NACM'S 129TH CREDIT CONGRESS & EXPO Cleveland MAY 18-21, 2025

Power Query Proficiency

Presented by: Rebekyah Brewer

Date: May 20, 2025

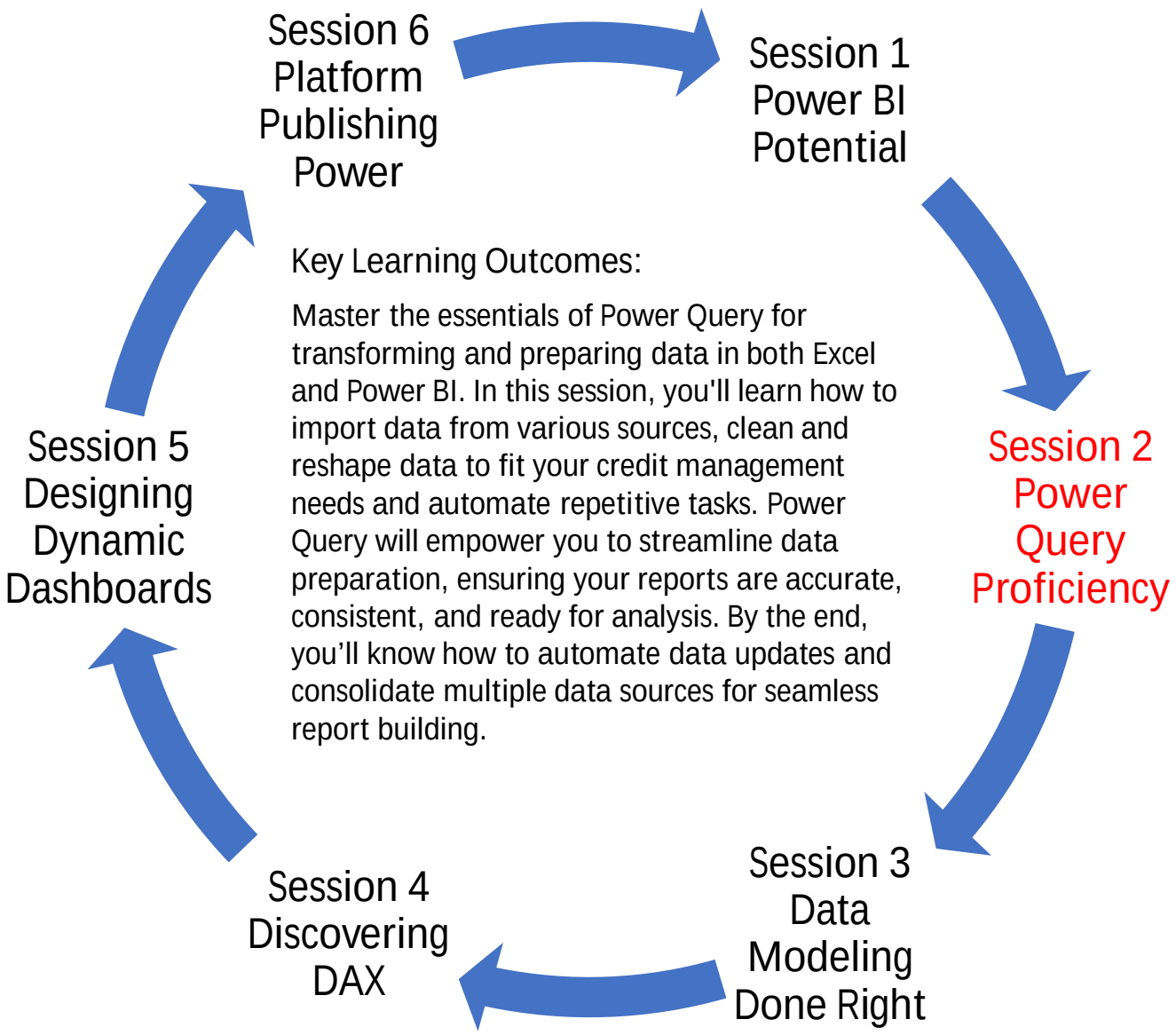
Session: #37043

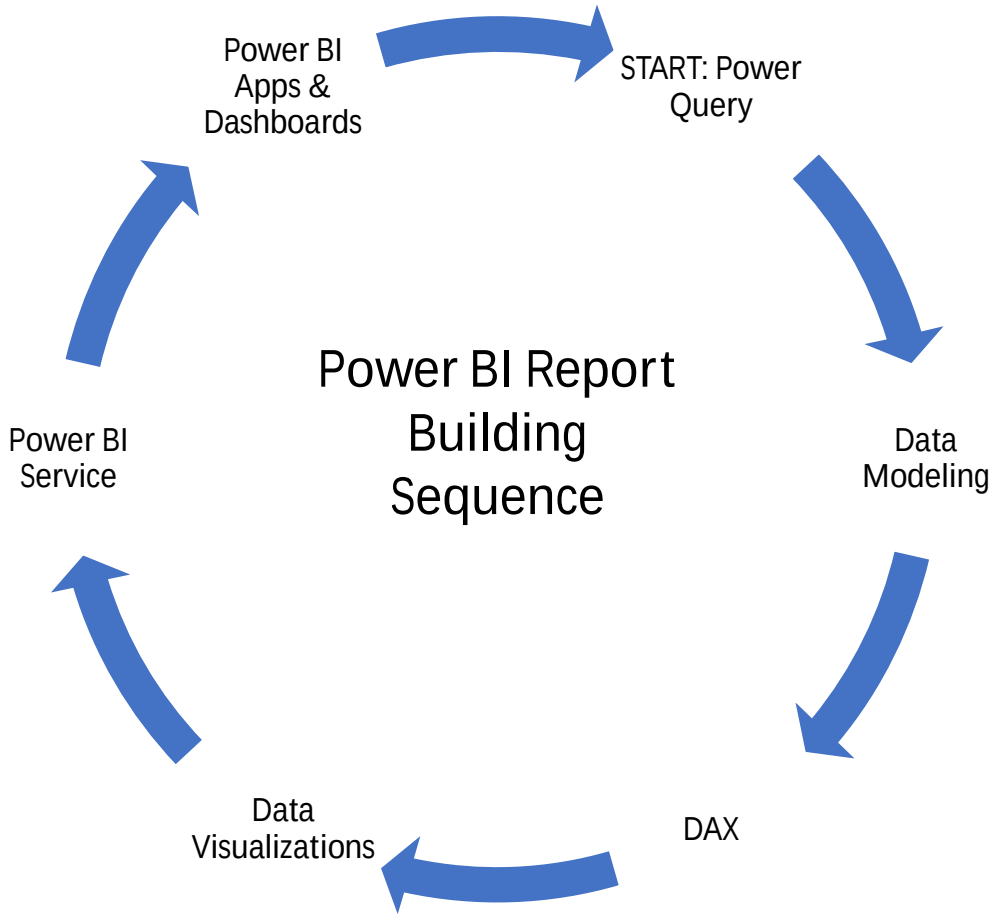
Session 2 Files Download:
Power Query Proficiency

