



DA-100 EXAM PREP

ANALYZING DATA WITH POWER BI

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



COURSE STRUCTURE



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

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 actual DA-100 exam experience, with detailed solution walkthroughs

COURSE OUTLINE

1

Preparing for the DA-100 Exam

Review the skills measured in the DA-100 exam 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 trial

3

Measured Skills Review

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

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 publish to Power BI Service
- Enhance reports to expose insights & perform advanced analysis
- Create and manage 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.

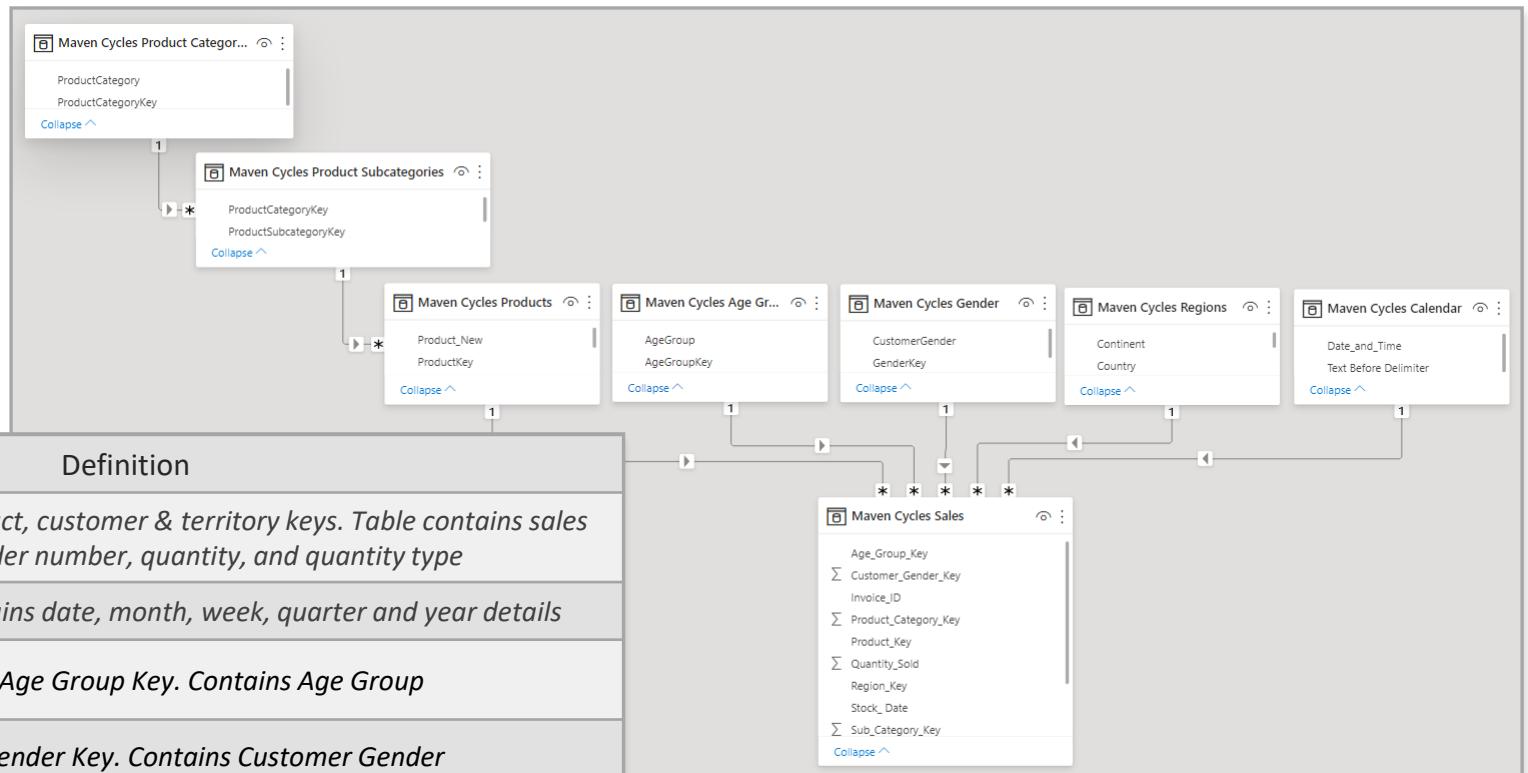
THE OBJECTIVES

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



THE MAVEN CYCLES DATA MODEL

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



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 DA-100 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 DA-100 exam*



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 DA-100 EXAM

EXAM STRUCTURE

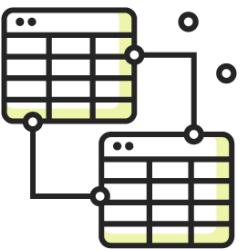
	Platform	Pearson VUE (remote connection)
	Question Format	Knowledge-Based + Case Study
 1 2 3	Number of Questions	40 - 60
	Time Limit	150 minutes (2.5 hours)
	Passing Score	70% (700/1000)
	Expiration	2 Years
	Cost	~\$165 USD* <small>*Price is based on the country in which the exam is proctored</small>

SKILLS MEASURED



Prepare the Data

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



Model the Data

- Design a Data Model
- Develop a data model
- Create measures using DAX
- Optimize Performance



Visualize the Data

- Create reports
- Create dashboards
- Enrich reports for usability



Analyze the Data

- Enhance reports to expose insights
- Perform advanced analysis



Deploy & Maintain Deliverables

- Manage datasets
- Create and manage workspaces

EXAM QUESTION TYPES

Knowledge-Based Questions

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

Question types include:

- Multiple choice & multi-response
- Fill in the blank (select from drop down options)
- Drag and drop (from a list of options)



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, you need to **identify the percentage of empty values** in each column as quickly as possible. Which *Data Preview* option should you select?

- a. Show whitespace
- b. Column profile
- c. Column distribution
- d. Column quality

Drag & Drop

You are building a dataset from a JSON file containing an array of documents. You need to **import attributes as columns from all the documents** in the JSON file. The solution must ensure that date attributes can be used as date hierarchies. What *three actions* should you perform in sequence?

Actions

Expand the columns

Expand the records

Add columns that us data type conversions

Set the data types

Convert the list to a table

Answer Area

Convert the list to a table

Expand the columns

Set the data types



EXAMPLE: KNOWLEDGE-BASED QUESTION

Drop-down Selection

You need to create a calculated table to return the 100 highest spending customers.
How should you complete the DAX expression?

Top 100 Customers = ▼

ASC(
TOPN(
DESC(
FILTER(
SUMMARIZE(

100, ▼

ASC(
TOPN(
DESC(
FILTER(
SUMMARIZE(

[Sales], ▼

ASC(
TOPN(
DESC(
FILTER(
SUMMARIZE(

(‘Transactions’,
‘Transactions’[Customer ID],
“Sales”
SUM(‘Transactions’[Sales])
,

EXAMPLE: CASE STUDY

Overview

Maven Analytics is an online learning platform using Microsoft Power BI for internal reporting. The company plans to leverage data from Microsoft SQL Server databases, Microsoft Excel files, text files, and several other data sources. Maven Analytics uses Azure Active Directory (Azure AD) to authenticate users.

Sales Data

Maven Analytics uses online sales data with the SQL schema shown in the table here →

In the **Date** table, the *date_id* column has a format of yyyyymmdd, and the *month* column has a format of yyymm.

The *week* column in the **Date** table and the *week_id* column in the **Weekly_Returns** table both have a format of yyyyww.

The *sales_id* column in the **Sales** table represents a unique transaction.

The *region_id* column can be managed by only one sales manager.

Table name	Column name	Data type
Sales_Region	region_id	Integer
	name	Varchar
Region_Manager	region_id	Integer
	manager_id	Integer
	sales_manager_id	Integer
Sales_Manager	name	Varchar
	username	Varchar
	sales_id	Integer
	sales_date_id	Integer
	sales_amount	Floating
Sales	customer_id	Integer
	sales_ship_date_id	Integer
	region_id	Varchar
Customer_Date	customer_id	Integer
	first_name	Varchar
	last_name	Varchar
Date	date_id	Integer
	date	Date
	month	Integer
	week	Integer
	year	Integer
Weekly_Returns	week_id	Integer
	total_returns	Floating
	sales_region_id	Varchar
Targets	target_id	Integer
	sales_target	Decimal
	date_id	Integer
	region_id	Integer

EXAMPLE: CASE STUDY

Data Concerns

You are concerned with data quality and completeness, and plan to verify the data for negative sales amounts.

Reporting Requirements

Maven Analytics has identified the following technical requirements:

- Executives require a visual showing sales by region
- Regional managers require a visual showing weekly sales and returns
- Sales managers must be able to see the data for their respective region only
- Sales managers require a visual to analyze sales performance against targets
- The Sales department requires reports showing the number of sales transactions

Question #1

In order to provide sales managers with the required access. What should you include in the solution?

- A. Create a security role that has a table filter on the Sales_Manager table where username = UserName()
- B. Create a security role that has a table filter on the Region_Manager table where sales_manager_id = UserPrincipalName()
- C. Create a security role that has a table filter on the Sales_Manager table where name = UserName()
- D. Create a security role that has a table filter on the Sales_Manager table where username = sales_manager_id

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)



Pass Tech Checks



Close all programs



Browser Open



Clear Desk



No phones, headphones, external monitors



No Talking
(except proctor questions)



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)



Enhanced security protocol
(digital photo & digital signature)



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



No Hats or hooded shirts



No Talking
(except proctor questions)



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

SCHEDULING THE EXAM

- 1 Head to the DA-100 site* and create an account

Schedule exam

Exam DA-100: Analyzing Data with Microsoft Power BI

Languages: English, Chinese (Simplified), Korean, Japanese
Retirement date: none

This exam measures your ability to accomplish the following technical tasks: prepare the data; model the data; visualize the data; analyze the data; and deploy and maintain deliverables.

For non-students interested in technology

[Schedule with Pearson VUE >](#)

For job seekers impacted by COVID-19

[Learn about our commitment to support people impacted by COVID-19.](#)

[Schedule for USD15 >](#)

Official practice test for Analyzing Data with Microsoft Power BI
All objectives of the exam are covered in depth so you'll be ready for any question on the exam.

*Make sure account name matches your ID

- 3 Launch the exam on testing day!

View or [launch](#) an online exam

Exams available for view or launch

- [DA-100: Analyzing Data with Microsoft Power BI](#)

- 2 Select the exam language & details

In-person or remote

How do you want to take your exam? [Exam delivery option descriptions](#)

- At a local test center
 Online from my home or office
 I have a Private Access Code

Payment

Description	Details	Price
Exam DA-100: Analyzing Data with Microsoft Power BI Language: English Exam Length: 150 minutes	Appointment Friday, April 30, 2021 Check-in time: 15:30 EDT Start time: 16:00 EDT	165.00

Exams for	Order Total
Name: Aaron Parry	Subtotal: 165.00
MS ID: MS0991352143	Tax: 0.00
	TOTAL DUE: USD 165.00
	VISA ****0315
	USD 165.00

Date & time

Select Date
[Why can't I find an available appointment?](#)

April 2021						
Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Select a date from the calendar. On	Morning	Afternoon
	00:00	12:15
	00:15	12:45
	00:30	13:15
	00:45	13:45

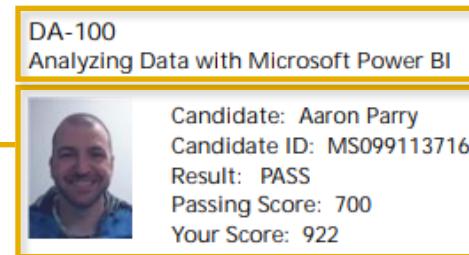


PRO TIP: To avoid surprises, take 5 minutes to **run a system test** before your remote exam day to make sure your computer and internet speed is sufficient

EXAM RESULTS

BASIC EXAM INFO

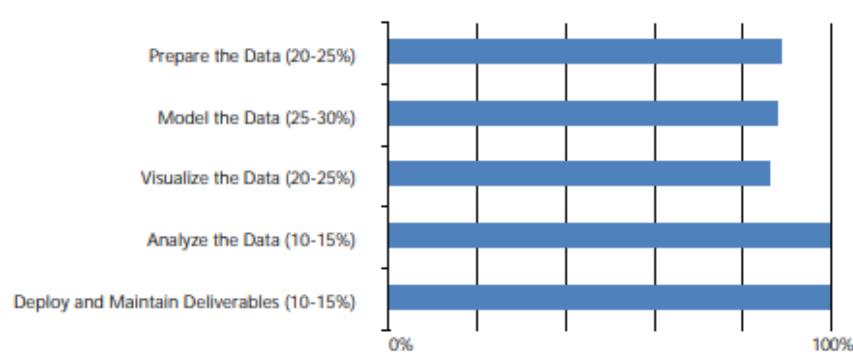
Exam details including name, id, result, and score



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

EXAM DETAILS

Exam type, location, and registration information



PERFORMANCE BY EXAM SECTION

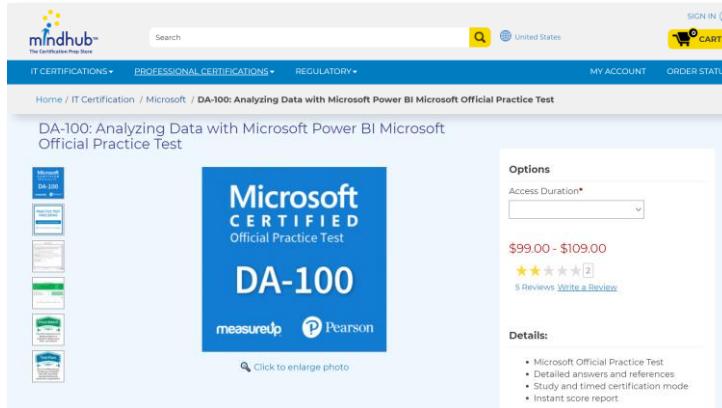
*Bar chart that represents your section-level performance
(Bar length cannot be used to calculate number of questions answered correctly)*

PERFORMANCE COMPARISON

How your performance compares to others who have taken the DA-100 exam

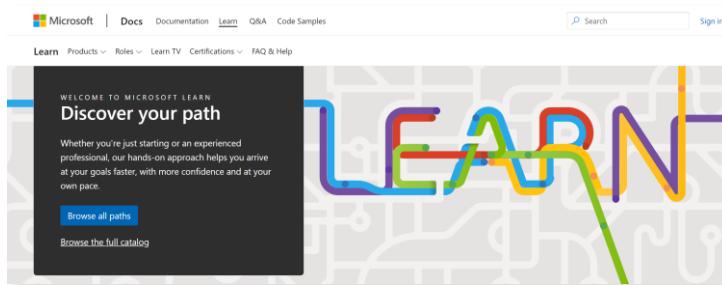
Content Area	Opportunity for Improvement	Comparable	Strength
Prepare the Data (20-25%)			✓
Model the Data (25-30%)			✓
Visualize the Data (20-25%)		✓	
Analyze the Data (10-15%)			✓
Deploy and Maintain Deliverables (10-15%)			✓

HELPFUL RESOURCES

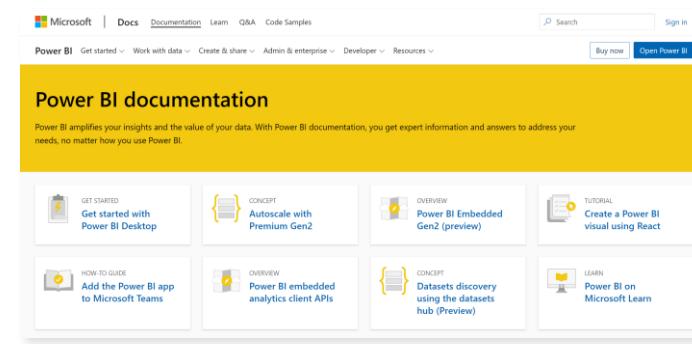


MeasureUp offers the official Microsoft Power BI practice test, which includes practice questions, detailed answers & references, study & certification testing modes, and an instant score report

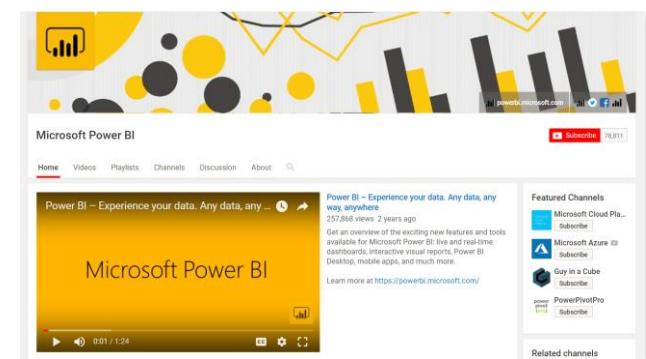
(mindhub.com/da-100-analyzing-data-with-microsoft-power-bi-microsoft-official-practice-test/p/MU-DA-100)



Microsoft Learn offers on-demand or instructor-led learning paths tailored to the DA-100
(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)*



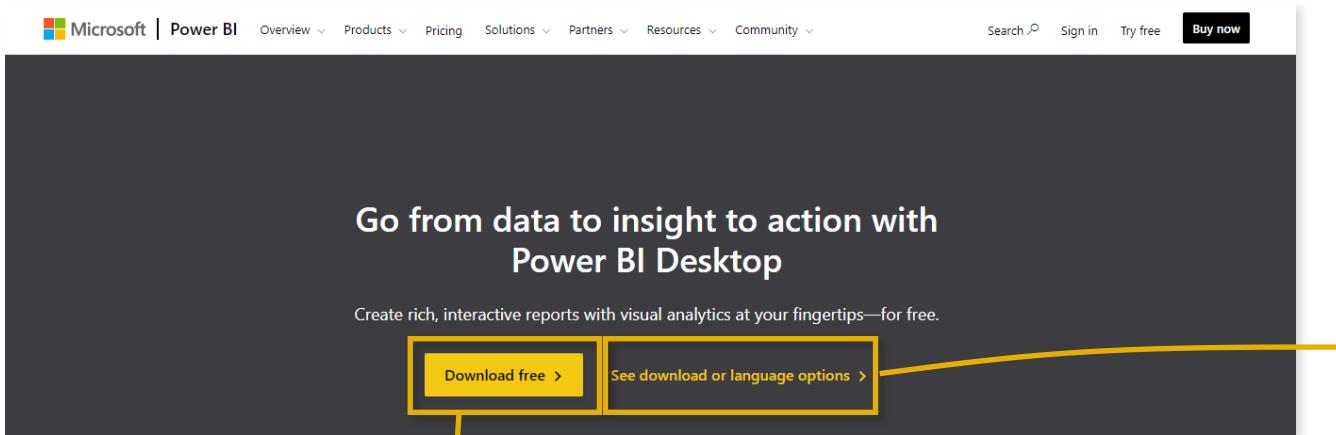
Skip questions if you get stuck, and revisit them later

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

SETTING UP DESKTOP & SERVICE

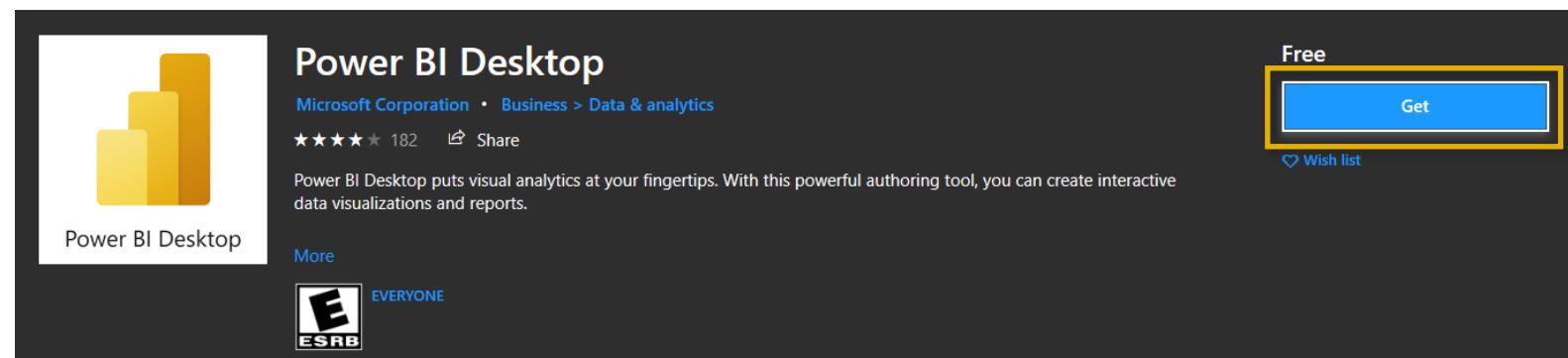
INSTALLING POWER BI DESKTOP

- 1 Head to powerbi.microsoft.com/desktop and click “Download free”



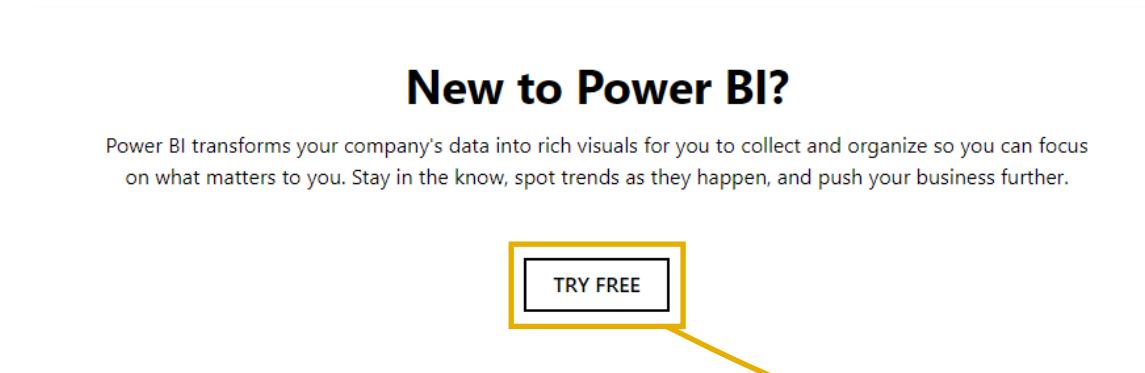
Use **See download or language options** to update Power BI Desktop

- 2 Click on “Get” to download Power BI Desktop from the Microsoft Store

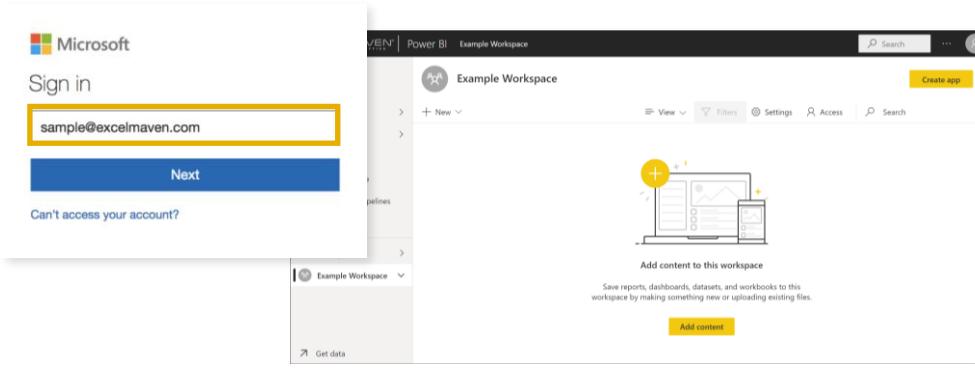


CREATING A POWER BI SERVICE ACCOUNT

- 1 Head to powerbi.microsoft.com/landing/signin and click “TRY FREE”



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



- 2 Create & verify your account

Thank you for choosing **Microsoft Power BI**

- 1 Let's set up your account
Enter your work or school email address, we'll check if you need to create a new account.

