*Can't I just **merge queries** or use **LOOKUP** or **RELATED** functions to pull those attributes into the fact table itself, so that I have everything in one place??*

-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	Nationeel	Nationeel Grape Fruit Roll	52382137179	17
1/7/1997	869	2	7	1	1997	Tuesday	January	Q1	Nationeel	Nationeel 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

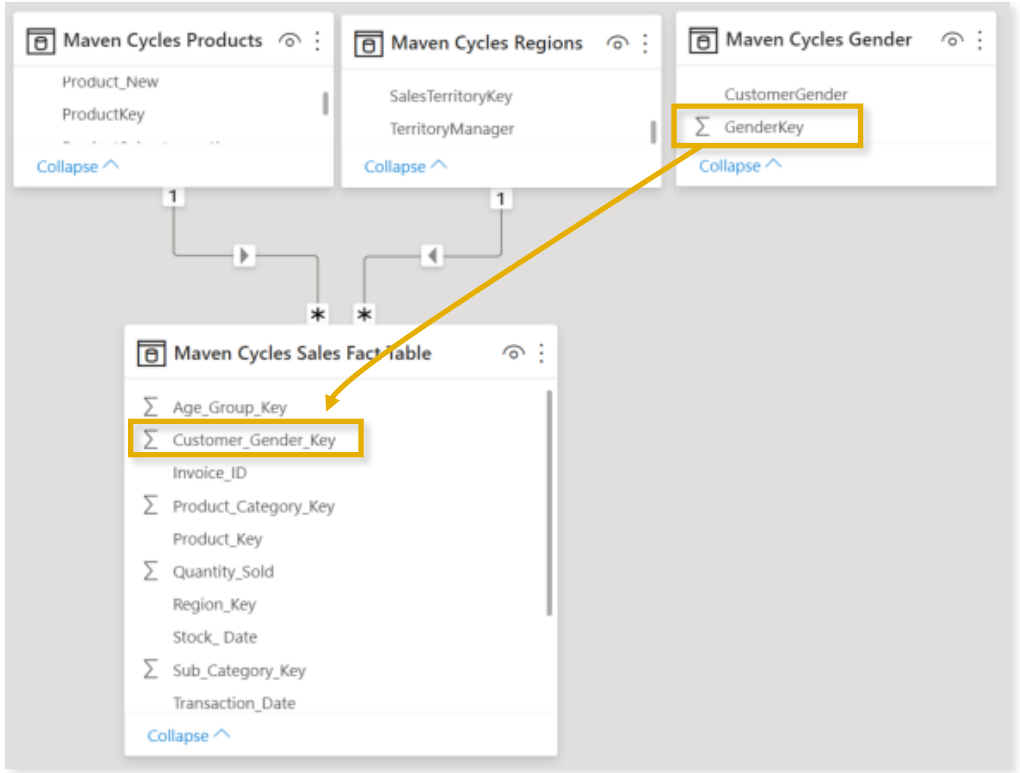
Sure, you can ***but it's inefficient!***

- Merging data in this way creates **redundant data** and utilizes **significantly more memory and processing power** than creating relationships between multiple small tables

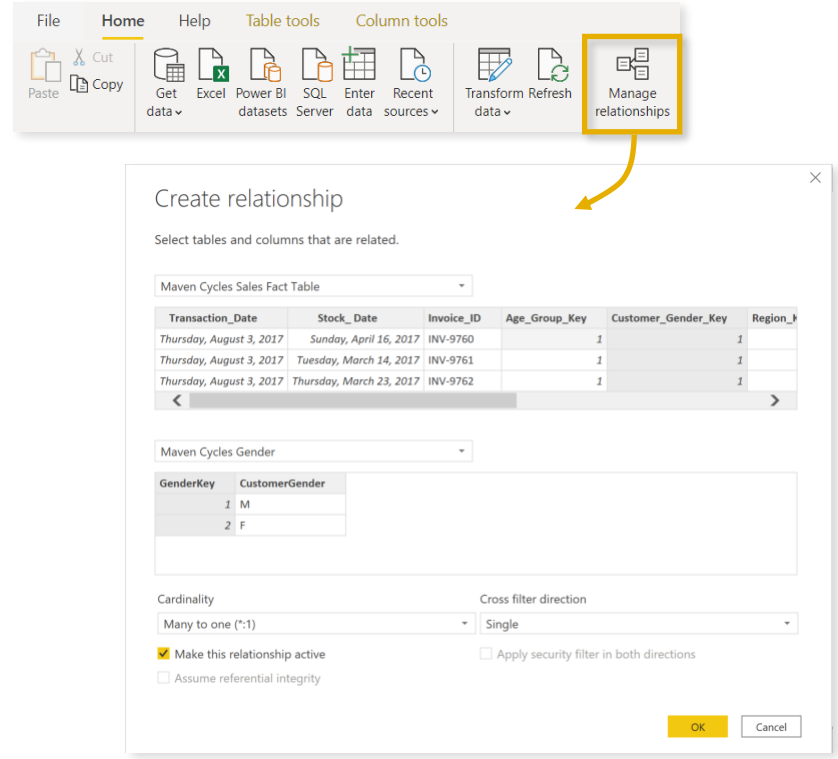
CREATING TABLE RELATIONSHIPS

- Data Model 101
- Table Roles
- **Table Relationships**
- Data Model Types
- Date Tables
- Optimization Tips

Option 1: Click and drag to connect primary and foreign keys within the **Relationships** pane



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



RELATIONSHIP CARDINALITY

Data Model 101

Table Roles

Table Relationships

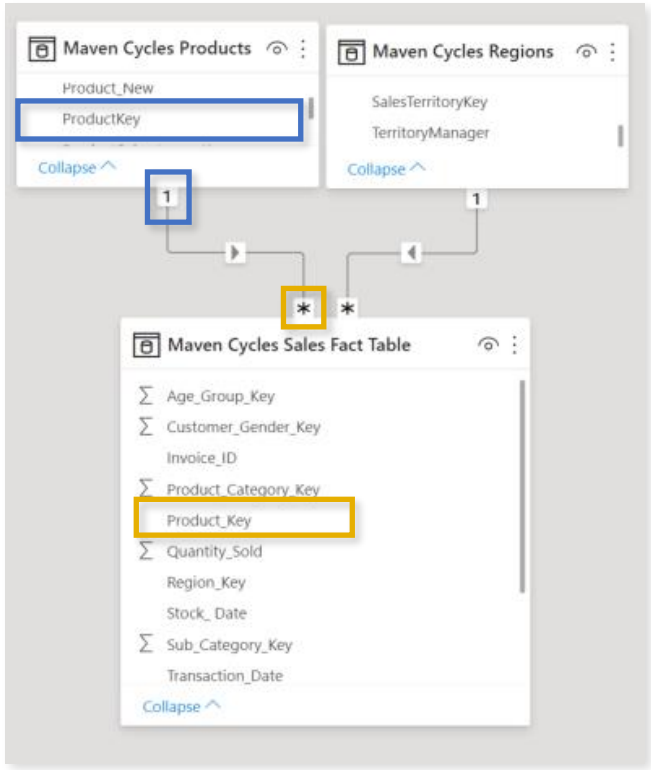
Data Model Types

Date Tables

Optimization Tips

Cardinality refers to the *uniqueness of values* in a column

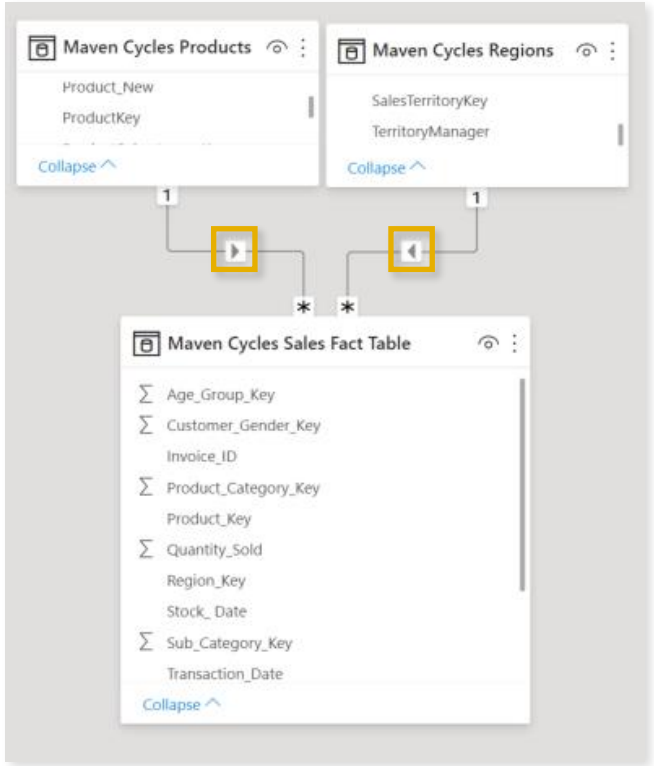
- As a best practice, all relationships in the data model should follow a "**one-to-many**" cardinality: **one** instance of each *primary key*, but potentially **many** instances of each *foreign key*



*In this case, there is only **ONE instance of each Product Key** in the Products table (noted by the "**1**"), since each row contains **attributes of a single product** (Name, Unit Cost, Unit Price, Price Range, etc.)*

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

FILTER FLOW



Here we have a single data table, **Sales**, connected to **Products** & **Regions** lookups

Note the filter directions (arrows) in each relationship; by default, **these will point from the "one" side of the relationship** *(lookups)* **to the "many" side** *(data)*

- When you filter a table, that filter context is passed along to all related "*downstream*" tables (following the direction of the arrow)
- Filters **cannot** flow "*upstream*" (against the direction of the arrow)

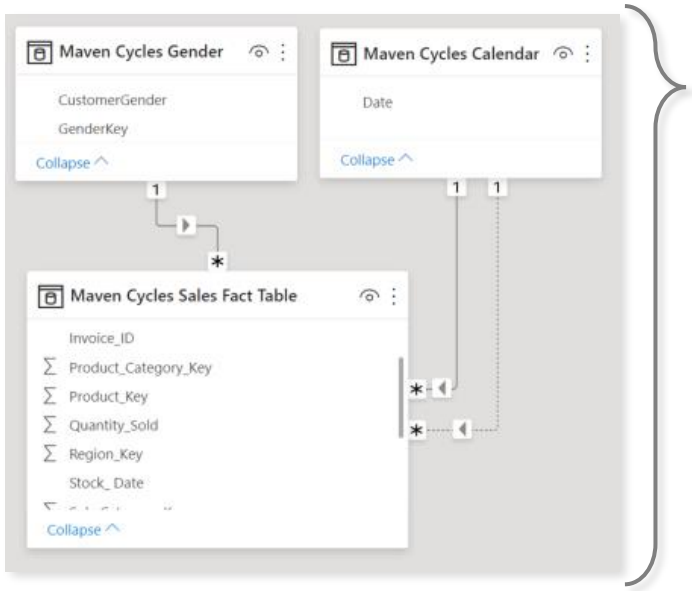
PRO TIP: Arrange your lookup tables **above** your data tables in your model as a visual reminder that filters flow "**downstream**"

**In some cases, filters may default to "two-way" depending on your Power BI Desktop settings*

ACTIVE & INACTIVE RELATIONSHIPS

There are two key types of physical table relationships: **Active** & **Inactive**

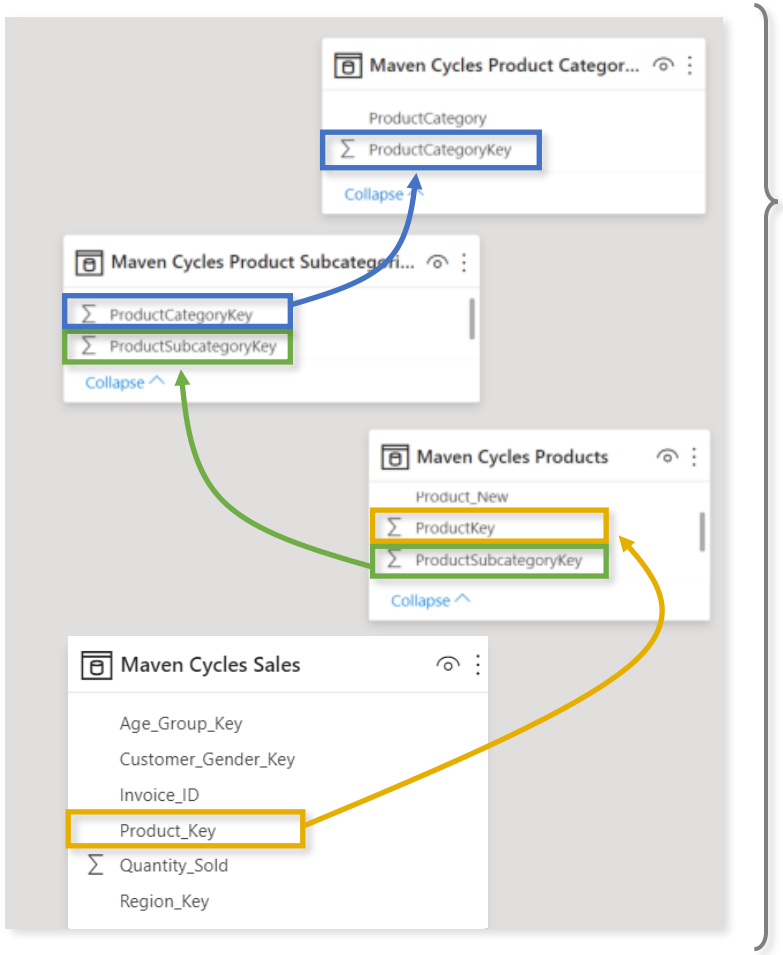
- **Active** relationships are the primary means of filter propagation in your data model
- **Inactive** relationships filter propagation during a calculation defined using DAX expressions



These are **physical** relationships:

- Visible links between tables (typically 1:* cardinality)
- Can be **active** or **inactive**
  - **Active** shown with solid line
  - **Inactive** shown by dotted line
- Can be accessed using DAX functions like **RELATED**, **RELATEDTABLE** or **USERELATIONSHIP** (inactive only)
- Best way to connect tables (but not always possible)

CREATING “SNOWFLAKE” SCHEMAS



The **Sales** table can connect to **Products** using the ***Product Key*** field, but cannot connect directly to the **Subcategories** or **Categories** tables

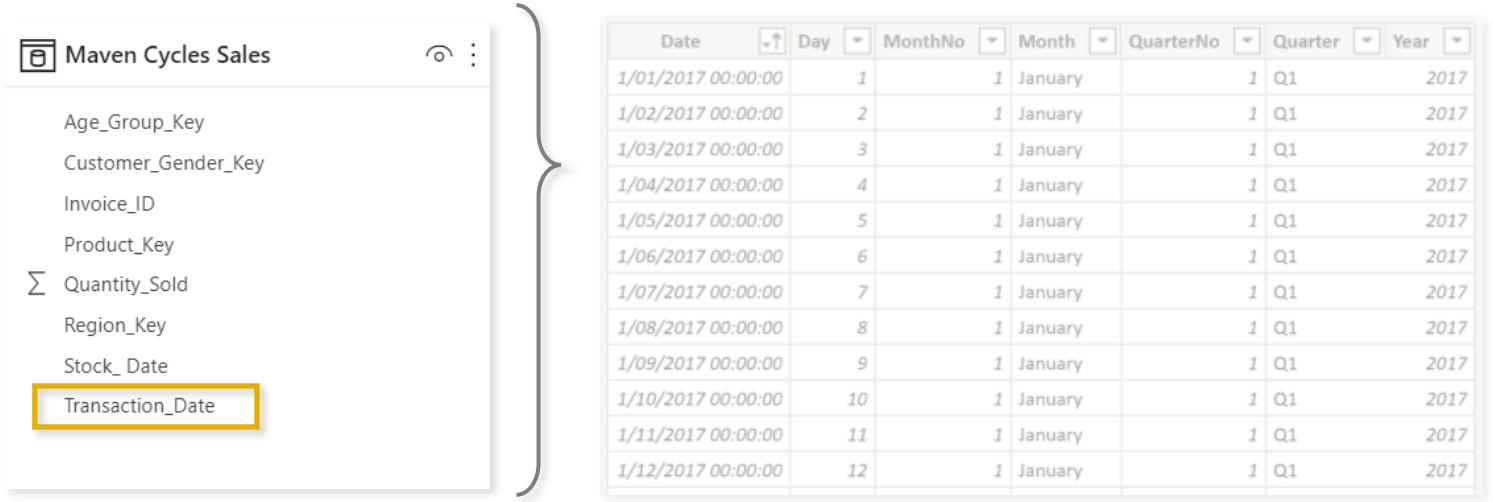
By creating relationships from **Products** to **Subcategories** (using ***ProductSubcategoryKey***) and **Subcategories** to **Categories** (using ***ProductCategoryKey***), we have essentially connected **Sales_Data** to each lookup table; filter context will now flow all the way down the chain

PRO TIP: Models with chains of dimension tables are often called “**snowflake**” schemas (whereas “**star**” schemas usually have individual lookup tables surrounding a central data table)

AUTOMATIC DATE TABLES

By default, Power BI automatically creates a **hidden date table** for any table that contains a **Date** or **DateTime** column on the one-side of a relationship

- Auto-generated calendars include *all* dates through the end of the year, regardless of the actual date range in the table



Date	Day	MonthNo	Month	QuarterNo	Quarter	Year
1/01/2017 00:00:00	1	1	January	1	Q1	2017
1/02/2017 00:00:00	2	1	January	1	Q1	2017
1/03/2017 00:00:00	3	1	January	1	Q1	2017
1/04/2017 00:00:00	4	1	January	1	Q1	2017
1/05/2017 00:00:00	5	1	January	1	Q1	2017
1/06/2017 00:00:00	6	1	January	1	Q1	2017
1/07/2017 00:00:00	7	1	January	1	Q1	2017
1/08/2017 00:00:00	8	1	January	1	Q1	2017
1/09/2017 00:00:00	9	1	January	1	Q1	2017
1/10/2017 00:00:00	10	1	January	1	Q1	2017
1/11/2017 00:00:00	11	1	January	1	Q1	2017
1/12/2017 00:00:00	12	1	January	1	Q1	2017

*Automatically creates a **hidden** date table containing all these columns*

PROS & CONS: AUTOMATIC DATE TABLES

- Data Model 101
- Table Roles
- Table Relationships
- Data Model Types
- **Date Tables**
- Optimization Tips

PROS:

- Automatically generated
- Enables (some) time intelligence functionality by default
- Simplifies data model creation and management
- Does not require an advanced understanding of DAX

CONS:

- Hidden from view, cannot be modified/customized
- Generated for every date field across every lookup/dimension table *(**bloats model size**)*
- Can't be enabled or disabled at the table-level
- Hierarchies aren't automatically generated *(if grouped by month, would summarize that month across ALL years)*
- Each automatic date table can *only* filter the table it corresponds to *(cannot traverse table relationships)*

PRO TIP: Turn OFF the **auto date/time** feature in Power BI Desktop and either import a date dimension table or create your own using **CALENDAR** functions

DATE TABLE REQUIREMENTS

If you import or create your own date table, **it must meet these requirements:**

- ✓ Must contain *all* the days for all years represented in your fact tables
- ✓ Must have at least one field set as a **Date** or **DateTime** datatype
- ✓ Cannot contain duplicate dates or datetime values
- ✓ If using a time component within a date column, all times must be identical *(i.e. 12:00)*
- ✓ Should be marked as a **date table** *(not required but a best practice)*

HEY THIS IS IMPORTANT!

If **Time** is present in your date field, split the time component into a new column (this adheres to relationship requirements and decreases column cardinality)

MODEL OPTIMIZATION TIPS

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
- Identify poorly performing measures and relationships
- Create 1:* relationships
- Separate Time component form a DateTime filed
- Aggregate data

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

DAX CALCULATIONS

In this section we'll cover **Data Analysis Expressions**, or **DAX**, which is the formula language that drives front-end calculated tables, columns, and measures in Power BI

TOPICS WE'LL COVER:

- **Calculated Columns & Measures**
- **Aggregation Functions**
- **CALCULATE**
- **CALCULATE Modifiers**
- **Table Functions**
- **Time Intelligence**

COMMON QUESTIONS:

- *You need to create a calculated table that returns the top 50 revenue generating products. How should you complete the following DAX calculation?*
- *How would you complete the following DAX calculated table expression, so it returns a table of the top 25 customers based on Total Revenue?*
- *How would you set up a quick measure to create a monthly rolling average for Transactions?*

MEET DAX

- **Calculated Columns & Measures**
- Aggregation Functions
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

Data Analysis Expressions, known as **DAX**, is the formula language that drives Power BI

With DAX, you can:

- Add **calculated columns** and **measures** to your model, using intuitive syntax
- Go beyond the capabilities of traditional "grid-style" formulas, with powerful and flexible functions built specifically to work with tabular data models

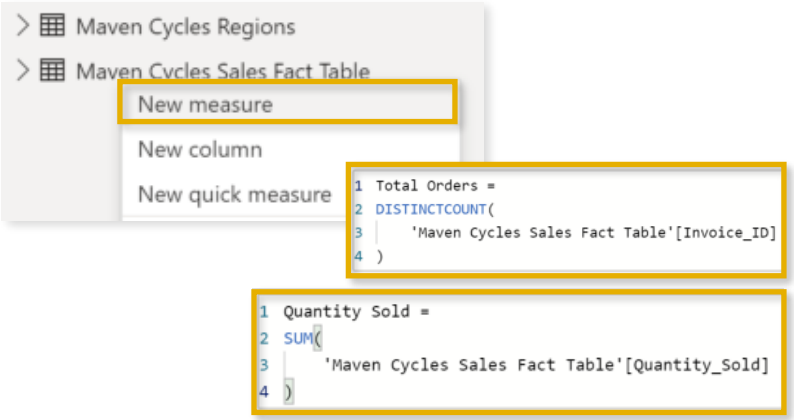
Two ways to use DAX

1) *Calculated Columns*

```
1 Week Number =
2 WEEKNUM(
3     'Maven Cycles Calendar'[Date],
4     2
5 )
```

Date	Year	Quarter	Month	Week Number	Day of Week	Month Name
Thursday, January 1, 2015	2015	1	1	1	4	Jan
Friday, January 2, 2015	2015	1	1	1	5	Jan
Saturday, January 3, 2015	2015	1	1	1	6	Jan
Sunday, January 4, 2015	2015	1	1	1	7	Jan
Monday, January 5, 2015	2015	1	1	2	1	Jan
Tuesday, January 6, 2015	2015	1	1	2	2	Jan
Wednesday, January 7, 2015	2015	1	1	2	3	Jan
Thursday, January 8, 2015	2015	1	1	2	4	Jan
Friday, January 9, 2015	2015	1	1	2	5	Jan

2) *Measures*



CALCULATED COLUMNS

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

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

- They refer to **entire tables** or **columns** *(no "A1-style" references)*
- They generate values for each row, which are **visible within tables in the Data view**
- They understand **row context**; they're great for defining properties based on information in each row, but generally useless for aggregation (*SUM, COUNT, etc.*)
- They increase the size of your data model!

HEY THIS IS IMPORTANT!

As a rule of thumb, use calculated columns when you want to "stamp" static, fixed values to each row in a table *(or use the Query Editor!)*

DO NOT use calculated columns for aggregation formulas, or to calculate fields for the "Values" area of a visualization *(use **measures** instead)*

PRO TIP: Creating calculated columns as close to the source as possible helps reduce data model size and improve performance

MEASURES

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

Measures are DAX formulas used to generate new calculated values

- Like calculated columns, measures reference **entire tables** or **columns** (*no A1-style or "grid" references*)
- *Unlike* calculated columns, **measure** values aren't visible within tables; they can only be "*seen*" within a visualization like a chart or matrix (*similar to a calculated field in an Excel pivot*)
- Measures are evaluated based on **filter context**, which means they recalculate when the fields or filters around them change (*like when new row or column labels are pulled into a matrix or when new filters are applied to a report*)

HEY THIS IS IMPORTANT!