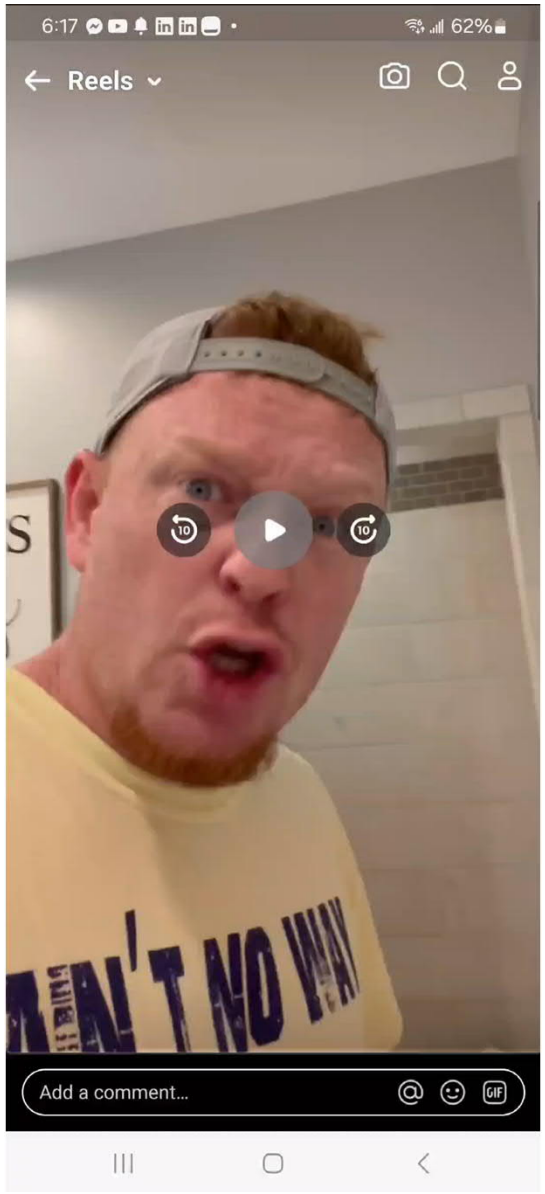
Password: NACM2025

https://tinyurl.com/NACMPowerQuery

Link Expiration: 6/18/2025


Session 6
Platform
Publishing
Power
Session 1
Power BI
Potential
Session 2
Power
Query
Proficiency
Session 3
Data
Modeling
Done Right
Session 4
Discovering
DAX
Session 5
Designing
Dynamic
Dashboards
Key Learning Outcomes:
Master the essentials of Power Query for transforming and preparing data in both Excel and Power BI. In this session, you'll learn how to import data from various sources, clean and reshape data to fit your credit management needs and automate repetitive tasks. Power Query will empower you to streamline data preparation, ensuring your reports are accurate, consistent, and ready for analysis. By the end, you'll know how to automate data updates and consolidate multiple data sources for seamless report building.


Power BI Report Building Sequence
START: Power Query
Data Modeling
DAX
Data Visualizations
Power BI Service
Power BI Apps & Dashboards


6:17
62%
Reels
Add a comment...

Power Query Proficiency Overview

I. Understanding Power Query Basics

II. Data Preparation and Loading

III Essential Data Transformation Techniques

IV Advanced Power Query Techniques

V. Practical Application & Tips

VI. Q&A

I. Understanding Power Query Basics

- Prerequisites
- Who is Power Query For?
- What is Power Query?
- Benefits of Learning Power Query within Excel
- Why use Power Query?
- Navigating the Interface

Prerequisites – Individual

Intermediate Excel Experience

Power Query is integrated into Excel and Power BI, so a solid understanding of Excel functionalities will be helpful:

- Basic formulas (e.g., TEXT functions, IF statements).
- Understanding of Pivot Tables and how data is structured in tables.
- Experience working with tables, named ranges and structured references.
- Experience working with v-lookups, matching, index…

Understanding Power BI or Excel Data Models

If your goal is to use Power Query in Power BI, it's important to understand:

- Fact and Dimension Tables (for structuring clean models). (Next Session is Data Modeling Done Right)
- Relationships between tables (One-to-Many, One-to-One, Many-to-Many).

Prerequisites – Technical

Software Requirements

- Power BI Desktop (Free) – Power Query is built into Power BI for data transformation.
- Excel (2016 and later, or Microsoft 365) – Power Query is available in the "Get & Transform" section.
- Windows OS (Windows 10 or later recommended) – Power Query in Power BI is optimized for Windows.

Optional:

- Power BI Service (Pro or Premium Per User License) – If publishing reports online, you'll need a Power BI account

Prerequisites – Technical

Understanding Data Connectivity & Storage Options

Power Query allows you to import data from various sources, and where your data is stored affects how you access it.