By proceeding you acknowledge that if you use your organization's email, your organization may have rights to access and manage your data and account.
[Learn More](#)
- 2 Next
- 3 Create your account
- 3 You're all set

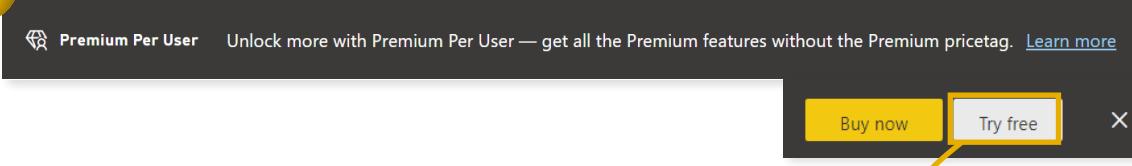


HEY THIS IS IMPORTANT!

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

Try Premium per user for free

To use this feature, upgrade to a Premium per user license. When you upgrade, you'll have access to all Premium features including paginated reports, deployment pipelines, and new AI capabilities.

Try Premium per user free for 60 days. [Learn more](#)



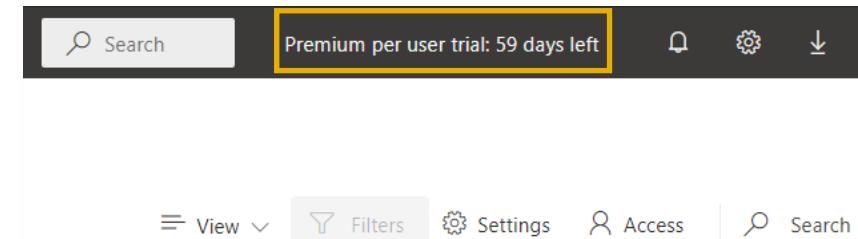
3

A Premium per user license is yours for 60 days

You get to use all Premium per user features for the next 60 days. If you love it, upgrade to a Premium per user license.



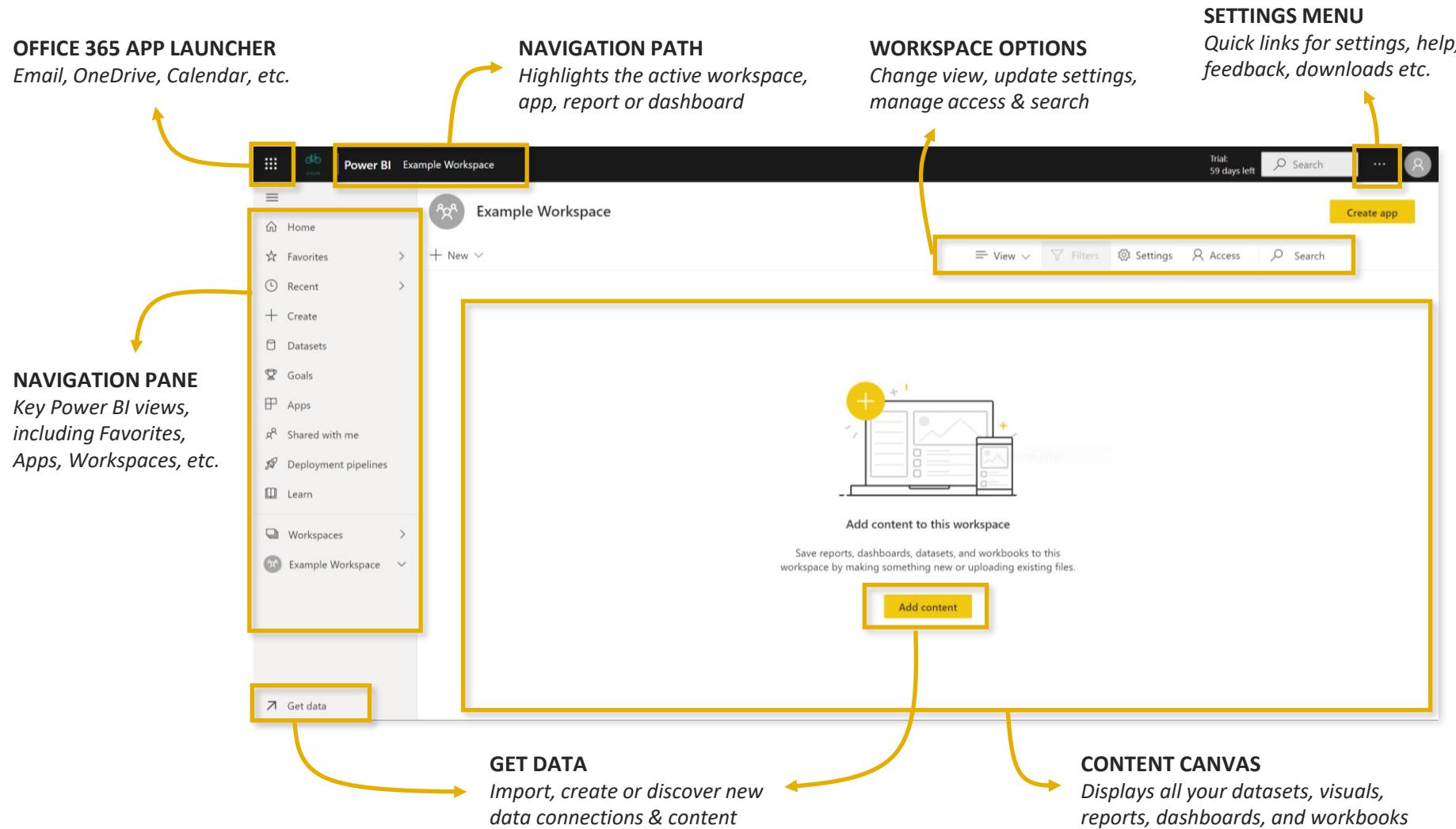
Your **trial period** lasts for **60 days**, and you'll see your remaining time in the header bar each time you log in



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.

THE POWER BI SERVICE (APP) INTERFACE



WORKSPACES

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

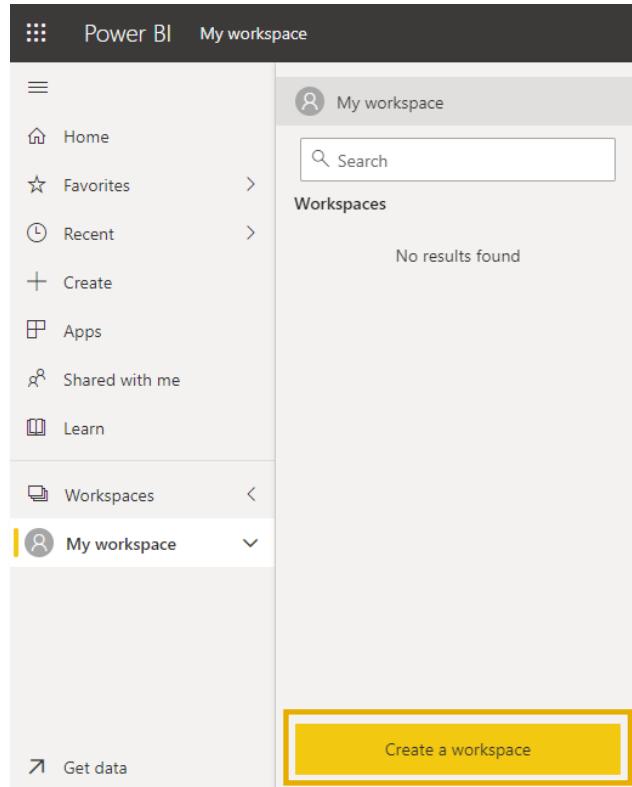
My Workspace

- **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

Create a workspace

YOU'RE CREATING AN UPGRADED WORKSPACE
Enjoy new features, better sharing options, and improved security controls.
[Revert to classic](#) | [Learn more](#)



Workspace name

Name this workspace

Description

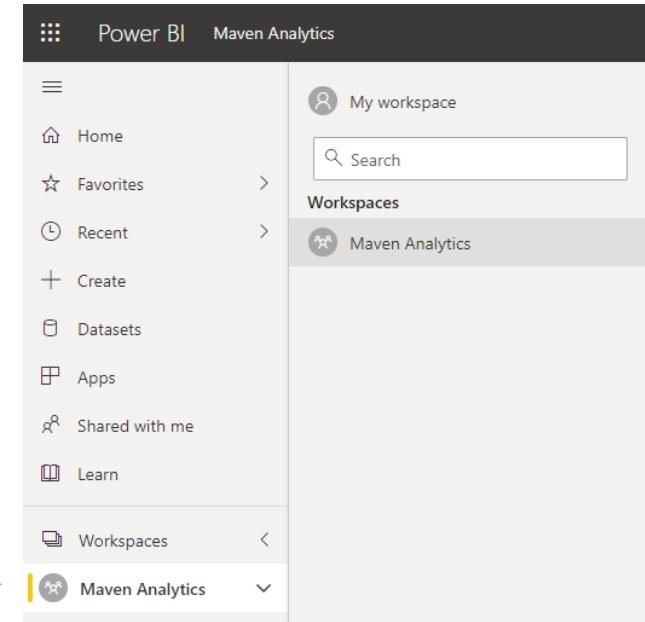
Describe this workspace

[Learn more about workspace settings](#)

Advanced

Save

Cancel



HEY THIS IS IMPORTANT!

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

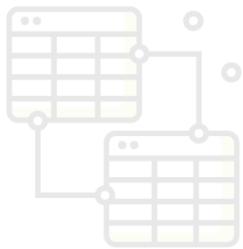
MEASURED SKILLS REVIEW

PREPARING THE DATA



Prepare the Data

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



Model the Data

- Design a Data Model
- Develop a data model
- Create measures using DAX
- Optimize Performance



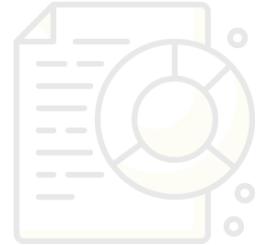
Visualize the Data

- Create reports
- Create dashboards
- Enrich reports for usability



Analyze the Data

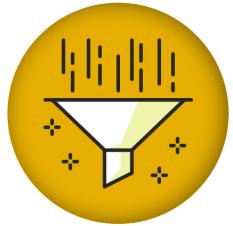
- Enhance reports to expose insights
- Perform advanced analysis



Deploy & Maintain Deliverables

- Manage datasets
- Create and 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:

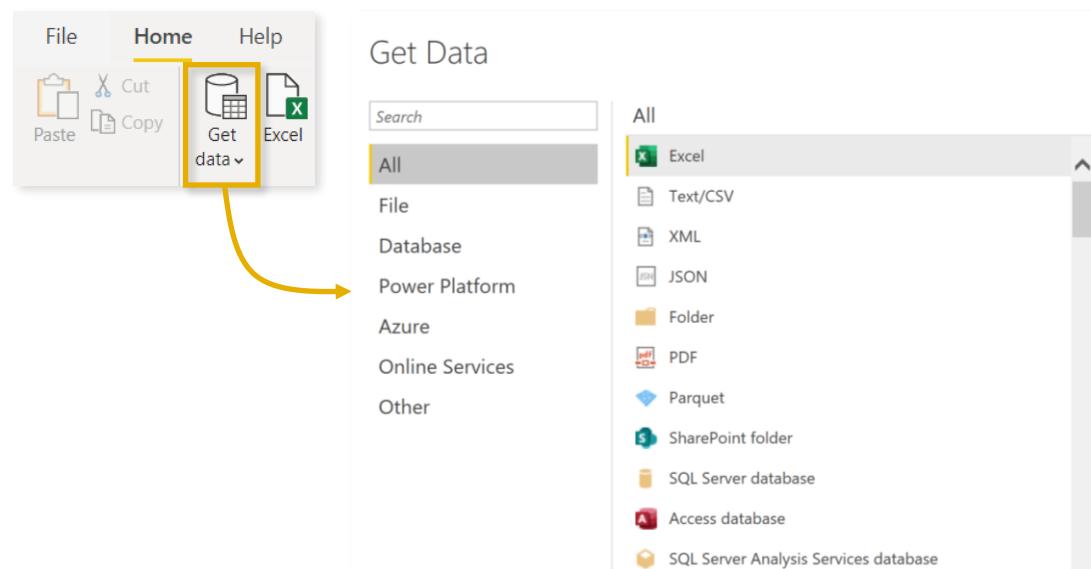
- *You need to provide users with summary sales data, what three actions should you perform in sequence?*
- *You have a 100M record MS SQL Server Orders table and need to import a sample. You add a WHERE clause to the SQL statement, does this meet the goal?*
- *What would you need to configure to easily change a data source between development, test, and production databases?*



CONNECTING TO DATA

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.)



Connecting to
Data

Data Sources

Storage Modes

Parameters



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.

The screenshot shows the Power Query Editor window with the following components highlighted:

- FORMULA BAR:** Labeled "This is 'M' code". It contains the M code: `= Table.TransformColumnTypes(#"Promoted Headers",{{"Transaction_Date", type date}})`.
- QUERY PANE:** Shows the "Maven Cycles Sales" table with columns: Transaction_Date, Stock_Date, A²c, Invoice_ID, Age_Group_Key, Customer_G. It includes a column profile table below it.
- COLUMN PROFILE:** Based on first 1,000 rows. It displays statistics for each column: Valid (100%), Error (0%), and Empty (0%).
- TABLE NAME & PROPERTIES:** Shows the table name "Maven Cycles Sales" and properties like "Valid", "Error", and "Empty".
- APPLIED STEPS:** Like a macro! It lists the steps: "Source", "Promoted Headers", and "Changed Type".



DATA SOURCE: JSON FILE

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

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

Get Data

Search:

All

- All
- File
- Database
- Azure
- Online Services
- Other

All

- Excel
- Text/CSV
- XML
- JSON**
- Folder
- SharePoint folder
- SQL Server database
- Access database
- SQL Server Analysis Services database
- Oracle database

1 Convert the JSON list to a table

= Table.FromRecords({Source})

ABC 123	people
1	List

2 Expand the attributes in the list to columns

= Table.ExpandRecordColumn#"Expanded people", "people", {"firstName", "lastName", "gender", "age"}

ABC 123	people.firstName	ABC 123	people.lastName	ABC 123	people.gender	ABC 123	people.age
1	Joe	ABC 123	Jackson	ABC 123	male	ABC 123	28
2	James	ABC 123	Smith	ABC 123	male	ABC 123	32

3 Change the data type for each column

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

ABC 123	people.firstName	ABC 123	people.lastName	ABC 123	people.gender	ABC 123	people.age
1	Joe	ABC 123	Jackson	ABC 123	male	ABC 123	28
2	James	ABC 123	Smith	ABC 123	male	ABC 123	32



DATA SOURCE: POWER BI DATA SOURCE FILES (PBIDS)

Connecting to
Data

Data Sources

Storage Modes

Parameters

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

Options and settings



Data source settings

Data source settings

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

Data sources in current file

Global permissions

Search data source settings

Power BI dataflows

Change Source... Export PBIDS

To connect to a PBIDS file:

1. Open the PBIDS file, enter credentials, launch “Data Source Navigator” window

Name

Date modified

Type

Name	Date modified	Type
DA-100 Workbook.pbix	4/22/2021 7:46 PM	Microsoft Power BI Desktop Document
Financial Reporting Dataflow.pbids	4/22/2021 8:11 PM	Microsoft Power BI Desktop Data Source

Navigator

Display Options

order_item_refund_id	created_at	order_item_id	order_id	refur
1	4/6/2012 12:32:43 PM	57	57	
2	4/13/2012 2:09:43 AM	74	74	
3	4/15/2012 3:03:48 AM	71	71	
4	4/17/2012 9:00:37 PM	118	118	
5	4/22/2012 9:53:49 PM	116	116	
6	5/4/2012 12:59:07 PM	147	147	
7	5/12/2012 3:41:14 AM	186	186	
8	5/16/2012 2:06:01 PM	191	191	
9	5/24/2012 5:00:09 PM	179	179	
10	5/30/2012 6:20:44 PM	199	199	
11	6/6/2012 2:22:14 PM	271	271	
12	6/10/2012 8:13:12 PM	290	290	
13	6/20/2012 8:13:12 PM	335	335	
14	6/28/2012 7:54:22 PM	382	382	
15	6/29/2012 4:50:01 PM	357	357	
16	7/5/2012 2:34:53 PM	409	409	
17	7/7/2012 8:36:44 AM	368	368	
18	7/7/2012 8:54:28 PM	391	391	
19	7/8/2012 3:59:09 AM	424	424	
20	7/10/2012 3:31:05 PM	393	393	
21	7/11/2012 11:16:10 PM	442	442	
22	7/19/2012 7:57:16 AM	472	472	
23	7/20/2012 7:51:51 PM	470	470	

Load Transform Data Cancel



DATA SOURCE: MICROSOFT DATAVERSE

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

Connecting to
Data

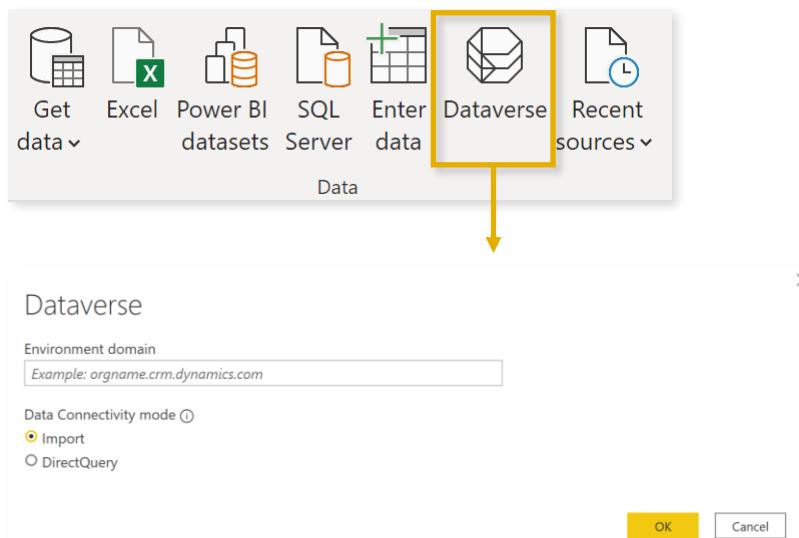
Data Sources

Storage Modes

Parameters

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!