As a rule of thumb, use measures *(vs. calculated columns)* when a single row can't give you the answer *(in other words, when you need to **aggregate**)*

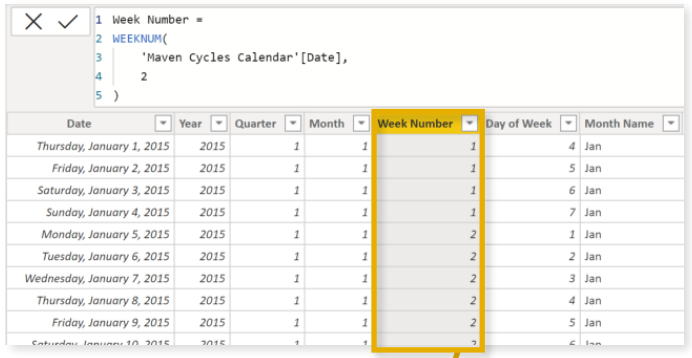
PRO TIP: Use measures to create **numerical, calculated values** that can be analyzed in the "**values**" field of a report visual

RECAP: CALCULATED COLUMNS VS. MEASURES

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

CALCULATED COLUMNS

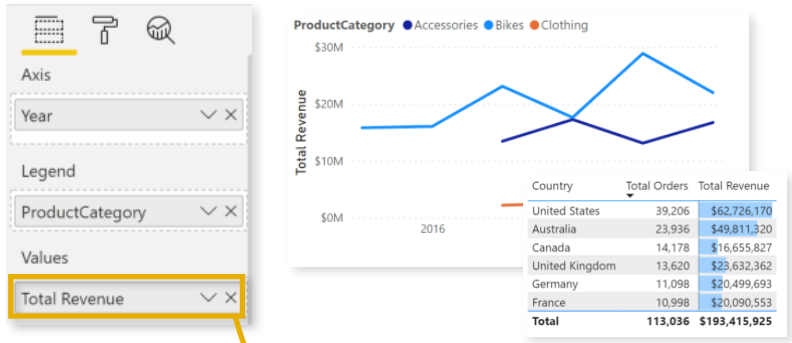
- Values are calculated based on information from each row of a table (has **row context**)
- Appends static values to each row in a table and stores them in the model (*which increases file size*)
- Recalculate on data source refresh or when changes are made to component columns
- Primarily used as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



*Calculated columns "live" in **tables***

MEASURES

- Values are calculated based on information from any filters in the report (has **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
- Almost *always* used within the **values** field of a visual



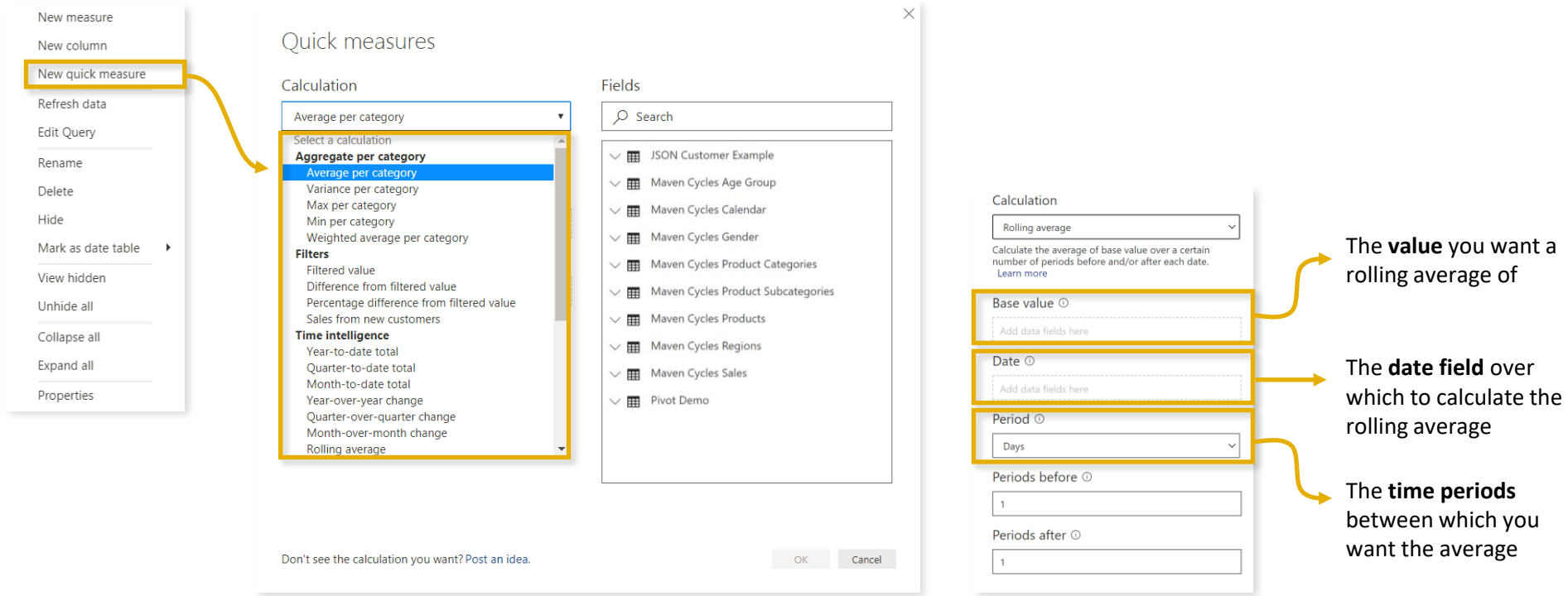
*Measures "live" in **visuals***

QUICK MEASURES

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

Quick measures are pre-built formula templates that allow you to drag and drop fields rather than write DAX from scratch

- They are helpful for defining complex measures *(like weighted averages or time intelligence formulas)*



COMMON DAX FUNCTION CATEGORIES

Calculated Columns & Measures

- **Aggregation Functions**
- **CALCULATE**
- **CALCULATE Modifiers**
- **Table Functions**
- **Time Intelligence**

MATH & STATS Functions

*Basic **aggregation** functions as well as "**iterators**" evaluated at the row-level*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

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

LOGICAL Functions

*Functions that return information about values based on a given **conditional expression***

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions to manipulate **text strings** or **control formats** for dates, times or numbers*

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

***Lookup** functions based on related tables and **filtering** functions for dynamic calculations*

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- RELATED
- RELATEDTABLE
- DISTINCT
- VALUES
- EARLIER/EARLIEST
- HASONEVALUE
- HASONEFILTER
- ISFILTERED
- USERELATIONSHIP
- TOPN

DATE & TIME Functions

*Basic **date & time** functions as well as advanced **time intelligence** operations*

Common Examples:

- DATEDIFF
- YEARFRAC
- YEAR/MONTH/DAY
- HOUR/MINUTE/SECOND
- TODAY/NOW
- WEEKDAY/WEEKNUM

Time Intelligence Functions:

- DATESYTD
- DATESQTD
- DATESMTD
- DATEADD
- DATESINPERIOD

***NOTE:** This is NOT a comprehensive list (does not include trigonometry functions, parent/child functions, information functions, or other less common functions)*

BASIC AGGREGATION FUNCTIONS

- Calculated Columns & Measures
- **Aggregation Functions**
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

SUM()	*Evaluates the sum of a column*	=**SUM**(ColumnName)
AVERAGE()	*Returns the average (arithmetic mean) of all the numbers in a column*	=**AVERAGE**(ColumnName)
MAX()	*Returns the largest value in a column or between two scalar expressions*	=**MAX**(ColumnName) *or* =**MAX**(Scalar1, *[Scalar2]*)
MIN()	*Returns the smallest value in a column or between two scalar expressions*	=**MIN**(ColumnName) *or* =**MIN**(Scalar1, *[Scalar2]*)
DIVIDE()	*Performs division and returns the alternate result (or blank) if div/0*	=**DIVIDE**(Numerator, Denominator, *[AlternateResult]*)

DAX SYNTAX

- Calculated Columns & Measures
- **Aggregation Functions**
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

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

Referenced **TABLE NAME**

Referenced **COLUMN NAME**

```
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*)

***Note:** This is a **"fully qualified"** column, since it's preceded by the table name -- table names with spaces must be surrounded by **single quotes**:*

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

PRO TIP: For column references, use the fully qualified name (i.e. Table[Column]); for measure references, just use the measure name (i.e. [Measure])

ITERATOR ("X") FUNCTIONS

- Calculated Columns & Measures
- **Aggregation Functions**
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

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

=**SUMX**(Table, Expression)

*Aggregation to apply to calculated rows**

Examples:

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

Table in which the expression will be evaluated

Examples:

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

Expression to be evaluated for each row of the given table

Examples:

- *[Total Orders]*
- *Sales[RetailPrice] * Sales[Quantity]*

PRO TIP: Imagine the function **adding a temporary new column** to the table, calculating the value in each row (based on the expression) and then applying the aggregation to that new column

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

DIVIDE

- Calculated Columns & Measures
- **Aggregation Functions**
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

DIVIDE() — *Safe Divide function with ability to handle divide by zero cases*

=**DIVIDE**(Numerator, Denominator, *[AlternateResult]*)

Numerator of the equation

***Examples**:*
- *4,7,10, etc.*
- *[Total Returns]*
- *[Sales Amount]*

Denominator of the equation

***Examples**:*
- *2, 3, 8, etc.*
- *[All Returns]*
- *CALCULATE([Sales Amount], ALLSELECTED('Products'[Category])*

Optional parameter to specify a result in case of divide by zero

***Examples**:*
- *"-"*
- *"N/A"*
- *0*

CALCULATE

- Calculated Columns & Measures
- Aggregation Functions
- **CALCULATE**
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

CALCULATE() — *Evaluates a given expression or formula under a set of defined filters*

=**CALCULATE**(Expression, *[Filter1], [Filter2],…*)

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

Examples:

- *[Total Orders]*
- *SUM(Returns[ReturnQuantity])*

*List of simple Boolean (True/False) filter expressions (**Note:** these require simple, fixed values; you cannot create filters based on other measures)*

Examples:

- *Territory_Lookup[Country] = **"USA"***
- *Calendar[Year] > **1998***

PRO TIP: CALCULATE works just like SUMIF or COUNTIF in Excel, except it can evaluate measures based on ANY sort of calculation *(not just sum, count, etc.)*; it may help to think of it like "CALCULATEIF"

CALCULATE (EXAMPLE)

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

We've defined a new measure named "**Australian Orders**", which evaluates the "**Total Orders**" measure when the *Country* in the **Regions** table equals "**Australia**"

```
1  Australian Orders = CALCULATE([Total Orders], 'Maven Cycles Regions'[Country] = "Australia")
```

Country	Total Orders	Australian Orders
United States	39,206	23,936
Australia	23,936	23,936
Canada	14,178	23,936
United Kingdom	13,620	23,936
Germany	11,098	23,936
France	10,998	23,936
Total	**113,036**	**23,936**

*Wait, why do we see **repeating values** when we view a matrix with different countries on rows?*

*Shouldn't these cells have different filter contexts for **Canada**, **Germany**, **France**, etc.?*

HEY THIS IS IMPORTANT!

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

In this example, the "France" row has filter context of Country = "**France**" (*defined by the row label*) ***and*** Country= "**Australia**" (*defined by the CALCULATE function*)

Both can't be true at the same time, "**France**" filter is overwritten, and the "**Australia**" filter (from CALCULATE) takes priority

CALCULATE **CREATES NEW** FILTER CONTEXT

- Calculated Columns & Measures
- Aggregation Functions
- **CALCULATE**
- CALCULATE Modifiers
- Table Functions
- Time Intelligence

CALCULATE

Filters modified by CALCULATE

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

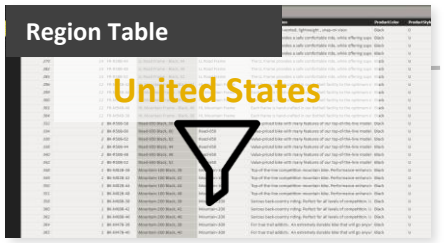
[**Country**] = "Australia"

STEP 1

Filter context is detected & applied

Country	Total Orders	Australian Orders
United States	39,206	23,936
Australia	23,936	23,936
Canada	14,178	23,936
United Kingdom	13,620	23,936
Germany	11,098	23,936
France	10,998	23,936
Total	**113,036**	**23,936**

Regions[**Country**] = "United States"

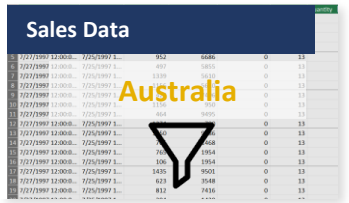


STEP 2

Filters flow "downstream" to all related tables

1

*



STEP 3

Measure formula evaluates against the filtered table

```
Orders = COUNTROWS('Maven Cycles Sales Fact Table')
```

*Count of the rows in the **Sales Data** table, filtered down to only rows where the country is "Australia"* = **23,936**

COMMON CALCULATE MODIFIERS

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- **CALCULATE Modifiers**
- Table Functions
- Time Intelligence

Modifiers are used to alter the way CALCULATE creates filter context, and are added as *filter* arguments within a CALCULATE function

- Modifiers are typically used to change filter context, access inactive table relationships, or change the way filters propagate *(i.e. one-way to bidirectional)*

Modify **Filters**

Common Examples:

- ALL
- ALLSELECTED
- ALLNOBLANKROW
- ALLEXCEPT
- KEEPFILTERS
- REMOVEFILTERS

Use **Relationships**

Common Examples:

- USERELATIONSHIP

Change **Filter Propagation**

Common Examples:

- CROSSFILTER

CALCULATE MODIFIERS

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- **CALCULATE Modifiers**
- Table Functions
- Time Intelligence

=**CALCULATE**(Expression, *[Filter1], [Filter2],…*)

CALCULATE filter expressions accept both Boolean & table functions (individually or at the same time!), but all filter arguments are automatically converted into a table

```
1  Store 3 Sales =
2  CALCULATE(
3      [Customer Sales],
4          'Store Lookup'[store_id] = 3
5  )
```

DAX interprets this as a table!

- Any time you use write a function that contains a logical statement (IN, >,<, =, etc.) you're **creating a table** *(internally processed with FILTER & ALL)*

```
1  All Store Sales =
2  CALCULATE(
3      [Customer Sales],
4      ALL(
5          'Store Lookup'
6      )
7  )
```

Filter arguments can be table functions too

- In this case we're using the ALL table function to remove all filters from the 'Store Lookup' table

USERELATIONSHIP

Calculated Columns & Measures
Aggregation Functions
CALCULATE
CALCULATE Modifiers
Table Functions
Time Intelligence

USERELATIONSHIP()

Specifies an existing relationship to be used in the evaluation of a DAX expression, defined by naming, as arguments, the two columns that serve as endpoints

=USERELATIONSHIP(ColumnName1, ColumnName2)

***Foreign** (or primary) key of the relationship*

Examples:

- *Food Inventory[Baked_Date]*
- *Calendar[Transaction Date]*

***Primary** (or foreign) key of the relationship*

Examples:

- *Calendar[Transaction Date]*
- *Food Inventory[Baked_Date]*

HEY THIS IS IMPORTANT!

USERELATIONSHIPS can only be used in functions which accept a filter parameter *(CALCULATE, TOTALYTD, etc.)*

PRO TIP: *If you have **multiple date columns** connected to a single calendar table, USERELATIONSHIP is a great way to force measures to use **inactive relationships** without having to manually activate them in your model*

ALL

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

ALL()

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

=ALL(Table *or* ColumnName, *[ColumnName1], [ColumnName2],…*)

The table or column that you want to clear filters on

Examples:

- *'Sales'*
- *'Products'[ProductName]*

List of columns that you want to clear filters on (optional)

Notes:

- *If your first parameter is a table, you can't specify additional columns*
- *All columns must include the table name, and come from the same table*

Examples:

- *'Products'[Product Type]*
- *'Products'[Product Size]*

PRO TIP: Instead of adding filter context, ALL **removes it;** use it when you need unfiltered values that won't react to changes in filter context *(i.e. **% of Total**, where the denominator needs to remain fixed)*

FILTER

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- CALCULATE Modifiers
- **Table Functions**
- Time Intelligence

FILTER() — *Returns a table that represents a subset of another table or expression*

=**FILTER**(Table, FilterExpression)

Table to be filtered

Examples:

- *Gender Lookup*
- *Customer Lookup*

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

Examples:

- *Calendar[Year]* **= *2020***
- *Products[Price]* **> *[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 almost always used as an *input* to other functions, like CALCULATE or SUMX

PRO TIP: Since FILTER iterates through each row in a table, it can be slow and processor-intensive; don't use FILTER if a CALCULATE function will accomplish the same thing

TOPN

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- CALCULATE Modifiers
- **Table Functions**
- Time Intelligence

TOPN()

Returns a given number of top rows according to a specified expression

=**TOPN**(N_Value, TableName, *[OrderBy Expression], [Order]*)

The number of rows to return

Examples:

- *100*
- *50*
- *10*
- *5*

Name of a table or table expression that you want to return rows from

Examples:

- *'Maven Cycles Sales'*
- *ALL('Maven Cycles Sales')*
- *SUMMARIZE('Sales',*
  'Sales'[Customer_ID],
  "Sales",
  SUM('Sales by Store'[Sales]))

Optional expression that's used to sort the table

Examples:

- *'Sales'*

Optional expression that defines the sort order

Examples:

- *DESC*
- *ASC*

PRO TIP: TOPN is a great technique to use when you want to **show the top "X"** number in visuals and not add a visual-level Top N filter

TIME INTELLIGENCE FORMULAS

- Calculated Columns & Measures
- Aggregation Functions
- CALCULATE
- CALCULATE Modifiers
- Table Functions
- **Time Intelligence**

Time Intelligence functions allow you to easily calculate common time comparisons:

Performance To-Date

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

Use **DATESQTD** *for Quarters or* **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 (i.e. previous month, rolling 10-day)*

Running Total