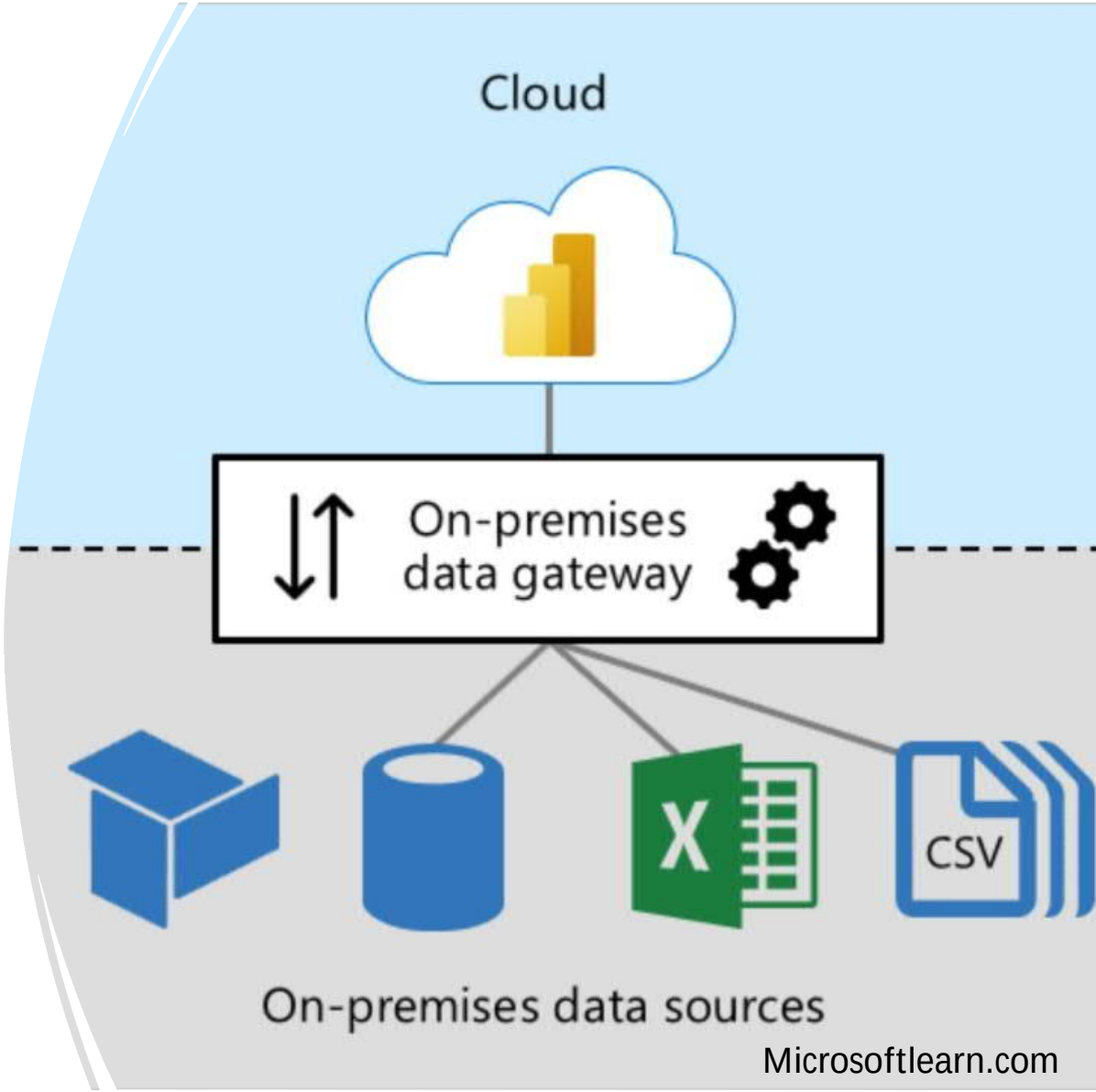
1. Local Storage vs. Cloud Storage

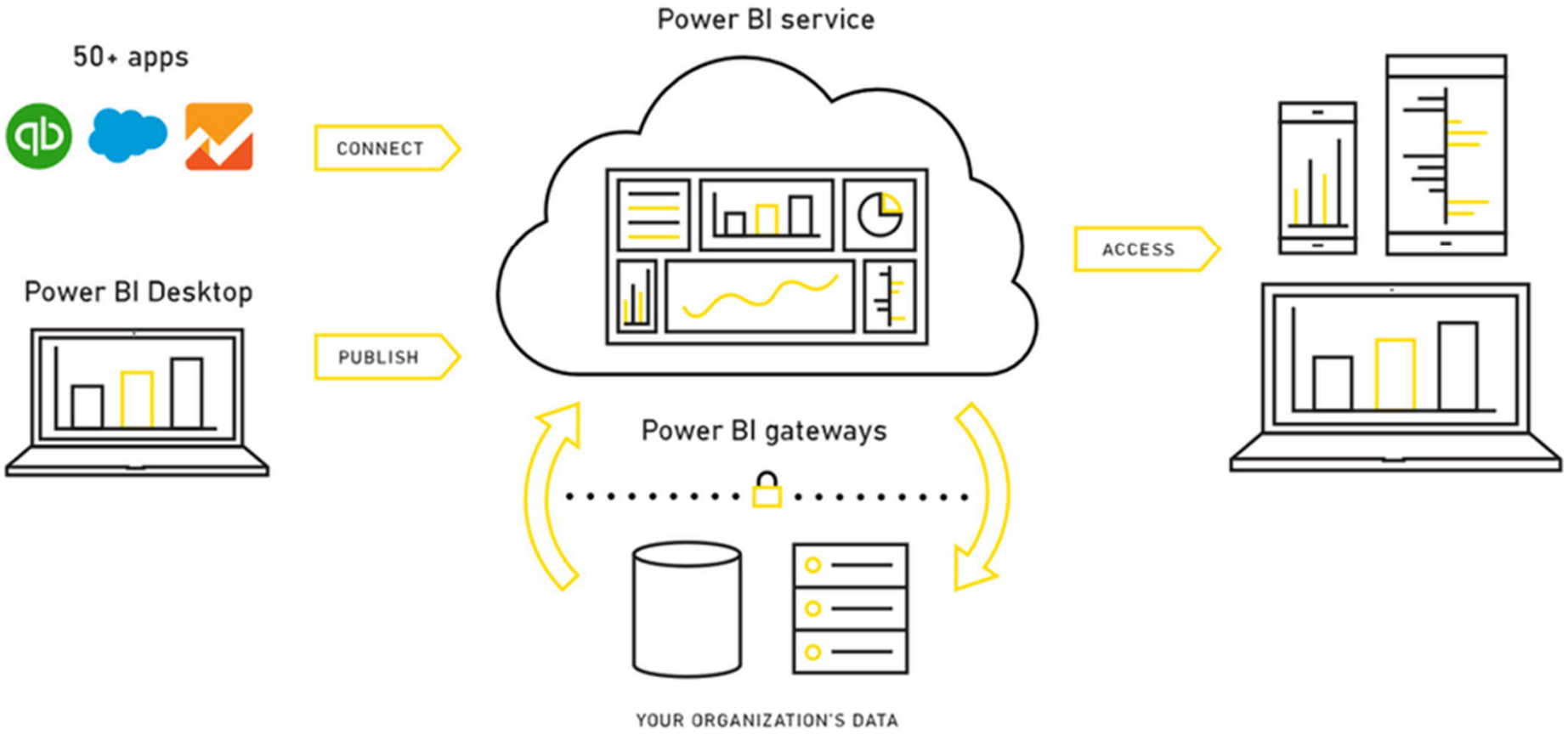
Local Files/Network Drives (Excel, CSV, etc.) – Power Query connects easily, but refresh automation requires a gateway. (On premises or private)
SharePoint & OneDrive – Recommended for cloud-based access; OneDrive links are easier to manage but might have refresh delays in Power BI.
Databases (SQL, MySQL, etc.) – Requires database credentials and proper configuration.
Webpages – Already in the cloud, easy to connect to.