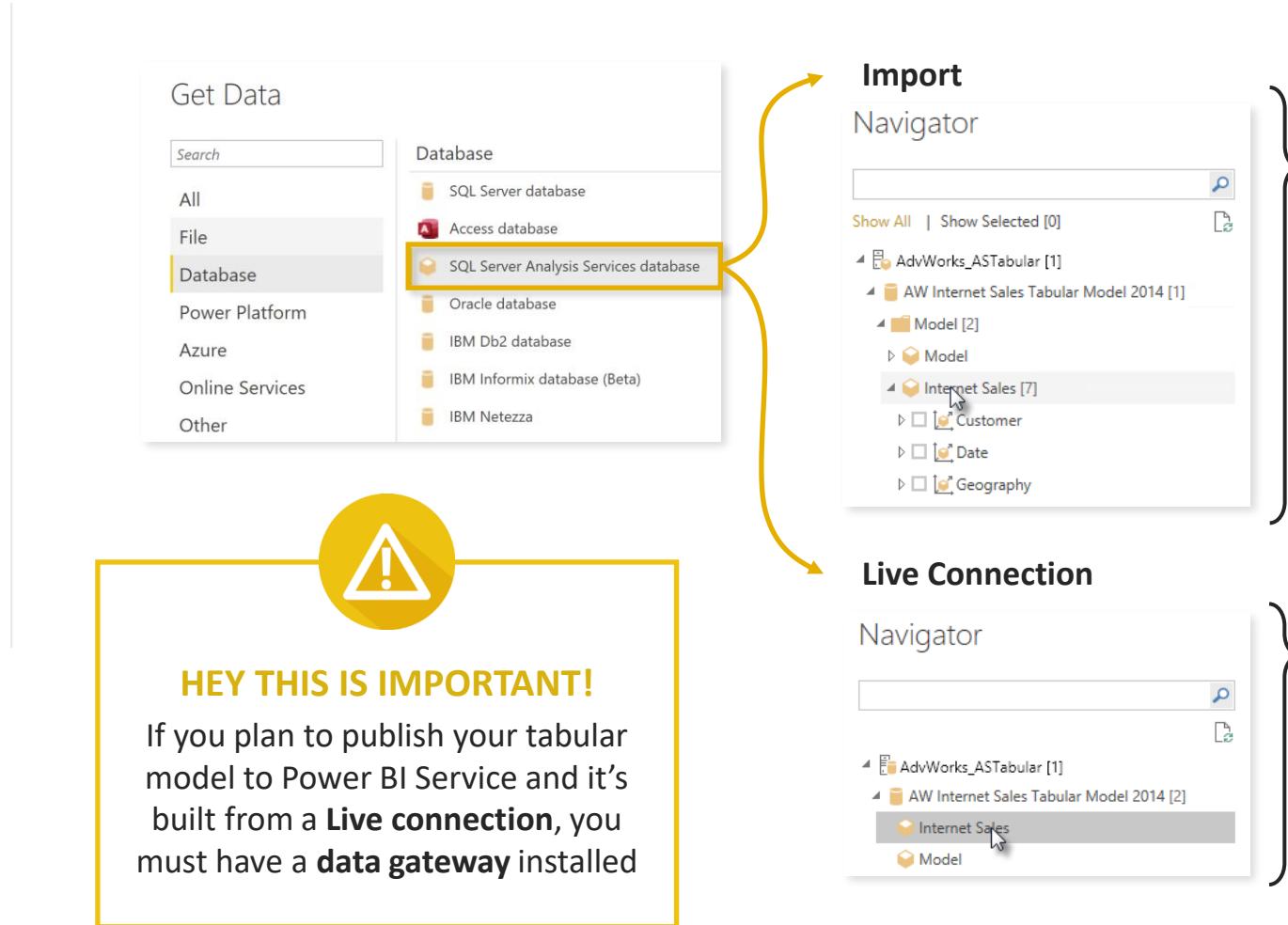
To connect to Microsoft Dataverse, you'll need to know the server address, which typically has the following format: *https://company.crm.dynamics.com*. And maker permissions to access the portal and read permissions to access data within tables

DATA SOURCE: SQL SERVER ANALYSIS SERVICES TABULAR (SSAS)



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

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



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

The screenshot shows the SharePoint Home page. At the top, there's a blue header bar with the SharePoint logo and navigation links. Below it, a large blue banner with white text says "SharePoint". Underneath, there are two buttons: "+ Create site" and "+ Create news post". The "Frequent sites" section lists two items: "Maven Internal Reporting Group" (marked with a star) and "MySQL Dataflow Demo Group" (marked with a star). At the bottom, there are two notifications: "AP You viewed MASTER Maven Partner Raw Data on 4/5/2021" and "AP You viewed tickets 10 hours ago".

Document Libraries

The screenshot shows a SharePoint Document Library titled "DA-100 Example". The library interface includes a toolbar with "+ New", "Upload", "Edit in grid view", "Sync", "Add shortcut to OneDrive", and more. On the left, a navigation pane lists "Shared with us", "Notebook", "Pages", "Incremental Refresh Test", and "DA-100 Example" (which is highlighted with a yellow box). The main area displays a grid of items with columns for Name, Modified, and Modified By. Two items are listed: "Test 1" and "Test 2", both modified yesterday at 7:34 PM by Aaron Parry.

Name	Modified	Modified By
Test 1	Yesterday at 7:33 PM	Aaron Parry
Test 2	Yesterday at 7:34 PM	Aaron Parry



DATA SOURCE: SHAREPOINT ONLINE

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

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

Get Data

SharePoint X

- All
- SharePoint folder**
- All
- File
- Online Services
- Other

1

Enter the site root URL

SharePoint folder

Site URL A^BC

https://mavenanalytics.sharepoint.com/sites/PowerBIExample/

2

Combine & Transform the data

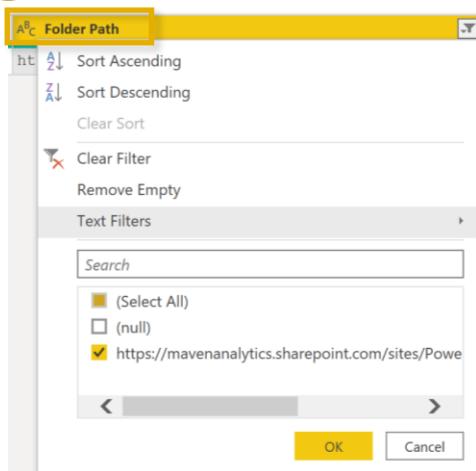
https://mavencycles.sharepoint.com/sites/DA-100ExampleSite

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	products.csv	.csv	null	6/8/2021 20:51:15	6/8/2021 20:51:15	Record	https://mavencycles.sharepo
Binary	inventory.csv	.csv	null	6/8/2021 20:51:15	6/8/2021 20:51:15	Record	https://mavencycles.sharepo
Binary	stores.csv	.csv	null	6/8/2021 20:51:17	6/8/2021 20:51:17	Record	https://mavencycles.sharepo
Binary	inventory_data_dictionary.csv	.csv	null	6/8/2021 20:51:18	6/8/2021 20:51:18	Record	https://mavencycles.sharepo
Binary	products_data_dictionary.csv	.csv	null	6/8/2021 20:51:18	6/8/2021 20:51:18	Record	https://mavencycles.sharepo
Binary	sales_data_dictionary.csv	.csv	null	6/8/2021 20:51:43	6/8/2021 20:51:43	Record	https://mavencycles.sharepo
Binary	stores_data_dictionary.csv	.csv	null	6/8/2021 20:51:43	6/8/2021 20:51:43	Record	https://mavencycles.sharepo

Combine ▾ Load Transform Data
Combine & Transform Data
Combine & Load

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

The screenshot shows the Power BI Desktop ribbon with the 'File' tab selected. The 'Data source settings' button in the 'Sources' group is highlighted with a yellow box and a yellow arrow pointing to it from the left. Below the ribbon is a window titled 'Data source settings'. The window contains a message about managing data sources, two radio button options ('Data sources in current file' and 'Global permissions'), a search bar, and a list of local CSV files. At the bottom are buttons for 'Change Source...', 'Edit Permissions...', 'Clear Permissions', and a 'Close' button.

Comma-Separated Values

Basic Advanced

File path: C:\Users\Chris\Desktop\Power BI Course Files\Adventure Works\Adventure

Open file as: Csv Document

File origin: 1252: Western European (Windows)

Line breaks: Apply all line breaks

Delimiter: Comma



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

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

Connecting to Data

Data Sources

Storage Modes

Parameters

The screenshot illustrates the process of creating a dynamic parameter in the Power Query Editor:

- Power Query Editor:** Shows the ribbon with "Manage Parameters" selected. A callout points to the "New Parameter" button.
- Manage Parameters Dialog:** A new parameter is being created:
 - Name:** AdventureWorks Calendar
 - Type:** Text (selected from a dropdown)
 - Current Value:** C:\Users\ampar\Desktop\AdventureWorksCSVFiles-15548
- Applied Steps:** Shows the steps taken so far:
 - Source
 - Changed Type
 - Promoted Headers
- Comma-Separated Values Dialog:** This dialog is used to define the file path for the dynamic source. It shows:
 - File path: AdventureWorks Calendar
 - Type: Text (selected)
 - Delimiter: Comma

Annotations explain the parameter fields:

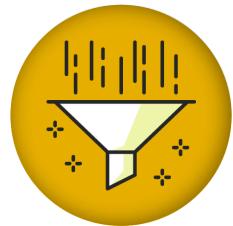
- Parameter name:** Points to the "Name" field in the Manage Parameters dialog.
- Parameter data type:** Points to the "Type" dropdown in the Manage Parameters dialog.
- Parameter value:** Points to the "Current Value" field in the Manage Parameters dialog.

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](#)

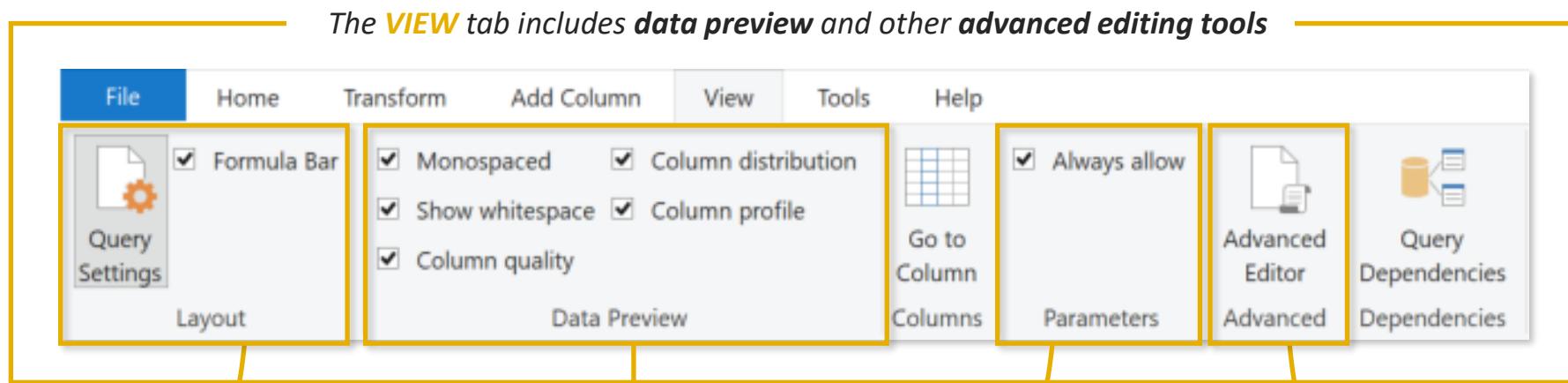
COMMON QUESTIONS:

- *What data preview option would you use to identify the percentage of empty values in a column as quickly as possible?*
- *You need to ensure a column contains records for each region, which of the following actions should you perform?*
- *You view a query with column profile and distribution enabled, which of the following choices are correct?*



VIEW MENU

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



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

Enable **parameter creation**

Open the **M code advanced editor**

Data profiling tools like column quality, distribution, and profile



DATA PROFILING TOOLS (COLUMN QUALITY)

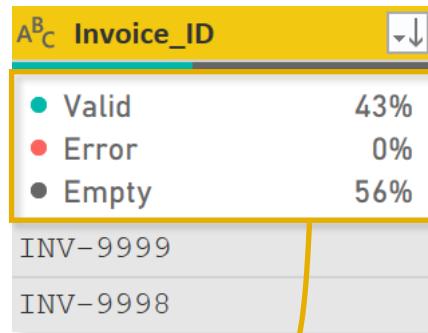
Data profiling tools like *column quality*, *column distribution*, and *column profile* provide a visual way to **explore data** and get a sense of your **dataset composition**

View Menu

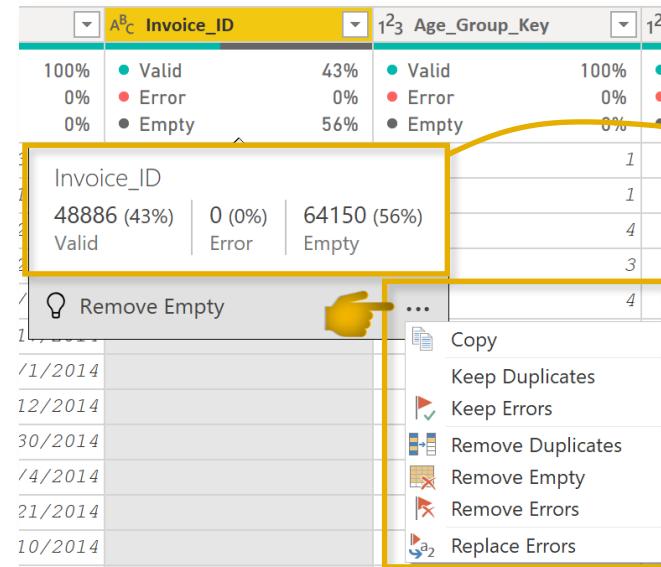
Column Quality

Column Distribution

Column Profile



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



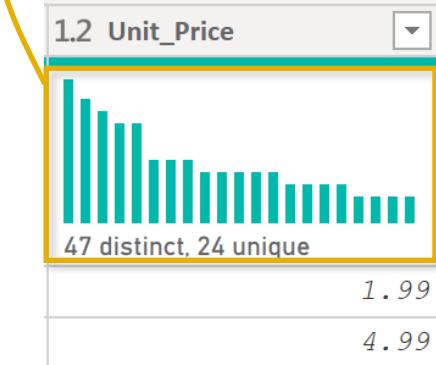
PRO TIP: Use the contextual menu to solve **column quality** issues instead of having to right-click or find the right Power Query option



DATA PROFILING TOOLS (COLUMN DISTRIBUTION)

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

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



Suggested action based on column distribution results

Transaction_Date	Stock_Date	A ^B C
155 distinct, 2 unique	276 distinct, 55 unique	10
8/30/2014 INV		
10/17/2014 INV		
11/28/2014 INV		
8/22/2014 INV		
	12/9/2014 INV	
7 1/2/2015		
8 1/2/2015		
9 1/2/2015		
10 1/3/2015		
11 1/3/2015		
12 1/3/2015		

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

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



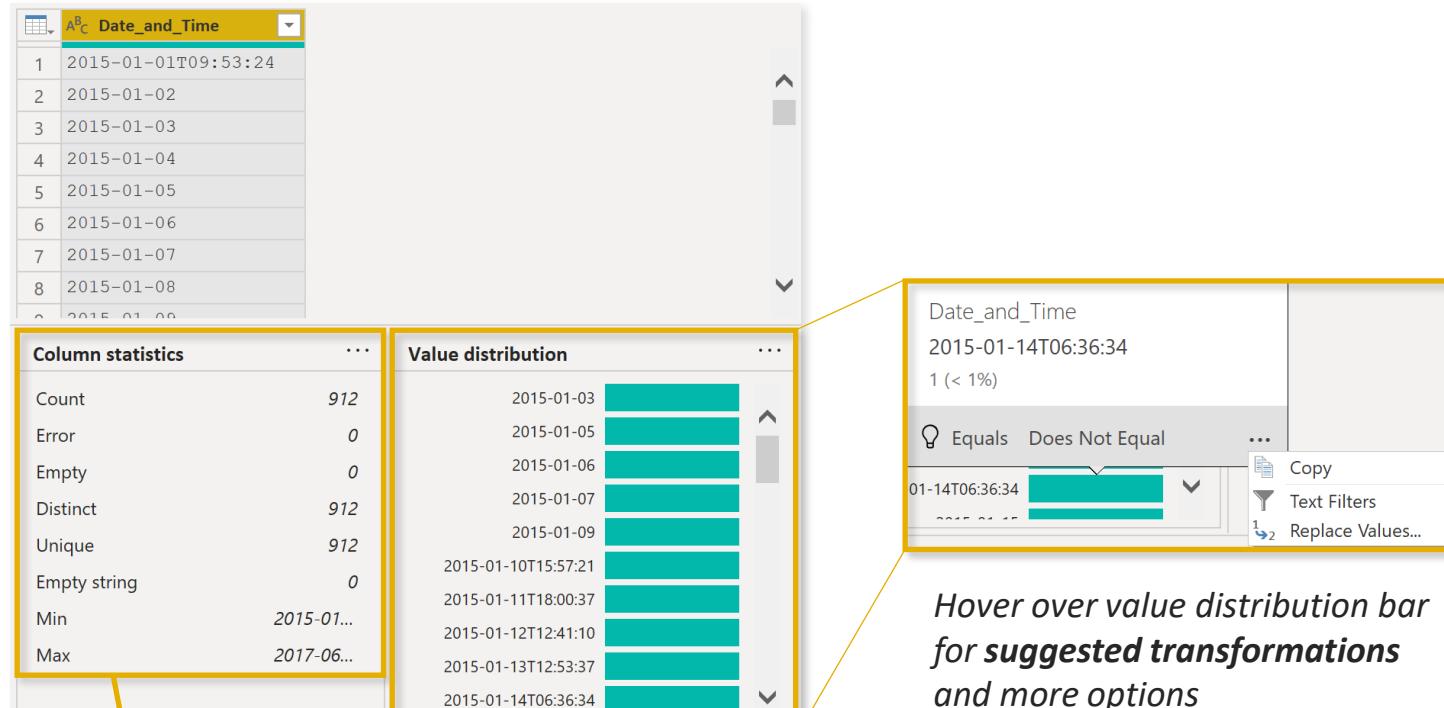
PRO TIP: Use column distribution to identify primary keys within your lookup tables



DATA PROFILING TOOLS (COLUMN PROFILE)

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

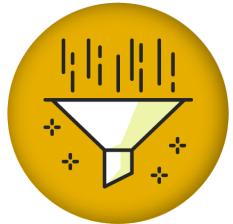
- [View Menu](#)
- [Column Quality](#)
- [Column Distribution](#)
- [Column Profile](#)



Column statistics provide a bit more detail than distribution view

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

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:

- *You create the following step by using Power Query, what will the value be when the step is applied?*
- *You import a dataset and want to remove "id" from columns with that suffix, how would you complete the following M code?*
- *You need to prepare the data to support year-over-year calculations, which actions should you perform in sequence?*



BASIC TABLE TRANSFORMATIONS

Cleaning Data

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

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

Identifying Keys

Shaping Tables

Combining
Queries

Modifying M Code

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

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

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

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



BASIC TABLE TRANSFORMATIONS

Cleaning Data

Identifying Keys

Shaping Tables

Combining Queries

Modifying M Code

The screenshot shows the Power BI ribbon with the 'Transform' tab selected. The ribbon includes sections for File, Home, Transform, Add Column, View, Tools, and Help. Under the Transform tab, there are several groups of tools:

- Manage Columns:** Includes 'Choose Columns', 'Remove Columns', 'Keep Rows', 'Remove Rows', and 'Reduce Rows'.
- Sort:** Includes 'Sort' (with A-Z and Z-A options) and 'All'.
- Data Type:** Includes 'Data Type: Date' dropdown, 'Use First Row as Headers', and 'Replace Values'.
- Transform:** Includes 'Split Column', 'Group By', 'Replace Values', 'Append Values', 'Merge Queries', 'Append Queries', 'Combine Files', and 'Combine'.

Annotations with arrows point to specific features:

- An arrow points from the 'Promote header row' text to the 'Use First Row as Headers' option in the Data Type group.
- An arrow points from the 'Choose or remove columns' text to the 'Remove Columns' and 'Remove Other Columns' options in the Manage Columns group.
- An arrow points from the 'Keep or remove rows' text to the 'Remove Top Rows', 'Remove Bottom Rows', 'Remove Alternate Rows', 'Remove Duplicates', 'Remove Blank Rows', and 'Remove Errors' options in the Sort group.
- An arrow points from the 'Duplicate, move & rename columns' text to the context menu that appears when right-clicking a column header, specifically highlighting the 'Transform' option.

Table Data: A table titled 'Stock Date' is shown with columns 'A_B_Invoice_ID' and 'Stock Date'. The table contains 18 rows of data, with the last two rows highlighted in yellow. The context menu for the first row shows options like 'Copy', 'Remove', 'Remove Other Columns', 'Duplicate Column', etc.

A_B_Invoice_ID	Stock Date
1	7
2	7
3	7
4	7
5	7
6	7
7	7
8	7
9	7
10	7
11	7
12	7
13	7
14	7
15	7
16	7
17	7
18	7
	6/26/2017 INV-9983
	5/28/2017 INV-9982



ADDING INDEX COLUMNS

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

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



The screenshot shows the Power BI desktop interface. The ribbon is open at the 'Add Column' tab. The 'Index Column' button is highlighted with a yellow box and a yellow arrow points from it to the 'Index' column in the data grid below. The data grid contains 16 rows of data with columns: Transaction_Date, Stock_Date, ABC, and Invoice_ID. The 'ABC' column is labeled 'Index' and shows sequential values from 1 to 16. The 'Invoice_ID' column shows unique identifiers for each row.