```
=CALCULATE([Measure],
        DATESINPERIOD(Calendar[Date], MAX(Calendar[Date]), -10, DAY))
```

PRO TIP: Use DATESBETWEEN to return dates between two given dates

VISUALIZING THE DATA

Prepare the Data

- Get data from different sources
- Clean, transform, and load data

Model the Data

- Design a data model
- Develop a data model
- Create model calculations with DAX
- Optimize model performance

Visualize & **Analyze** the Data

- Create reports
- Create dashboards
- Enrich reports for usability
- Enhance reports for usability & storytelling
- Identify patterns & trends

Deploy & **Maintain** Assets

- Manage files & datasets
- Manage workspaces

CREATING REPORTS & DASHBOARDS

In this section we'll cover **creating reports & dashboards**, including adding & formatting visuals, interacting with reports, and publishing to Power BI Service

TOPICS WE'LL COVER:

- **Inserting Visuals**
- **Formatting Visuals**
- **Report Navigation**
- **Tooltips**
- **Custom Visuals**
- **R & Python Visuals**
- **Accessibility**
- **Dashboards**

COMMON QUESTIONS:

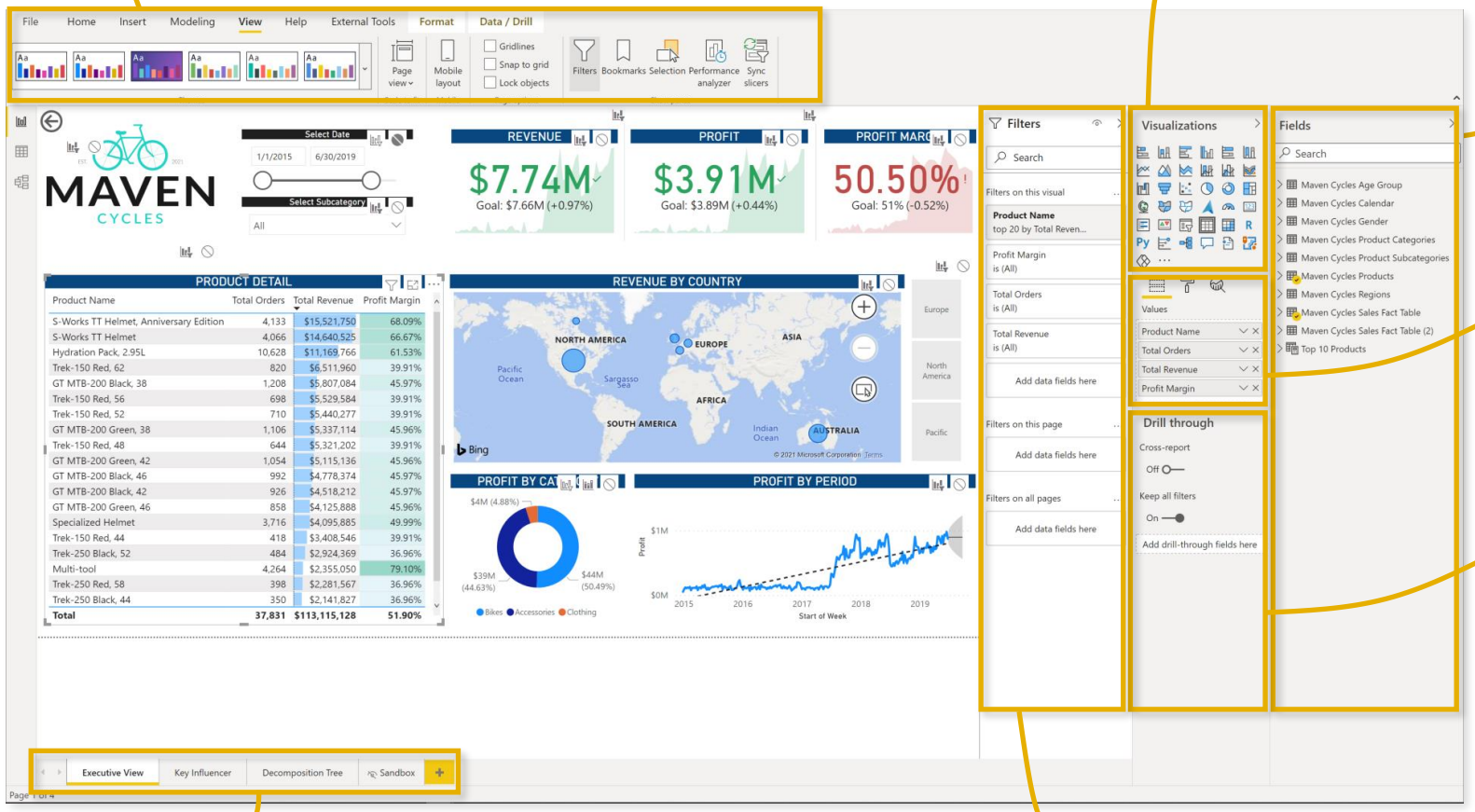
- *How would you adjust visual interactions so a segment selection on one chart shows the portion of sales on the other chart?*
- *What feature allows you to see additional detail about a data point when you hover?*
- *You need to build a report that is optimized for viewing on the Power BI mobile app. What steps are needed to build a mobile report?*

THE POWER BI REPORT VIEW

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

VIEW OPTIONS
Themes, Layouts, Gridlines, Filter/ Bookmarks/Selection Panes, etc.

VISUALIZATION OPTIONS
Charts, Slicers, Maps, Matrices, etc.



FIELD LIST
Tables, Columns, Measures

FIELDS/FORMAT/ ANALYICTS PANE
Visual-specific configuration & formatting tools

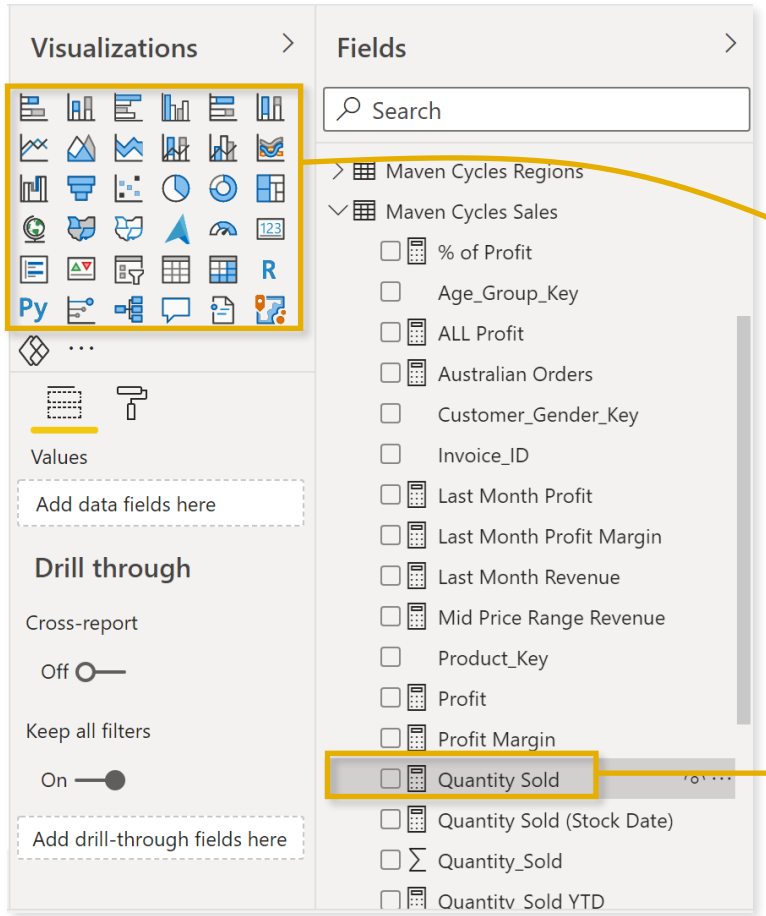
DRILL THROUGH FILTERS
Options for page-level drill through filters

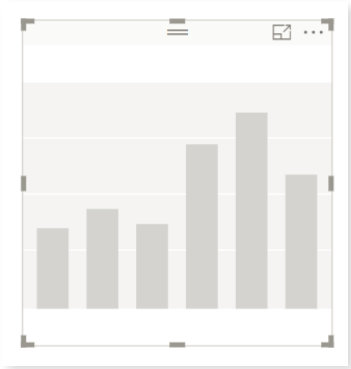
REPORT PAGES
Similar to Excel tabs; each is a blank reporting canvas

FILTERS PANE
Visual-Level, Page-Level, and Report-Level Filters

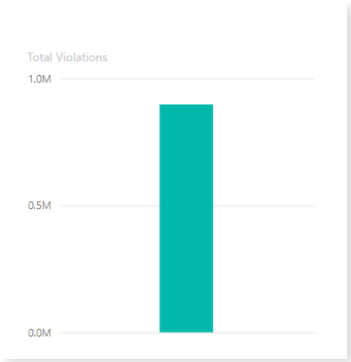
INSERTING OBJECTS & BASIC CHARTS

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards





(or)



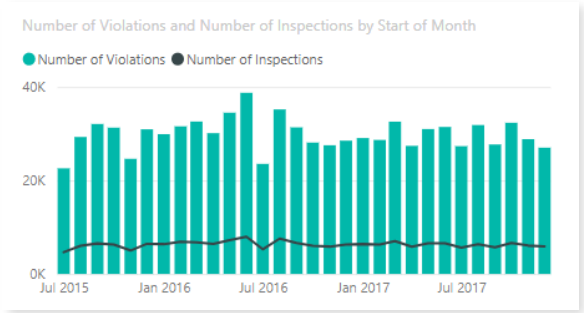
Select a **visualization type** to create a blank chart template on the canvas

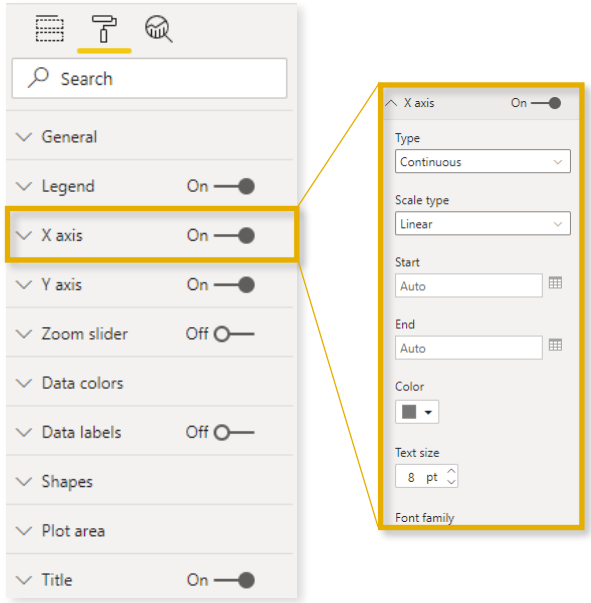
Drag **fields** into the report canvas to automatically generate a new visual *(typically a column chart, by default)*

FORMATTING OPTIONS

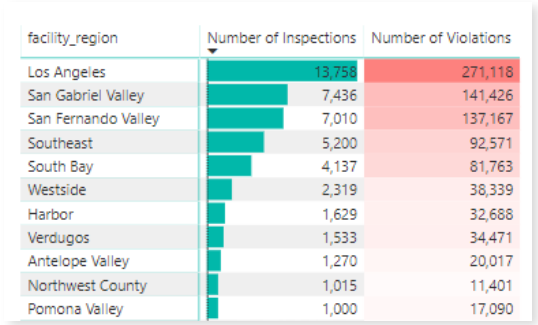
- Inserting Visuals
- **Formatting Visuals**
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

Example: ***Line & Column Chart***

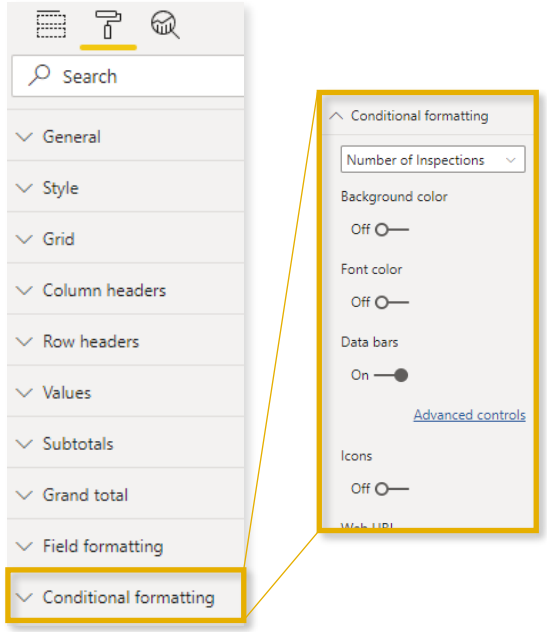




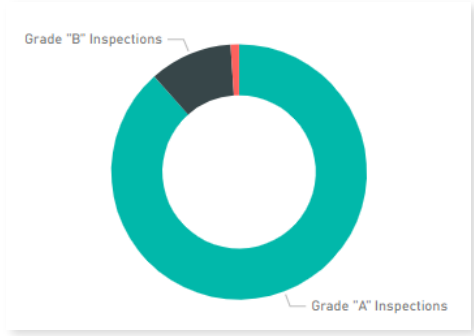
Example: ***Matrix***

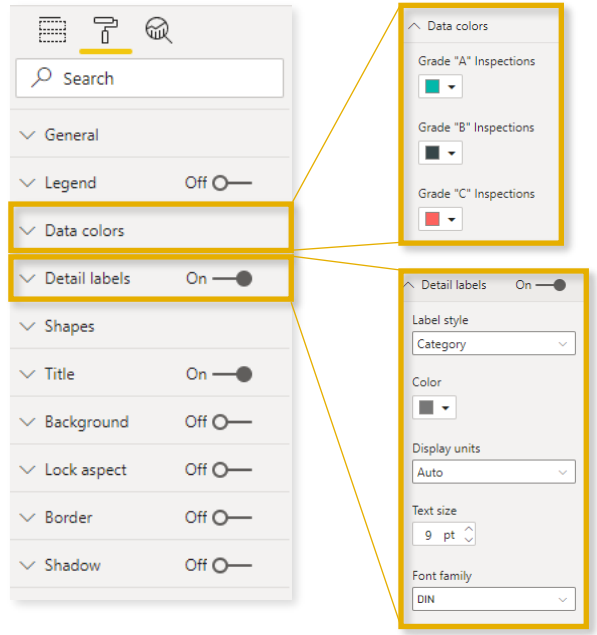


facility_region	Number of Inspections	Number of Violations
Los Angeles	13,758	271,118
San Gabriel Valley	7,436	141,426
San Fernando Valley	7,010	137,167
Southeast	5,200	92,571
South Bay	4,137	81,763
Westside	2,319	38,339
Harbor	1,629	32,688
Verdugos	1,533	34,471
Antelope Valley	1,270	20,017
Northwest County	1,015	11,401
Pomona Valley	1,000	17,090



Example: ***Donut Chart***

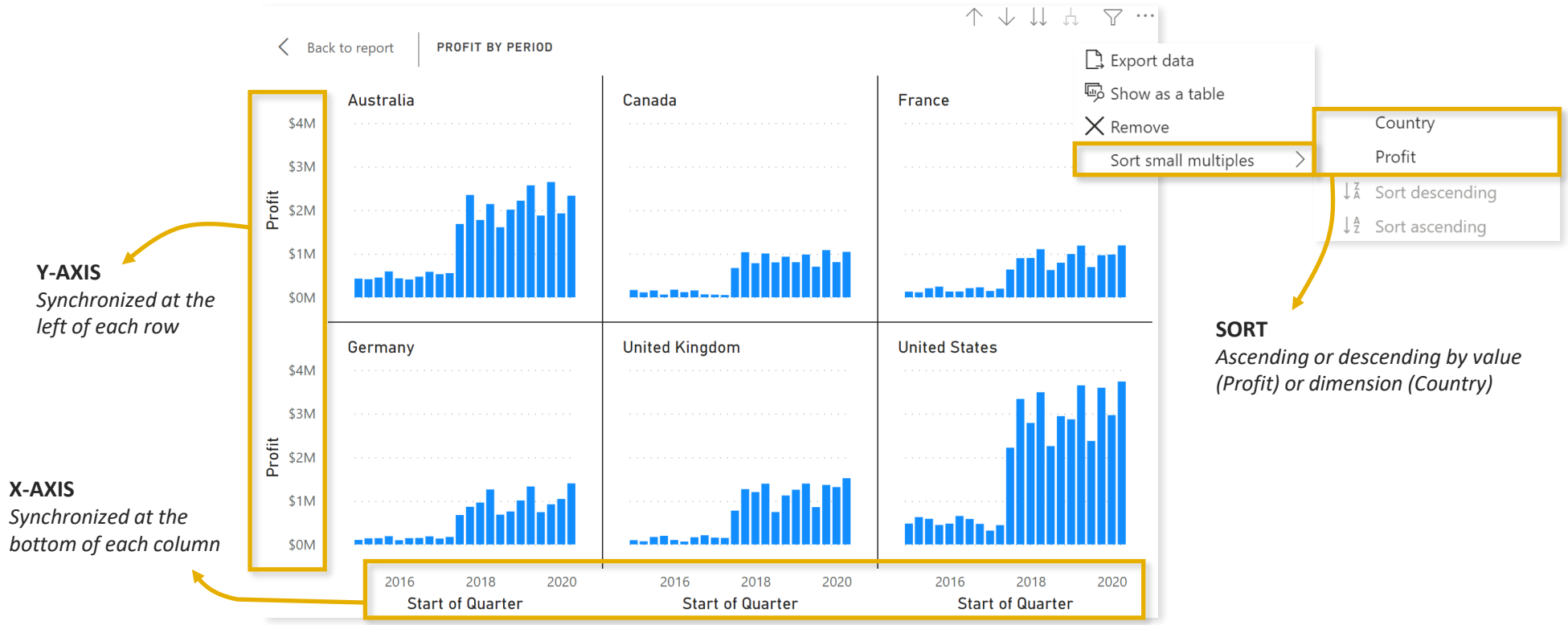




PRO TIP: CONFIGURE SMALL MULTIPLES

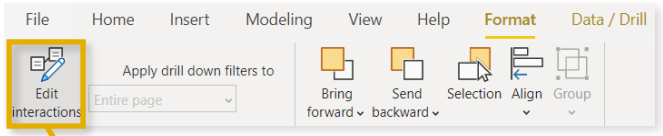
Small multiples, or trellising, splits a visual into multiple versions of itself that are presented side by side, with the data divided across a dimension

- **NOTE**: Currently, you can only create small multiples on bar, column, line, and area charts

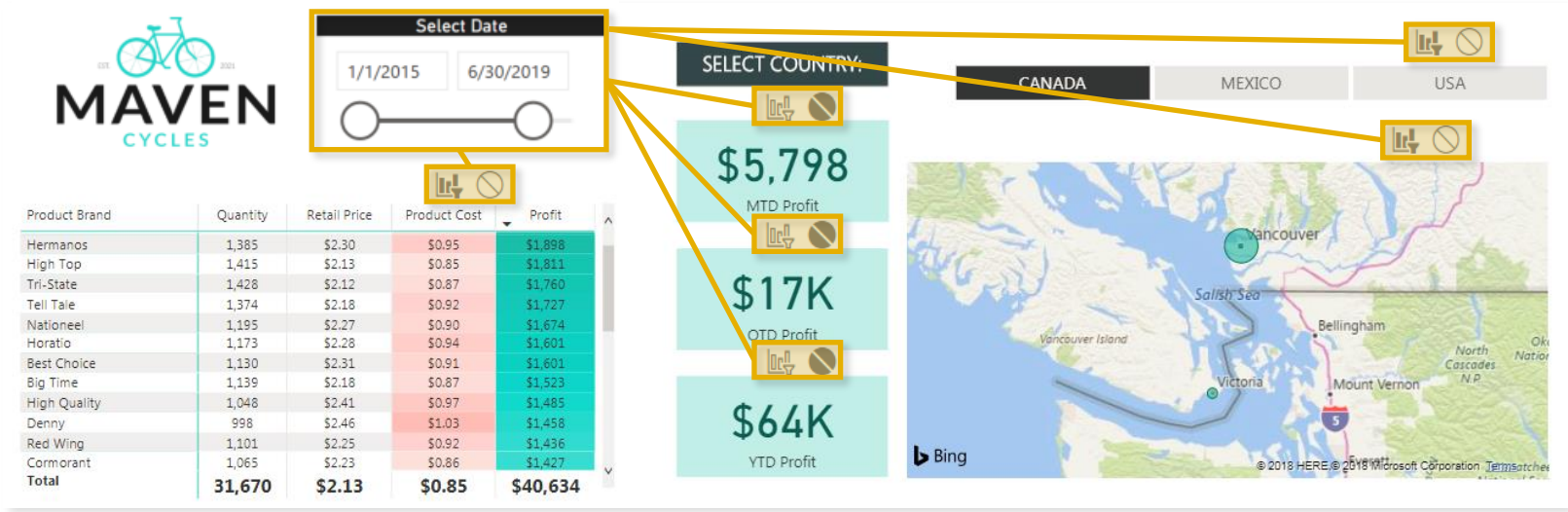


EDITING REPORT INTERACTIONS

Report interactions allow you to define how filters applied to *one* visual impact the *others*



*By selecting the date slicer and enabling "Edit interactions" from the **Format** tab, we can manually determine which visuals should "react" when the date range changes*

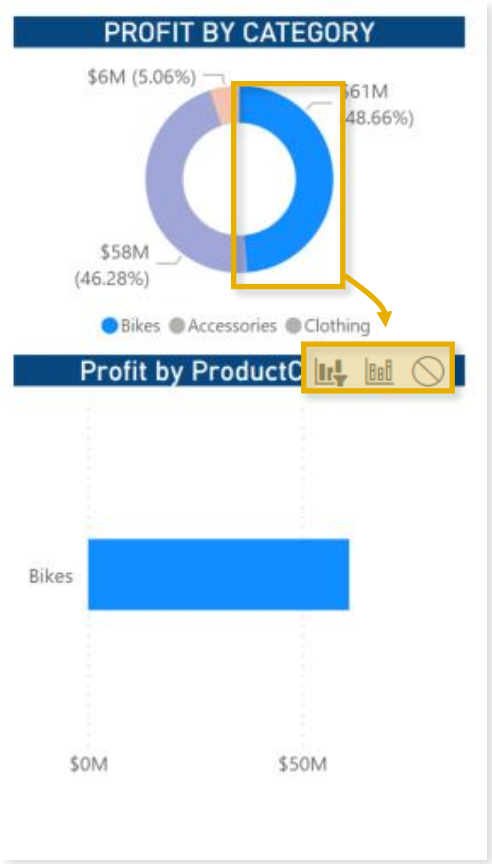


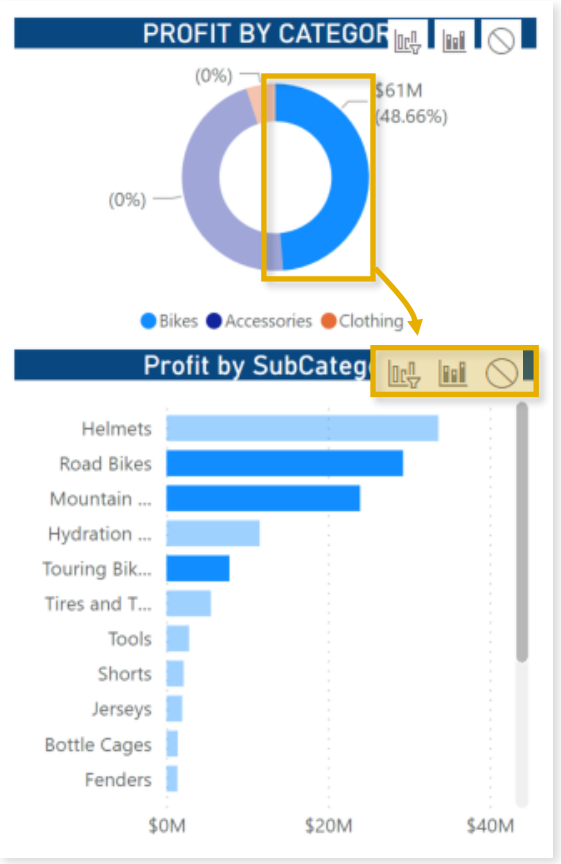
*In this case the **Product matrix**, **Country slicer** and **Map** will filter in response to date slicer changes ( ), but the **MTD**, **QTD**, and **YTD Profit** cards will not ( )*

EDITING REPORT INTERACTIONS (CONT.)

For certain types of visuals, a third option allows you to "**highlight**" subsegments of the data, rather than simply filtering vs. not filtering

*When the interaction mode is set to "**filter**", selecting the "**Bikes**" category in the donut chart produces a filtered list of subcategories in the chart*



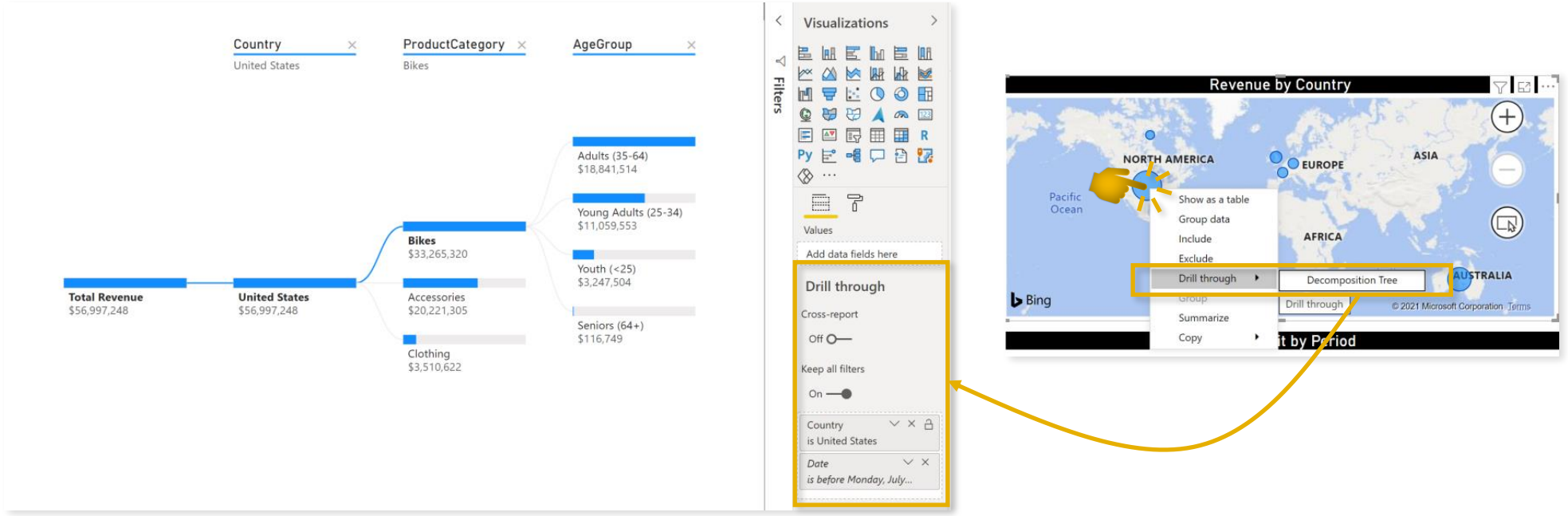


*When the interaction mode is set to "**highlight**", selecting the "**Bikes**" category in the donut chart highlights the relevant subsegments in the chart*

DRILL-THROUGH FILTERS

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

Drill-through filters allow users to jump to different report pages (*like bookmarks*), while simultaneously filtering based on the *specific item selected*

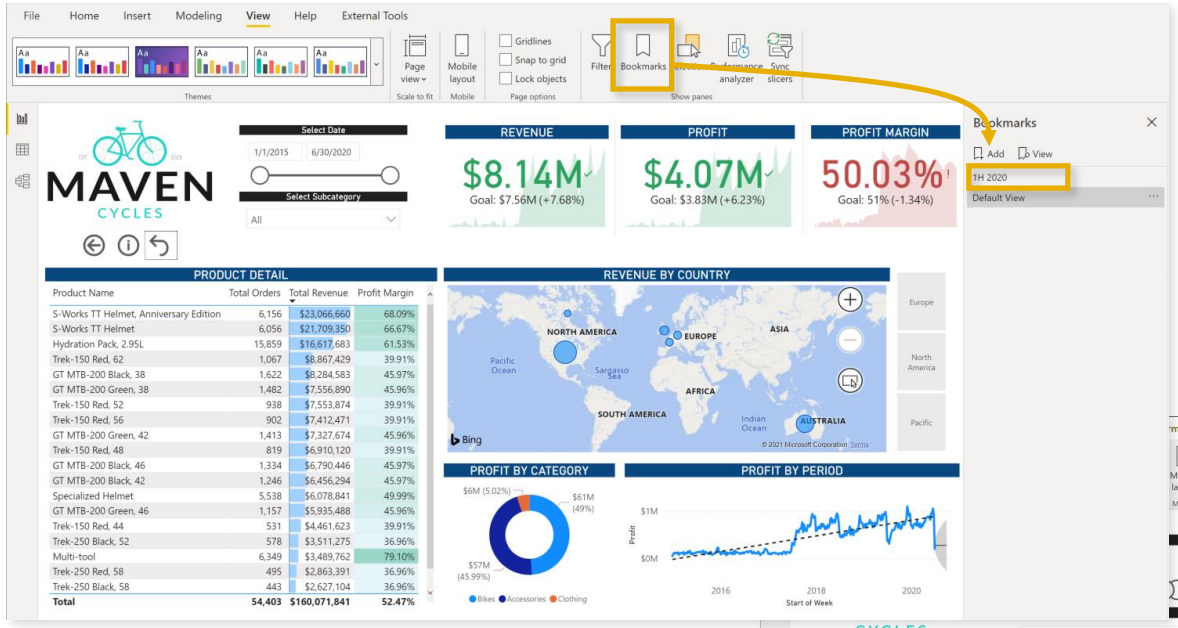


*Here we've built a report page ("**Decomposition Tree**") featuring country, product, and age detail, and added a Drillthrough filter for **Country***

*Users can now right-click any report visual containing country name, and jump straight to a pre-filtered version of this page ("**United States**" shown in the example above)*

ADDING & LINKING BOOKMARKS

- Inserting Visuals
- Formatting Visuals
- **Report Navigation**
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

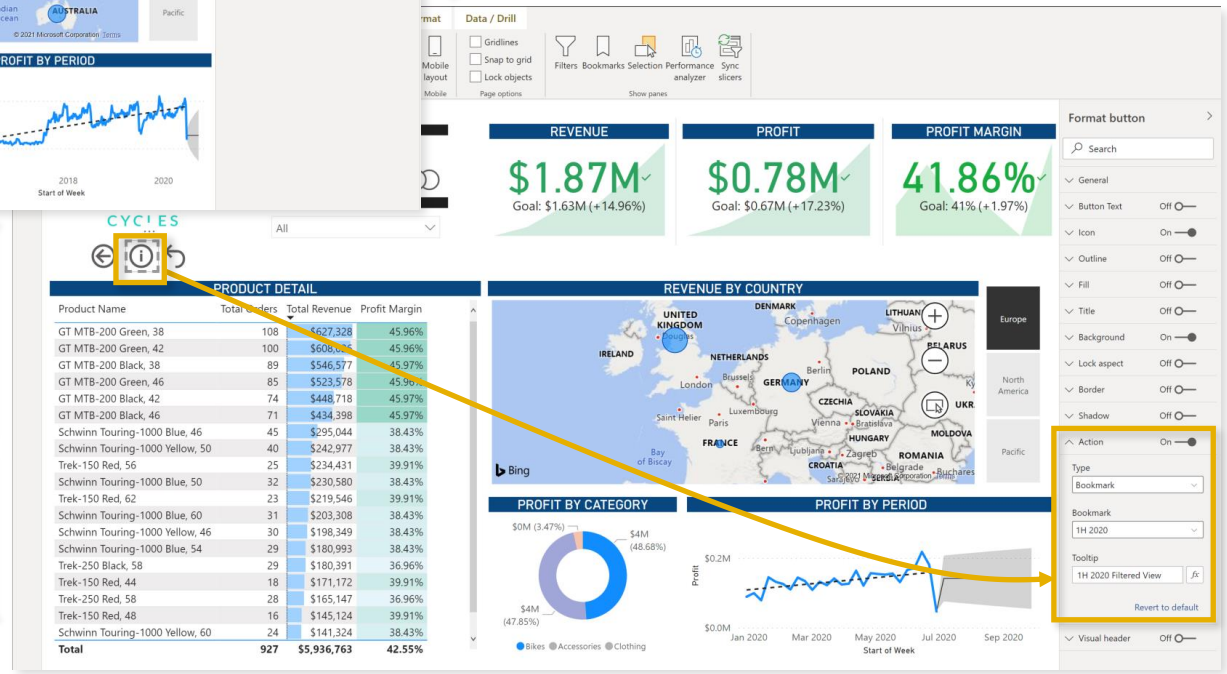


In this example, we created a prefiltered view of 1H 2020 for the executive team's upcoming planning session.

To do this, we add a new **bookmark** (*View > Bookmarks Pane > Add*) and name it "**1H 2020**"

On the report page, we add our ⓘ button, and link it to bookmarks using the object "***Action***" properties

Now we're able to create a *narrative* from the data, and really bring our insights to life!



TOOLTIPS

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- **Tooltips**
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

Tooltips in Power BI are a way to add additional data when you hover over a visual

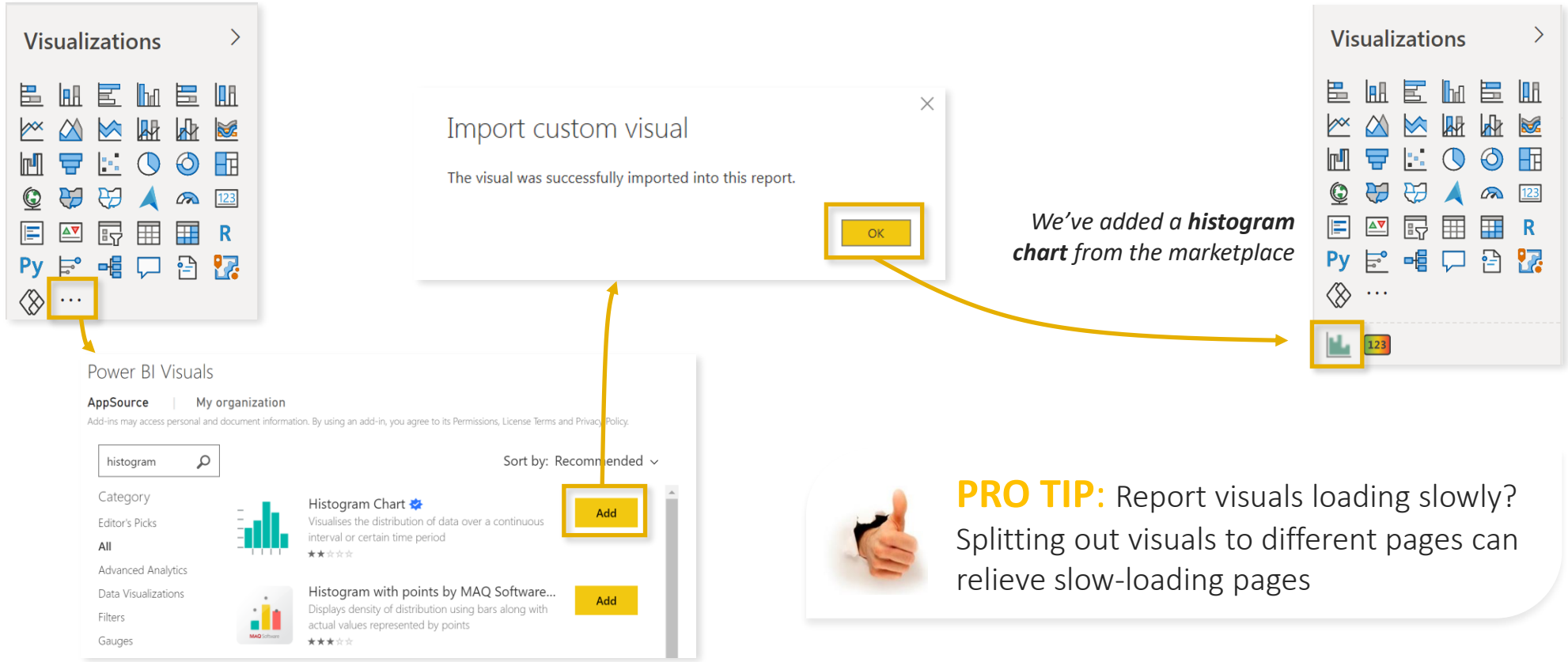
*Map that shows the **Total Revenue** by country, filtered to Europe*

***Tooltip** shows the Total Revenue along with Total Orders, Profit, & % of Profit for France*

IMPORTING CUSTOM VISUALS

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