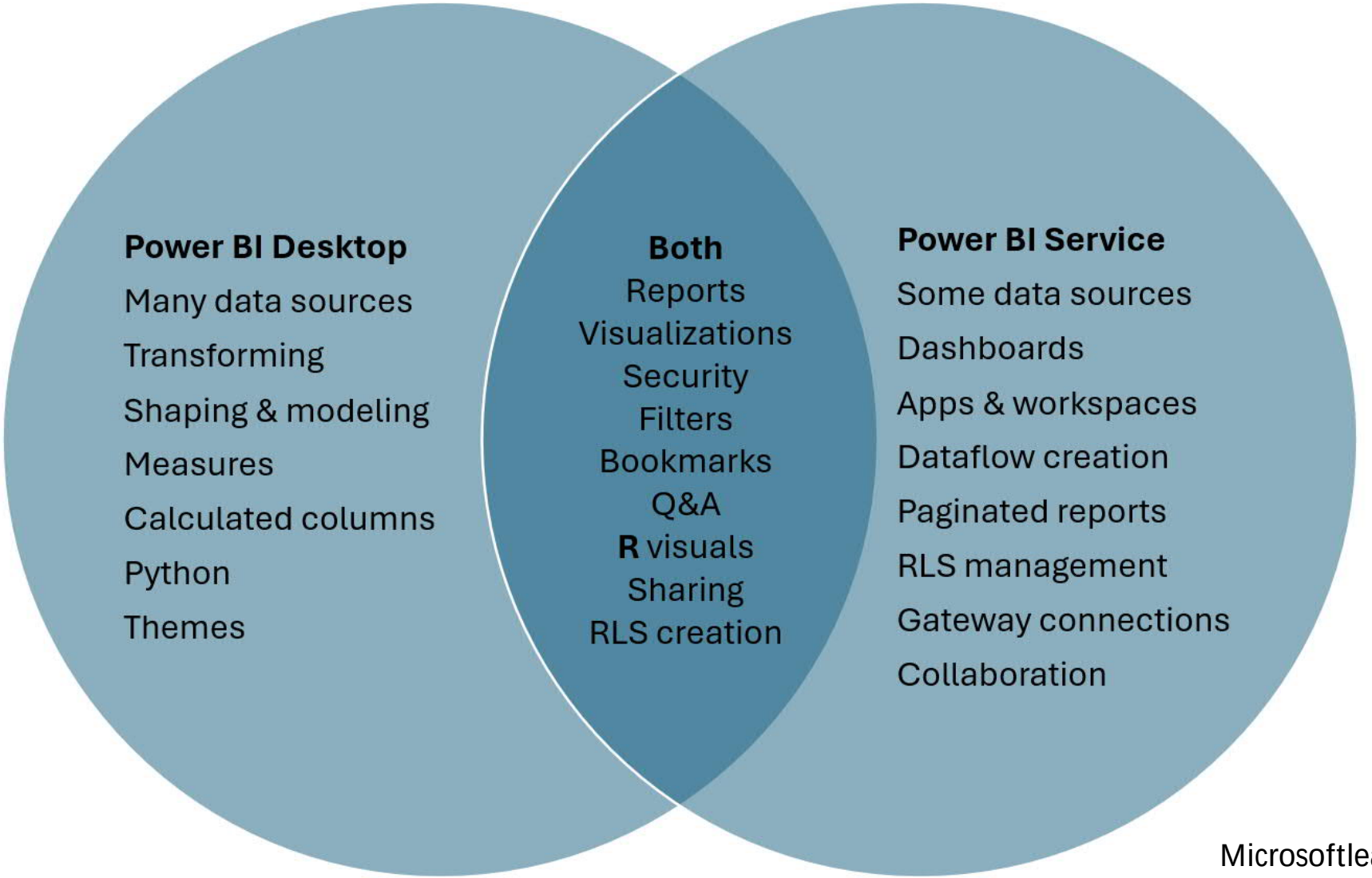
Understanding Data Connectivity & Storage Options cont'd.

2. Gateway vs. OneDrive/SharePoint

- On-Premises Data Gateway (Required for Local Files & Databases)
  - If your data is stored locally (e.g., Excel on your computer, SQL Server), you need a gateway to refresh data in Power BI Service.
  - The gateway allows Power BI to connect to on-premises data sources for scheduled refreshes.
  - Example: If you store an Excel file on your local computer and publish a Power BI report, without a gateway, the report won't refresh online.




50+ apps
CONNECT
Power BI service
ACCESS
Power BI Desktop
PUBLISH
Power BI gateways
YOUR ORGANIZATION'S DATA



Microsoftlearn.com

Prerequisites – Technical

OneDrive/SharePoint (No Gateway Required)

- If your files are stored in OneDrive for Business or SharePoint Online, Power BI can access and refresh them without needing a gateway.

- Best Practice: Use SharePoint Online file paths instead of OneDrive local sync folders for better reliability.
- Example: Uploading an Excel file to SharePoint Online lets Power BI refresh it automatically without additional setup.

Use web path to your link to SharePoint files address instead of file path. Otherwise, when other users attempt your Power Query it will fail because of access and your online files will not be able to refresh.