	Transaction_Date	Stock_Date	ABC	Invoice_ID
1	8/5/2017	7/7/2017	1	INV-9999
2	8/5/2017	7/10/2017	2	INV-9998
3	8/5/2017	7/31/2017	3	INV-9997
4	8/5/2017	7/9/2017	4	INV-9996
5	8/5/2017	6/11/2017	5	INV-9995
6	8/5/2017	2/13/2017	6	INV-9994
7	8/5/2017	6/23/2017	7	INV-9993
8	8/5/2017	3/12/2017	8	INV-9992
9	8/5/2017	3/19/2017	9	INV-9991
10	8/5/2017	6/16/2017	10	INV-9990
11	8/5/2017	7/12/2017	11	INV-9989
12	8/5/2017	7/25/2017	12	INV-9988
13	8/5/2017	7/29/2017	13	INV-9987
14	8/5/2017	2/26/2017	14	INV-9986
15	8/5/2017	4/8/2017	15	INV-9985
16	8/5/2017	6/13/2017	16	INV-9984



ADDING CONDITIONAL COLUMNS

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

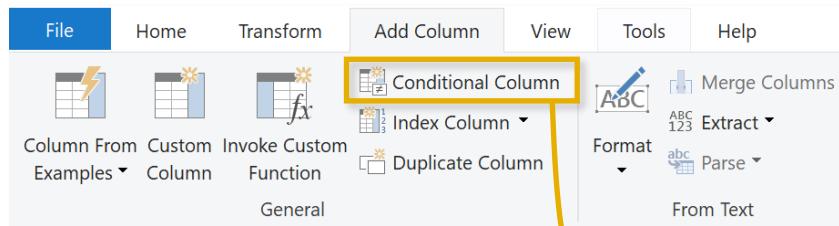
Cleaning Data

Identifying Keys

Shaping Tables

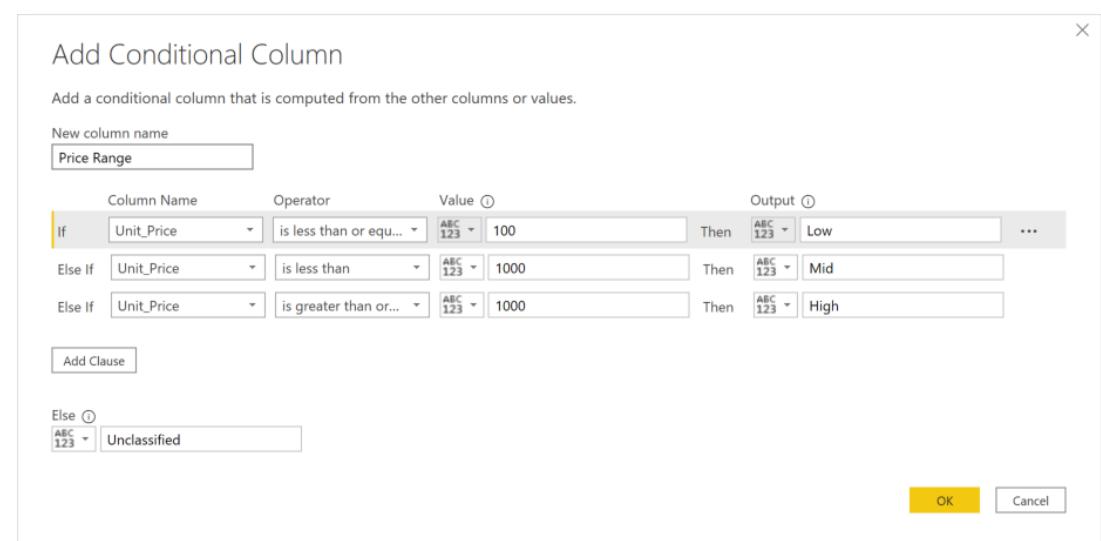
Combining
Queries

Modifying M Code



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

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



The screenshot shows the Power BI desktop interface. The ribbon is open at the 'Tools' tab, with the 'Add Column' option highlighted. A yellow box highlights the 'Column From Examples' button under the 'General' section. An arrow points down to the 'Add Column From Examples' dialog box. This dialog has 'Query Settings' and 'PROPERTIES' sections. In the 'PROPERTIES' section, 'Name' is set to 'AdventureWorks_Calen'. Below the dialog is a table with four rows of data from the 'Date_and_Time' column. A yellow box highlights the first row's value '2015-01-01'. An arrow points from this value to a 'Column1' properties dialog, which also shows '2015-01-01'. Another arrow points from this dialog to a 'Text Before Delimiter' table, which contains dates from January 1 to 6, 2015. A final arrow points from this table to an 'APPLIED STEPS' pane, which lists 'Source', 'Changed Type', 'Promoted Headers', and 'Inserted Text Before Delimiter'.

New column created based on example

Creates column using text before delimiter transformation

Text Before Delimiter

2015-01-01
2015-01-02
2015-01-03
2015-01-04
2015-01-05
2015-01-06

APPLIED STEPS

- Source
- Changed Type
- Promoted Headers
- Inserted Text Before Delimiter



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

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.

Cleaning Data

Identifying Keys

Shaping Tables

Combining Queries

Modifying M Code

The screenshot illustrates the Power BI interface for transforming data. On the left, a table of raw transaction data is shown with columns: Transaction_Date, Product_Key, and Stock_Date. A yellow arrow points from the 'Group By' button in the 'Transform' ribbon to a 'Group By' dialog box. This dialog box shows 'ProductKey' selected for grouping, 'TotalQuantity' as the new column name, and 'Sum' as the operation for the 'OrderQuantity' column. An 'OK' button is visible. To the right of the dialog is a summary table with columns: Product_Key and TotalQuantity, containing 15 rows of data where each product's total quantity is summed up.

Product_Key	TotalQuantity
1	271
2	273
3	252
4	244
5	227
6	247
7	255
8	269
9	229
10	249
11	268
12	226
13	245
14	314
15	248

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

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

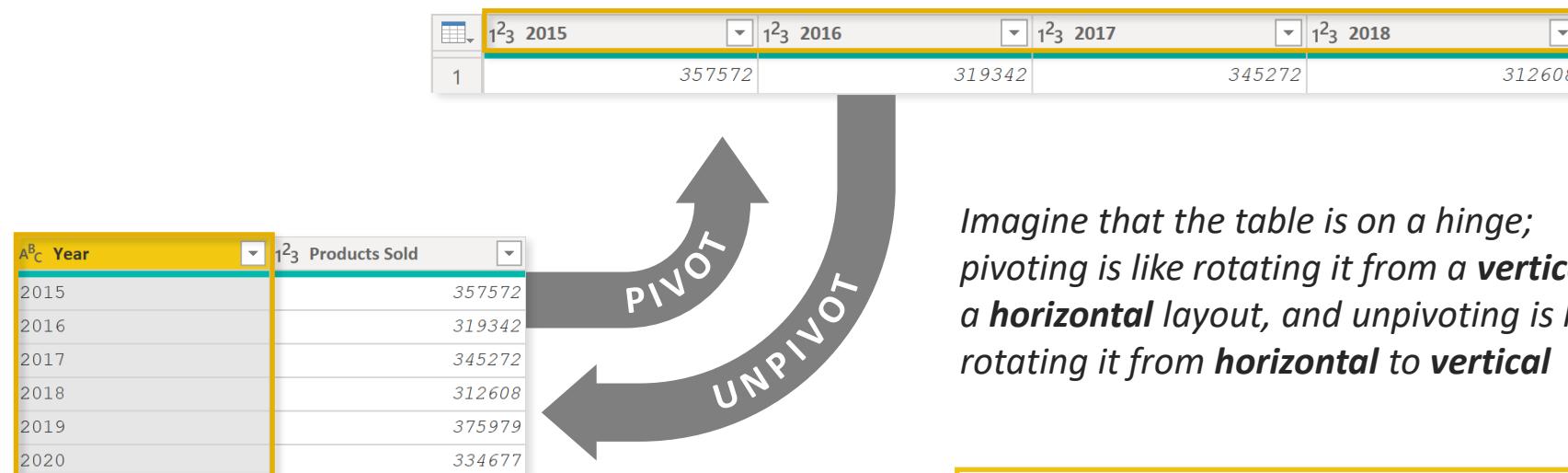
Cleaning Data

Identifying Keys

Shaping Tables

Combining Queries

Modifying M Code



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

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

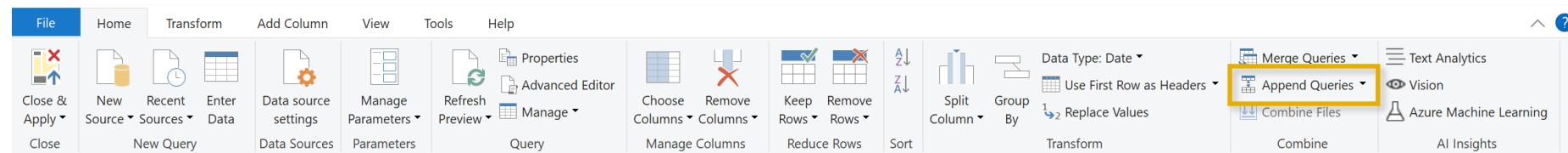
Cleaning Data

Identifying Keys

Shaping Tables

Combining
Queries

Modifying M Code



Append

Concatenate rows from two tables into a single table.

Two tables Three or more tables

First table

Maven Cycles Sales - 2019

Second table

Maven Cycles Sales - 2020

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

The screenshot shows the Microsoft Power BI desktop application. The ribbon is visible at the top with the Home tab selected. In the center, the Power Query Editor window is open, displaying a table titled "Maven Cycles Sales". The "Queries" pane on the left lists nine queries, with "Maven Cycles Sales" currently selected. The main area shows a table with columns: Transaction_Date, Product_Key, Stock_Date, Invoice_ID, and Age_Group_Key. The "Applied Steps" pane on the right shows a list of transformations applied to the query, including "Promoted Headers", "Changed Type", "Reordered Columns", and "Sorted Rows". Arrows point from the text instructions to the corresponding sections in the Power Query Editor interface.



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**

The screenshot shows the Power Query Editor interface. On the left, there's a context menu for the 'Product_Key' column, with 'Sort Ascending' selected. In the center, a table is displayed with several rows of data. At the top, the formula bar shows the M code: = Table.Sort(#"Reordered Columns",{{"Product_Key", Order.Ascending}}). On the right, the 'Query Settings' pane is open, showing the 'APPLIED STEPS' section with 'Sorted Rows' listed.

Transaction_Date	Product_Key	Stock_Date	Invoice_ID	Age_Group_Key
7/30/2020	201	6/9/2020	INV-113948	
7/8/2017	201	5/14/2017	INV-8430	
9/3/2017	201	7/8/2017	INV-13571	
7/30/2020	201	3/26/2020	INV-113947	
7/1/2017	201	3/16/2017	INV-8098	
8/30/2017	201	6/29/2017	INV-13085	
7/28/2017	201	2/18/2017	INV-9337	
9/13/2017	201	5/24/2017	INV-14876	
8/19/2017	201	5/18/2017	INV-11782	
9/15/2017	201	4/7/2017	INV-15248	
7/28/2020	201	7/16/2020	INV-113767	
8/27/2017	201	3/22/2017	INV-12775	
8/20/2017	201	5/10/2017	INV-11927	
8/17/2017	201	4/13/2017	INV-11510	

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

The screenshot shows the 'Query Settings' pane from the Power Query Editor. It displays the 'PROPERTIES' section with the name 'Maven Cycles Sales' and the 'APPLIED STEPS' section, which contains the step 'Sorted Rows'.

Query Settings

PROPERTIES

Name: Maven Cycles Sales

APPLIED STEPS

Source, Promoted Headers, Changed Type, Reordered Columns, **Sorted Rows**

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



EDITING & ADDING APPLIED STEPS

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**

Cleaning Data

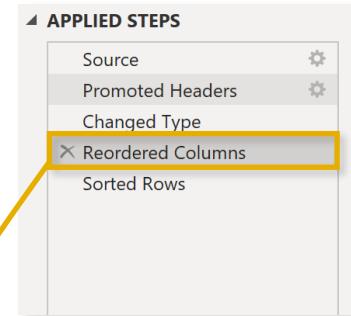
Identifying Keys

Shaping Tables

Combining
Queries

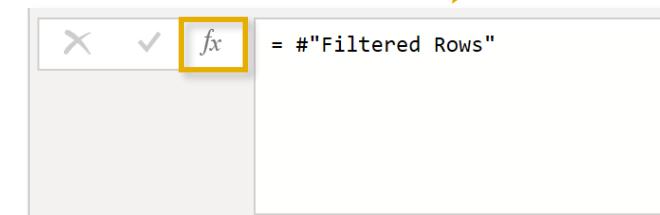
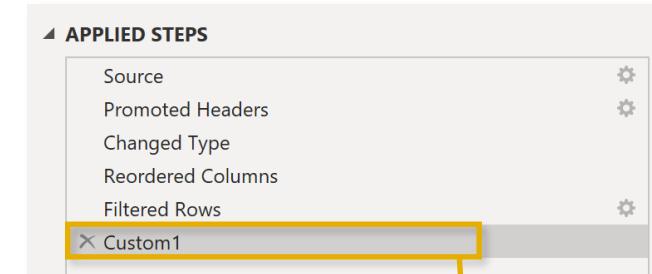
Modifying M Code

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

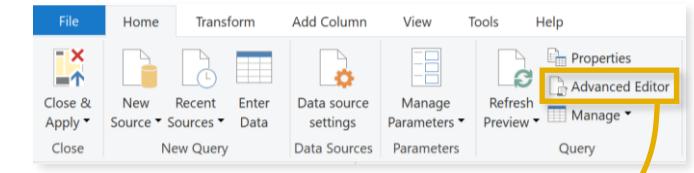
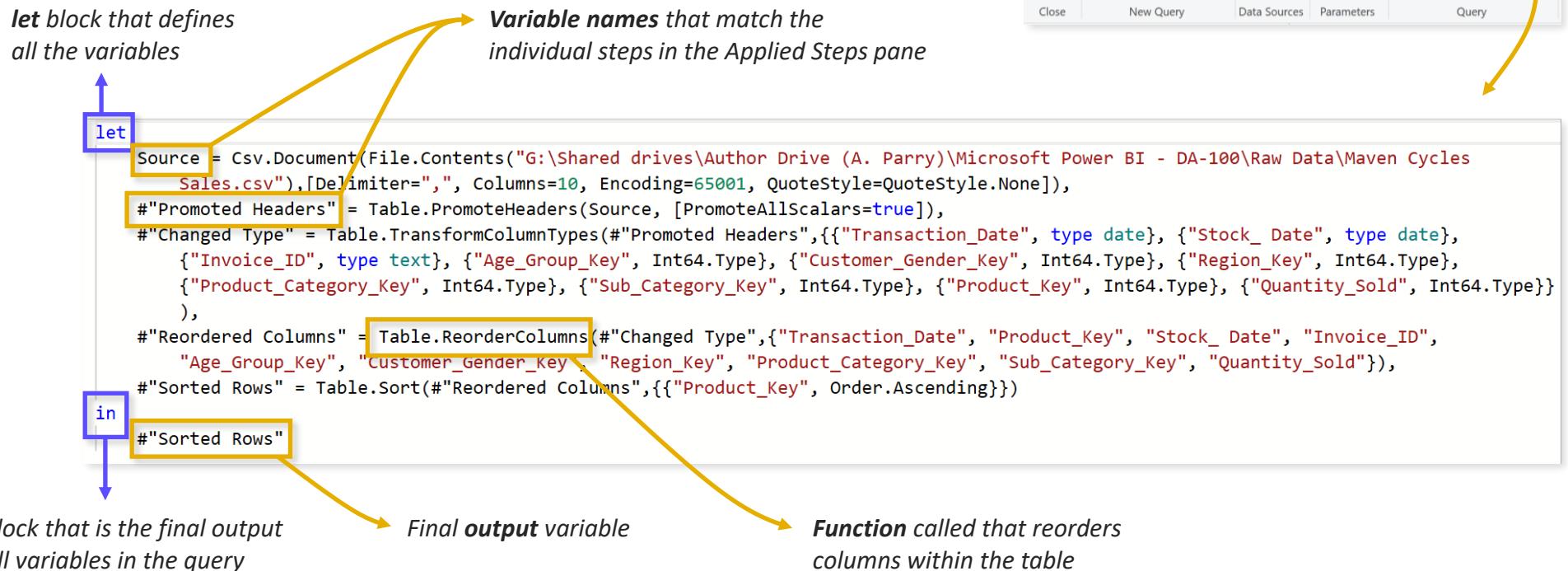




ADVANCED EDITOR (INTRO TO M CODE)

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





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)

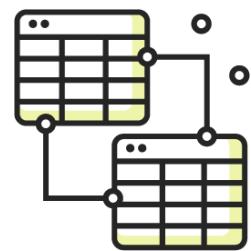


MODELING THE DATA



Prepare the Data

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



Model the Data

- Design a Data Model
- Develop a data model
- Create measures using DAX
- Optimize Performance



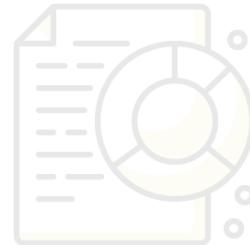
Visualize the Data

- Create reports
- Create dashboards
- Enrich reports for usability



Analyze the Data

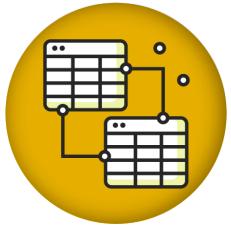
- Enhance reports to expose insights
- Perform advanced analysis



Deploy & Maintain Deliverables

- Manage datasets
- Create and 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 three tables in your model and need to show campaign totals for last year. What two actions should you take to minimize data model size?*
- *You have a large dataset (1M+ rows) with a DateTime column named Date. You need to reduce the model size, what should you do?*
- *You have two tables named Customers and Invoices. You need to define the relationship between the tables that optimizes query performance, what do you do?*



WHAT'S A "DATA MODEL"?

Data Model 101

Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips

The screenshot shows three separate tables in a data model:

- Maven Cycles Products:** Contains columns: Product_New, ProductKey, and a collapse button.
- Maven Cycles Regions:** Contains columns: SalesTerritoryKey, TerritoryManager, and a collapse button.
- Maven Cycles Sales Fact Table:** Contains columns: Age_Group_Key, Customer_Gender_Key, Invoice_ID, Product_Category_Key, Product_Key, Quantity_Sold, Region_Key, Stock_Date, Sub_Category_Key, Transaction_Date, and a collapse button.

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

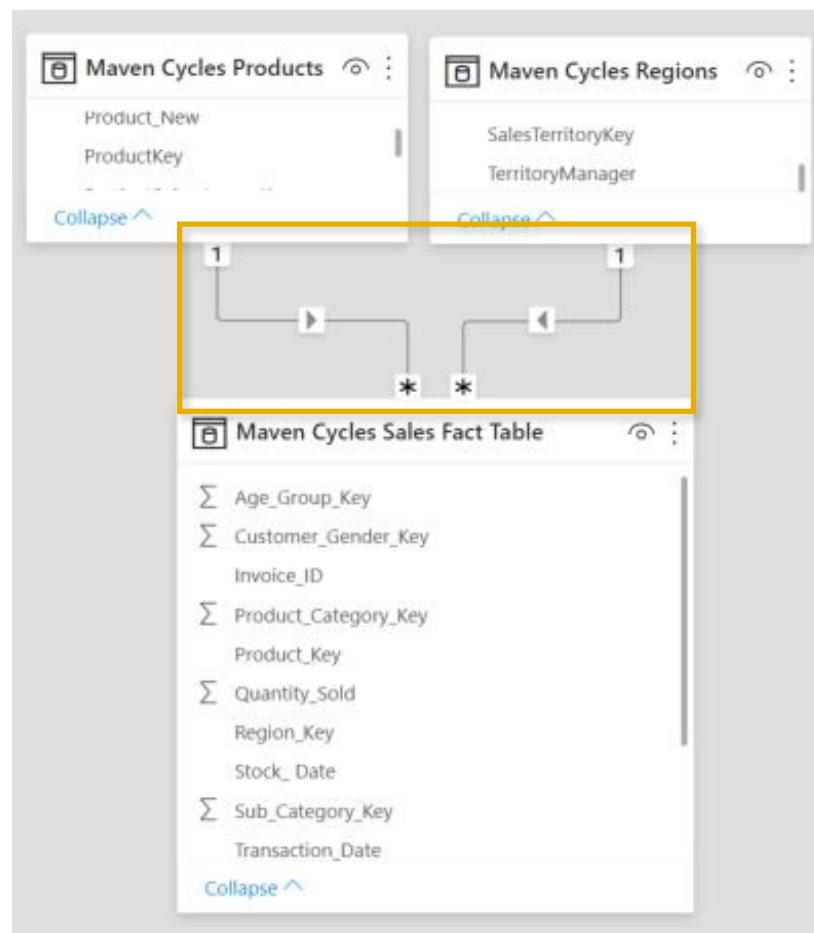
Table Roles

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

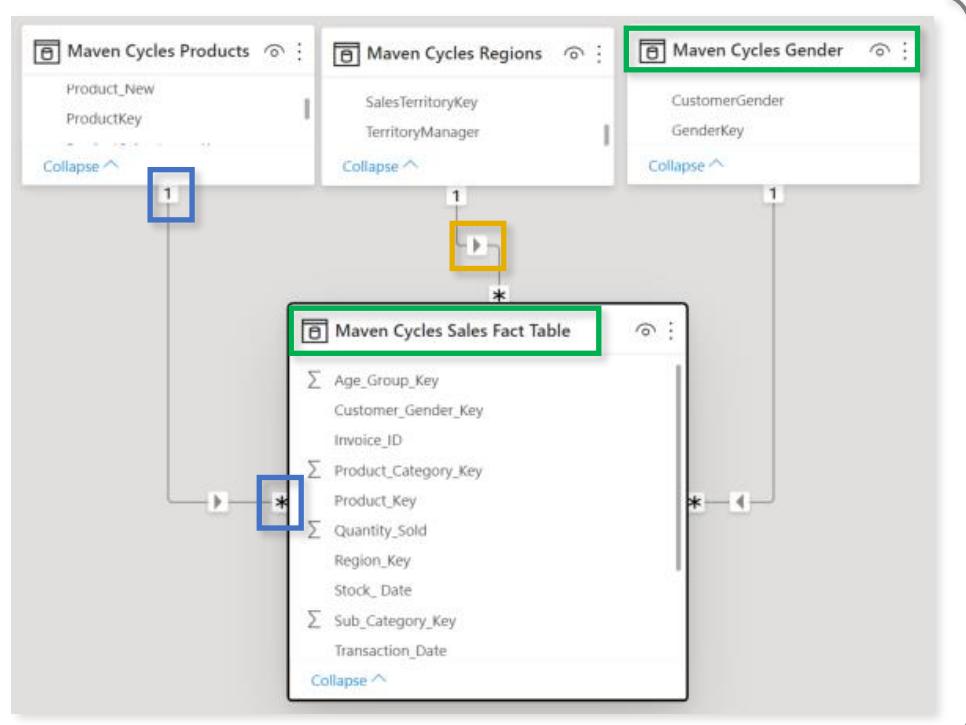
Table Roles

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

Data Model 101

Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips

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	2019144754	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

date	day_of_month	month	year	weekday	week_of_year	week_endings	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 **foreign keys**; they contain *multiple* instances of each value, and are used to match the **primary keys** in related lookup tables