Import custom visuals from files, your organization, or the AppSource marketplace *(requires sign-in)* directly into Power BI



PRO TIP: Report visuals loading slowly? Splitting out visuals to different pages can relieve slow-loading pages

R & PYTHON VISUALS

You can create visuals from queries and datasets generated in **R** or **Python**

Getting Started:

1. Install R on local machine
2. Confirm R home directory

Limitations:

- Plots limited to 150,000 rows
- Visual output limit of 2MB
- Resolution is 72 DPI
- Calculations > 5 minutes will time out

Getting Started:

1. Install Python on local machine
2. Install necessary packages/libraries *(Matplotlib, NumPy)*
3. Confirm Python home directory

Limitations:

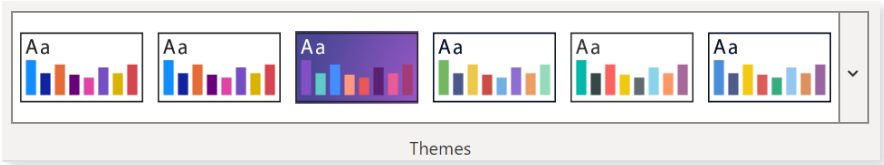
- You won't be able to use Python scripts in reports created with Enhanced Metadata (Preview) feature enabled

For information about known limitations head to:
https://docs.microsoft.com/en-us/power-bi/create-reports/desktop-r-visuals#known-limitations *or* ***https://docs.microsoft.com/en-us/power-bi/connect-data/desktop-python-scripts#known-limitations***

ACCESSIBILITY FEATURES

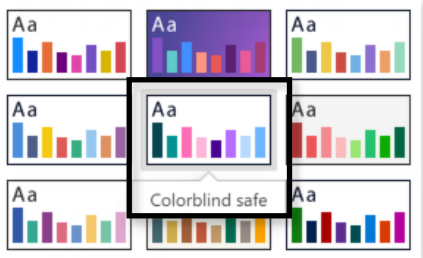
- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

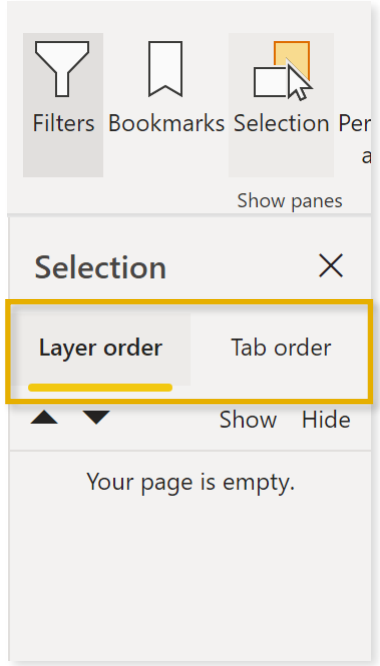
Power BI's different **accessibility features** allow consumers to easily navigate reports



Themes allow you to set a default theme for your entire report

PRO TIP: Use the **colorblind safe** theme to make sure all users can distinguish your report colors





The **selection** pane allows you to adjust the tab and layer order for reports

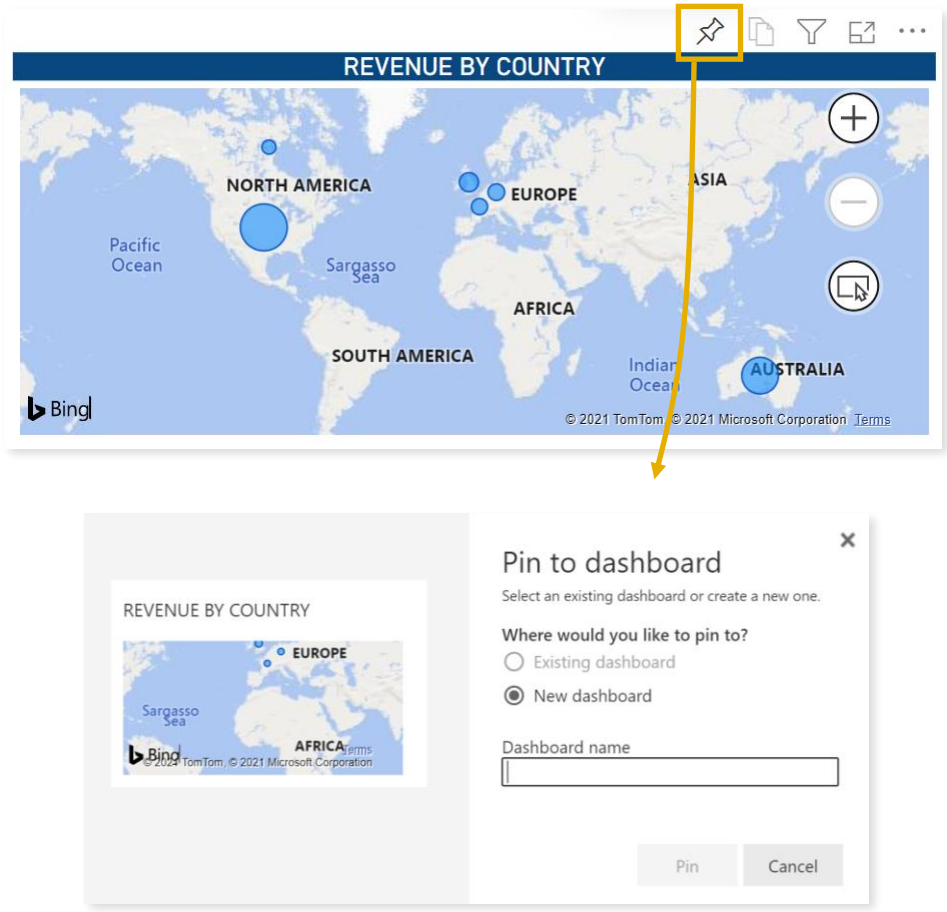
PRO TIP: Use this on complex reports with overlapping objects

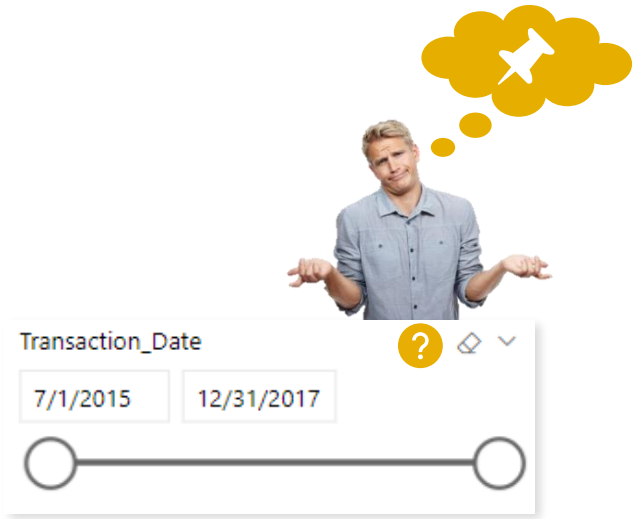
- **Tab order** allows you to override the default creation order and specify how "tab" cycles between visuals
- **Layer order** allows you to change the default layer order and send objects back or bring the forward

PINNING TILES TO A DASHBOARD

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

Use the pushpin icon to **pin an individual visual** to a dashboard





HEADS UP!

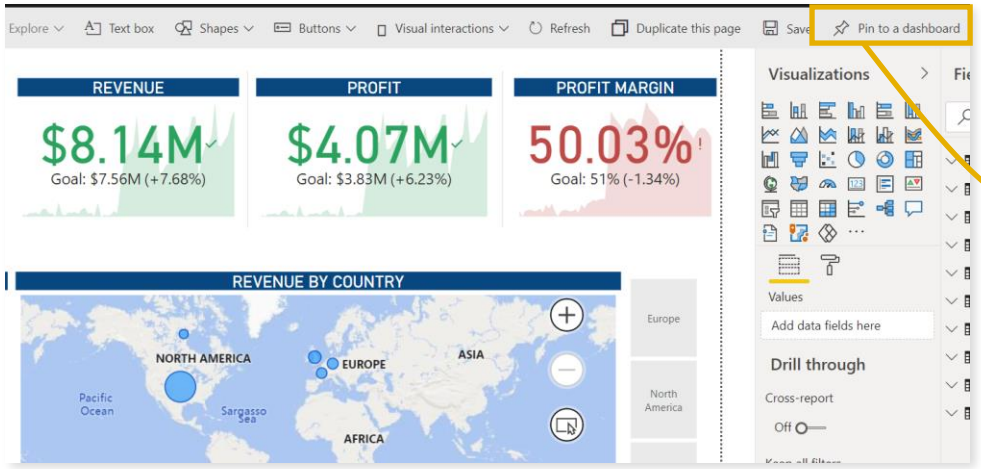
Slicers have some limitations in Power BI, and cannot:

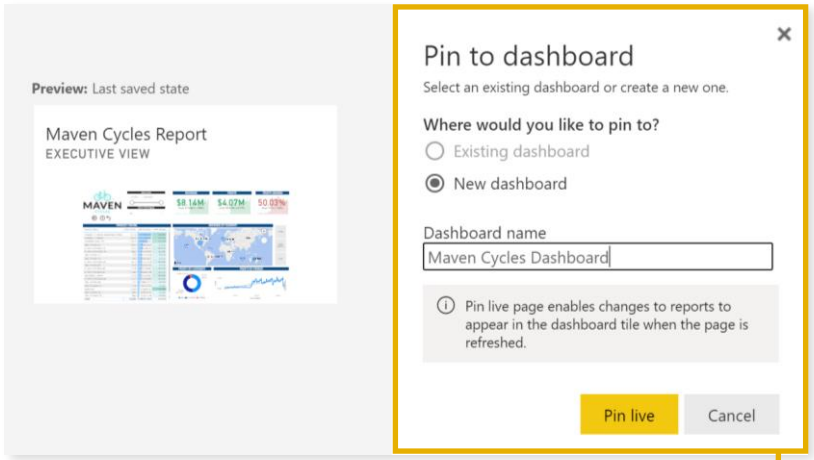
- *Drill down non-hierarchical fields*
- *Support visual level filters*
- *Be pinned to a dashboard individually (they can be pinned as part of a **live page**)*

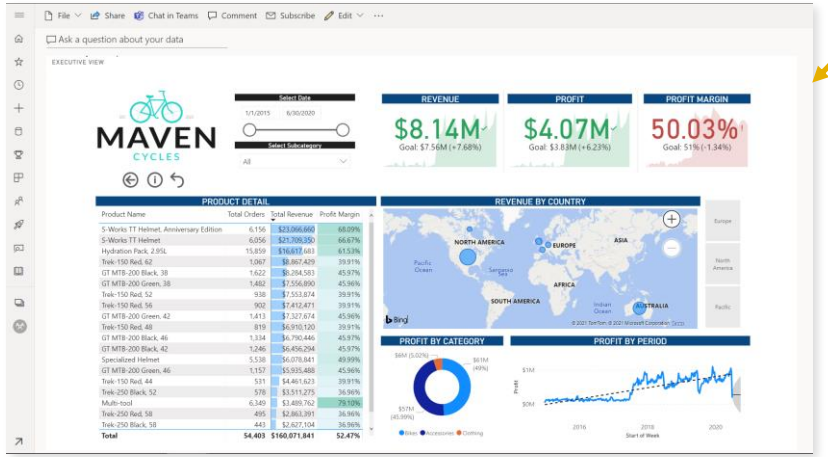
PINNING ENTIRE REPORTS TO A DASHBOARD

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

Use the **Pin Live Page** option to pin an entire report to a dashboard



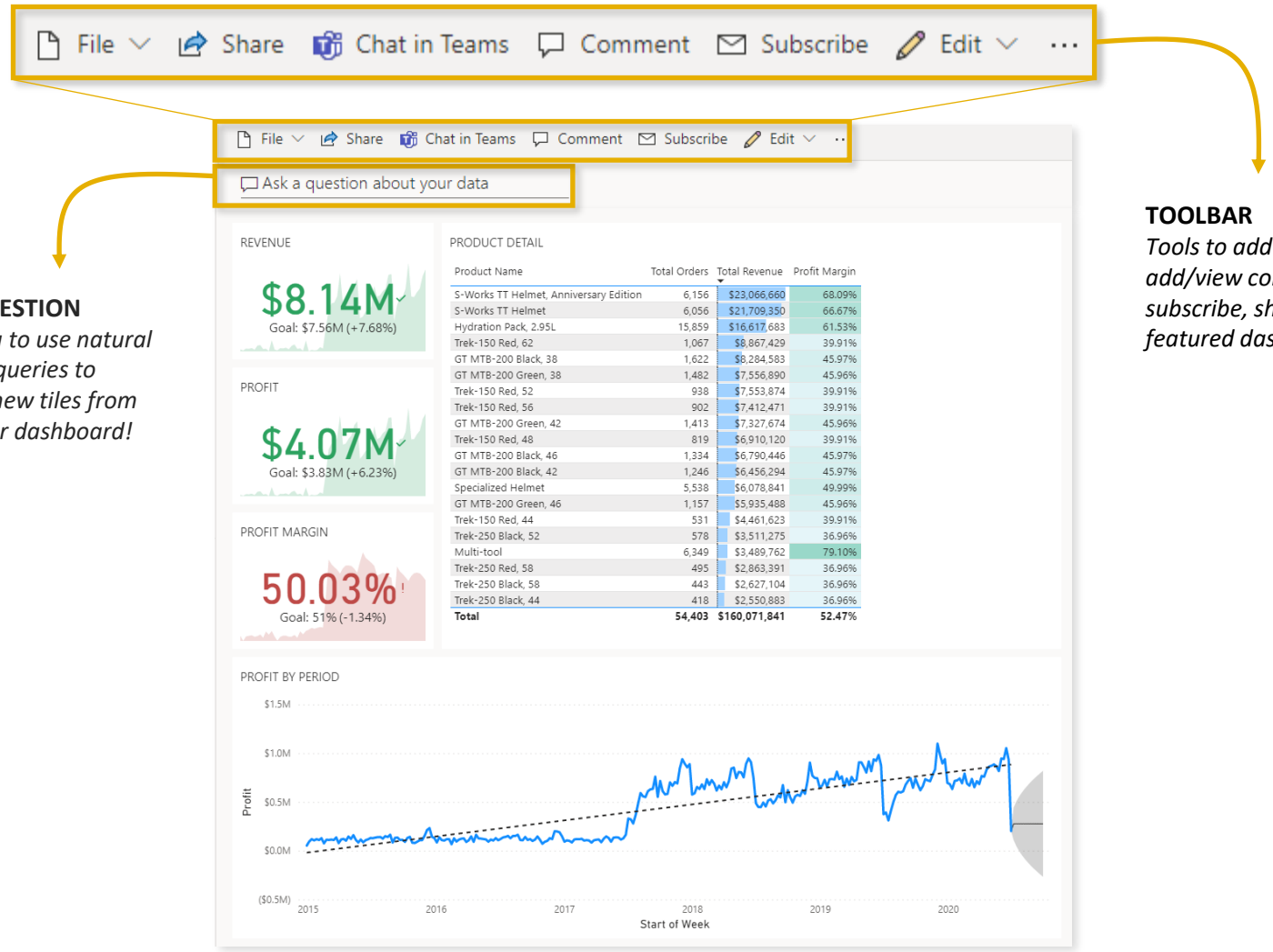




PRO TIP: Pinning a live page will **import slicers,** preserve functionality, and reflect any updates to the report

THE DASHBOARD INTERFACE

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

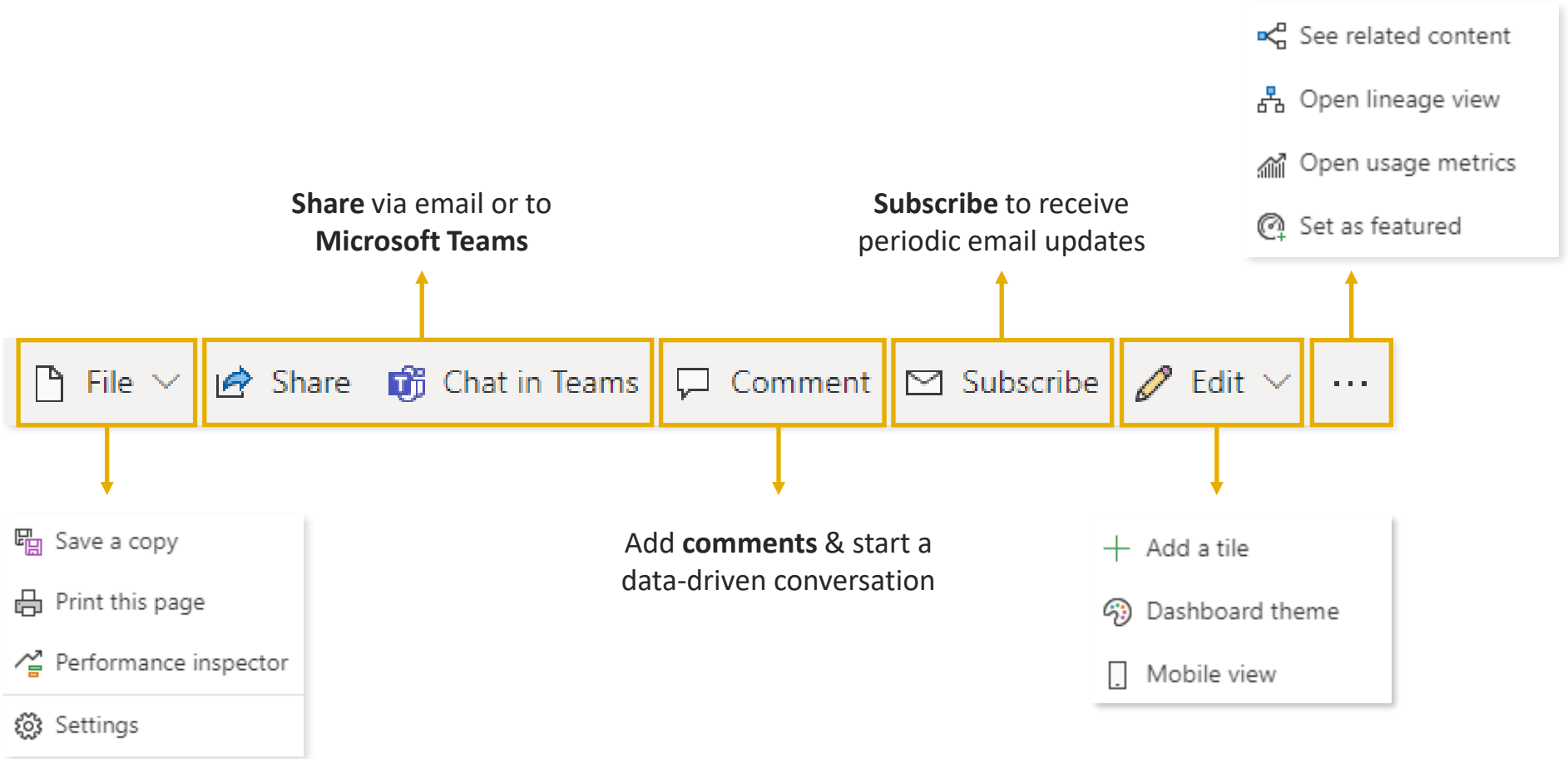


ASK A QUESTION
Allows you to use natural language queries to generate new tiles from within your dashboard!

TOOLBAR
Tools to add new tiles, add/view comments, subscribe, share, set as featured dashboard, etc.

DASHBOARD TOOLS

- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- Dashboards

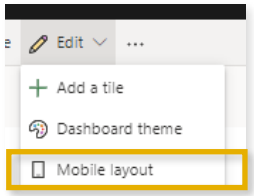


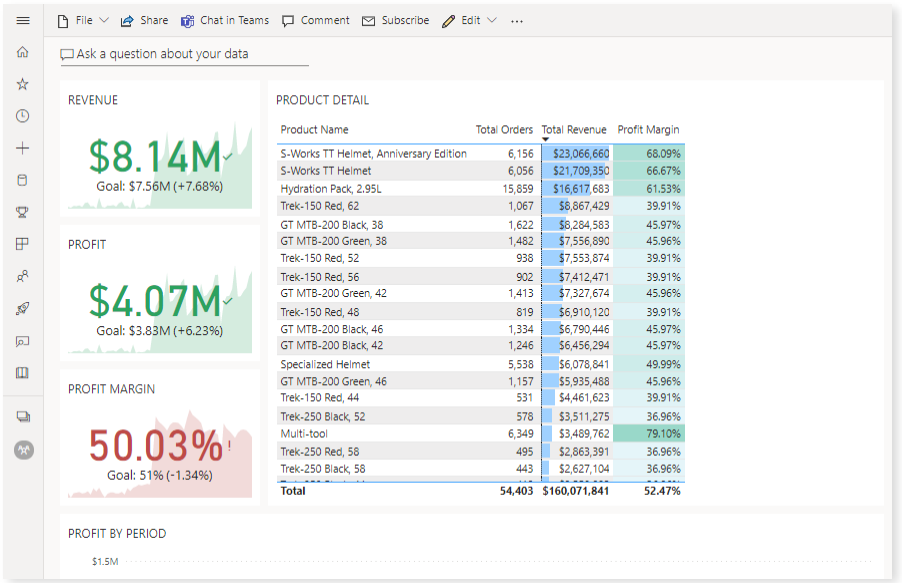
WEB VS. MOBILE LAYOUT

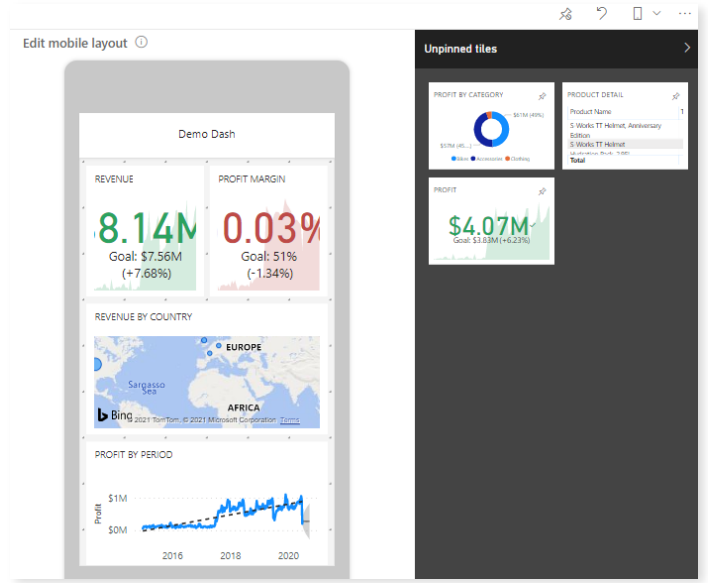
- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- **Dashboards**

Mobile Layout allows you to design reports optimized for mobile viewing (*vs. web*)

- **NOTE:** You can't **build** content within the Mobile Layout view; you need to build in **Web Layout** and assemble select visuals to share via the Power BI mobile app





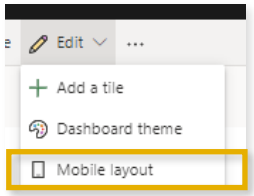


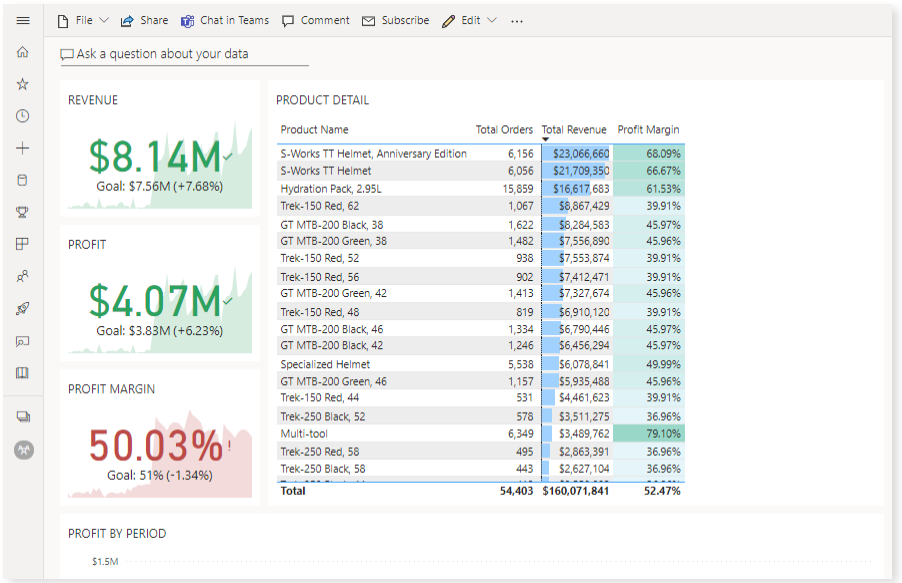
WEB VS. MOBILE LAYOUT

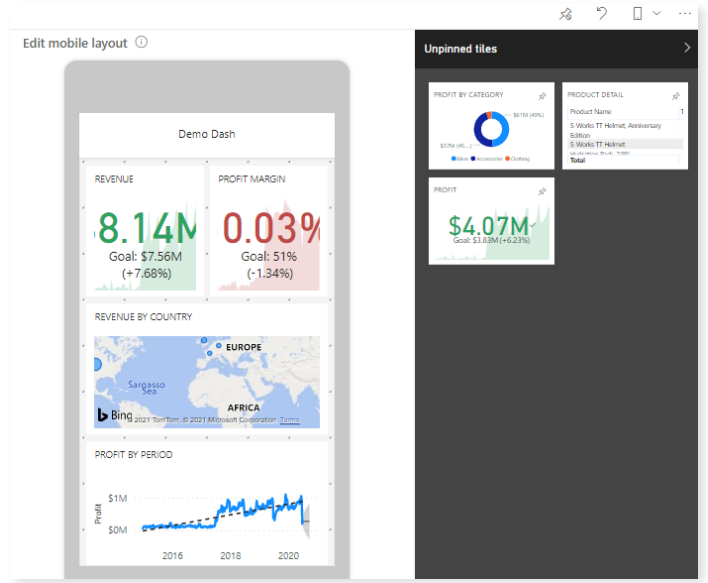
- Inserting Visuals
- Formatting Visuals
- Report Navigation
- Tooltips
- Custom Visuals
- R & Python Visuals
- Accessibility
- **Dashboards**

Mobile Layout allows you to design reports optimized for mobile viewing (*vs. web*)

- **NOTE:** You can't **build** content within the Mobile Layout view; you need to build in **Web Layout** and assemble select visuals to share via the Power BI mobile app







PRO TIP: PAGINATED REPORTS

Paginated Reports are called *paginated* because they're formatted to fit well on a page and are designed to be printed or shared

Benefits & requirements:

- Control the layout for **pixel perfect** reports
- **Print** all the data, no matter how long
- **Premium capacity** needed to publish & share
- **Pro** or **PPU** license to publish
- Apply **sensitivity labels** in Power BI Service
- **Report Builder** required to build *(separate tool)*
- Same foundation as **SSRS** *(backwards compatible)*

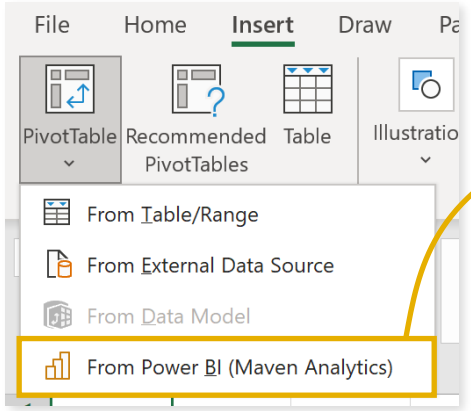
Orders & Revenue by Customer

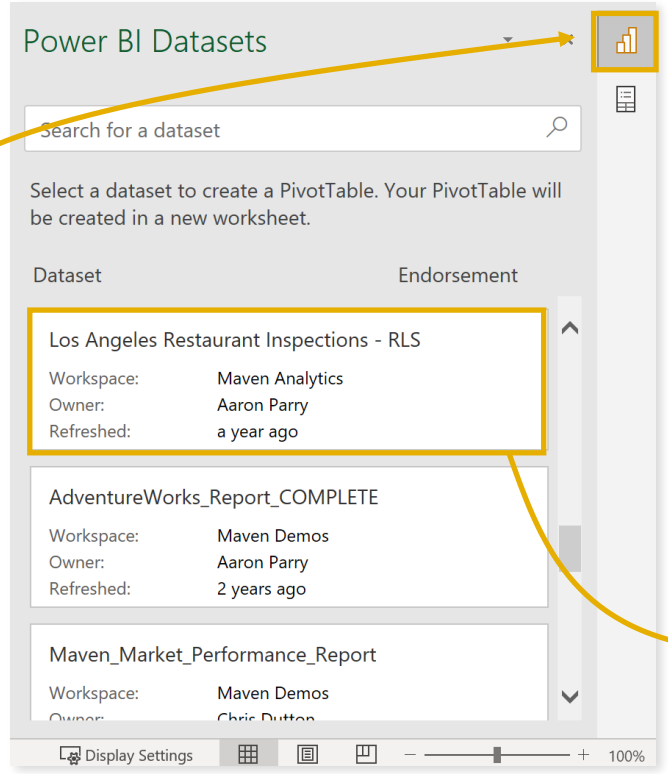
Customer Name	Product Name	Orders	Total Revenue
Abby Rana	Total	3	73.92
	Fender Set - Mountain	1	43.96
	Mountain Bottle Cage	1	19.98
	Water Bottle - 30 oz.	1	9.98
Adriana Gonzalez	Total	9	9195.69
Alejandro Beck	Total	5	2975.22
Alexa Cox	Total	3	88.02
Alexa Watson	Total	1	17.28
Alexander Jackson	Total	1	49.98
Alexandra Evans	Total	2	68.49
Alexandria Stewart	Total	1	1700.99
Alisha Liu	Total	3	2214.51

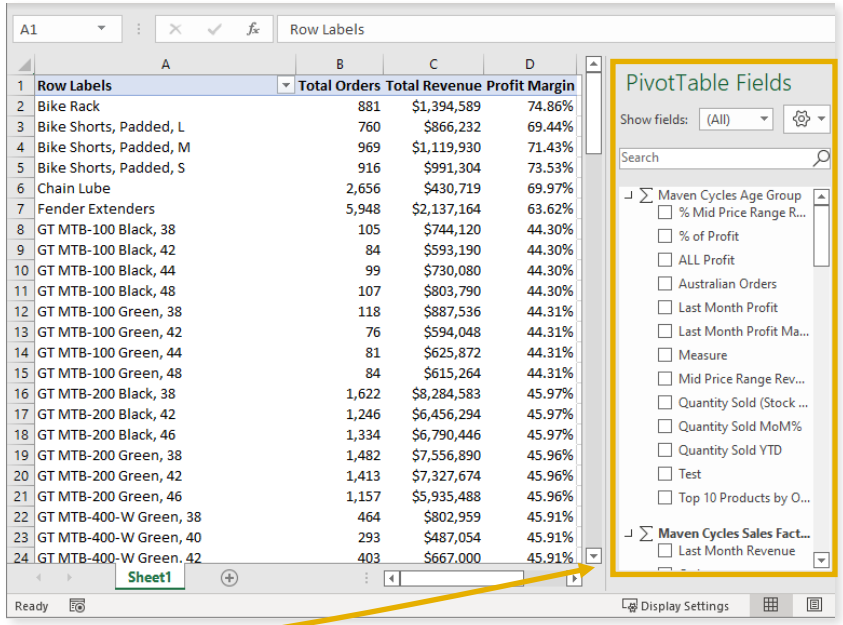
https://docs.microsoft.com/en-us/power-bi/paginated-reports/paginated-reports-report-builder-power-bi

PRO TIP: CREATING A PIVOT TABLE FROM A POWER BI DATASET

You can create **PivotTables** from **Power BI datasets** to leverage the best of both PivotTables & Power BI *(available for Microsoft 365 only)*







HEY THIS IS IMPORTANT!

Create measures first! PivotTables from Power BI datasets do not support drag-and-drop aggregation

ANALYZING THE DATA

Prepare the Data

- Get data from different sources
- Clean, transform, and load data

Model the Data

- Design a data model
- Develop a data model
- Create model calculations with DAX
- Optimize model performance

Visualize & **Analyze** the Data

- Create reports
- Create dashboards
- Enrich reports for usability
- Enhance reports for usability & storytelling
- Identify patterns & trends

Deploy & **Maintain** Assets

- Manage files & datasets
- Manage workspaces

ENHANCE REPORTS

In this section we'll cover tools and techniques that can be used to **enhance reports** to expose insights and perform advanced analysis

TOPICS WE'LL COVER:

- **Basic Chart Types**
- **Analytics Options**
- **Q&A**
- **Filtering Options**
- **AI Visuals**