Prerequisites – Technical

Understanding Refresh & Automation

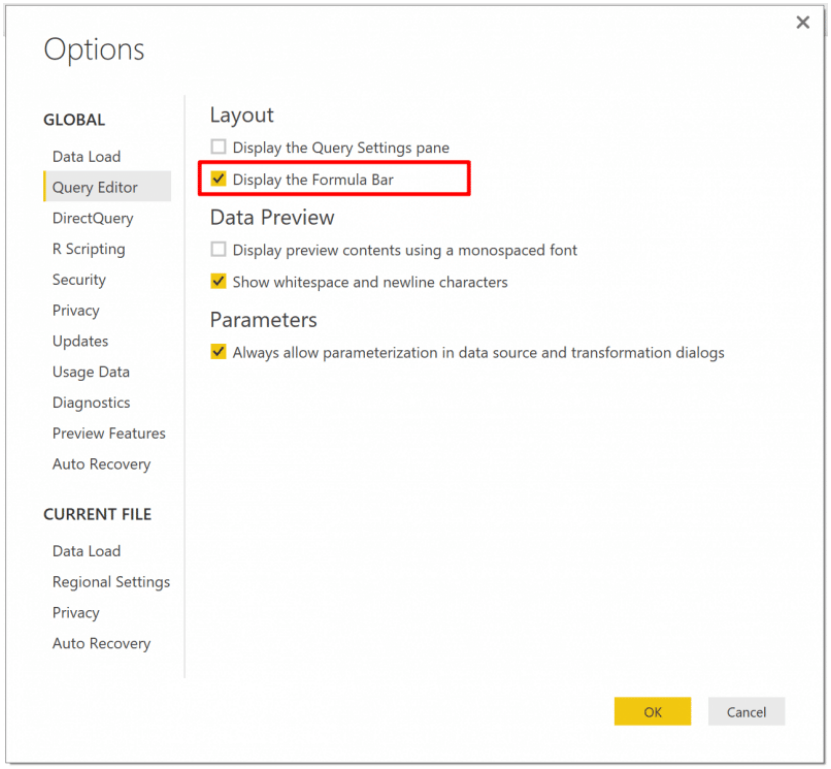
- Manual Refresh – Run refresh in Power BI Desktop.
- Scheduled Refresh (Power BI Service) – Requires a gateway for local files but not for cloud sources (SharePoint/OneDrive).
- Incremental Refresh – Available in Power BI Premium or PPU, useful for large datasets.

Essential Settings PQ

Activate Formula Bar

Open the Query Editor
Click View tab
Check the Formula Bar checkbox

- Or-
Click File tab
Select Options & Settings
Select Options
Select Global
Select Query Editor
Check Display the Formula Bar



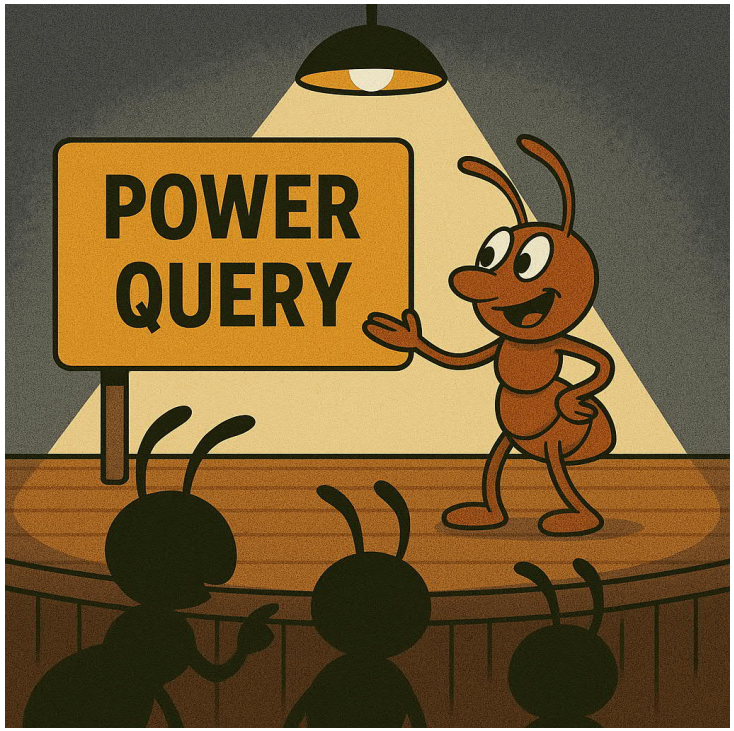
Who is Power Query For?

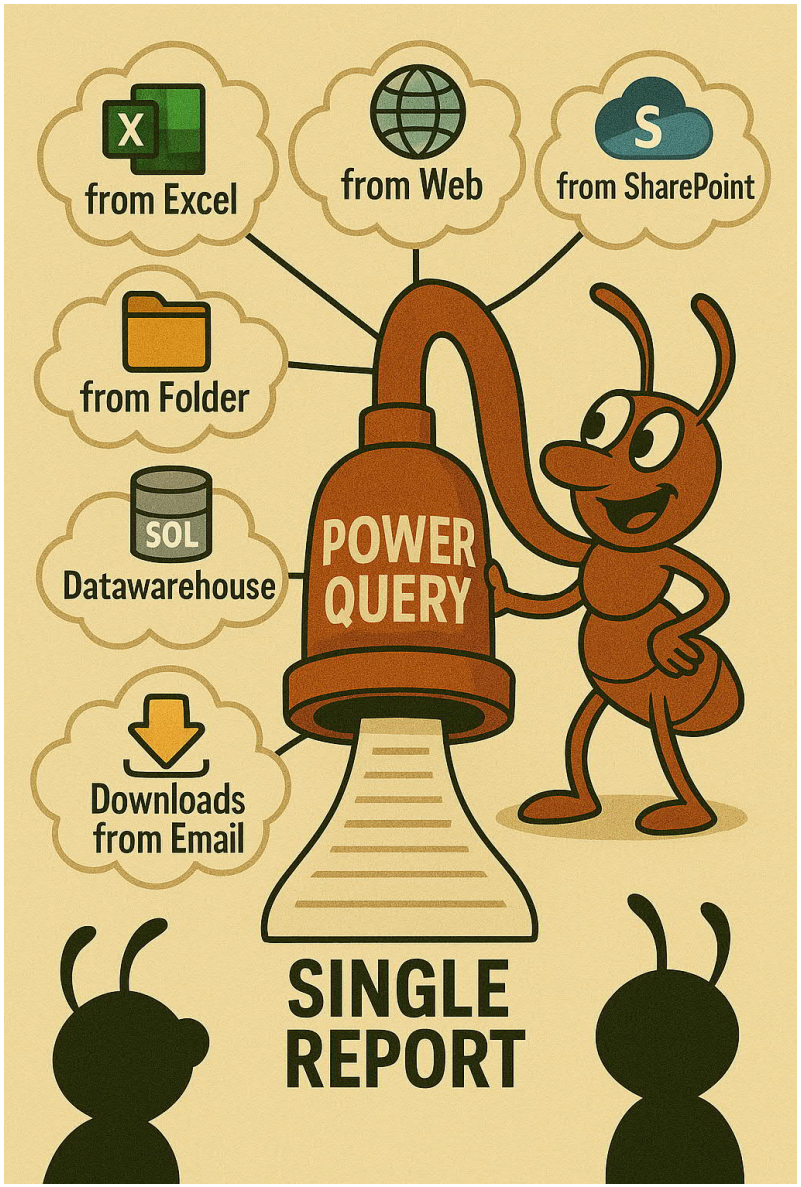
	User Group	How Power Query Benefits Them
1	Business Analysts	Allows them to pull, clean and transform data from various sources without advanced coding skills.
2	Data Analysts aka Data Wizards	Simplifies data reshaping, cleaning & transformation for further analysis.
3	Financial Analysts and Accountants	Automates repetitive tasks like merging data, filtering, and transforming columns.
4	Excel & Power BI Users	Provides seamless integration to clean and prepare data for pivot tables, reports and visualizations.
5	Non-Technical Business Users	Offers a user-friendly point-and-click interface for performing complex data transformations easily.
6	Developers	Helps prototype data transformations or prepare data for complex models, especially Power BI or ETL processes.