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

Data Model 101

Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips



*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

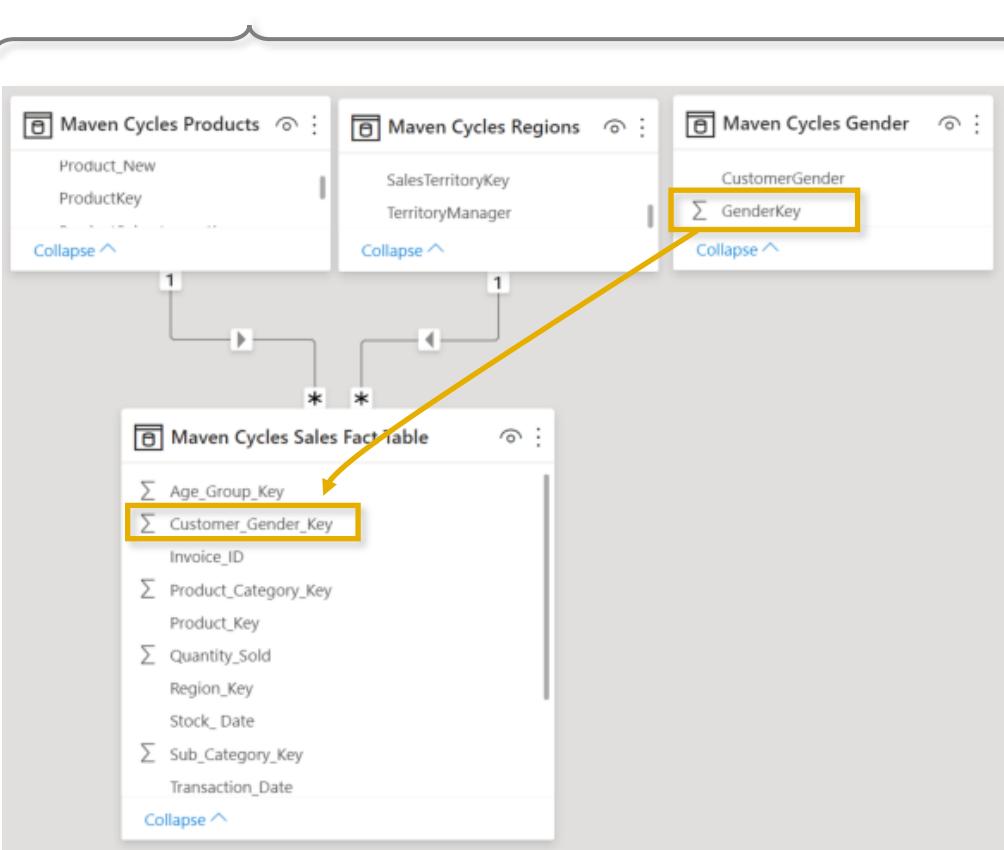
Table Roles

Table Relationships

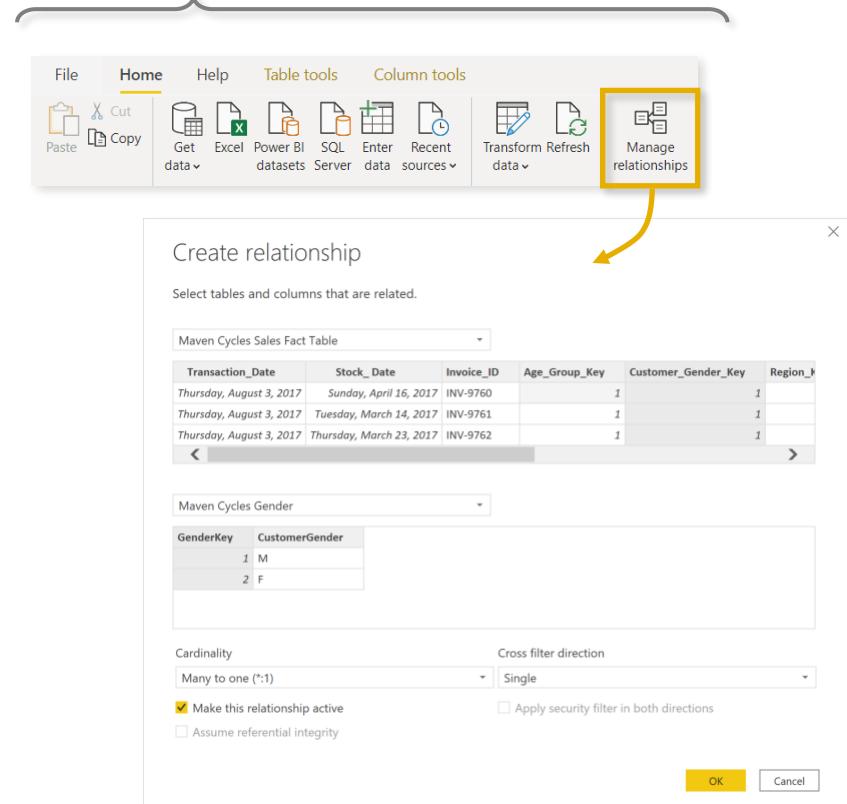
Data Model Types

Date Tables

Optimization Tips



Option 2: Add or detect relationships using the “Manage Relationships” dialog box





RELATIONSHIP CARDINALITY

Data Model 101

Table Roles

Table Relationships

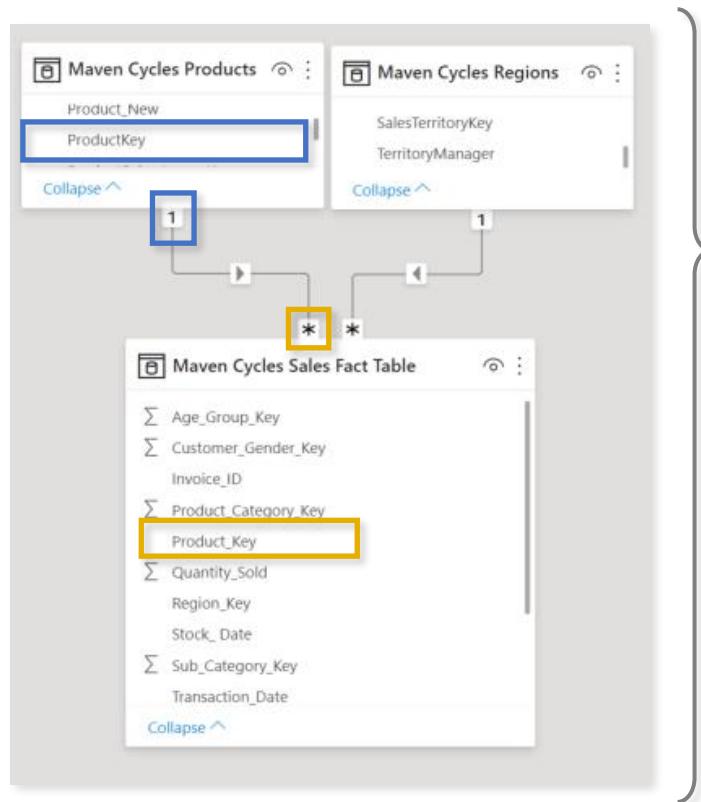
Data Model Types

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

Data Model 101

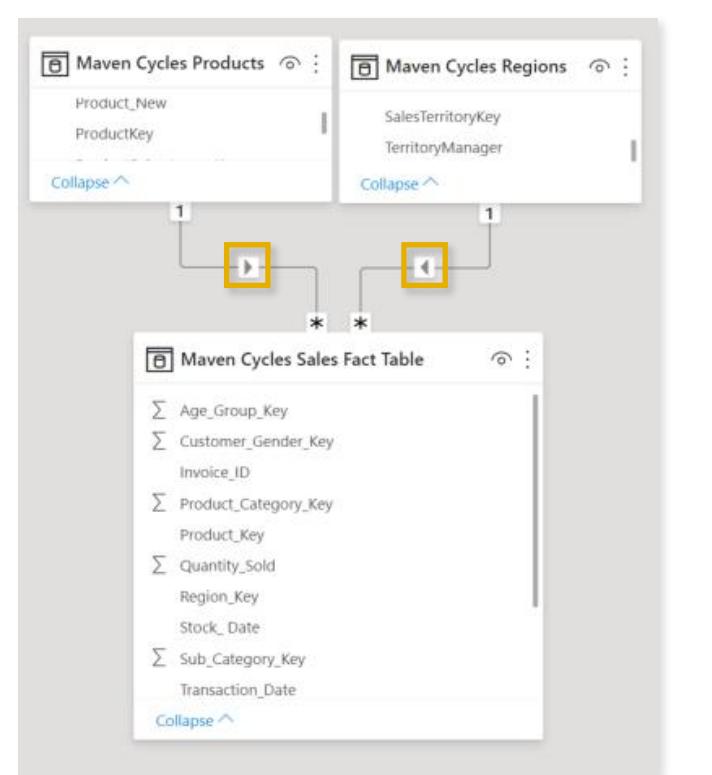
Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips



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”



ACTIVE & INACTIVE RELATIONSHIPS

Data Model 101

Table Roles

Table Relationships

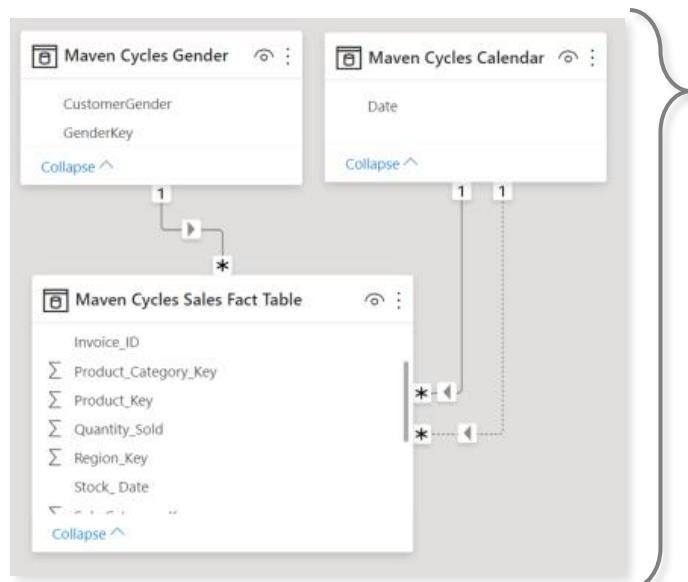
Data Model Types

Date Tables

Optimization Tips

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

Data Model 101

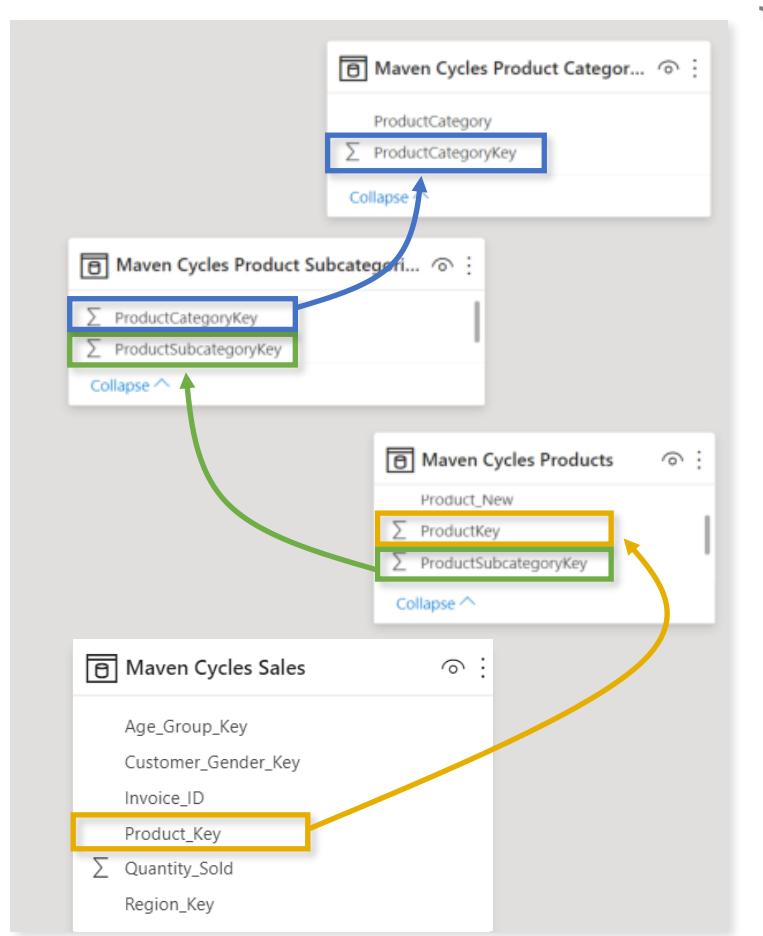
Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips



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

Data Model 101

Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips

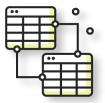
By default, Power BI automatically creates a **hidden date table** for any table that contains a **Date** or **DateTime** column on the one-side of a relationship

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

	Maven Cycles Sales	⋮
Age_Group_Key		
Customer_Gender_Key		
Invoice_ID		
Product_Key		
Quantity_Sold		
Region_Key		
Stock_Date		
Transaction_Date		

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



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

Data Model 101

Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips



DATE TABLE REQUIREMENTS

Data Model 101

Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips

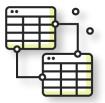
If you import or create your own date table, **it must meet these requirements:**

- ✓ 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

Data Model 101

Table Roles

Table Relationships

Data Model Types

Date Tables

Optimization Tips

Normalization is the process of organizing the tables and columns in a relational database to reduce redundancy and preserve data integrity

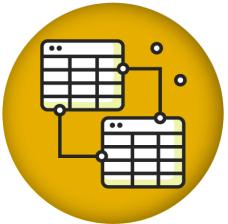
It's commonly used to:

- 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 to return the 100 highest spending customers. How should you complete the following DAX expression?
- You need to build a measure that shows revenue for the last 50 transactions, how should you complete the following DAX expression?
- You need to create a monthly rolling average for Sales, how should you configure the quick measure?



MEET DAX

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

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

The screenshot shows the Power BI Data View interface. A new column is being created with the following DAX formula:

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

The resulting table includes the original Date column and the newly added Week Number column, which contains values 1 through 5 corresponding to the dates from January 1 to January 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
Saturday, January 10, 2015	2015	1	1	2	6	Jan

2) Measures

The screenshot shows the Power BI Model view interface. A new measure is being created with the following DAX formula:

```
1 Total Orders =  
2 DISTINCTCOUNT(  
3     'Maven Cycles Sales Fact Table'[Invoice_ID]  
4 )
```

Below this, another measure is shown with the following formula:

```
1 Quantity Sold =  
2 SUM(  
3     'Maven Cycles Sales Fact Table'[Quantity_Sold]  
4 )
```



CALCULATED COLUMNS

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

- 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

Measures are DAX formulas used to generate new calculated values

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

- 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*)

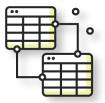


PRO TIP: Use measures to create numerical, calculated values that can be analyzed in the “values” field of a report visual



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*)



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**

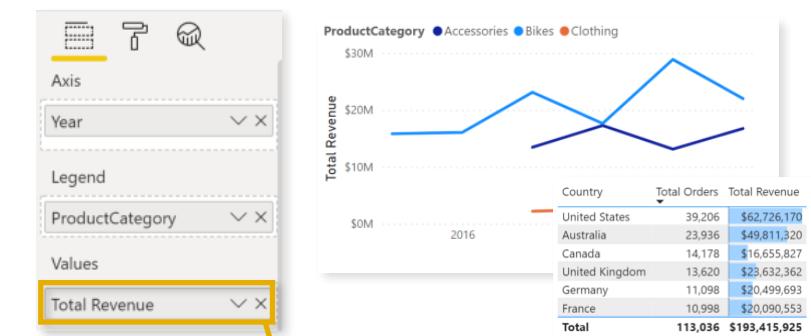
```
1 Week Number =
2 WEEKNUM(
3     'Maven Cycles Calendar'[Date],
4     2
5 )
```

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
Saturday, January 10, 2015	2015	1	1	2	6	Jan

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

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*)

Calculated Columns & Measures

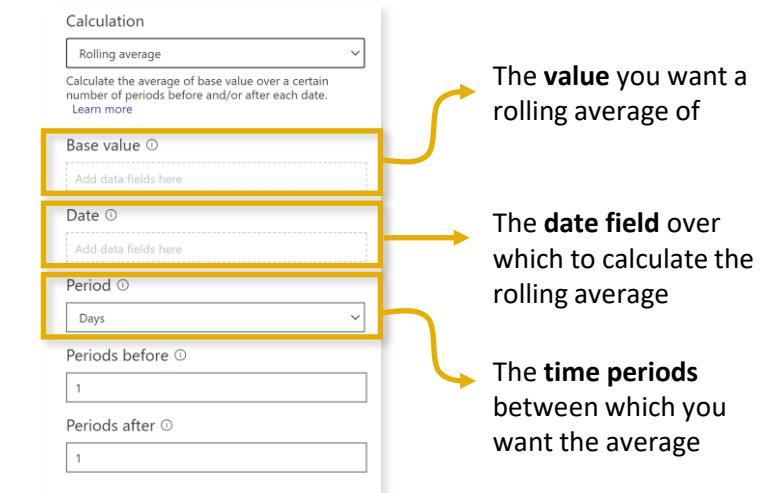
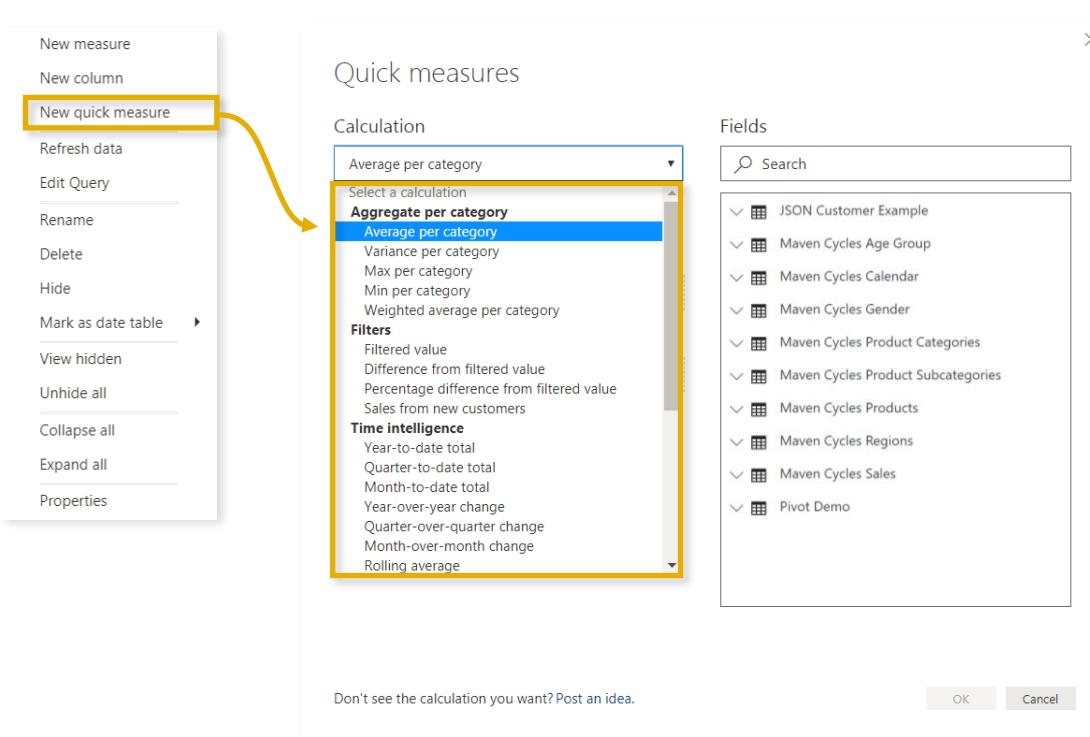
Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence





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

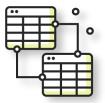
*Basic **date & time** functions as well as advanced **time intelligence** operations*

Common Examples:

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

Time Intelligence Functions:

- DATESYTD
- DATESQTD
- DATESMTD
- DATEADD
- DATESINPERIOD



BASIC AGGREGATION FUNCTIONS

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE
Modifiers

Table Functions

Time Intelligence

SUM()

Evaluates the sum of a column

=SUM(Column**Name**)

AVERAGE()

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

=AVERAGE(Column**Name**)

MAX()

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

=MAX(Column**Name**) or =MAX(Scalar1, [Scalar2])

MIN()

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

=MIN(Column**Name**) 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

Total Quantity: =SUM(Transactions[quantity])

FUNCTION NAME

Calculated columns don't always use functions, but measures do:

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

Referenced
TABLE NAME

Referenced
COLUMN NAME

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

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.*)

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE
Modifiers

Table Functions

Time Intelligence

=SUMX(Table, Expression)

Aggregation to apply to calculated rows*

Table in which the expression will be evaluated

Expression to be evaluated for each row of the given table

Examples:

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

Examples:

- ‘Sales’
- FILTER(‘Sales’, RELATED(‘Products’[Category])=“Clothing”))

Examples:

- [Total Orders]
- Sales[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



DIVIDE

DIVIDE()

Safe Divide function with ability to handle divide by zero cases

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE
Modifiers

Table Functions

Time Intelligence

=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

CALCULATE()

Evaluates a given expression or formula under a set of defined filters

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE
Modifiers

Table Functions

Time Intelligence

=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)