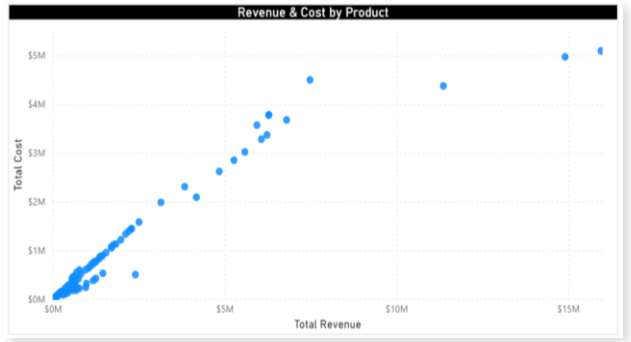
COMMON QUESTIONS:

- *You need to add a visualization to a report that helps the team understand factors that drive a specific metric. What visualization should you add?*
- *Which of the following visualizations should be used to show the relationship between cost and revenue and help identify possible outliers?*
- *Which of the following options is the best way to create a visual to show the top 10 States based on population?*

CHART TYPES BASED ON ANALYSIS

- Basic Chart Types
- Analytics Options
- Q&A
- Filtering Options
- AI Visuals

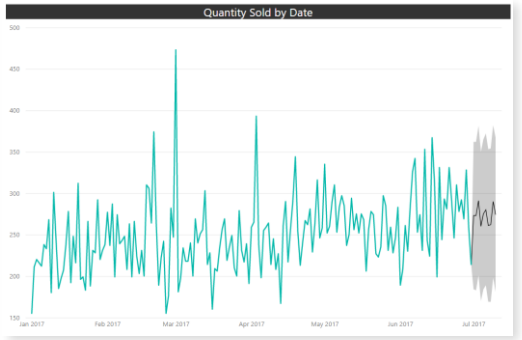
Scatter charts *show the relationship between two numerical values*



Common Uses:

- Show patterns in large sets of data
- Show linear & non-linear trends
- Cluster analysis
- Outlier identification

Line charts *are used to track changes over periods of time*



Common Uses:

- Show changes in values over time
- Add multiple lines to compare trends between series (categories)

Clustered Column charts *compare values across different categories*

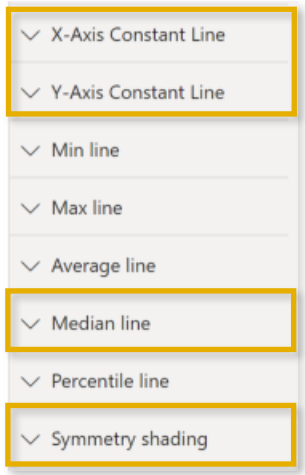
Common Uses:

- Show distribution of data points
- Comparisons across categories

CHART ANALYTICS OPTIONS

- Basic Chart Types
- Analytics Options
- Q&A
- Filtering Options
- AI Visuals

Scatter charts

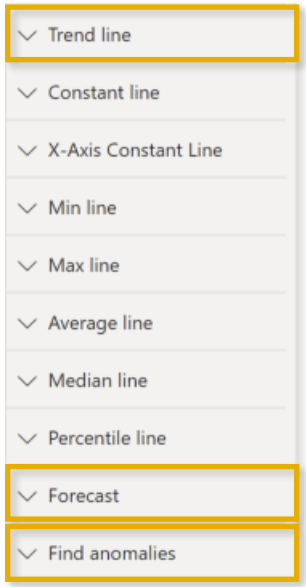


X & Y-Axis Constant line *adds a constant line to your visual*

Median line *adds a line that separates the higher half of data from lower half*

Symmetry shading *makes it easier to see if the Y or X axis value is bigger. Border is where the values are the same*

Line charts

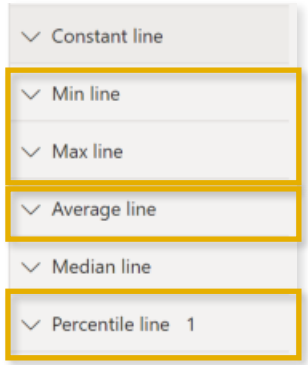


Trend line *adds a trend line to your visual*

Forecast *adds a forecast to your visual based on a specific number of periods*

Find anomalies *mark datapoint outside the expected range of values*

Clustered Column charts



Min & Max line *adds min/max context to your visual*

Average line *adds the arithmetic mean of the values in the visual*

Percentile *add a dynamic reference line based on a percentage*

PRO TIP: CLUSTERING

Basic Chart Types

Analytics Options

Q&A

Filtering Options

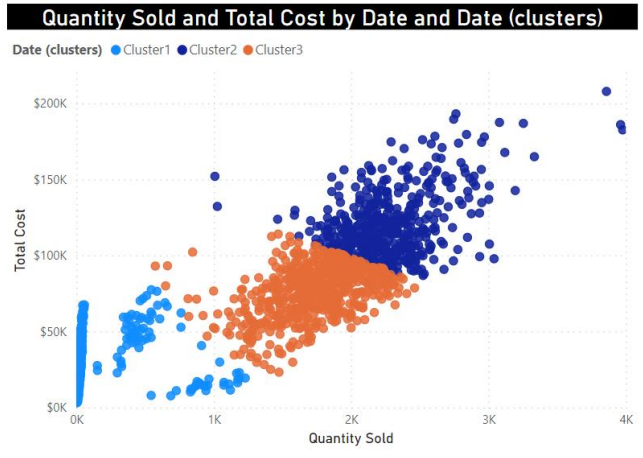
AI Visuals

Clustering allows you to group observations in a dataset with similar characteristics

- Clustering is a form of unsupervised machine learning
- Power BI's default clustering algorithm is the *Expectation Maximization (EM)** method

Typical Use Cases:

- ✓ Identify relationships that may not be derived through casual observation
- ✓ Detection of anomalies in the dataset
- ✓ Identify performance by segments *(i.e. customer segmentation)*
- ✓ Clustering dimensions based on factors *(i.e. store locations by sales, orders, ratings, etc.)*



PRO TIP: Remember, there's no "right" answer or single optimization metric when it comes to clustering; the best outputs help you **answer the question at hand** and **make practical, data-driven business decisions**

* https://docs.microsoft.com/en-us/analysis-services/data-mining/microsoft-clustering-algorithm-technical-reference

PRO TIP: GROUPING & BINNING

Basic Chart Types

Analytics Options

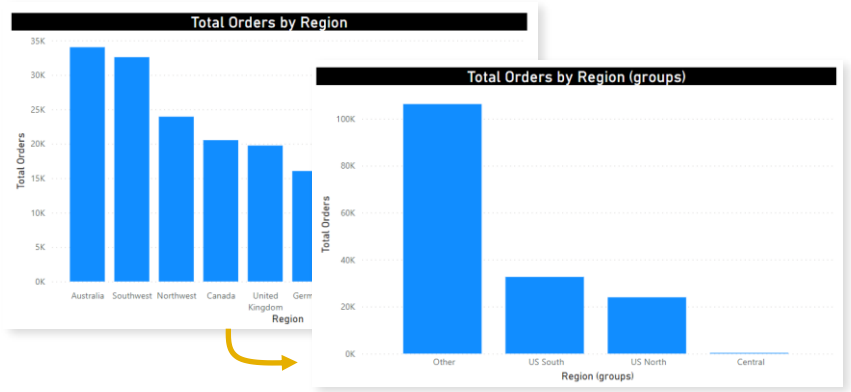
Q&A

Filtering Options

AI Visuals

Grouping is the process of creating logical categories with *text* data, while **binning** is the process of creating logical ranges for *numerical* data

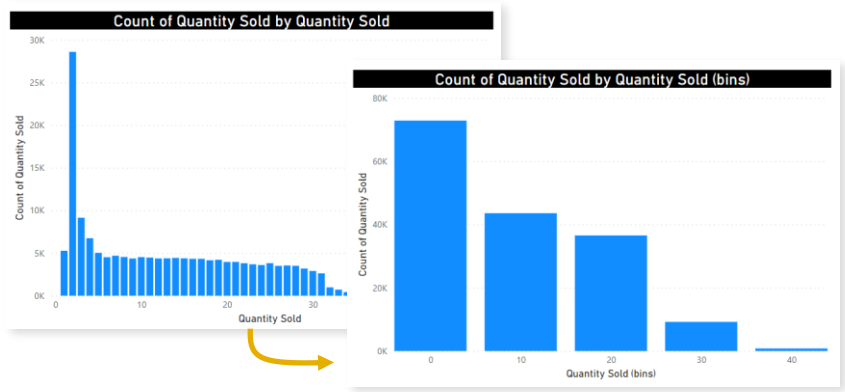
GROUPING



Common Uses:

- Group 2 or more categories together
- Refine how groups are presented
- Add/remove categories from existing groups

BINNING



Common Uses:

- Creating logical ranges of numerical data
- Use bins to “right-size” data
- Create equal size ranges
- **NOTE:** Bins can be created for calculated columns but not for measures

Q&A

- Basic Chart Types
- Analytics Options
- Q&A
- Filtering Options
- AI Visuals

Q&A in lets you explore your data “in your own words” using **natural language queries**

Q&A Tooling Interface

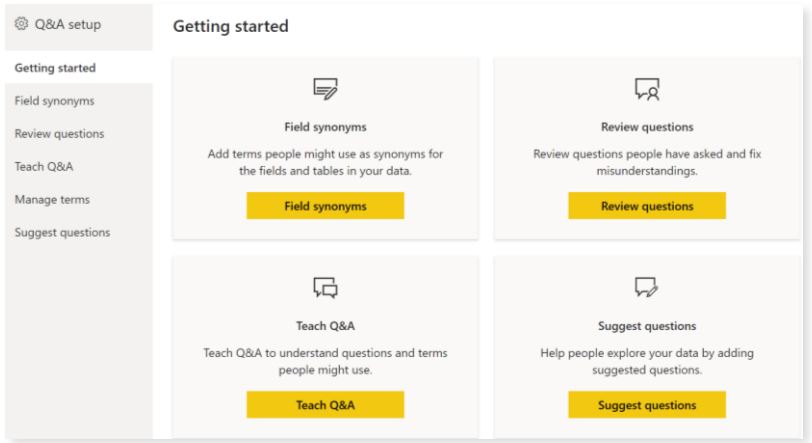
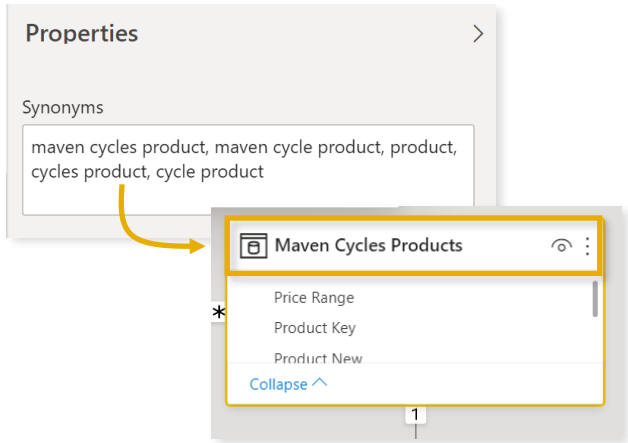


Table & Column Synonyms



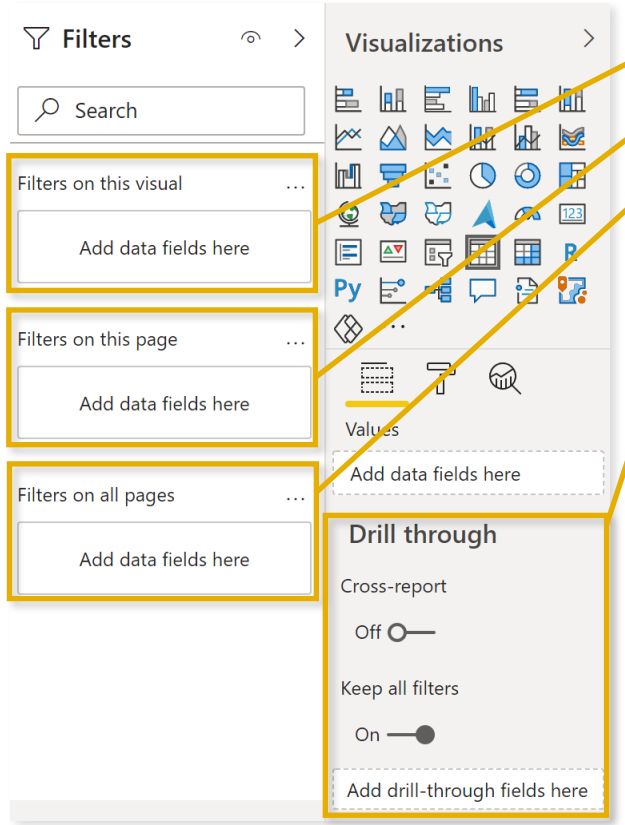
Best Practices:

- ✓ Use new Q&A tooling to interact with queries and make improvements
- ✓ Add missing relationships between tables
- ✓ Rename tables and columns
- ✓ Fix incorrect data types
- ✓ Normalize your model (single table or column per question)
- ✓ Add synonyms to tables and columns

FILTERING OPTIONS

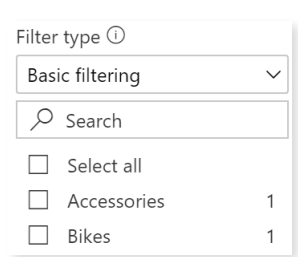
- Basic Chart Types
- Analytics Options
- Q&A
- **Filtering Options**
- AI Visuals

There are four primary **filter options** in Power BI reports:

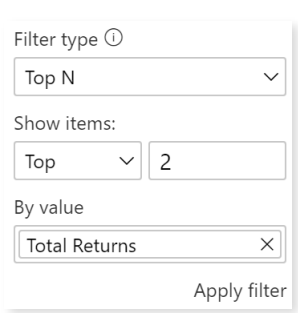


1. **Visual Level:** Applies only to the ***specific visual*** in which it is defined
2. **Page Level:** Applies to ***all visuals on the specific page*** in which it is defined
3. **Report Level:** Applies to ***all visuals*** across ***all pages*** of the report
4. **Drill through:** Applies to ***specific pages***, and ***updates*** based on the item clicked

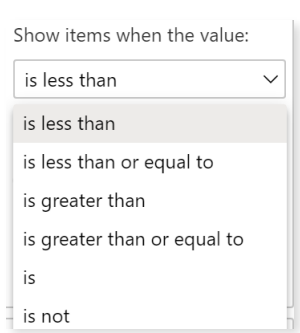
Filter settings include Basic, Advanced, and Top N options



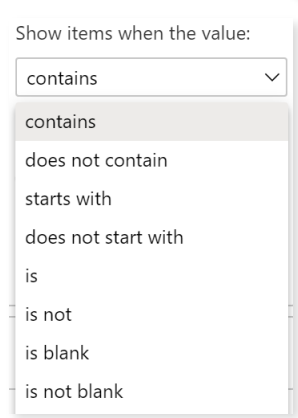
Basic Options



Top N Options



***Advanced** (Values)*



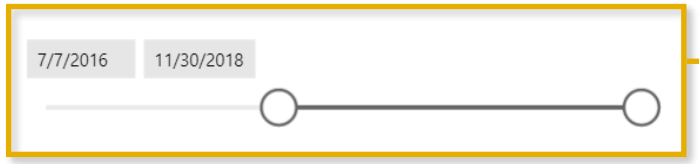
***Advanced** (Text)*

In the newest versions of Power BI, these tools can be found in both the **Home and **Insert** menus*

SLICERS

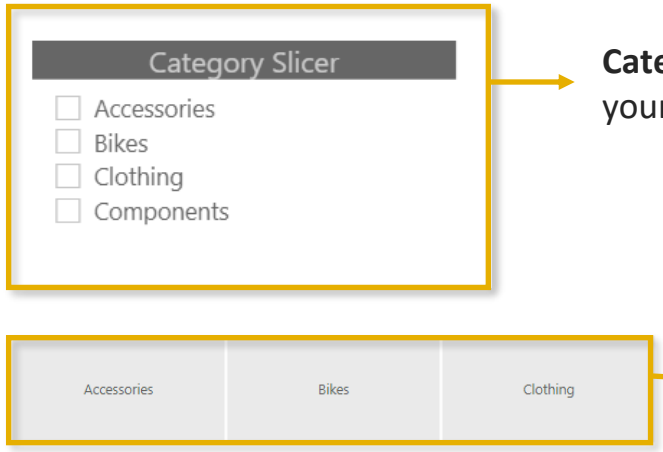
Slicers provide an interactive way for users to sort and filter a report

- Basic Chart Types
- Analytics Options
- Q&A
- **Filtering Options**
- AI Visuals



Date slicer formatted as a slider

Can also be a list, range, or “in the past X time period” option



Categorical slicers can also be created from dimensions or attributes within your model *(Category, Store Location, Gender, etc.)*

AI VISUALS (KEY INFLUENCERS)

Basic Chart Types

Analytics Options

Q&A

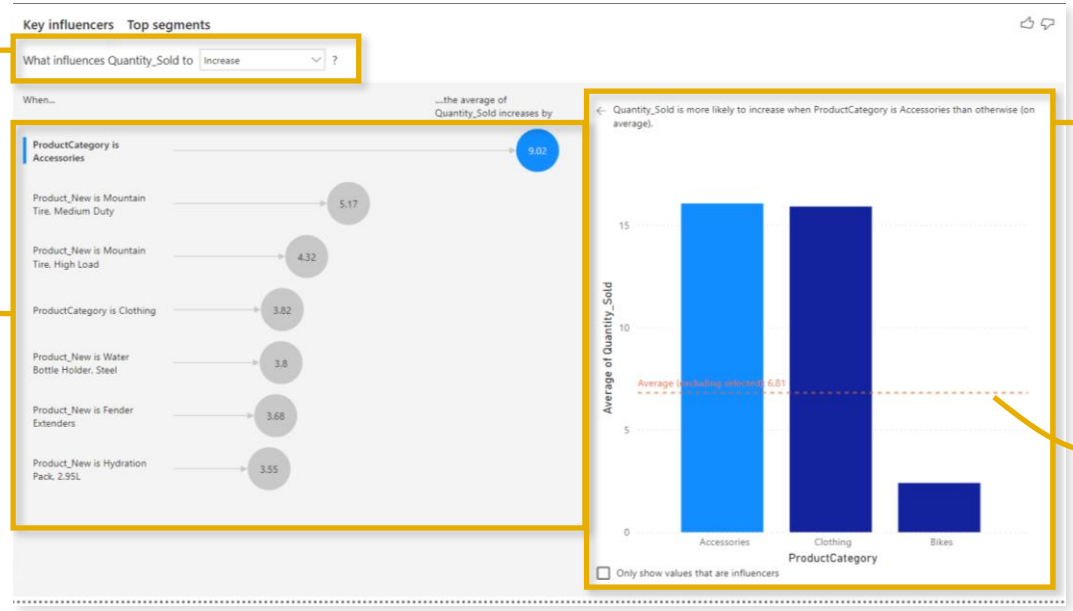
Filtering Options

AI Visuals

The **key influencers** visual helps you understand the factors that drive a specific metric

DROP DOWN BOX
The value or metric under investigation

LEFT PANE
Visual that shows a list of the top key influencers



RIGHT PANE
*Column chart display all values for the key influencer **theme** selected in the left pane*

AVERAGE LINE
Shows the percentage of the other themes that increase quantity sold

AI VISUALS (DECOMPOSITION TREE)

Basic Chart Types

Analytics Options

Q&A

Filtering Options

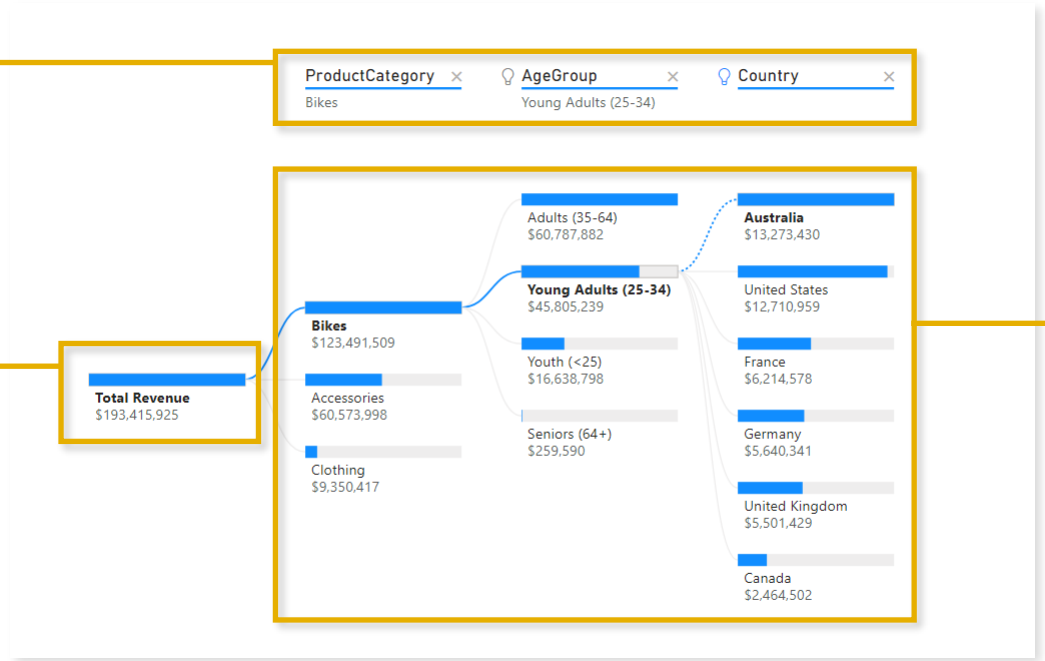
AI Visuals