What is Power Query?

- Power Query is a data engine and editor. It is a powerful data connectivity and transformation tool available in Microsoft Excel and Power BI. Using Power Query, users can extract, transform, and load (ETL) processing of data. It enables users to:

  - Connect to Various Data Sources: Import data from multiple sources, such as Excel files, databases, web services, and more.
  - Transform and Clean Data: Automate the process of cleaning, reshaping, and preparing data without manual effort or repetitive tasks.
  - Streamline Workflows: Build repeatable processes with no coding required, using an intuitive, user-friendly interface.
  - Enable Advanced Transformations: Use M code (Power Query Formula Language) for complex scenarios when needed.

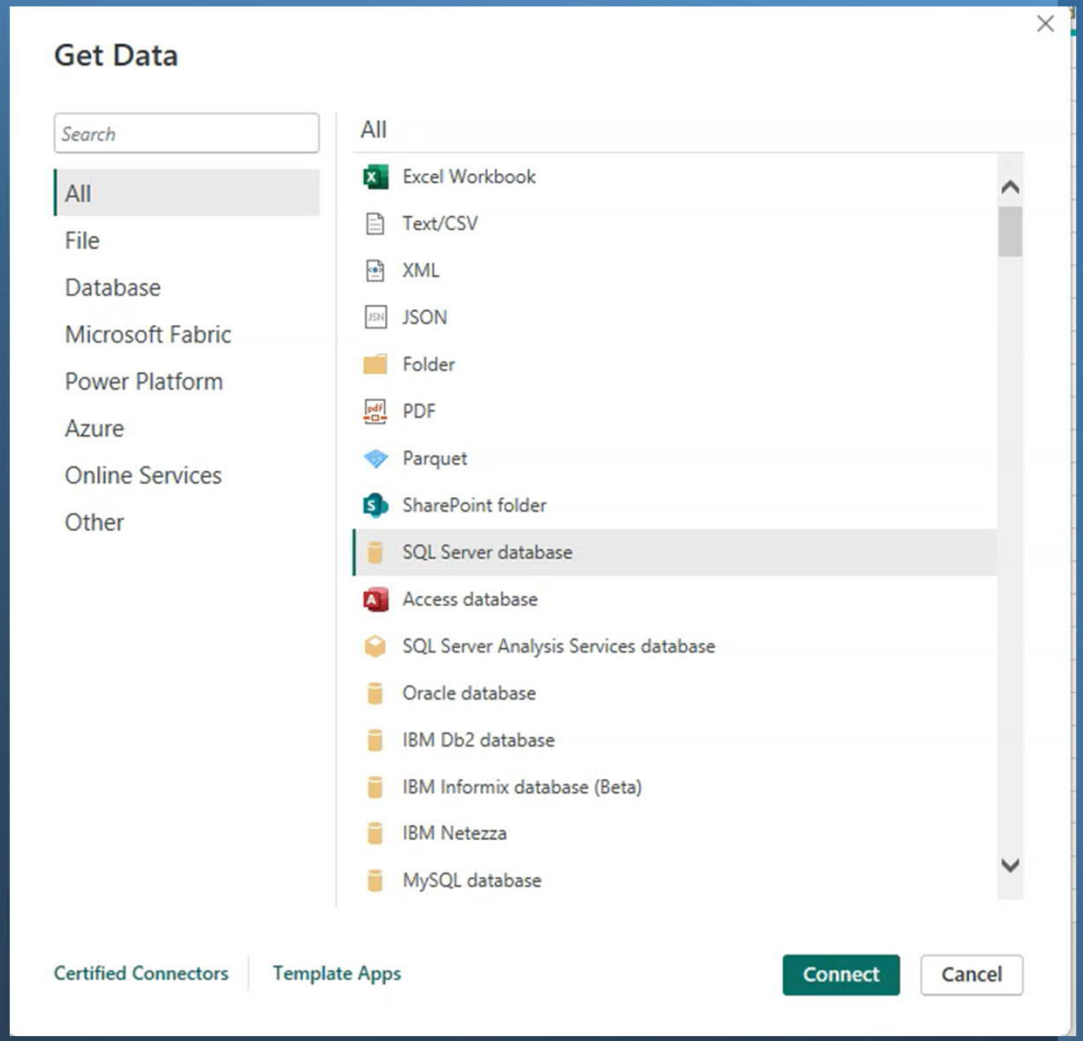


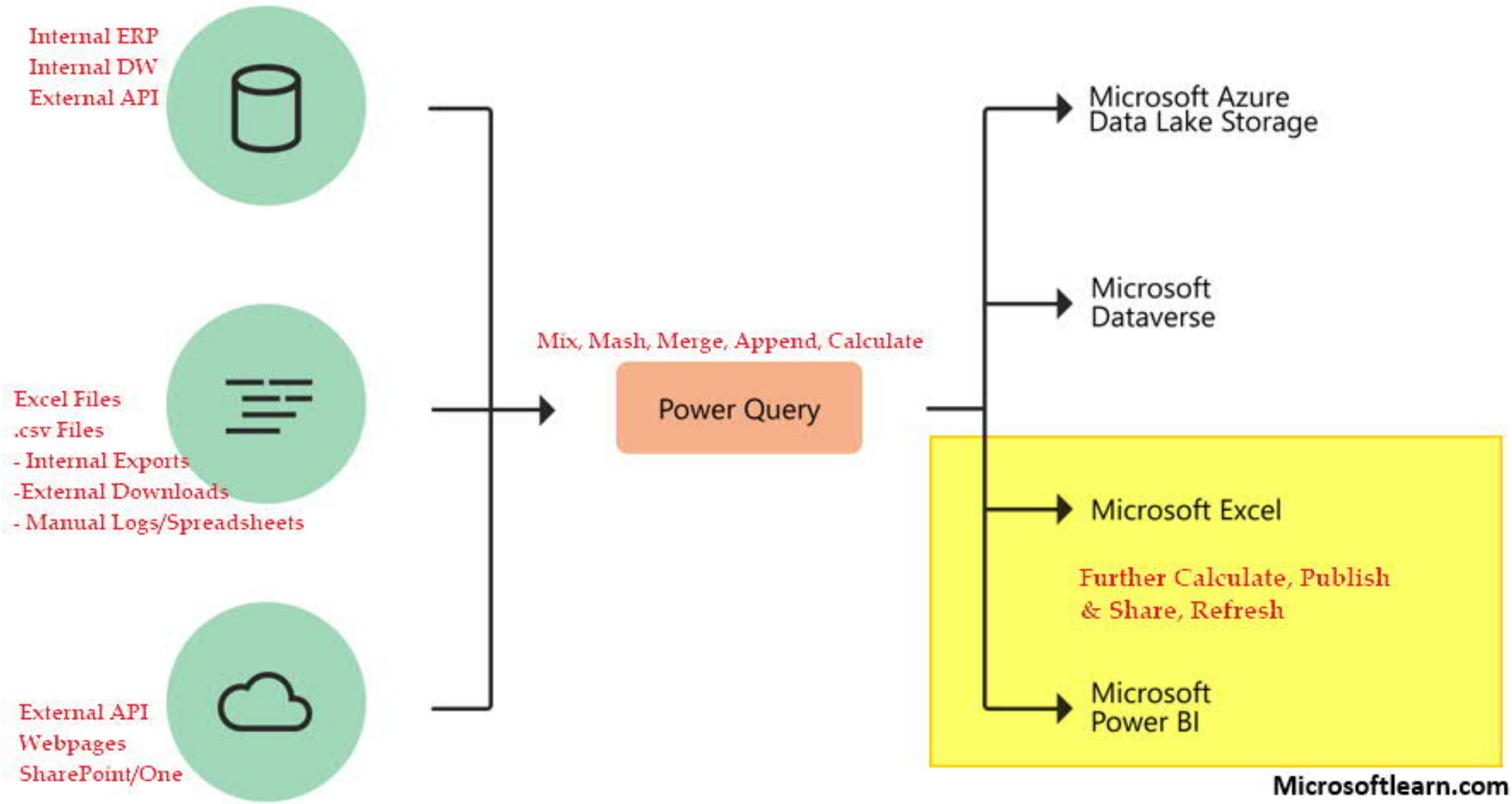


+

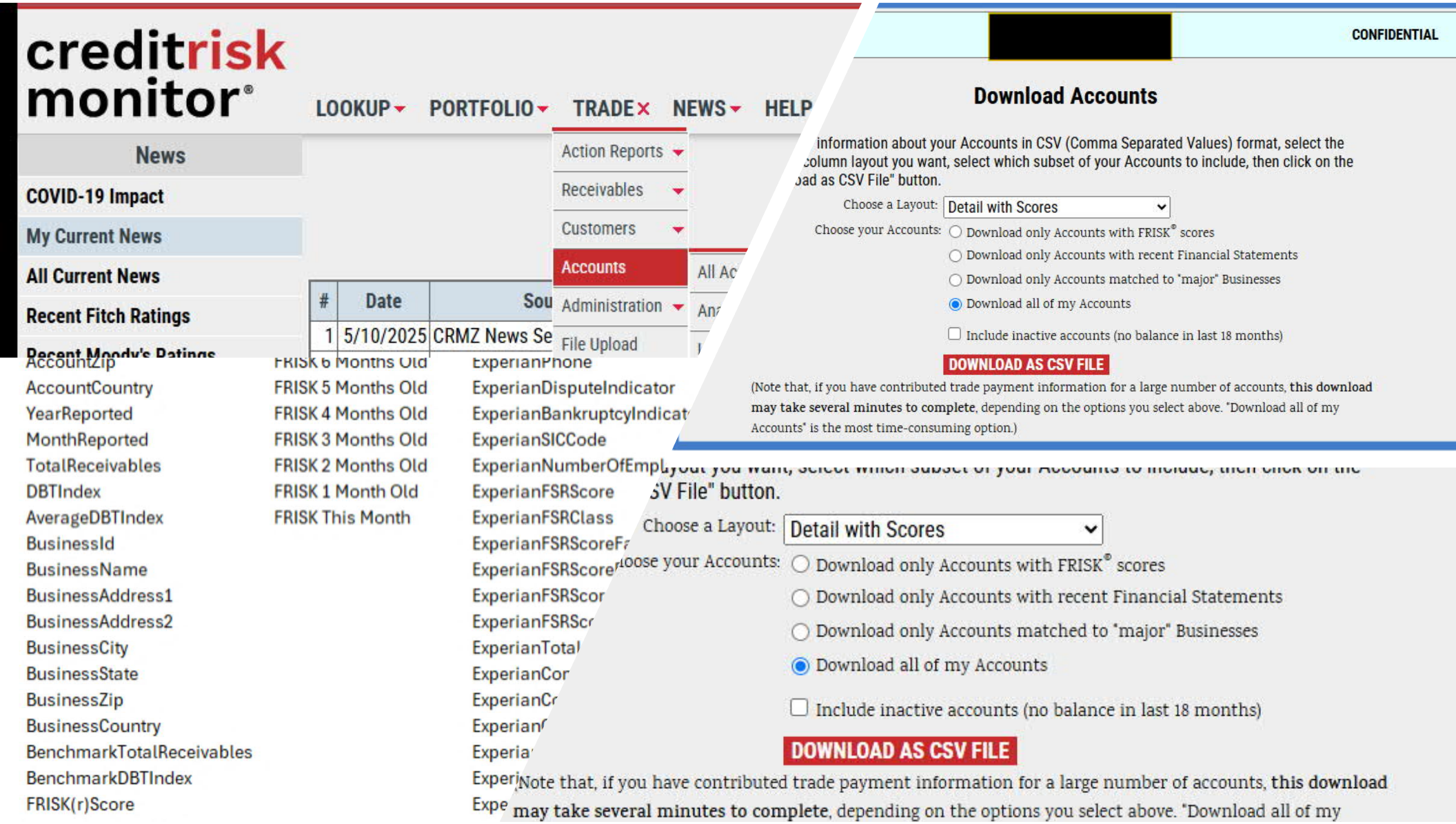
Automate Transformations

- Filters
- Adding/Removing
- Renaming Columns
- Replacing Fields
- Extracting Characters
- Adding Calculations
- Appending/Merging