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

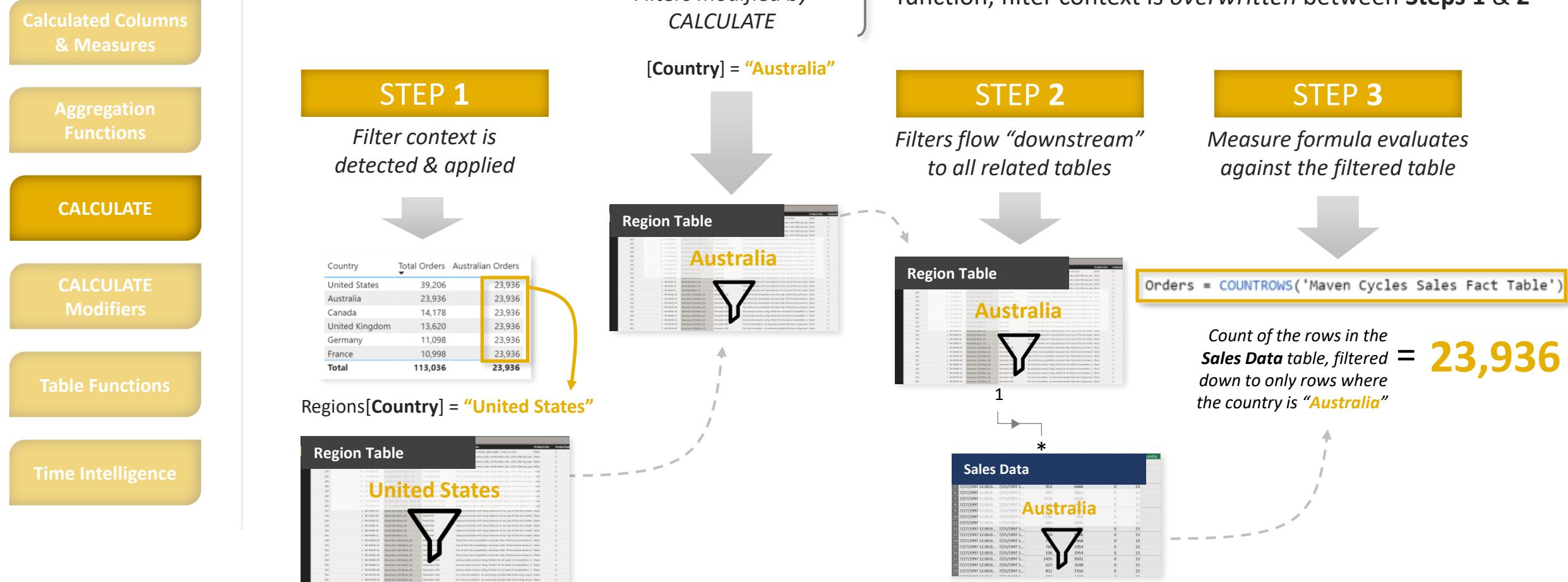
Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence





COMMON CALCULATE MODIFIERS

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE
Modifiers

Table Functions

Time Intelligence



CALCULATE MODIFIERS

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE
Modifiers

Table Functions

Time Intelligence

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

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

=USERELATIONSHIP(Column**Name1**, Column**Name2**)

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

ALL()

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

=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

FILTER()

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

=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

TOPN()

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

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

The number of rows to return

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

Optional expression that's used to sort the table

Optional expression that defines the sort order

Examples:

- 100
- 50
- 10
- 5

Examples:

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

Examples:

- 'Sales'

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

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

Calculated Columns & Measures

Aggregation Functions

CALCULATE

CALCULATE Modifiers

Table Functions

Time Intelligence

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))



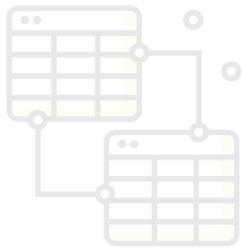
PRO TIP: Use DATESBETWEEN to return dates between two given dates

VISUALIZING THE DATA



Prepare the Data

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



Model the Data

- Design a Data Model
- Develop a data model
- Create measures using DAX
- Optimize Performance



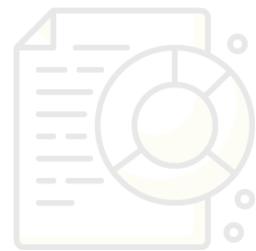
Visualize the Data

- Create reports
- Create dashboards
- Enrich reports for usability



Analyze the Data

- Enhance reports to expose insights
- Perform advanced analysis



Deploy & Maintain Deliverables

- Manage datasets
- Create and 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:

- *You have two charts, bar & column. How would you adjust visual interactions so a segment selection on the bar chart shows the portion of sales on the column chart?*
- *You have a line chart that shows the number of employees in a department, and you want to see the salary when you hover. How can you achieve this goal?*
- *You need to build a report that will be frequently viewed in portrait mode on mobile phones. What four steps should you perform in sequence?*



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.

The screenshot shows the Power BI Report View interface. At the top is the ribbon menu with options like File, Home, Insert, Modeling, View, Help, External Tools, Format, Data / Drill, and various icons for themes, filters, and sync. Below the ribbon are several visualizations: a card with metrics (\$7.74M Revenue, \$3.91M Profit, 50.50% Profit Margin), a world map showing revenue by country, a donut chart for profit by category, and a line chart for profit by period. To the left is a 'PRODUCT DETAIL' table. On the right are three panes: 'FILTERS' (containing filters for Product Name, Total Orders, Total Revenue, Profit Margin), 'VISUALIZATIONS' (listing various visual types and their configurations), and 'FIELDS' (listing tables, columns, and measures). At the bottom are 'REPORT PAGES' (Executive View, Key Influencer, Decomposition Tree, Sandbox) and a 'Page 1 of 1' indicator.

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

The screenshot shows the Power BI interface with the 'Visualizations' pane open on the left. It displays various chart icons and a Python icon. Below the icons are sections for 'Values' and 'Add data fields here'. On the right, the 'Fields' pane is open, showing a search bar and a list of fields under 'Maven Cycles Sales'. One field, 'Quantity Sold', is highlighted with a yellow box and an arrow pointing to it from the text below.

Visualizations >

Fields

Search

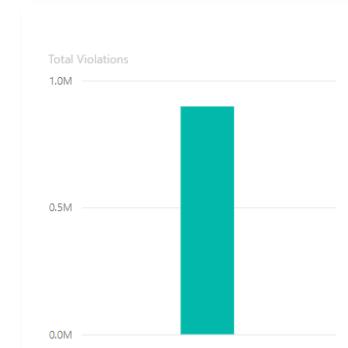
Maven Cycles Regions

Maven Cycles Sales

- % of Profit
- Age_Group_Key
- ALL Profit
- Australian Orders
- Customer_Gender_Key
- Invoice_ID
- Last Month Profit
- Last Month Profit Margin
- Last Month Revenue
- Mid Price Range Revenue
- Product_Key
- Profit
- Profit Margin
- Quantity Sold
- Quantity Sold (Stock Date)
- \sum Quantity_Sold
- Quantity Sold YTD

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

(or)



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**



Search

X axis On

Y axis On

Zoom slider Off

Data colors

Data labels Off

Shapes

Plot area

Title On

X axis

Type Continuous

Scale type Linear

Start Auto

End Auto

Color

Text size 8 pt

Font family

Search

General

Style

Column headers

Row headers

Values

Subtotals

Grand total

Field formatting

Conditional formatting

Number of Inspections

Background color Off

Font color Off

Data bars On

Icons Off

Advanced controls

Search

General

Legend Off

Data colors

Grade "A" Inspections

Grade "B" Inspections

Grade "C" Inspections

Detail labels On

Shapes

Title On

Background Off

Lock aspect Off

Border Off

Shadow Off

Detail labels

Label style Category

Color

Display units Auto

Text size 9 pt

Font family DIN



EDITING REPORT INTERACTIONS

Inserting Visuals

Formatting Visuals

Report Navigation

Tooltips

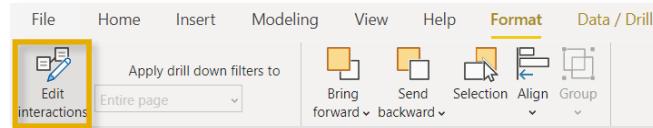
Custom Visuals

R & Python Visuals

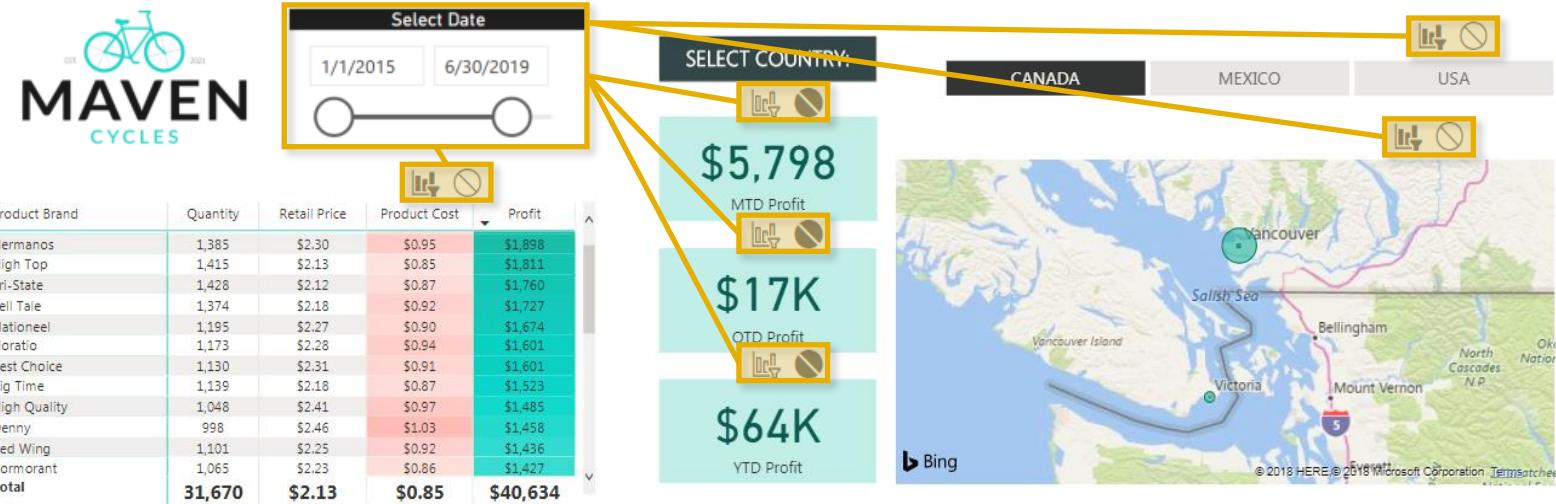
Accessibility

Dashboards

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.)

Inserting Visuals

Formatting Visuals

Report Navigation

Tooltips

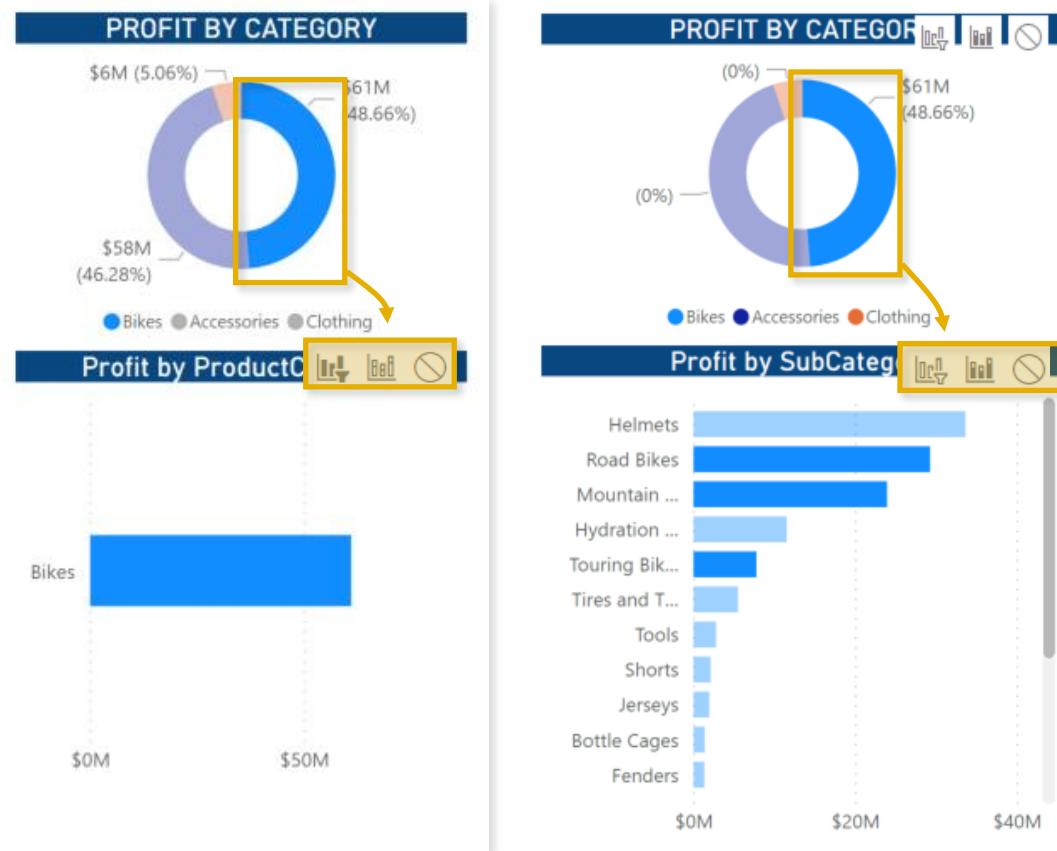
Custom Visuals

R & Python Visuals

Accessibility

Dashboards

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

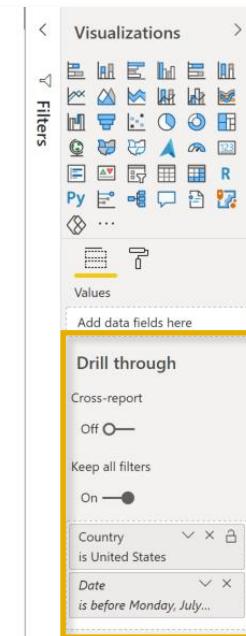
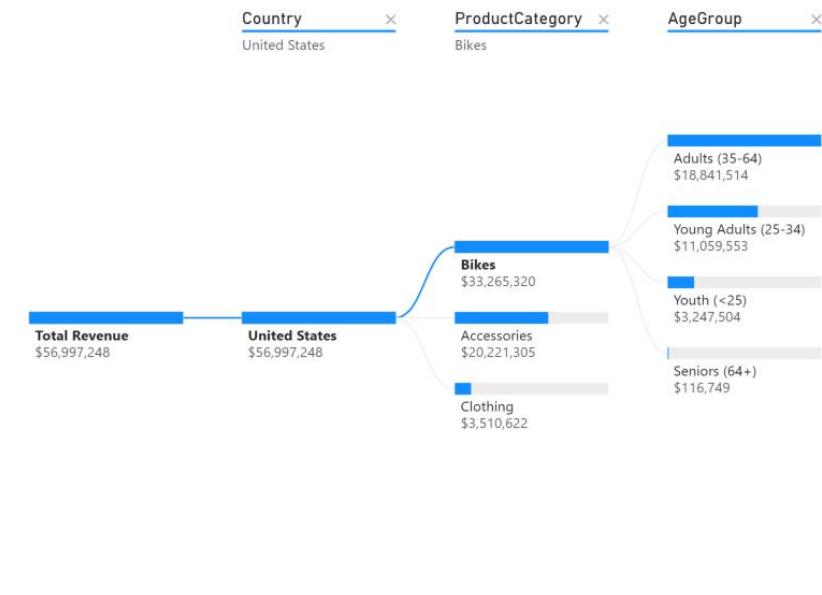
Custom Visuals

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

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!

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”



TOOLTIPS

Inserting Visuals

Formatting Visuals

Report Navigation

Tooltips

Custom Visuals

R & Python Visuals

Accessibility

Dashboards

Toolips 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

The screenshot shows the 'Visualizations' pane on the left with various icons for different types of charts and reports. A yellow box highlights the ellipsis (...) icon under 'Power BI Visuals'. Below the pane, the 'AppSource' section displays the 'histogram' visual by MAQ Software. The visual has a 4-star rating and is described as 'Visualises the distribution of data over a continuous interval or certain time period'. An 'Add' button is visible next to the visual's description.

Import custom visual

The visual was successfully imported into this report.



We've added a **histogram chart** from the marketplace

The screenshot shows the 'Visualizations' pane on the right with the 'Histogram' visual now listed under 'My organization'. A yellow box highlights the 'Histogram' icon in the list.



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



R & PYTHON VISUALS

Inserting Visuals

Formatting Visuals

Report Navigation

Tooltips

Custom Visuals

R & Python Visuals

Accessibility

Dashboards

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



ACCESSIBILITY FEATURES

Inserting Visuals

Formatting Visuals

Report Navigation

Tooltips

Custom Visuals

R & Python Visuals

Accessibility

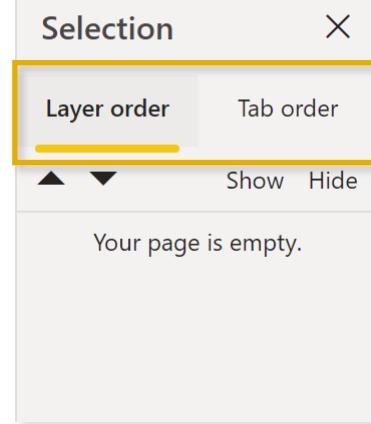
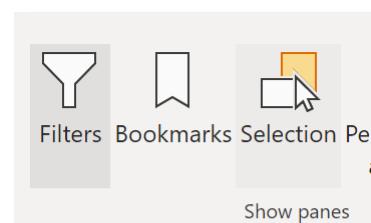
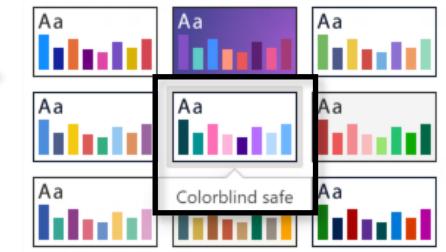
Dashboards

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

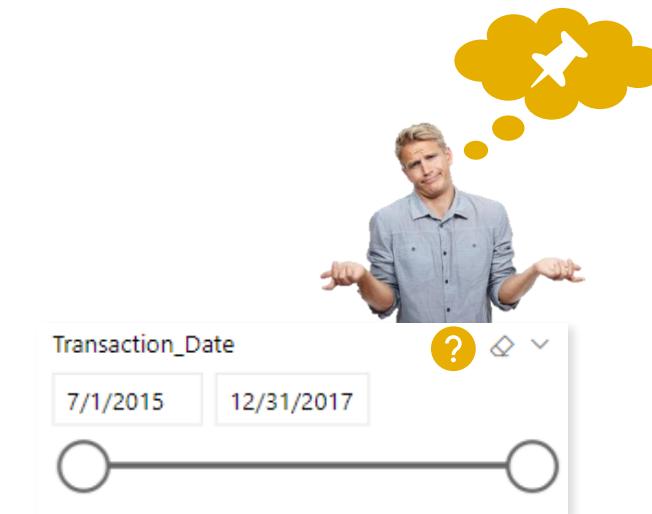
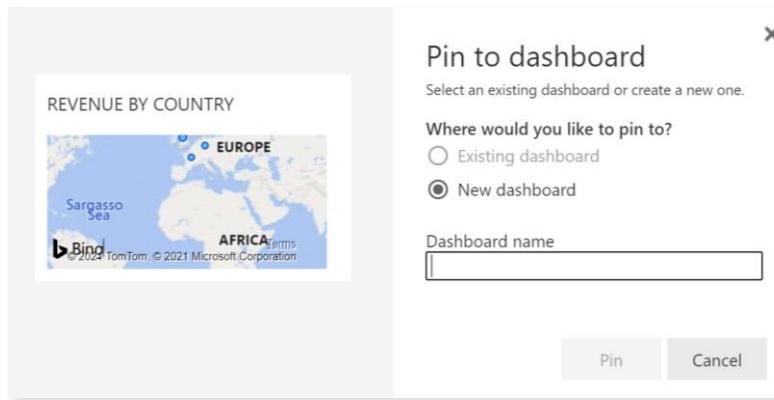
Custom Visuals

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

The screenshot illustrates the process of pinning a live report to a dashboard. On the left, a dashboard view shows three key performance indicators (KPIs): Revenue (\$8.14M), Profit (\$4.07M), and Profit Margin (50.03%). Below these are two maps: 'Revenue by Country' and 'Product Detail'. The 'Pin to a dashboard' button is highlighted in yellow. A callout bubble points to this button with the text: "PRO TIP: Pinning a live page will import slicers, preserve functionality, and reflect any updates to the report".

The central part of the interface shows the 'Pin to a dashboard' dialog box. It includes fields for selecting an existing dashboard or creating a new one, a preview of the pinned report, and a 'Pin live' button.

On the right, the final result is shown: a pinned dashboard titled 'Maven Cycles Report EXECUTIVE VIEW' containing the same KPIs, maps, and product detail table as the original report.



THE DASHBOARD INTERFACE

Inserting Visuals

Formatting Visuals

Report Navigation

Tooltips

Custom Visuals

R & Python Visuals

Accessibility

Dashboards

The screenshot shows a Microsoft Power BI dashboard. At the top is a toolbar with standard file operations like File, Share, Chat in Teams, Comment, Subscribe, Edit, and more. Below the toolbar is a search bar labeled "Ask a question about your data". The main area contains several data tiles:

- REVENUE:** A large green tile showing "\$8.14M" and "Goal: \$7.56M (+7.68%)".
- PRODUCT DETAIL:** A table listing products with columns for Product Name, Total Orders, Total Revenue, and Profit Margin.
- PROFIT:** A large green tile showing "\$4.07M" and "Goal: \$3.83M (+6.23%)".
- PROFIT MARGIN:** A large red tile showing "50.03%" and "Goal: 51% (-1.34%)".
- PROFIT BY PERIOD:** A line chart showing profit over time from 2015 to 2020, with a dashed trend line.