The **decomposition tree** visual allows you to perform exploratory analysis by successively breaking down a measure across multiple dimensions

- This is a great choice when you want to perform a **root cause analysis** or **ad hoc exploration**

EXPLAIN BY
One or more dimensions used to explore measure

ANALYZE
Measure or aggregate value that is analyzed



EXPANDED TREE
Shows expanded tree breaking down the measure by dimensions in the explain by columns

DEPLOYING & MAINTAINING ASSETS

Prepare the Data

- Get data from different sources
- Clean, transform, and load data

Model the Data

- Design a data model
- Develop a data model
- Create model calculations with DAX
- Optimize model performance

Visualize & **Analyze** the Data

- Create reports
- Create dashboards
- Enrich reports for usability
- Enhance reports for usability & storytelling
- Identify patterns & trends

Deploy & **Maintain** Assets

- Manage files & datasets
- Manage workspaces

DEPLOY & MAINTAIN ASSETS

In this section we'll cover tools and techniques to **deploy & maintain assets**, including managing datasets, refreshing data, securing datasets, and creating and managing workspaces

TOPICS WE'LL COVER:

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Endorse Content
- Storage Formats
- Sensitivity Labels

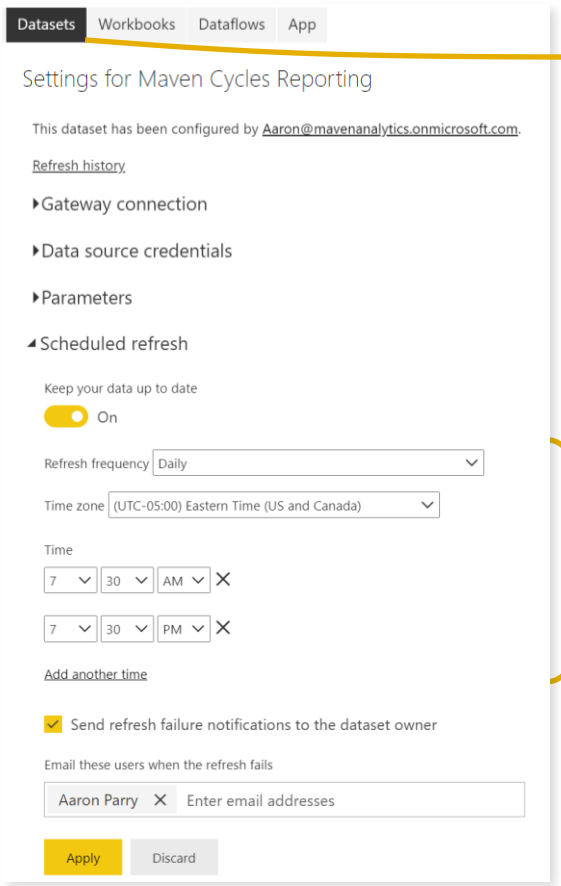
COMMON QUESTIONS:

- *Which of the following steps should you take to ensure a Dataflow is updated every morning?*
- *What DAX expression would you use to create an RLS role based on a manager's Region?*
- *Which of the following roles can add other members to a workspace?*

SCHEDULED REFRESH (DATASET)

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

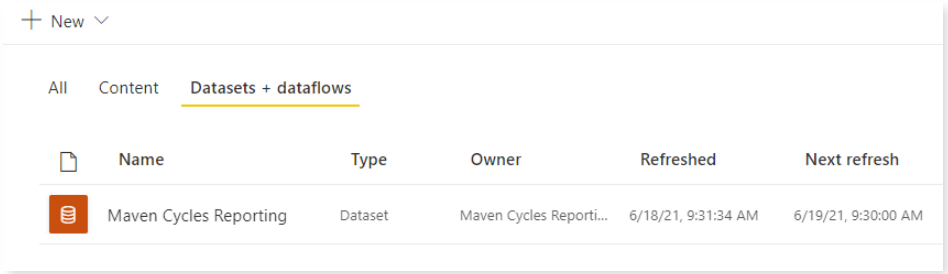
Scheduled refresh allows you to keep your Power BI reports up to date by automatically refreshing datasets based on a given frequency & time of day



*Select the dataset to refresh and click **Scheduled Refresh** or **Settings** from the ellipsis menu options*

*Set a **Frequency** (daily or weekly), **Time Zone**, and **Time** (half-hour increments AM/PM)*

Confirm dataset refresh is working properly (schedules align, errors, etc.)



HEY THIS IS IMPORTANT!

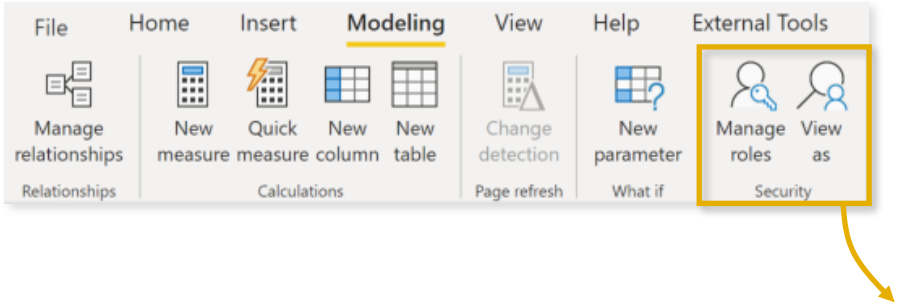
A data gateway is required to refresh on-premises & online data sources

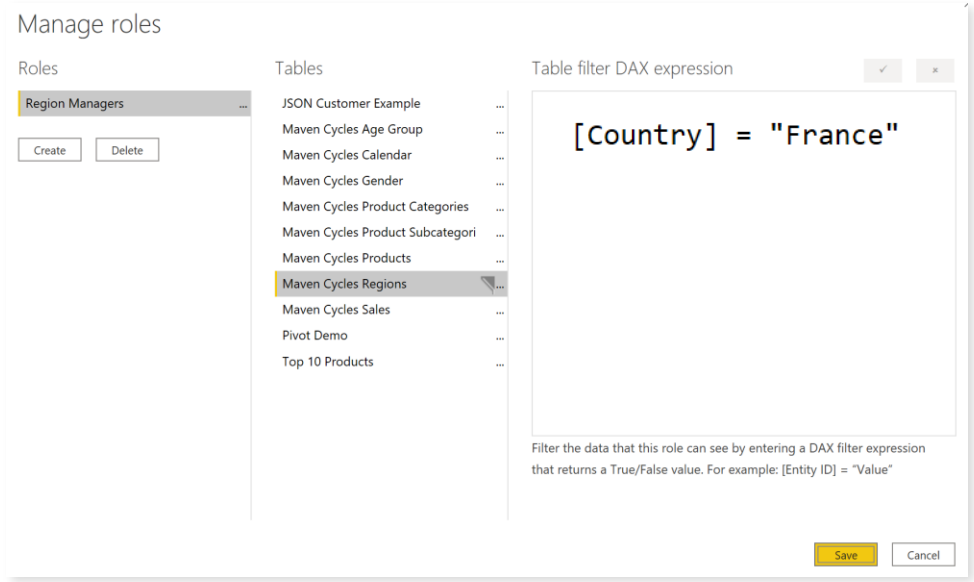
STATIC ROW-LEVEL SECURITY

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Static roles allow you to define filtered views for specific audiences (*territory managers, department leads, execs, etc.*) using simple DAX statements

- This is not the same as bookmarks or pre-filtered views; roles **filter data out of your model** and limit what audiences can access
- Static roles must first be configured in Power BI Desktop and then applied in Power BI Service





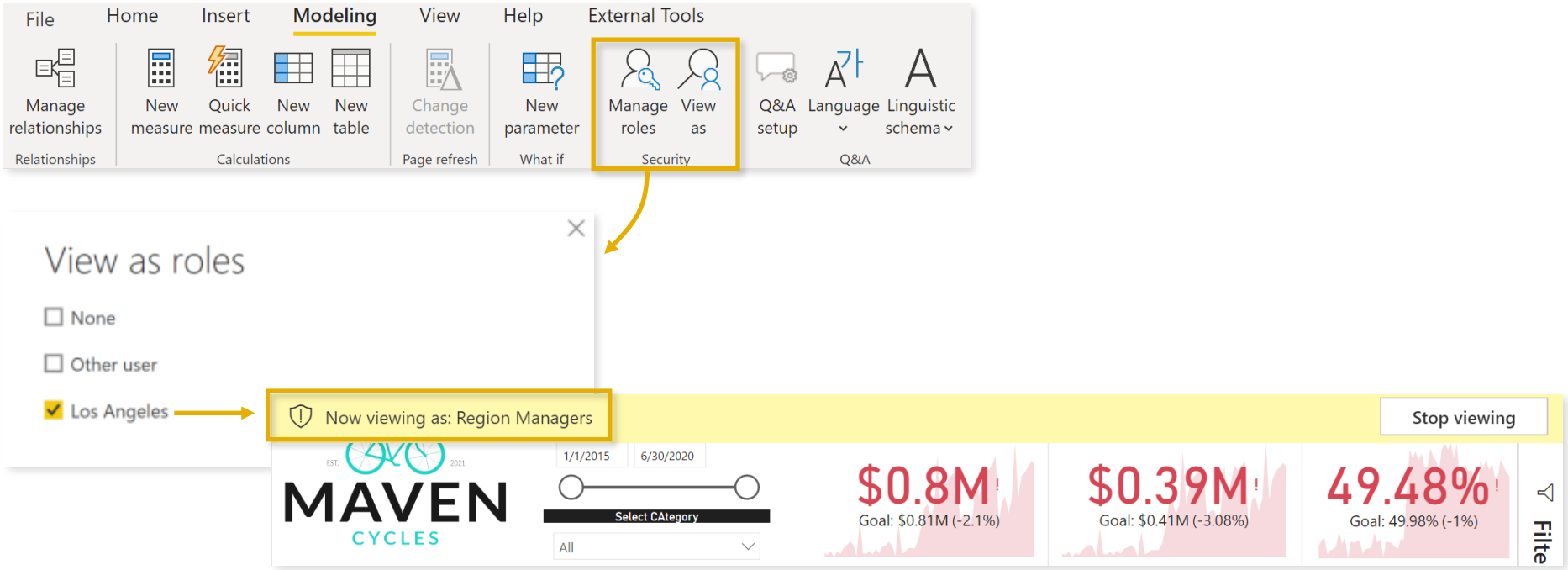
HEY THIS IS IMPORTANT!

If a user is part of two roles, the RLS roles are combined for the individual

TESTING ROW-LEVEL SECURITY

- Scheduled Refresh
- **Row Level Security**
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

As a best practice, use the **View as** option to confirm that the security role properly limits the data prior to publishing to Power BI Service



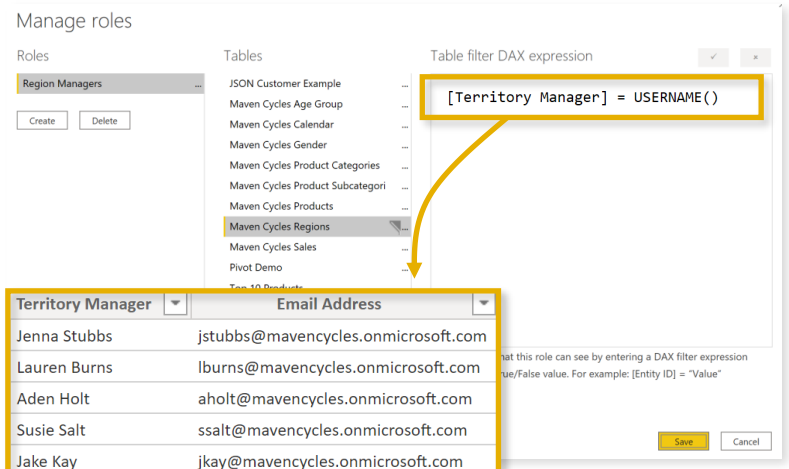
DYNAMIC ROW-LEVEL SECURITY

- Scheduled Refresh
- **Row Level Security**
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Dynamic roles allow you to define filtered views for a specific list of users with the DAX functions **USERNAME** or **USERPRINCIPALNAME**

- These DAX functions require adding an additional table into your data model
- Dynamic roles must first be configured in Power BI Desktop and then applied in Power BI Service

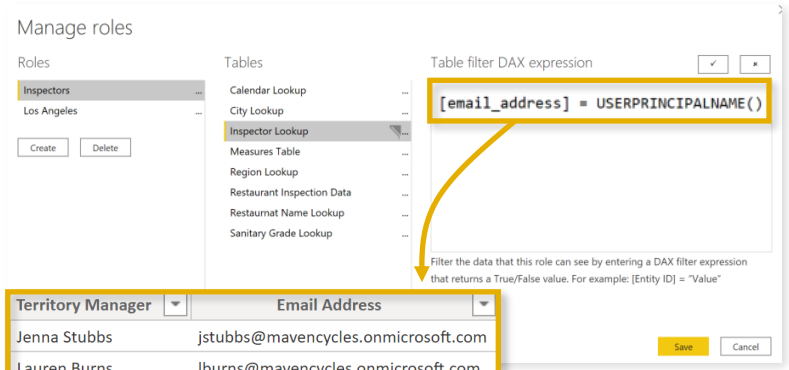
USERNAME



Territory Manager	Email Address
Jenna Stubbs	jstubbs@mavencycles.onmicrosoft.com
Lauren Burns	lburns@mavencycles.onmicrosoft.com
Aden Holt	aholt@mavencycles.onmicrosoft.com
Susie Salt	ssalt@mavencycles.onmicrosoft.com
Jake Kay	jkay@mavencycles.onmicrosoft.com
Kathy Meza	kmeza@mavencycles.onmicrosoft.com
Adam Juan	ajuan@mavencycles.onmicrosoft.com
David Hahn	dhahn@mavencycles.onmicrosoft.com
Benny Blanco	bblanco@mavencycles.onmicrosoft.com
Dirk Gently	dgently@mavencycles.onmicrosoft.com

Dynamic role will filter by the Username in this table

USERPRINCIPALNAME



Territory Manager	Email Address
Jenna Stubbs	jstubbs@mavencycles.onmicrosoft.com
Lauren Burns	lburns@mavencycles.onmicrosoft.com
Aden Holt	aholt@mavencycles.onmicrosoft.com
Susie Salt	ssalt@mavencycles.onmicrosoft.com
Jake Kay	jkay@mavencycles.onmicrosoft.com
Kathy Meza	kmeza@mavencycles.onmicrosoft.com
Adam Juan	ajuan@mavencycles.onmicrosoft.com
David Hahn	dhahn@mavencycles.onmicrosoft.com
Benny Blanco	bblanco@mavencycles.onmicrosoft.com
Dirk Gently	dgently@mavencycles.onmicrosoft.com

Dynamic role will filter by the Email Address in this table

USERNAME

USERNAME()

Returns the domain name and username of the current logged in user

=USERNAME( )

***USERNAME** doesn't have any parameters*

Examples:

- *[Name] = USERNAME()*

HEY THIS IS IMPORTANT!

USERNAME returns the domain and user's username in the format of *domain-name\user-name*

1. Person's **username** *(i.e., aaronp)*
2. Company **domain** *(i.e., mavencycles)*
3. **USERNAME** returns *(i.e., mavencycles\aaronp)*

USERPRINCIPALNAME

USERPRINCIPALNAME() — *Returns the user's name as their email address (i.e., aaron@mavendemo.com)*

=USERPRINCIPALNAME( )

USERPRINCIPALNAME *doesn't have any parameters*

Examples:

- *[Email] = USERPRINCIPALNAME()*

HEY THIS IS IMPORTANT!

User Principal Name (UPN) looks like an email address, but technically it's a combination of three items:

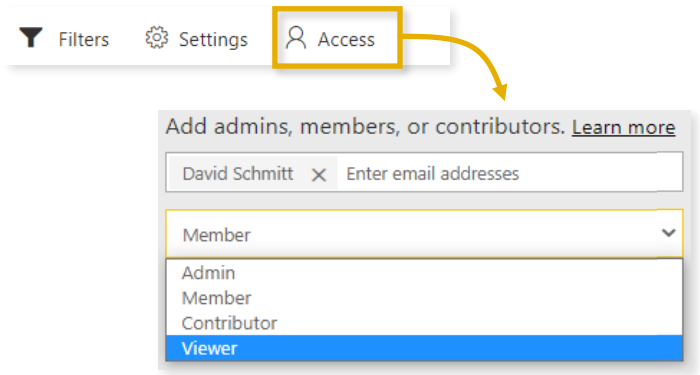
1. Person's **username** *(i.e., aaronp)*
2. "**@**" symbol
3. Company **domain** *(i.e., maven inspectional services)*

APPLYING RLS IN SERVICE

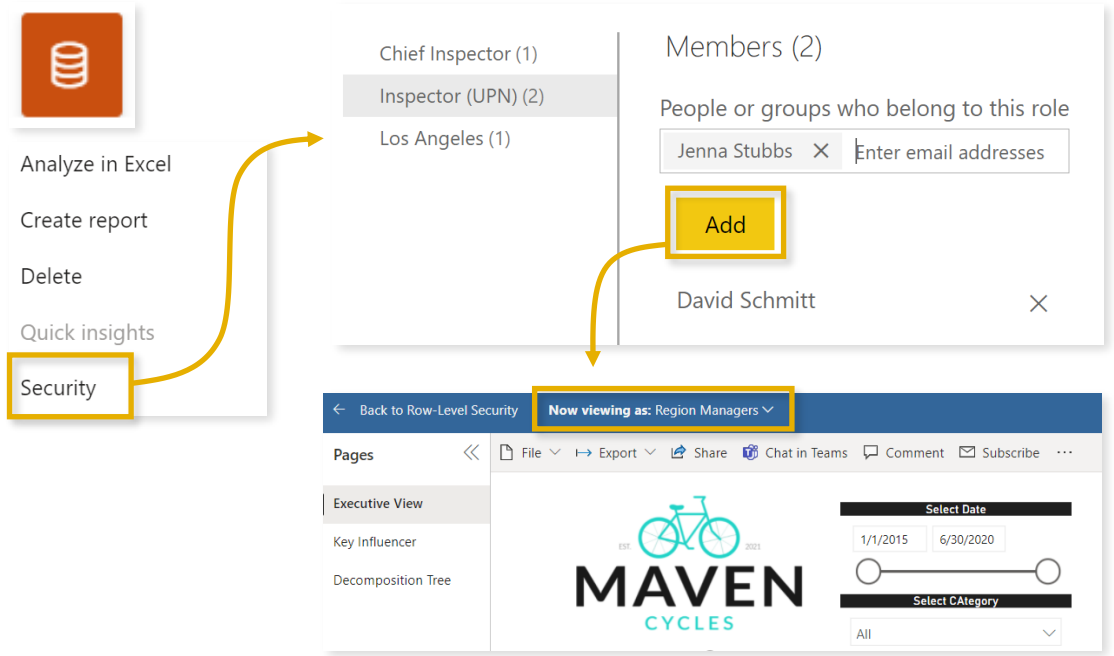
- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Once RLS rules have been defined in Power BI Desktop, they can be applied to users who you've shared reports with or have **Viewer** permissions in Power BI Service

1 Confirm that users impacted by RLS are assigned to the **Viewer** role



2 Add people or groups to the RLS rule



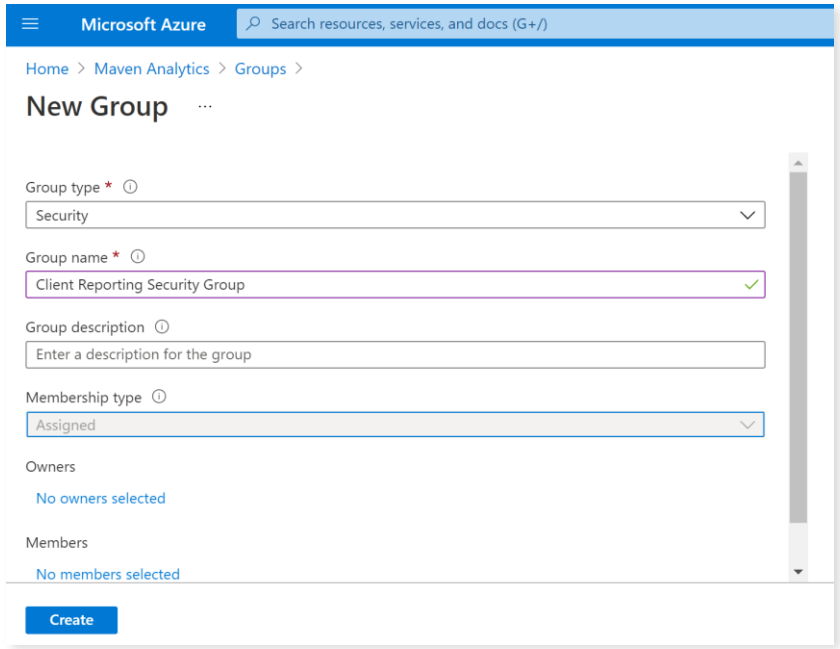
HEY THIS IS IMPORTANT!

If a user is assigned as an Admin, Member, or Contributor, **RLS will be overridden**

PRO TIP: RLS AZURE GROUPS

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Azure Active Directory **security groups** allow you to manage an entire group of users instead of a list of individual users



Security Group Benefits:

- ✓ Used to manage member and computer access to shared resources for a group of users
- ✓ Create specific security policies (permission levels) for different groups of users
- ✓ Allows you to set permissions for all members of a group at once
- ✓ Great for managing user access when people join and leave teams

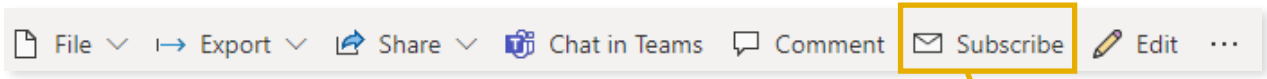
SUBSCRIPTIONS

- Scheduled Refresh
- Row Level Security
- **Sharing Options**
- Data Lineage
- Incremental Refresh
- Endorse Content
- Storage Formats
- Sensitivity Labels

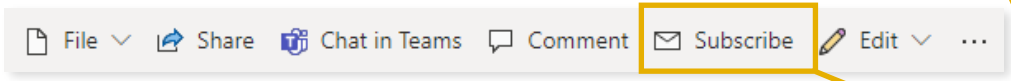
Subscribe to receive periodic email updates with a report, dashboard, or app snapshot

- Creating subscriptions requires a Pro or PPU license (self & others)
- Add email, subject and an optional message
- Set frequency & time (monthly, weekly, daily, hourly)
- Schedule the start and end dates

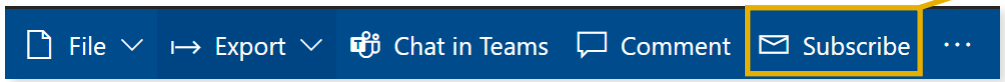
Report View

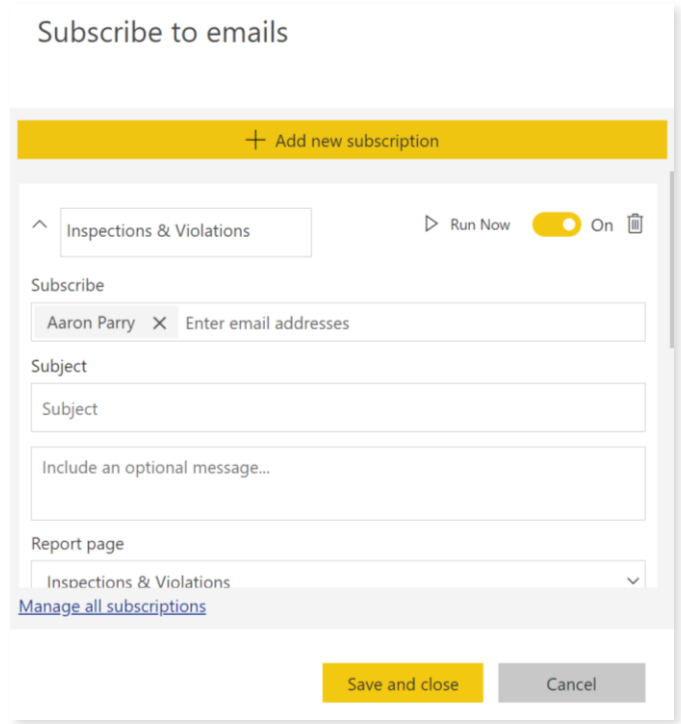


Dashboard View



App View

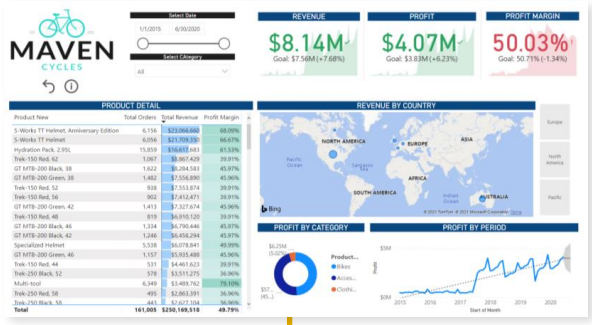




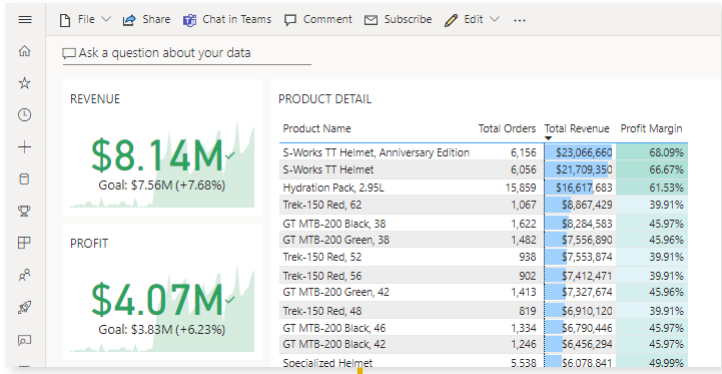
SHARING OPTIONS

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

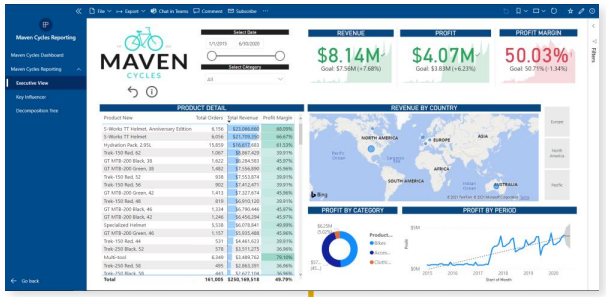
Reports



Dashboards



Apps



INDIVIDUALS

TEAMS

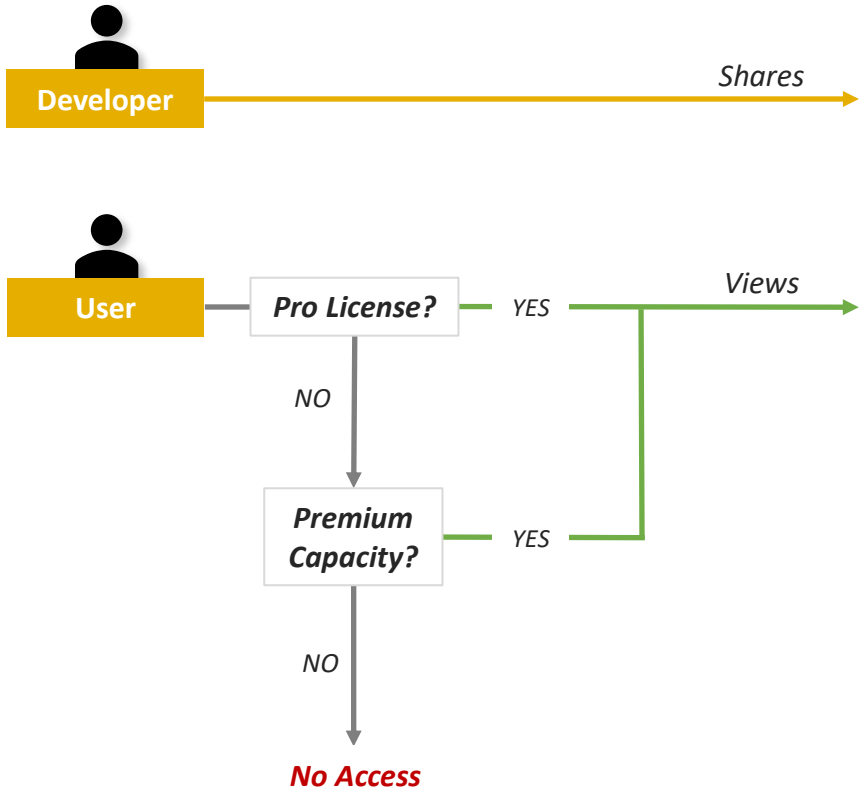
ORGANIZATIONS

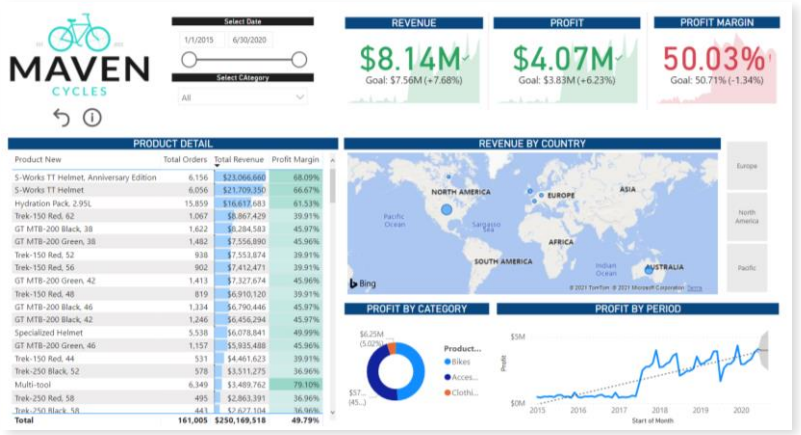
PRO TIP: When sharing reports & dashboards, the maximum number of recipients is **100** at a time (**500** total); if you need to share with more than 100 recipients, split into multiple sends or use groups

VIEWING RIGHTS

- Scheduled Refresh
- Row Level Security
- **Sharing Options**
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

When a developer shares a report or dashboard or publishes an app, users must have either a **pro license** or **access to premium capacity** in order to view





HEY THIS IS IMPORTANT!

Users with Pro licenses can share content with PPU but not the other way around

USER PERMISSIONS

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

When sharing workspaces, you can assign roles based on these **levels of permissions**:

VIEWER

- View workbooks, reports, and dashboards
- Can't access datasets, dataflows, or edit content

CONTRIBUTOR

Lifts Viewer restrictions, plus:

- Publish, create, edit, and delete content
- Can't add people to new roles or modify members
- Can share if the content has been previously shared but can't share new content

MEMBER

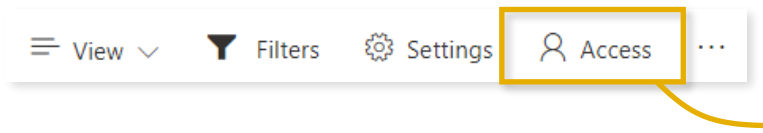
All Contributor rights, plus:

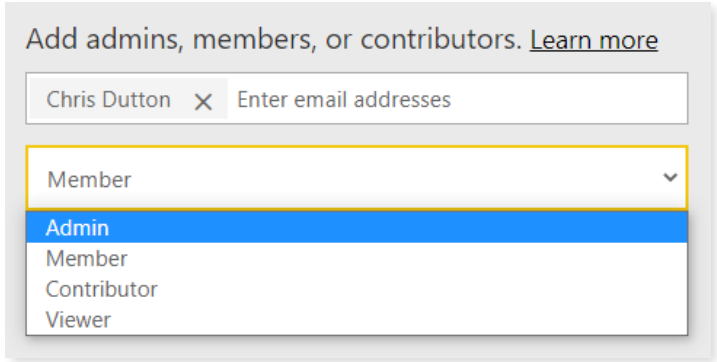
- Add members or users with lower permission levels
- Publish and update apps
- Share an item or app
- Allow others to reshare

ADMIN

All Member rights, plus:

- Update/delete workspaces
- Add or remove other users (including admins)



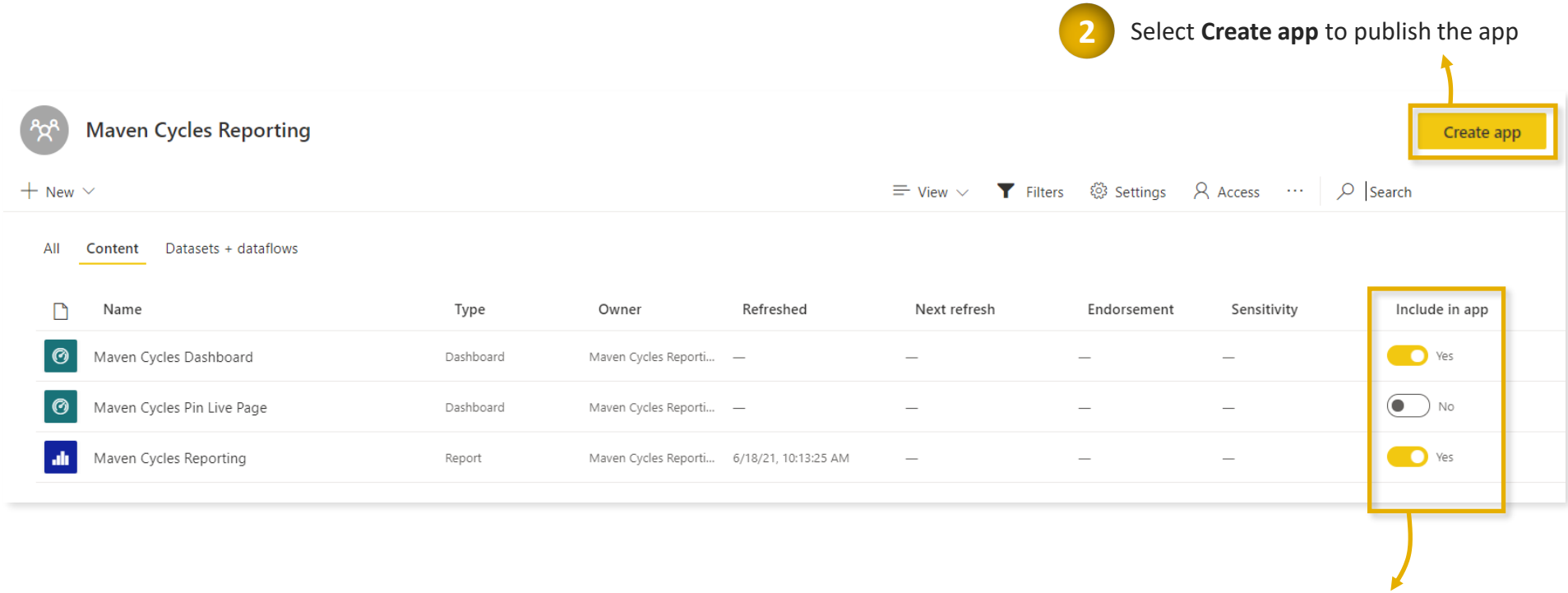


PRO TIP: When assigning privileges, use the **principle of least privilege** so users only have access to what they need

PUBLISHING APPS

- Scheduled Refresh
- Row Level Security
- **Sharing Options**
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

You can select reports and dashboards to **publish as an app** so large groups of people, both internal and external to your organization, can view them

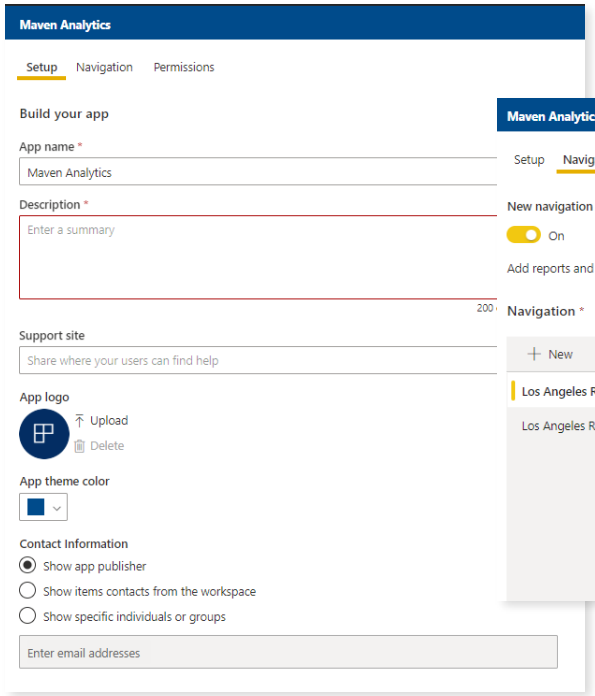


PUBLISHING APPS

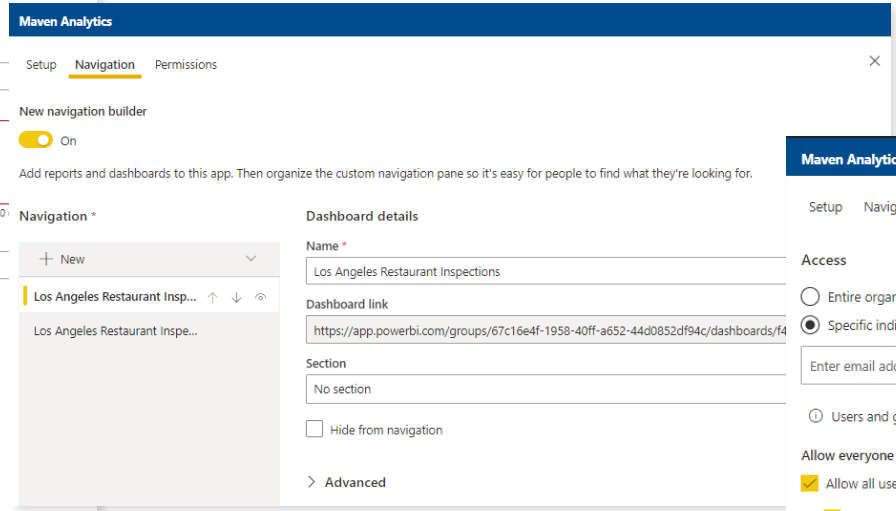
- Scheduled Refresh
- Row Level Security
- **Sharing Options**
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

After selecting Create app, configure the **setup**, **navigation**, and **permissions**

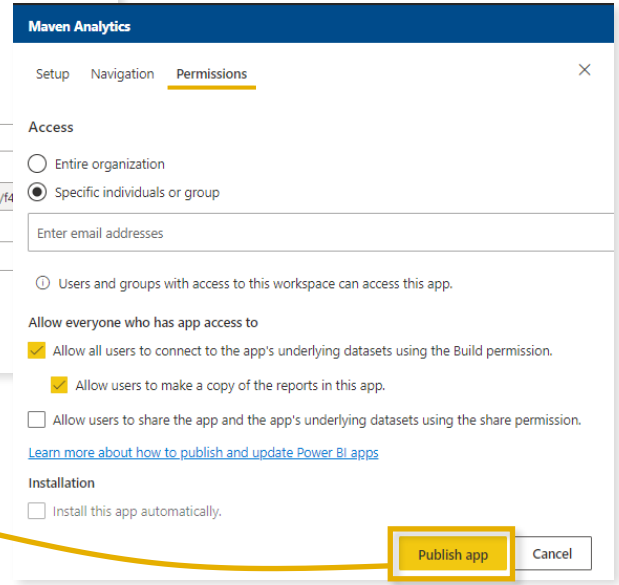
3 In the **Setup** tab, name the app, add a description, logo, and color theme



4 In the **Navigation** tab, use the navigation builder to customize the order, name, and navigation width



5 In the **Permissions** tab, add individuals, groups, or entire organizations to the App

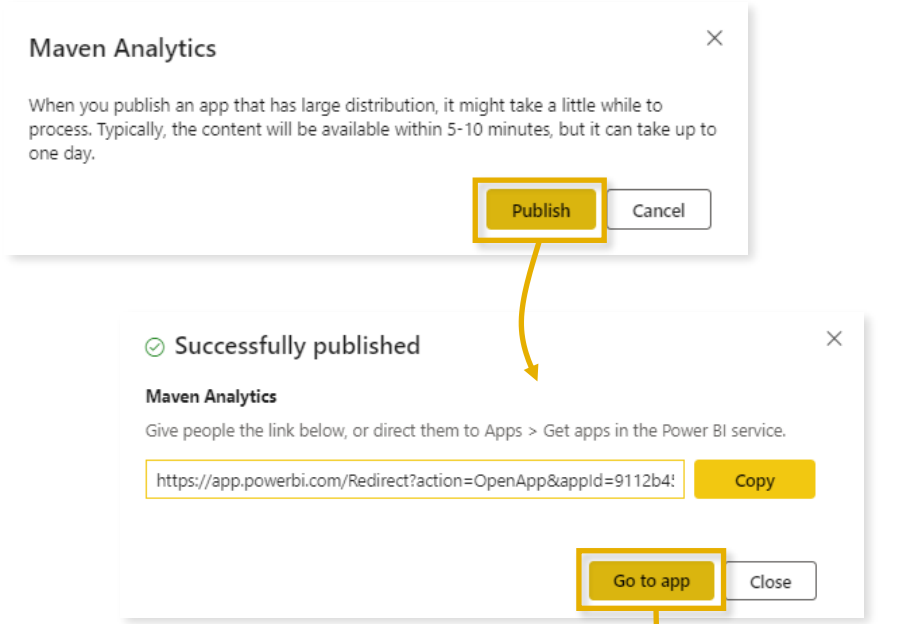


6 Publish the app!

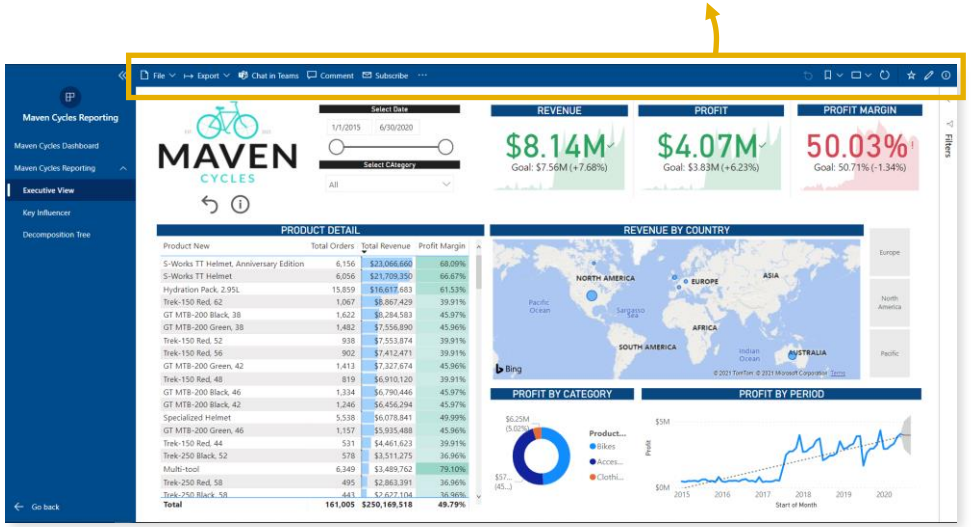
PUBLISHING APPS

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

After publishing the app, you'll be provided a **link for direct access**



Menu options allow users to do things like **print**, chat in **Teams**, add **comments**, **subscribe**, **favorite**, **edit** *(permissions depending)*, etc.



DEPLOYMENT PIPELINES

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Deployment pipelines allow you to manage the lifecycle of your organization's content (ALM) by developing and testing in Power BI before it's consumed by the end-user

Development

- First stage in Deployment Pipeline
- Used to design, build & upload new content with fellow creators

Test

- Share with testers & reviewers
- Test with larger volumes of data
- See how app looks for end-users

Production

- Share final version of content with users across organization

HEY THIS IS IMPORTANT!

This concept is possible by manually creating separate workspaces, but deployment pipelines automate the process!

DATA LINEAGE

- Scheduled Refresh
- Row Level Security
- Sharing Options
- **Data Lineage**
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Data lineage refers to the flow of data from a data source to a report and dashboard

Datasets

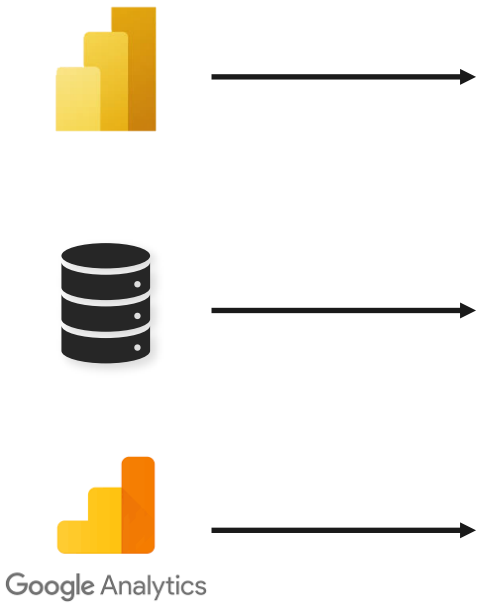
Data sources you've connected to

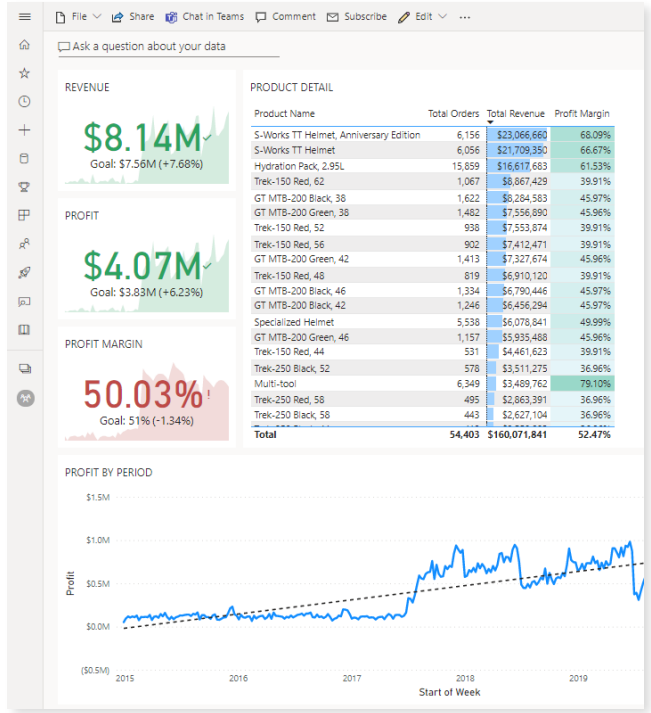
Reports

Visuals built from a single dataset

Dashboards

Reports & tiles built from multiple datasets



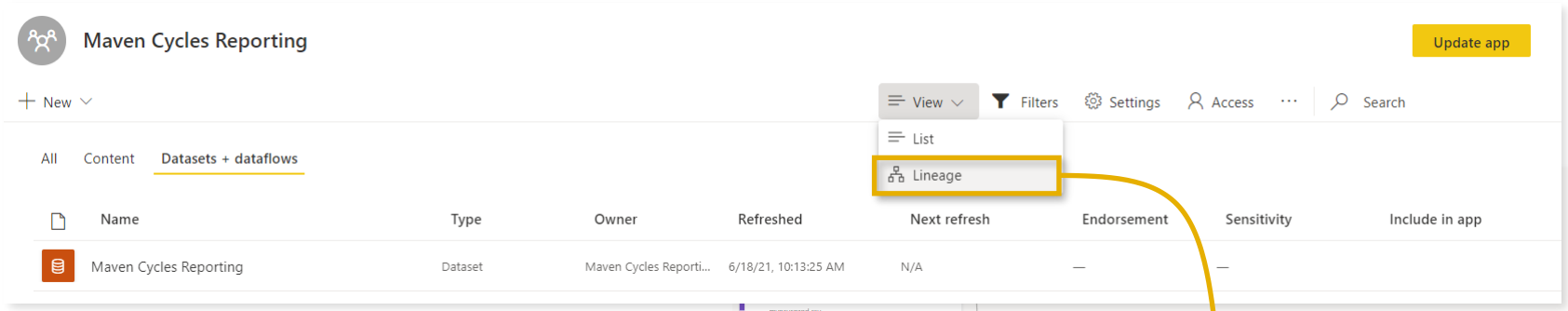


DATA LINEAGE VIEW

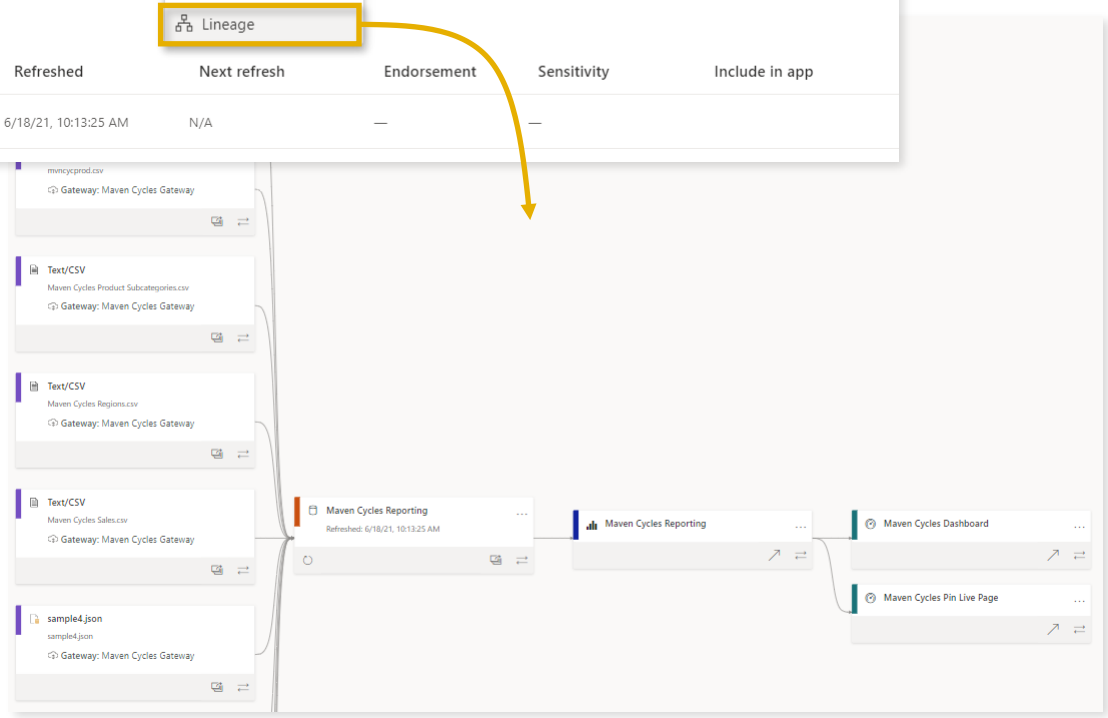
- Scheduled Refresh
- Row Level Security
- Sharing Options
- **Data Lineage**
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Lineage view shows how data is linked across multiple data sources & dependencies

List View



Lineage View



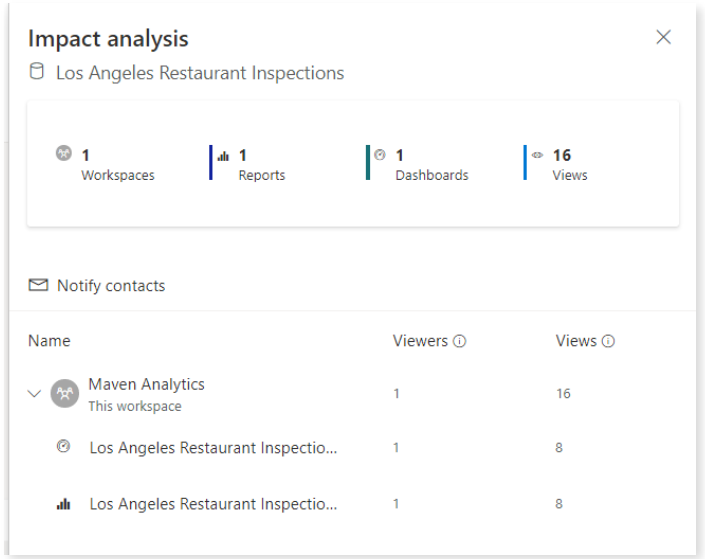
DATA LINEAGE TOOLS

- Scheduled Refresh
- Row Level Security
- Sharing Options
- **Data Lineage**
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Shows the **impact analysis** for the data source or dataset selected

Opens the **report** or dashboard selected

Highlights the data **lineage path** based on the selection



PRO TIP: Use the **lineage path** tool to help diagnose errors in reports and dashboards that get data from multiple data sources; this is especially useful when sharing dataflows and datasets across workspaces

INCREMENTAL REFRESH

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- **Incremental Refresh**