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

Share via email or to
Microsoft Teams

Subscribe to receive
periodic email updates



- Save a copy
- Print this page
- Performance inspector
- Settings

Add **comments** & start a
data-driven conversation

- Add a tile
- Dashboard theme
- Mobile view



WEB VS. MOBILE LAYOUT

Inserting Visuals

Formatting Visuals

Report Navigation

Tooltips

Custom Visuals

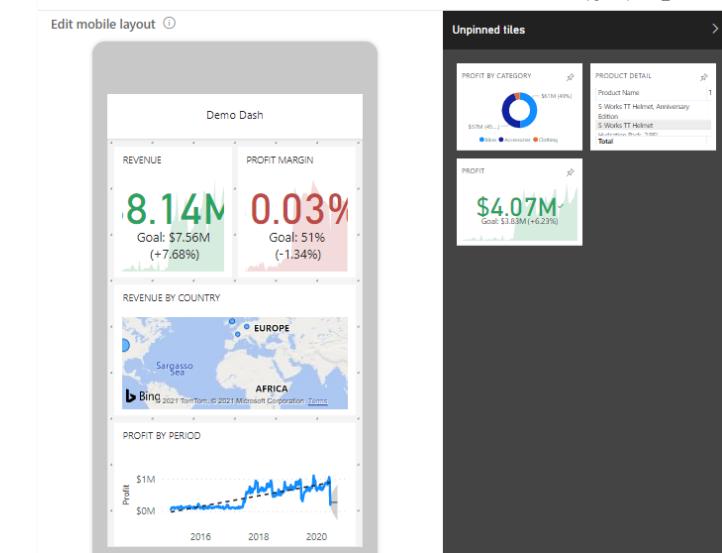
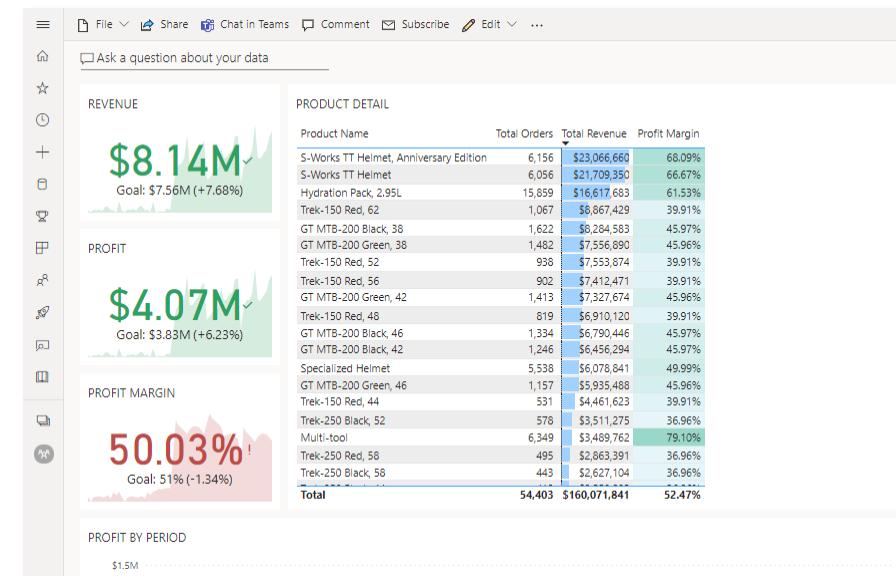
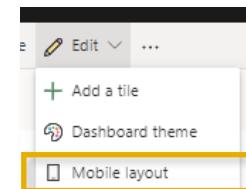
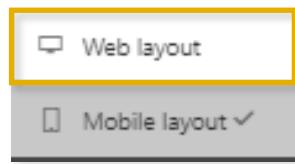
R & Python Visuals

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

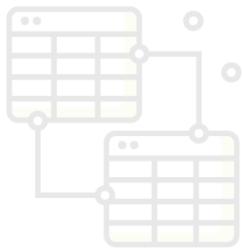


ANALYZING THE DATA



Prepare the Data

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



Model the Data

- Design a Data Model
- Develop a data model
- Create measures using DAX
- Optimize Performance



Visualize the Data

- Create reports
- Create dashboards
- Enrich reports for usability



Analyze the Data

- Enhance reports to expose insights
- Perform advanced analysis



Deploy & Maintain Deliverables

- Manage datasets
- Create and 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 single visualization to a report that helps the sales team understand sales drivers. What visualization should you use?*
- *You need to create a visualization that shows the relationship between quantity and price and highlights orders that have a similar price. What visual type should you use?*
- *You need to ensure that a matrix visual only shows the top 10 cities by sales. What should you do?*



CHART TYPES BASED ON ANALYSIS

Basic Chart Types

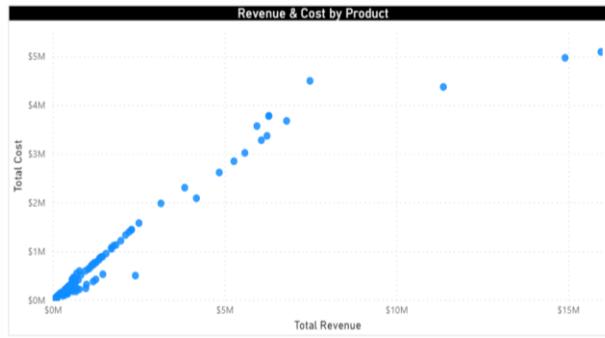
Analytics Options

Q&A

Filtering Options

AI Visuals

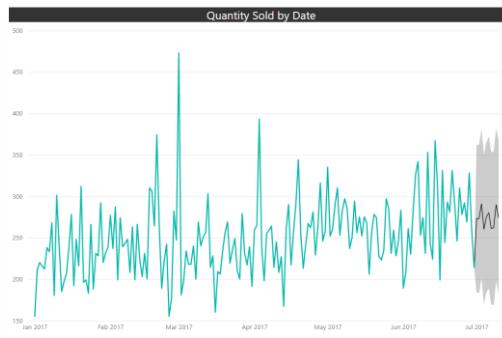
Scatter charts show the relationship between two numerical values



Common Uses:

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

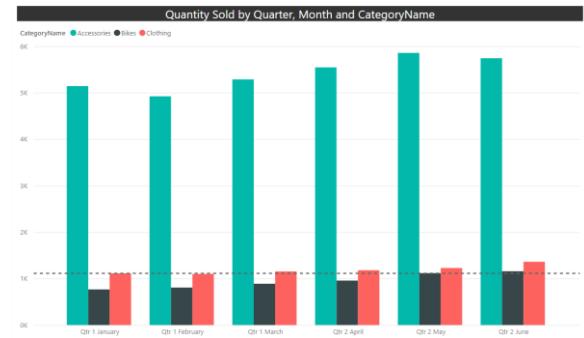
Line charts are used to track changes over periods of time



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-Axis Constant Line
- ✓ Y-Axis Constant Line
- ✓ Min line
- ✓ Max line
- ✓ Average line
 - ✓ Median line
 - ✓ Percentile line
- ✓ Symmetry shading

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
- ✓ Constant line
- ✓ X-Axis Constant Line
- ✓ Min line
- ✓ Max line
- ✓ Average line
- ✓ Median line
- ✓ Percentile line
- ✓ Forecast
- ✓ Find anomalies

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

- ✓ Constant line
- ✓ Min line
- ✓ Max line
- ✓ Average line
- ✓ Median line
- ✓ Percentile line 1

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



Q&A

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

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

Q&A Tooling Interface

The interface shows a sidebar with 'Q&A setup' and 'Getting started' options like 'Field synonyms', 'Review questions', 'Teach Q&A', 'Manage terms', and 'Suggest questions'. The main area has four cards:

- Field synonyms**: Add terms people might use as synonyms for the fields and tables in your data. **Field synonyms** button.
- Review questions**: Review questions people have asked and fix misunderstandings. **Review questions** button.
- Teach Q&A**: Teach Q&A to understand questions and terms people might use. **Teach Q&A** button.
- Suggest questions**: Help people explore your data by adding suggested questions. **Suggest questions** button.

Table & Column Synonyms

The properties panel shows 'Synonyms' for 'Maven Cycles Products':
maven cycles product, maven cycle product, product, cycles product, cycle product

A yellow arrow points from the 'Field synonyms' card in the Q&A interface to the 'Maven Cycles Products' entry in the synonyms list.

Properties
Synonyms
maven cycles product, maven cycle product, product, cycles product, cycle product

Maven Cycles Products
Price Range
Product Key
Product New
Collapse ^

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

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

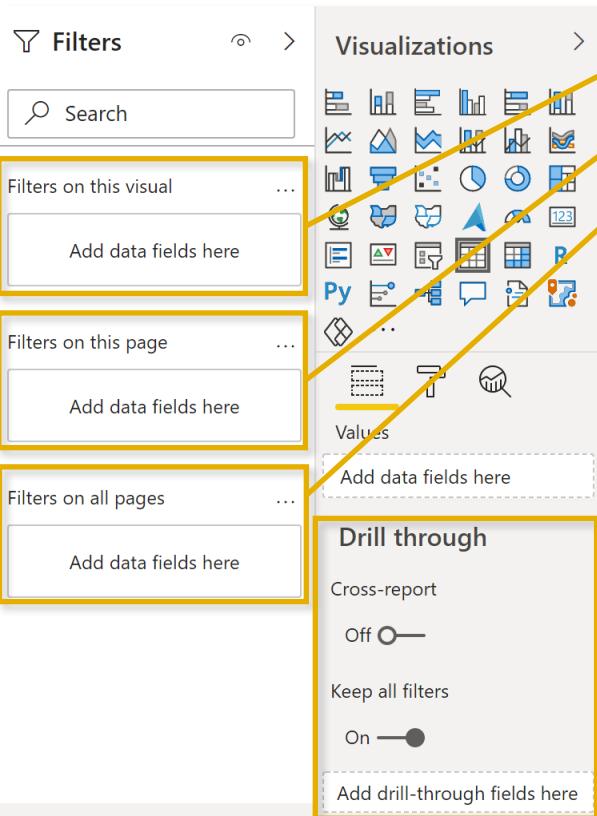
Basic Chart Types

Analytics Options

Q&A

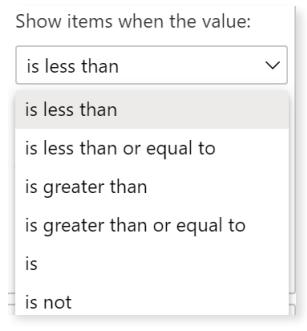
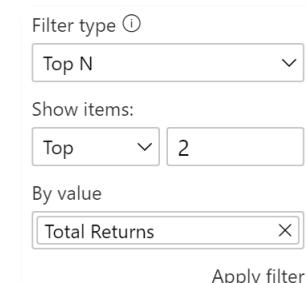
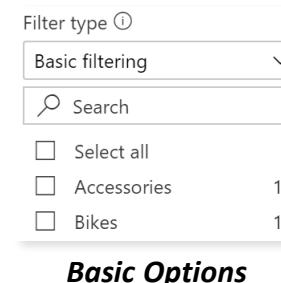
Filtering Options

AI Visuals



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



Advanced (Text)



SLICERS

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

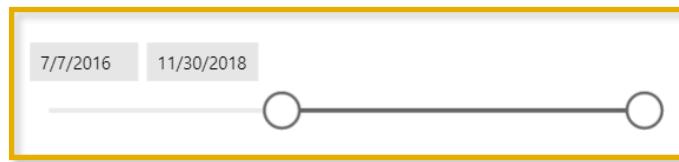
Basic Chart Types

Analytics Options

Q&A

Filtering Options

AI Visuals

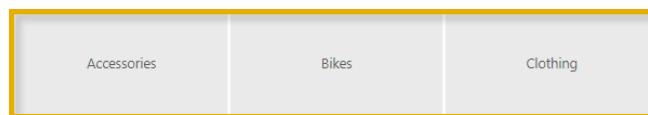


Date slicer formatted as a slider

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)

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

Basic Chart Types

Analytics Options

Q&A

Filtering Options

AI Visuals

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

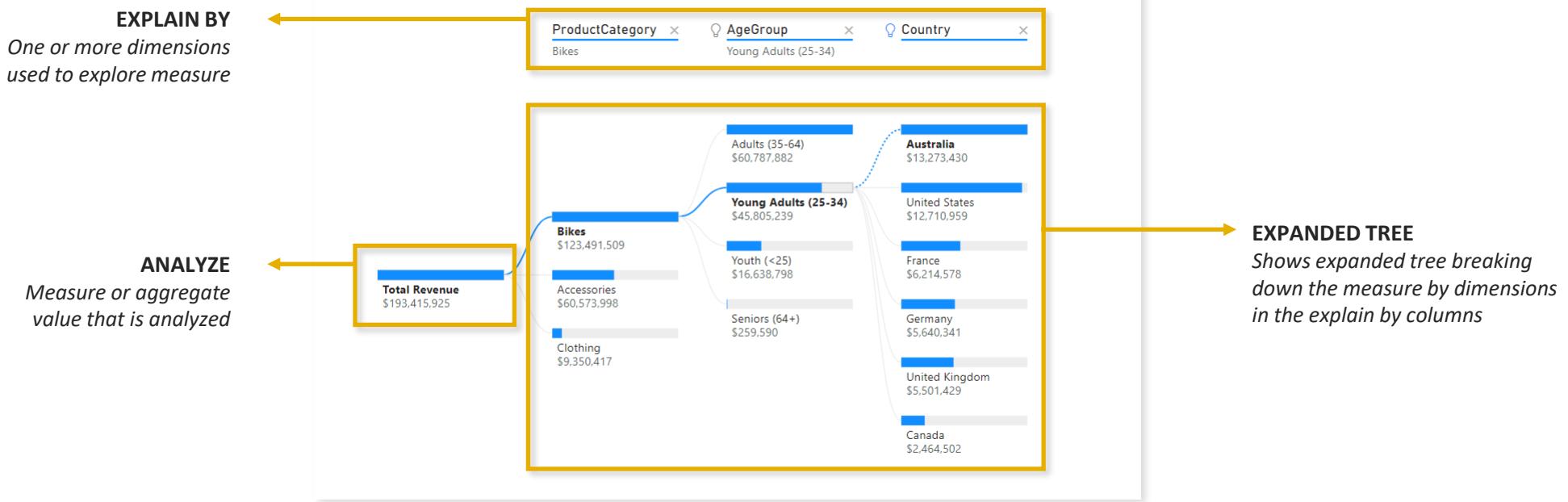
Q&A

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**

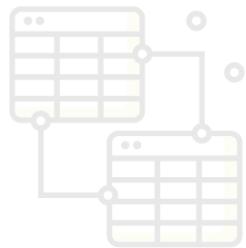


DEPLOY & MAINTAIN DELIVERABLES



Prepare the Data

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



Model the Data

- Design a Data Model
- Develop a data model
- Create measures using DAX
- Optimize Performance



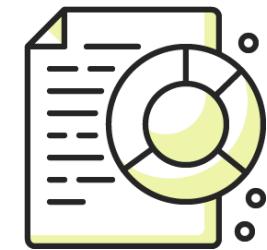
Visualize the Data

- Create reports
- Create dashboards
- Enrich reports for usability



Analyze the Data

- Enhance reports to expose insights
- Perform advanced analysis



Deploy & Maintain Deliverables

- Manage datasets
- Create and manage workspaces

DEPLOY & MAINTAIN DELIVERABLES



In this section we'll cover tools and techniques to **deploy & maintain deliverables**, 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:

- You publish a dataset to Power BI Service and need to ensure the data is updated every morning. Which two actions should you perform?
- You need to create a security role that has a table filter on Region_Manager. What DAX should your solution include?
- You need to provide a user with the ability to add members to a workspace. Which role should you assign the user?



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

Datasets Workbooks Dataflows App

Settings for Maven Cycles Reporting

This dataset has been configured by Aaron@mavenanalytics.onmicrosoft.com.

[Refresh history](#)

► Gateway connection

► Data source credentials

► Parameters

► Scheduled refresh

Keep your data up to date

On

Refresh frequency: Daily

Time zone: (UTC-05:00) Eastern Time (US and Canada)

Time: 7 30 AM X
7 30 PM X

Add another time

Send refresh failure notifications to the dataset owner

Email these users when the refresh fails

Aaron Parry Enter email addresses

Apply Discard

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

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

+ New ▾

All	Content	Datasets + dataflows	Type	Owner	Refreshed	Next refresh
	Maven Cycles Reporting	Dataset	Maven Cycles Report...	6/18/21, 9:31:34 AM	6/19/21, 9:30:00 AM	

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



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

The screenshot shows the Power BI Desktop ribbon with the 'Modeling' tab selected. The 'External Tools' section of the ribbon has a 'Manage roles' button highlighted with a yellow box and an arrow pointing to it from the callout box. To the right, a 'Manage roles' dialog box is open. It shows a list of roles under 'Region Managers' and a list of tables under 'Tables'. On the right, a 'Table filter DAX expression' field contains the DAX expression `[Country] = "France"`. At the bottom of the dialog, there is a note: 'Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"'.



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

The screenshot illustrates the process of testing row-level security in Power BI. It shows the Power BI desktop ribbon with the 'Modeling' tab selected. In the 'Security' section of the ribbon, the 'View as' button is highlighted with a yellow box and a yellow arrow pointing from the 'Los Angeles' selection in the 'View as roles' dialog below. The 'View as' dialog shows three options: 'None', 'Other user', and 'Los Angeles', with 'Los Angeles' checked. A yellow box and arrow also point to the status bar at the bottom of the dialog, which displays the message 'Now viewing as: Region Managers'. The main Power BI interface shows a report for 'MAVEN CYCLES' with various visualizations and data cards.



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

The screenshot shows the 'Manage roles' dialog in Power BI Desktop. A new role named 'Territory Manager' is being created. In the 'Table filter DAX expression' field, the formula `[Territory Manager] = USERNAME()` is entered. Below the dialog, a preview table titled 'Territory Manager' displays email addresses for various users: Jenna Stubbs, Lauren Burns, Aden Holt, Susie Salt, Jake Kay, Kathy Meza, Adam Juan, David Hahn, Benny Blanco, and Dirk Gently. The table has two columns: 'Territory Manager' and 'Email Address'. A callout arrow points from the 'Table filter DAX expression' field in the dialog to the 'Territory Manager' column header in the preview table.

Dynamic role will filter by the Username in this table

USERPRINCIPALNAME

The screenshot shows the 'Manage roles' dialog in Power BI Desktop. A new role named 'Inspector Lookup' is being created. In the 'Table filter DAX expression' field, the formula `[email_address] = USERPRINCIPALNAME()` is entered. Below the dialog, a preview table titled 'Inspector Lookup' displays email addresses for users: Jenna Stubbs, Lauren Burns, Aden Holt, Susie Salt, Jake Kay, Kathy Meza, Adam Juan, David Hahn, Benny Blanco, and Dirk Gently. The table has two columns: 'Inspector Lookup' and 'Email Address'. A callout arrow points from the 'Table filter DAX expression' field in the dialog to the 'Inspector Lookup' column header in the preview table. A note at the bottom right of the dialog states: "Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"."

Dynamic role will filter by the Email Address in this table



USERNAME

Scheduled Refresh

Row Level Security

Sharing Options

Data Lineage

Incremental Refresh

Storage Formats

Endorse Content

Sensitivity Labels

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*

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



Table filter DAX expression

[Territory Manager] = USERNAME()

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

Save

Cancel



USERPRINCIPALNAME

Scheduled Refresh

Row Level Security

Sharing Options

Data Lineage

Incremental Refresh

Storage Formats

Endorse Content

Sensitivity Labels

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)



Table filter DAX expression

[email_address] = USERPRINCIPALNAME()

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

Save

Cancel



APPLYING RLS IN SERVICE

Scheduled Refresh

Row Level Security

Sharing Options

Data Lineage

Incremental Refresh

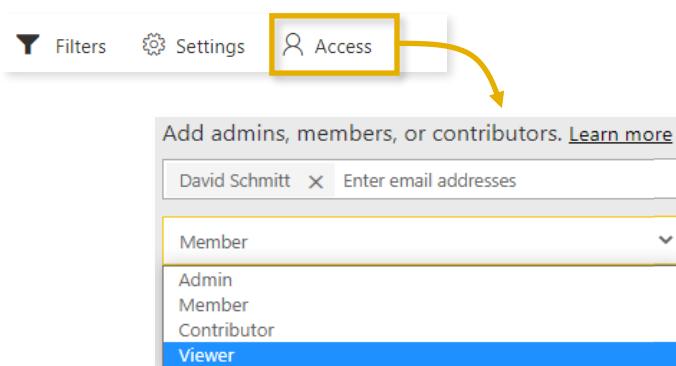
Storage Formats

Endorse Content

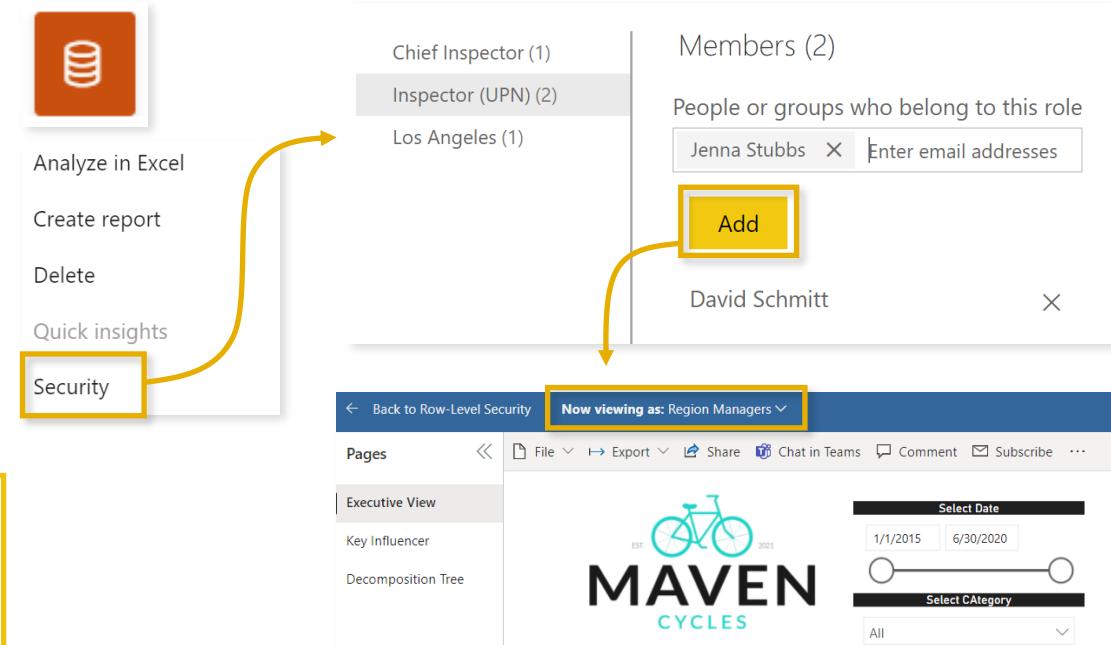
Sensitivity Labels

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

The screenshot shows the Microsoft Azure portal interface for creating a new security group. The top navigation bar includes the Microsoft Azure logo and a search bar. Below the navigation, the breadcrumb trail shows 'Home > Maven Analytics > Groups > New Group'. The main form fields are: 'Group type * ⓘ' (Security), 'Group name * ⓘ' (Client Reporting Security Group), 'Group description ⓘ' (Enter a description for the group), and 'Membership type ⓘ' (Assigned). Under the 'Owners' section, it says 'No owners selected'. Under the 'Members' section, it says 'No members selected'. At the bottom of the form is a blue 'Create' button.