- Storage Formats
- Endorse Content
- Sensitivity Labels

Incremental Refresh is the process of reloading only the part of a dataset that may change over time and adding it to the rest of the data set that no longer changes

- **Faster Refresh Times** - Typically used with large datasets to decrease processing time
- **More Reliable** - Decreases the time connections are made to external sources
- **Reduced Resource Usage** - Easier on the internal resources of your computer (i.e., memory)

Example 1: Macro Data *(i.e. population)*

- 20+ years of data
- Daily-level data
- 1 Billion records

Example 2: Micro Data *(i.e. sensor data)*

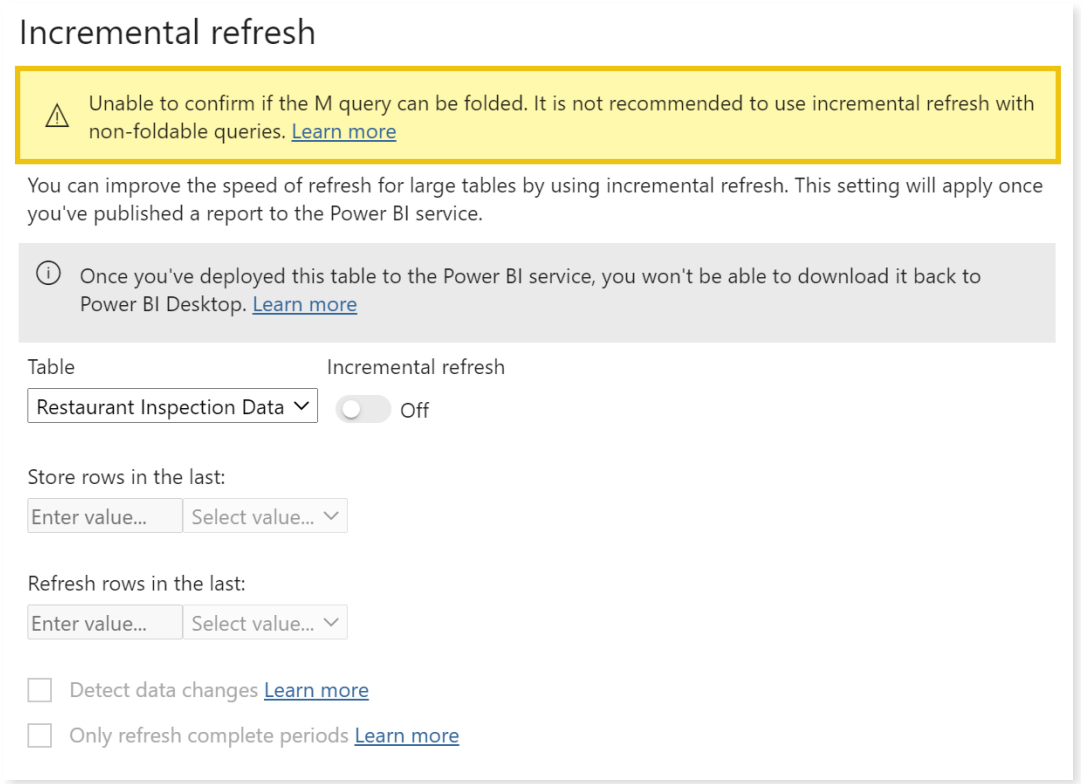
- 3 years of data
- Daily-level (hours, minutes, seconds)
- 100 Million records

QUERY FOLDING

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- **Incremental Refresh**
- Storage Formats
- Endorse Content
- Sensitivity Labels

Query folding is the ability for Power Query to generate a single query statement to retrieve and transform source data to improve the efficiency of the Power Query engines

- **IMPORTANT:** The whole purpose of incremental refresh is lost if a query cannot be folded



Sources that support Query Folding:

- Relational Databases
- Odata (SharePoint lists)
- Microsoft Exchange
- Azure Active Directory

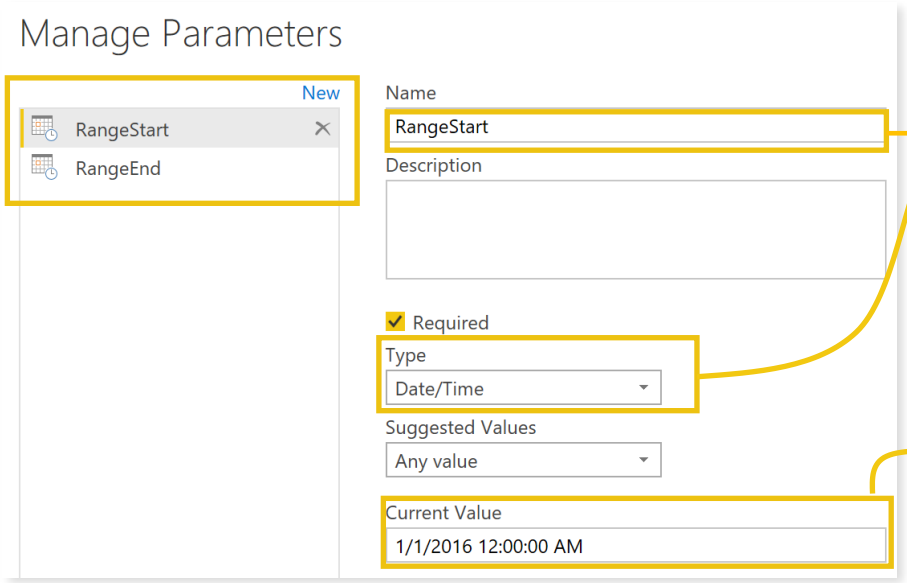
Sources that don't support Query Folding:

- Flat files (csv, xlsx, etc.)
- Azure Blob storage
- Web page data

CONFIGURING INCREMENTAL REFRESH

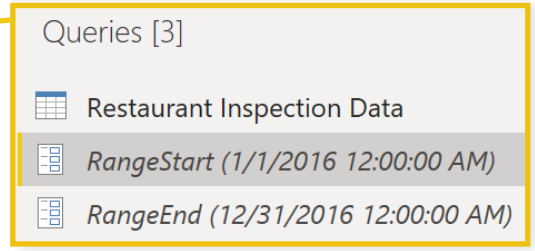
- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- **Incremental Refresh**
- Storage Formats
- Endorse Content
- Sensitivity Labels

1 Set **RangeStart** & **RangeEnd** parameters from the Query Editor in Power BI Desktop



IMPORTANT: *The Name & Type **must** reflect what's shown here; these parameters are case sensitive and are reserved by Power BI specifically for incremental refresh*

Current Value should be a date/time value within your date/time range; this will be overwritten when you later define Incremental refresh



***RangeStart** & **RangeEnd** parameters are added to the list of Query Editor queries*

HEY THIS IS IMPORTANT!

Current Value data type must be set to **Date/Time**

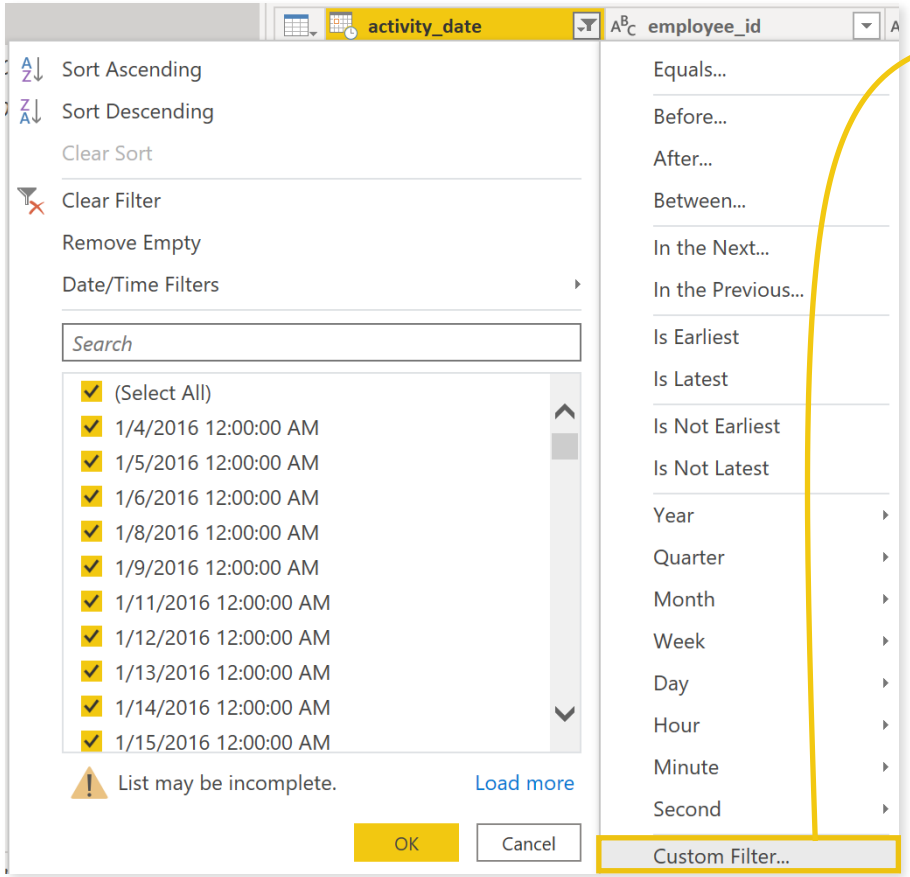
Additionally, your fact table **date column** data type must also be set to **Date/Time**

CONFIGURING INCREMENTAL REFRESH

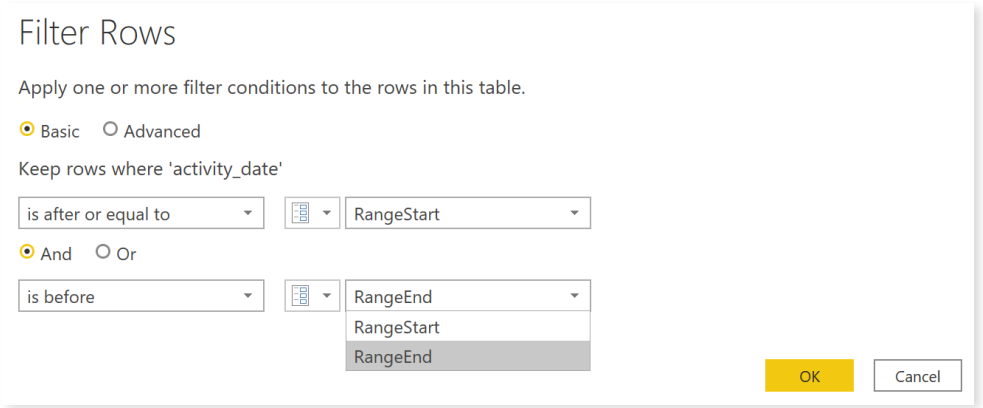
- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- **Incremental Refresh**
- Storage Formats
- Endorse Content
- Sensitivity Labels

2 Apply **RangeStart** & **RangeEnd** parameters to a date column using a *Custom Filter* from the filter options

Parameters are used to filter data imported into Power BI Desktop & dynamically partition the data into ranges



To avoid duplicated rows, only add an "=" sign on one side of the parameter; try using ">=" on StartRange and "<" on EndRange



HEY THIS IS IMPORTANT!

Since the date field is what determines the partial refresh of the underlying data source, incremental refresh only works with a **Date/Time** column

CONFIGURING INCREMENTAL REFRESH

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- **Incremental Refresh**
- Storage Formats
- Endorse Content
- Sensitivity Labels

3 Define the incremental refresh policy on the dataset *(right-click dataset)*

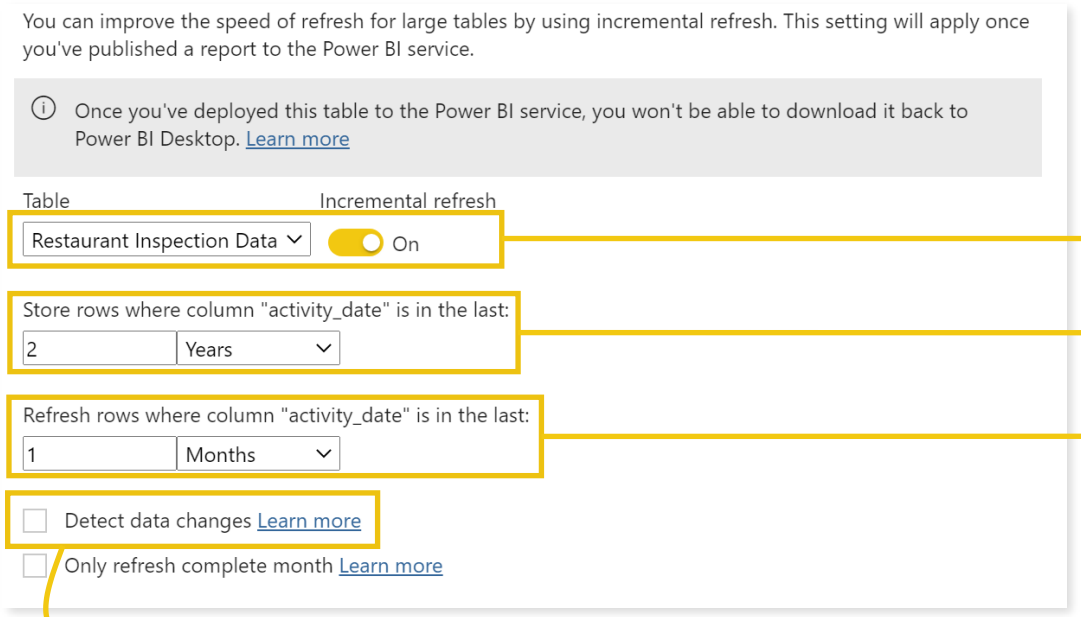


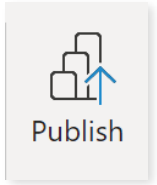
Table incremental refresh is applied to

*The number of rows you want to **store***
(think of this like "load only once and never load again")

*The number of rows you want to **refresh***
(think of this like "the rows I want to re-load each time")

*Detect data changes is an advanced setting that requires a separate "LastUpdateAt" column **(this isn't the same column used to partition the RangeStart & RangeEnd parameters)***

4 Publish to Service for the policy to take effect



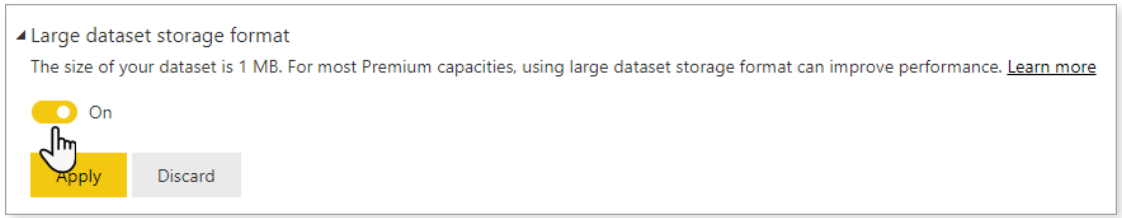
HEY THIS IS IMPORTANT!

Once you publish and configure incremental refresh in Power BI Service, you will not be able to download the dataset to Power BI Desktop

LARGE DATASET STORAGE FORMAT

Large dataset storage format is used for datasets over the 10GB refresh limit in Service

- This is available for Premium & Embedded capacities and Premium Per User



Steps to enable:

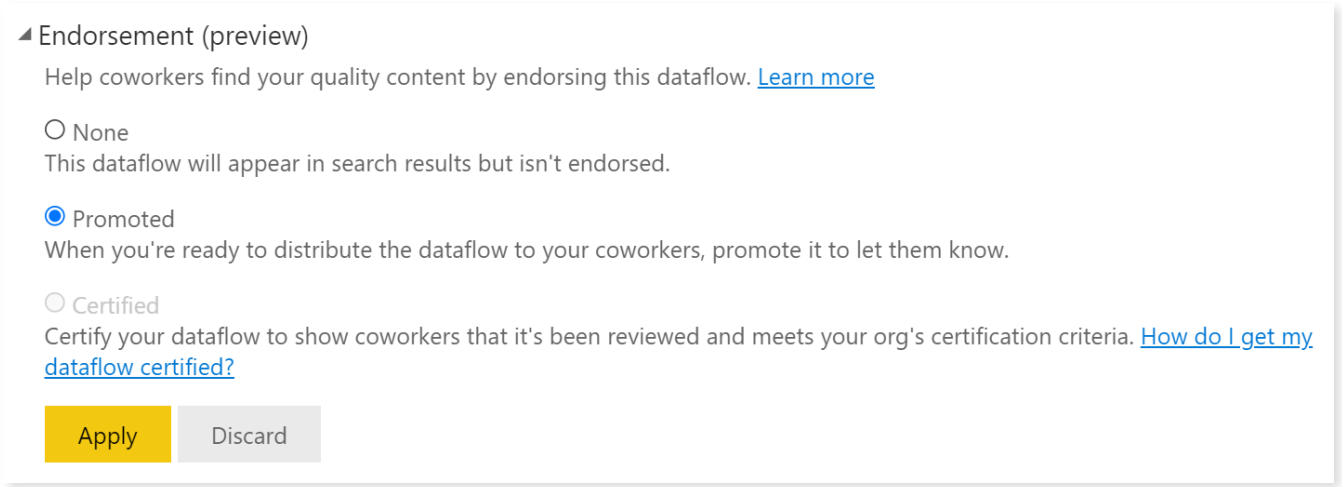
1. After creating a model in Power BI Desktop, configure **incremental refresh** if you expect your dataset will become larger and progressively consume more memory
2. Publish the model as a **dataset** to Power BI Service
3. In Power BI Service, go to Dataset > Settings > **Large dataset storage format**, click the slider to turn "On", and then "Apply"
4. Refresh the dataset to load **historical data** based on the incremental refresh policy *(the first refresh could take a while to load, but subsequent refreshes should be faster depending on your incremental refresh policy)*

ENDORSE CONTENT

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- **Endorse Content**
- Sensitivity Labels

Endorsement is a way to flag content that's ready for others to use

- Any content owner or member with write permissions can endorse content
- It's possible to endorse **datasets**, **dataflows**, **reports**, and **apps**

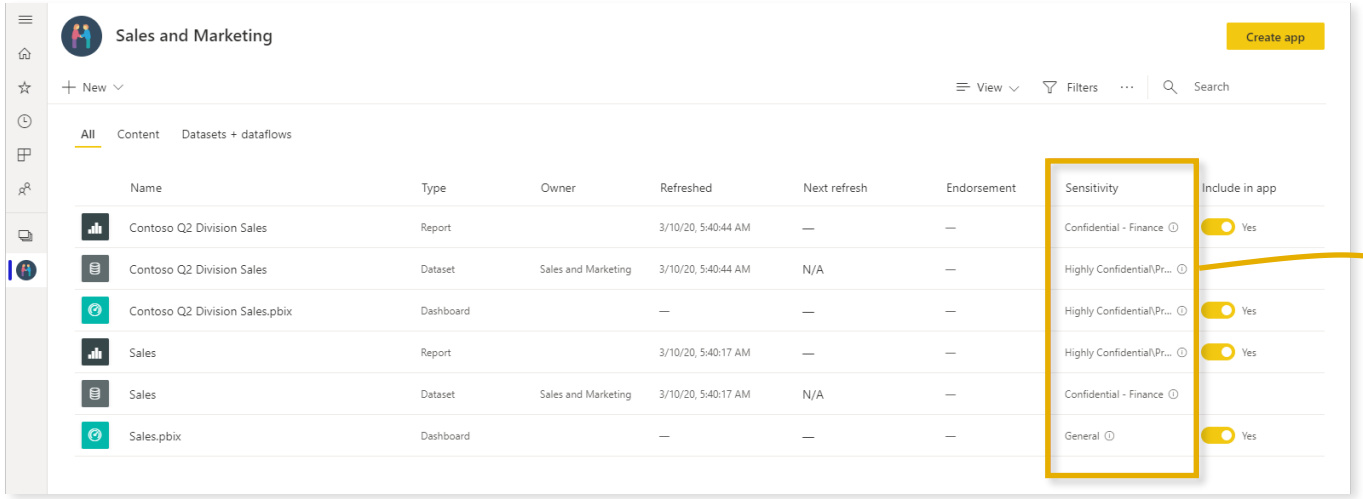


Certification means that the content meets the organizations quality standards and can be regarded as reliable, authoritative, and ready to use across the organization

SENSITIVITY LABELS

- Scheduled Refresh
- Row Level Security
- Sharing Options
- Data Lineage
- Incremental Refresh
- Storage Formats
- Endorse Content
- Sensitivity Labels

Sensitivity labels in Power BI provide a simple way to classify and safeguard sensitive content by labeling reports, dashboards, datasets, dataflows, and .pbix files



SENSITIVITY LABEL
Hover over label to see a description

To apply sensitivity labels:

1. You must have a Pro or Premium per User license and edit permission on the content
2. **Belong to a security group that has permission to apply sensitivity labels**
3. Sensitivity labels must be enabled for your organization
4. Subscribe to Azure Information Protection

HEY THIS IS IMPORTANT!

Sensitivity labels do not affect access to content in Power BI Service; access is solely managed by Power BI permissions

For more detail: ***https://docs.microsoft.com/en-us/power-bi/admin/service-security-sensitivity-label-overview***


LEARN ON!