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



Subscribe to emails

+ Add new subscription

Run Now On

Subscribe

Aaron Parry Enter email addresses

Subject

Subject

Include an optional message...

Report page

Inspections & Violations

Manage all subscriptions

Save and close Cancel



SHARING OPTIONS

Scheduled Refresh

Row Level Security

Sharing Options

Data Lineage

Incremental Refresh

Storage Formats

Endorse Content

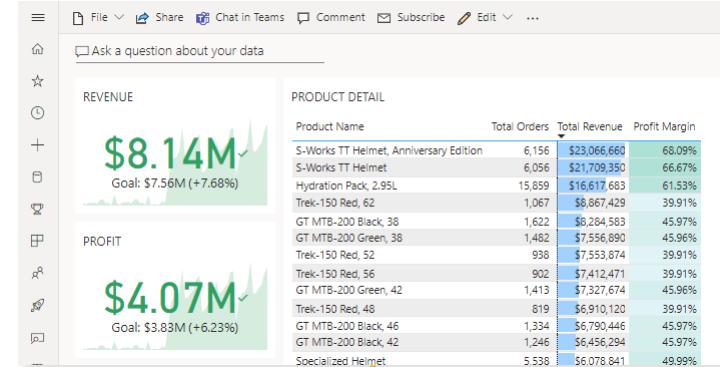
Sensitivity Labels



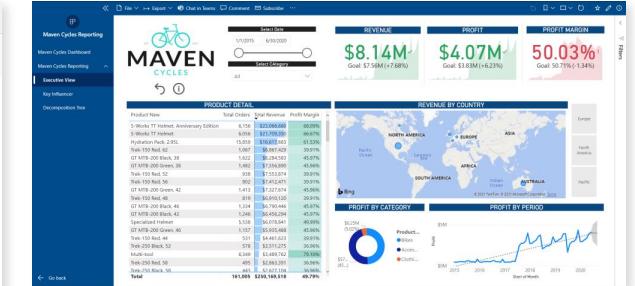
Reports



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

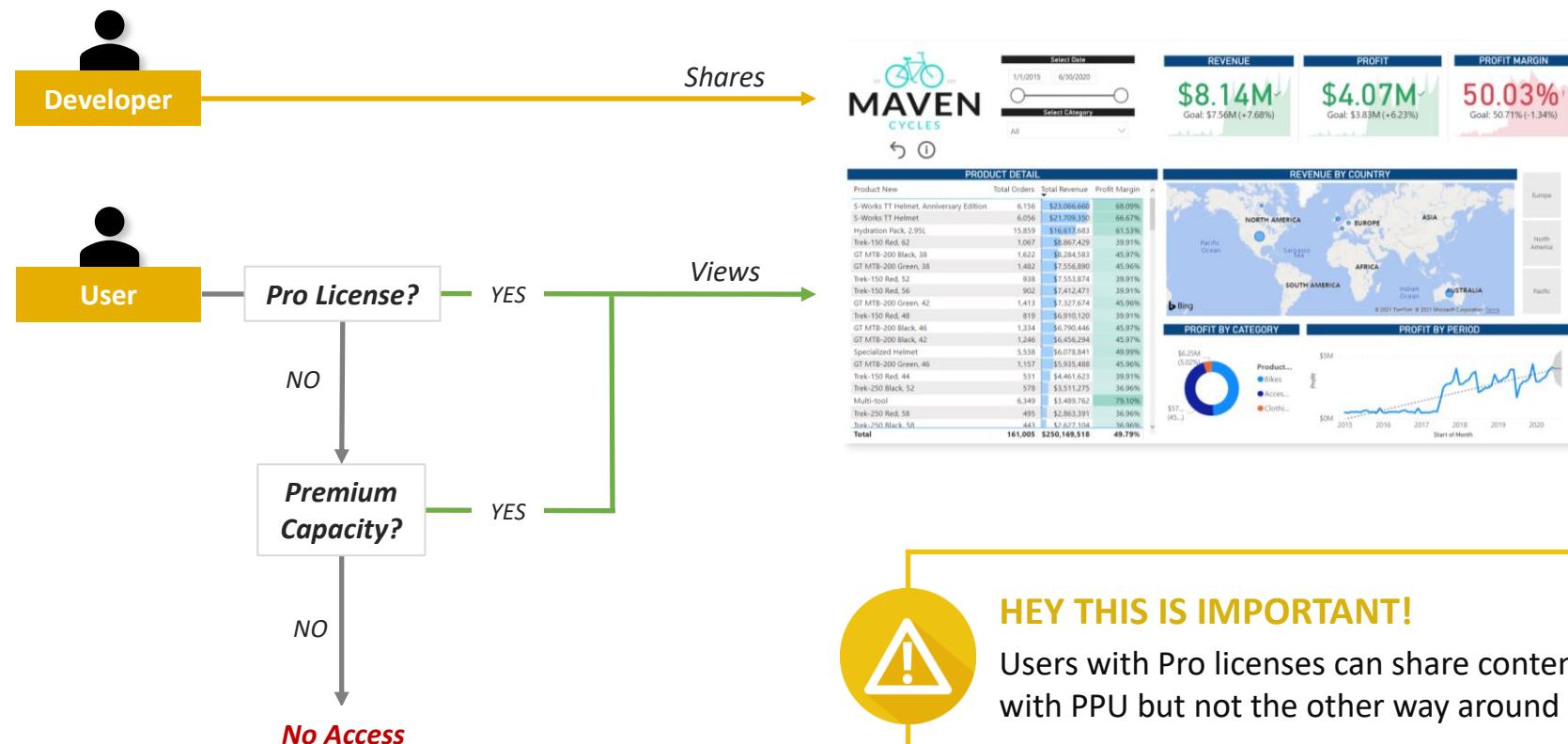
Incremental Refresh

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



Add admins, members, or contributors. [Learn more](#)

Chris Dutton Enter email addresses

Member

Admin

Member

Contributor

Viewer



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

2 Select **Create app** to publish the app

Create app

1 From within a workspace, select any **reports** and **dashboards** you want to include

The screenshot shows a workspace titled "Maven Cycles Reporting". The top navigation bar includes "View", "Filters", "Settings", "Access", and a search bar. Below the title, there are buttons for "+ New" and "Content". The "Content" tab is selected, showing a list of three items: "Maven Cycles Dashboard", "Maven Cycles Pin Live Page", and "Maven Cycles Reporting". Each item has columns for Name, Type, Owner, Refreshed, Next refresh, Endorsement, Sensitivity, and an "Include in app" toggle switch. The "Include in app" column contains three rows: "Yes" (selected), "No", and "Yes". A yellow box highlights the "Create app" button in the top right corner, and another yellow box highlights the "Include in app" column.

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Include in app
Maven Cycles Dashboard	Dashboard	Maven Cycles Reporti...	—	—	—	—	<input checked="" type="checkbox"/> Yes
Maven Cycles Pin Live Page	Dashboard	Maven Cycles Reporti...	—	—	—	—	<input type="checkbox"/> No
Maven Cycles Reporting	Report	Maven Cycles Reporti...	6/18/21, 10:13:25 AM	—	—	—	<input checked="" type="checkbox"/> Yes



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

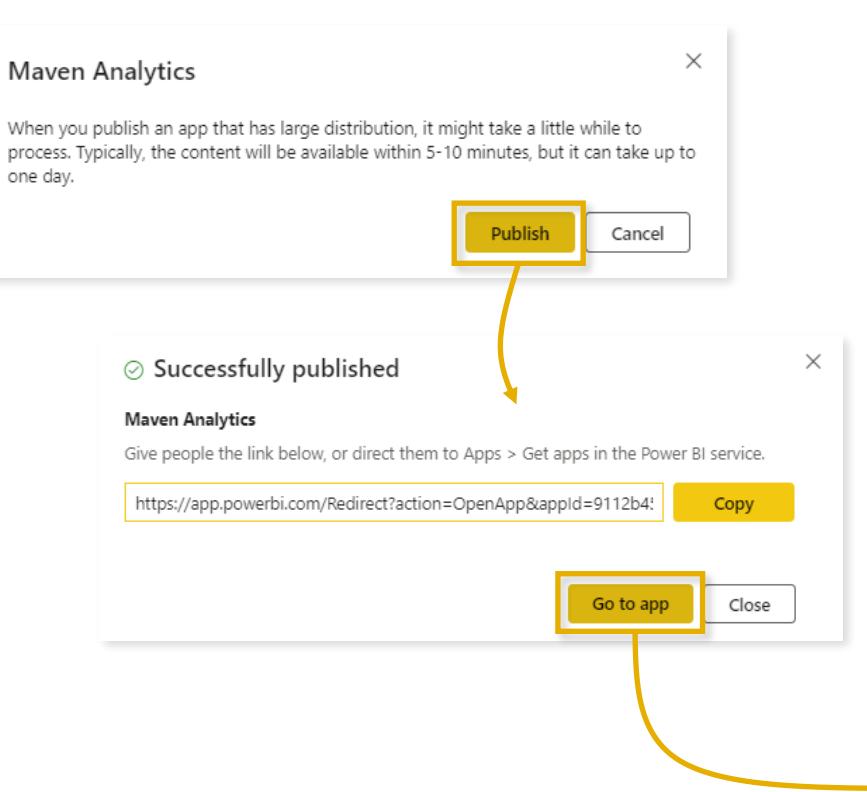
Incremental Refresh

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



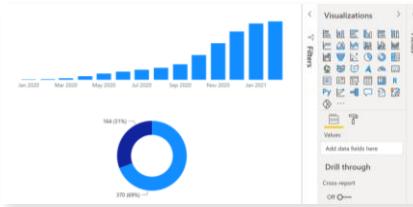
Datasets

Data sources you've connected to

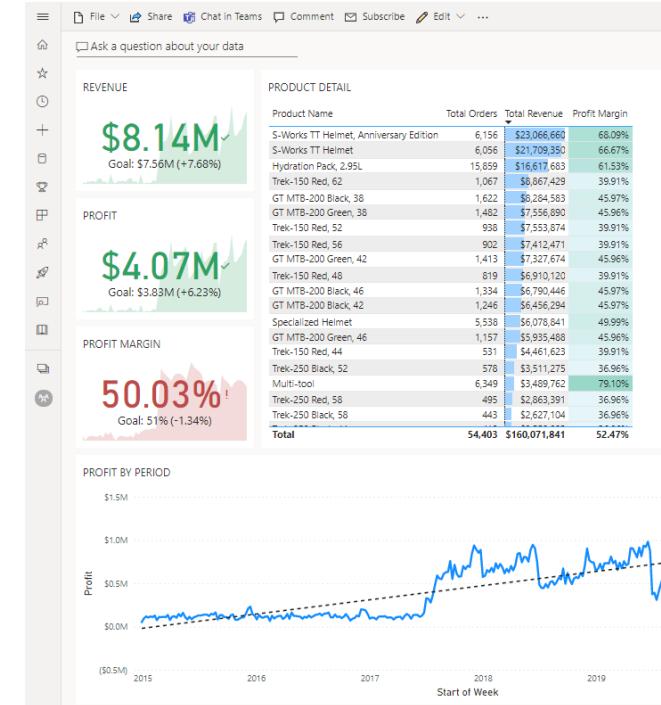


Reports

Visuals built from a single dataset



Google Analytics





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

Maven Cycles Reporting

+ New ▾

All Content Datasets + dataflows

Name Type Owner Refreshed Next refresh Endorsement Sensitivity Include in app

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Include in app
Maven Cycles Reporting	Dataset	Maven Cycles Report...	6/18/21, 10:13:25 AM	N/A	—	—	—

View ▾ Filters Settings Access ... Search

Lineage

mvnccyprod.csv
Gateway: Maven Cycles Gateway

Text/CSV
Maven Cycles Product Subcategories.csv
Gateway: Maven Cycles Gateway

Text/CSV
Maven Cycles Regions.csv
Gateway: Maven Cycles Gateway

Text/CSV
Maven Cycles Sales.csv
Gateway: Maven Cycles Gateway

sample4.json
sampled.json
Gateway: Maven Cycles Gateway

Maven Cycles Reporting
Refreshed: 6/18/21, 10:13:25 AM

Maven Cycles Reporting

Maven Cycles Dashboard

Maven Cycles Pin Live Page

Lineage View

```
graph TD; A[mvnccyprod.csv  
Gateway: Maven Cycles Gateway] --> B[Text/CSV  
Maven Cycles Product Subcategories.csv  
Gateway: Maven Cycles Gateway]; C[Text/CSV  
Maven Cycles Regions.csv  
Gateway: Maven Cycles Gateway]; D[Text/CSV  
Maven Cycles Sales.csv  
Gateway: Maven Cycles Gateway]; E[sample4.json  
sampled.json  
Gateway: Maven Cycles Gateway]; B --> F[Maven Cycles Reporting  
Refreshed: 6/18/21, 10:13:25 AM]; C --> F; D --> F; E --> F; F --> G[Maven Cycles Reporting]; F --> H[Maven Cycles Dashboard]; G --> I[Maven Cycles Pin Live Page]; H --> I;
```

DATA LINEAGE TOOLS

Scheduled Refresh

Row Level Security

Sharing Options

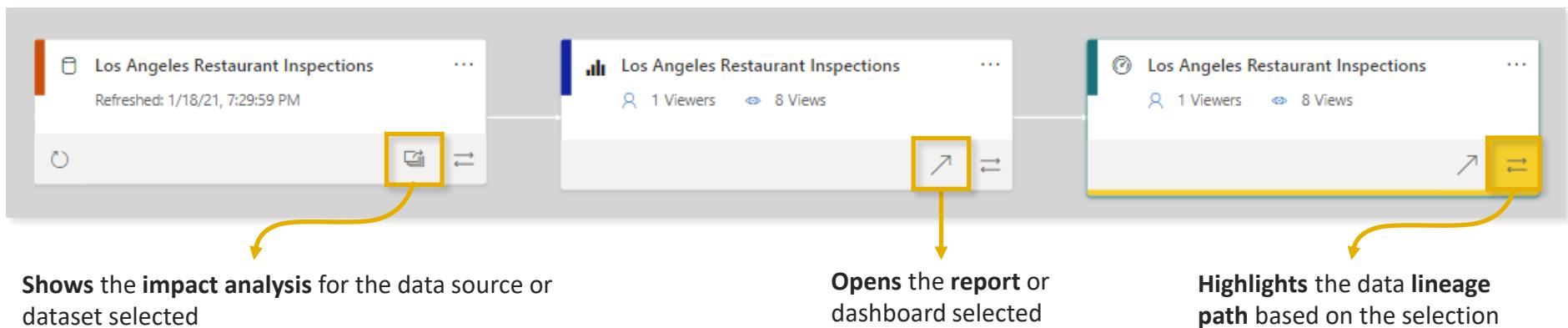
Data Lineage

Incremental Refresh

Storage Formats

Endorse Content

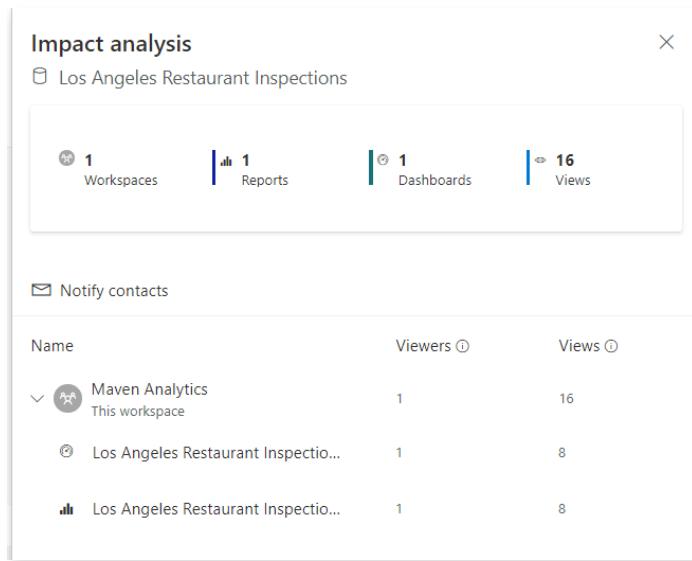
Sensitivity Labels



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

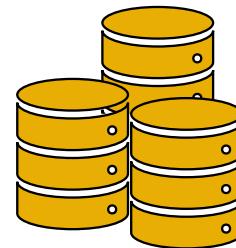
Storage Formats

Endorse Content

Sensitivity Labels

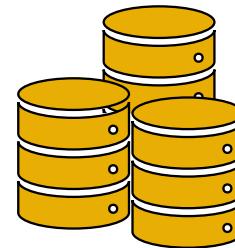
Incremental Refresh is the process of reloading only the part of a dataset that may change over time and adding it to the rest of the data set that no longer changes

- **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

Incremental refresh

⚠️ Unable to confirm if the M query can be folded. It is not recommended to use incremental refresh with non-foldable queries. [Learn more](#)

You can improve the speed of refresh for large tables by using incremental refresh. This setting will apply once you've published a report to the Power BI service.

ⓘ Once you've deployed this table to the Power BI service, you won't be able to download it back to Power BI Desktop. [Learn more](#)

Table Incremental refresh
Restaurant Inspection Data Off

Store rows in the last:

Enter value... Select value... ▾

Refresh rows in the last:

Enter value... Select value... ▾

Detect data changes [Learn more](#)

Only refresh complete periods [Learn more](#)

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

Manage Parameters

The screenshot shows the 'Manage Parameters' dialog in Power BI Desktop. A 'New' button is at the top right. Below it is a list of parameters: 'RangeStart' (selected) and 'RangeEnd'. The 'RangeStart' row contains fields for 'Name' (RangeStart), 'Description' (empty), 'Required' (checked), 'Type' (Date/Time), 'Suggested Values' (Any value), and 'Current Value' (1/1/2016 12:00:00 AM). A yellow box highlights the 'RangeStart' parameter.

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

Queries [3]	
	Restaurant Inspection Data
	RangeStart (1/1/2016 12:00:00 AM)
	RangeEnd (12/31/2016 12:00:00 AM)

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

The screenshot shows the 'activity_date' column's filter options. On the left, there are standard filtering actions like Sort Ascending, Sort Descending, Clear Sort, Clear Filter, Remove Empty, and Date/Time Filters. Below these is a search bar and a list of specific dates from January 4 to 15, 2016. At the bottom, a warning message says 'List may be incomplete.' and there are 'Load more' and 'OK/Cancel' buttons. On the right, a dropdown menu lists various filtering conditions, with 'Custom Filter...' being highlighted.

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

Filter Rows

Apply one or more filter conditions to the rows in this table.

Basic Advanced

Keep rows where 'activity_date'

is after or equal to RangeStart
 And Or
 is before RangeEnd
RangeStart
RangeEnd

OK Cancel

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)

You can improve the speed of refresh for large tables by using incremental refresh. This setting will apply once you've published a report to the Power BI service.

- (i)* Once you've deployed this table to the Power BI service, you won't be able to download it back to Power BI Desktop. [Learn more](#)

Table	Incremental refresh
Restaurant Inspection Data	<input checked="" type="radio"/> On
Store rows where column "activity_date" is in the last:	2 Years
Refresh rows where column "activity_date" is in the last:	1 Months
<input type="checkbox"/> Detect data changes Learn more	
<input type="checkbox"/> Only refresh complete month Learn more	

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)

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")*



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



4 Publish to Service for the policy to take effect



LARGE DATASET STORAGE FORMAT

Scheduled Refresh

Row Level Security

Sharing Options

Data Lineage

Incremental Refresh

Storage Formats

Endorse Content

Sensitivity Labels

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:

- After creating a model in Power BI Desktop, configure **incremental refresh** if you expect your dataset will become larger and progressively consume more memory
- Publish the model as a **dataset** to Power BI Service
- In Power BI Service, go to Dataset > Settings > **Large dataset storage format**, click the slider to turn “On”, and then “Apply”
- 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**

Endorsement (preview)

Help coworkers find your quality content by endorsing this dataflow. [Learn more](#)

None

This dataflow will appear in search results but isn't endorsed.

Promoted

When you're ready to distribute the dataflow to your coworkers, promote it to let them know.

Certified

Certify your dataflow to show coworkers that it's been reviewed and meets your org's certification criteria. [How do I get my dataflow certified?](#)

Apply

Discard

Certification means that the content meets the organization's 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

The screenshot shows a list of items in the 'Sales and Marketing' workspace. Each item has columns for Name, Type, Owner, Refreshed, Next refresh, Endorsement, Sensitivity, and Include in app. A yellow box highlights the 'Sensitivity' column for the 'Contoso Q2 Division Sales' report, which is labeled 'Confidential - Finance'. A yellow arrow points from this box to a callout box containing the text 'SENSITIVITY LABEL Hover over label to see a description'.

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Include in app
Contoso Q2 Division Sales	Report	Sales and Marketing	3/10/20, 5:40:44 AM	—	—	Confidential - Finance ⓘ	Yes
Contoso Q2 Division Sales	Dataset	Sales and Marketing	3/10/20, 5:40:44 AM	N/A	—	Highly Confidential Pr... ⓘ	Yes
Contoso Q2 Division Sales.pbix	Dashboard	—	—	—	—	Highly Confidential Pr... ⓘ	Yes
Sales	Report	Sales and Marketing	3/10/20, 5:40:17 AM	—	—	Highly Confidential Pr... ⓘ	Yes
Sales	Dataset	Sales and Marketing	3/10/20, 5:40:17 AM	N/A	—	Confidential - Finance ⓘ	—
Sales.pbix	Dashboard	—	—	—	—	General ⓘ	Yes

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



LEARN ON!