Get Data
Search
All
File
Database
Microsoft Fabric
Power Platform
Azure
Online Services
Other
All
Excel Workbook
Text/CSV
XML
JSON
Folder
PDF
Parquet
SharePoint folder
SQL Server database
Access database
SQL Server Analysis Services database
Oracle database
IBM Db2 database
IBM Informix database (Beta)
IBM Netezza
MySQL database
Certified Connectors
Template Apps
Connect
Cancel


Internal ERP
Internal DW
External API
Excel Files
.csv Files
- Internal Exports
-External Downloads
- Manual Logs/Spreadsheets
External API
Webpages
SharePoint/One
Mix, Mash, Merge, Append, Calculate
Power Query
Microsoft Azure
Data Lake Storage
Microsoft
Dataverse
Microsoft Excel
Further Calculate, Publish
& Share, Refresh
Microsoft
Power BI
Microsoftlearn.com


Data Sources for Credit Managers
Credit
Records
Filings
Bradstreet
Duns
Experian
Risk
Comptroller
Public
Legal
Financial
Analysis
Payment
Customer
Reviews
State
Codes
Secretary
Taxes
Court
Data
Bureau
Open
Property
Insurance
Sites
Corporate
Bankruptcy
Licenses
NACM
NAICS
Provider
SOS
Statements
Payable
Judgment
Trends
Business
Accounts
UCC
References
Trade
Industry
Lien
Certificates
Monitor
Reports


creditrisk monitor®
LOOKUP
PORTFOLIO
TRADE
NEWS
HELP
News
COVID-19 Impact
My Current News
All Current News
Recent Fitch Ratings
Recent Moody's Ratings
Action Reports
Receivables
Customers
Accounts
Administration
File Upload
All Ac
An
Date Sou
1 5/10/2025 CRMZ News Se
AccountZip
AccountCountry
YearReported
MonthReported
TotalReceivables
DBTIndex
AverageDBTIndex
BusinessId
BusinessName
BusinessAddress1
BusinessAddress2
BusinessCity
BusinessState
BusinessZip
BusinessCountry
BenchmarkTotalReceivables
BenchmarkDBTIndex
FRISK(r)Score
FRISK 6 Months Old
FRISK 5 Months Old
FRISK 4 Months Old
FRISK 3 Months Old
FRISK 2 Months Old
FRISK 1 Month Old
FRISK This Month
ExperianPhone
ExperianDisputeIndicator
ExperianBankruptcyIndicat
ExperianSICCode
ExperianNumberOfEmpl
ExperianFSRScore
ExperianFSRClass
ExperianFSRScoreFa
ExperianFSRScore
ExperianFSRScor
ExperianFSRScr
ExperianTotal
ExperianCor
ExperianCr
Experian
Experia
Experi
Expe
CONFIDENTIAL
Download Accounts
information about your Accounts in CSV (Comma Separated Values) format, select the
column layout you want, select which subset of your Accounts to include, then click on the
oad as CSV File" button.
Choose a Layout: Detail with Scores
Choose your Accounts:
Download only Accounts with FRISK® scores
Download only Accounts with recent Financial Statements
Download only Accounts matched to "major" Businesses
Download all of my Accounts
Include inactive accounts (no balance in last 18 months)
DOWNLOAD AS CSV FILE
(Note that, if you have contributed trade payment information for a large number of accounts, this download may take several minutes to complete, depending on the options you select above. "Download all of my Accounts" is the most time-consuming option.)
SV File" button.
Choose a Layout: Detail with Scores
oose your Accounts:
Download only Accounts with FRISK® scores
Download only Accounts with recent Financial Statements
Download only Accounts matched to "major" Businesses
Download all of my Accounts
Include inactive accounts (no balance in last 18 months)
DOWNLOAD AS CSV FILE
Note that, if you have contributed trade payment information for a large number of accounts, this download may take several minutes to complete, depending on the options you select above. "Download all of my

Vitals | A/R Details | Alerts | Decisions | Watch+ | Boost | Manage | Circles | Preferences

Boost New File | Boost Results

Boost New File

Boost is a web-based, self service tool that allows you to get in-depth insights on your customers and prospects. In a few easy steps you can append data to your customers and prospects including the delinquency risk score, buying behavior information, demographics, parent-child relationships, and payment information. See a sample Boost file.

Choose the type of Boost Insights:

- (●) **Risk Boost - $1.00 per match**
  Includes Delinquency Risk Score, Payment Score, Payment Details and more.
- ( ) **Boost with Spend - $1.50 per match**
  Includes purchase behavior across all segments (shipping, materials, operations and other), in addition to the risk boost.
- ( ) **Boost with Trend - $1.50 per match**
  Includes purchase trends for all segments, in addition to the risk boost.
- ( ) **Boost Plus - $2.00 per match**
  Includes both spend AND trend across all segments, plus the risk boost.
- ( ) **Credit Decisions and Credit Limits - $3.00 per match**
  Includes credit decision, recommended limits and Boost Plus fields.

Select your scorecard to apply:

Default

First, click "Browse" to choose a file and then click "Submit File" to continue to the Map Fields page.

New BOOST File: Choose File No file chosen

File Format: (●) Comma Delimited ( ) Tab Delimited ( ) Pipe Delimited

Files need to be in .CSV format.
Maximum file size is 5.0 MB - call us if you you have a larger file. Click here to learn about required fields.

Submit File

Lien and Legal Reports

NACM offers statewide and regional legal reports to help credit and finance professionals manage accounts receivable and even prevent bad debt losses.

Subscription Details

Reports are published and distributed weekly for the states listed below (North Carolina, Florida, and Virginia). The report format is a native, optimized, and searchable PDF format. (Opens fast and takes up less space on your PC).

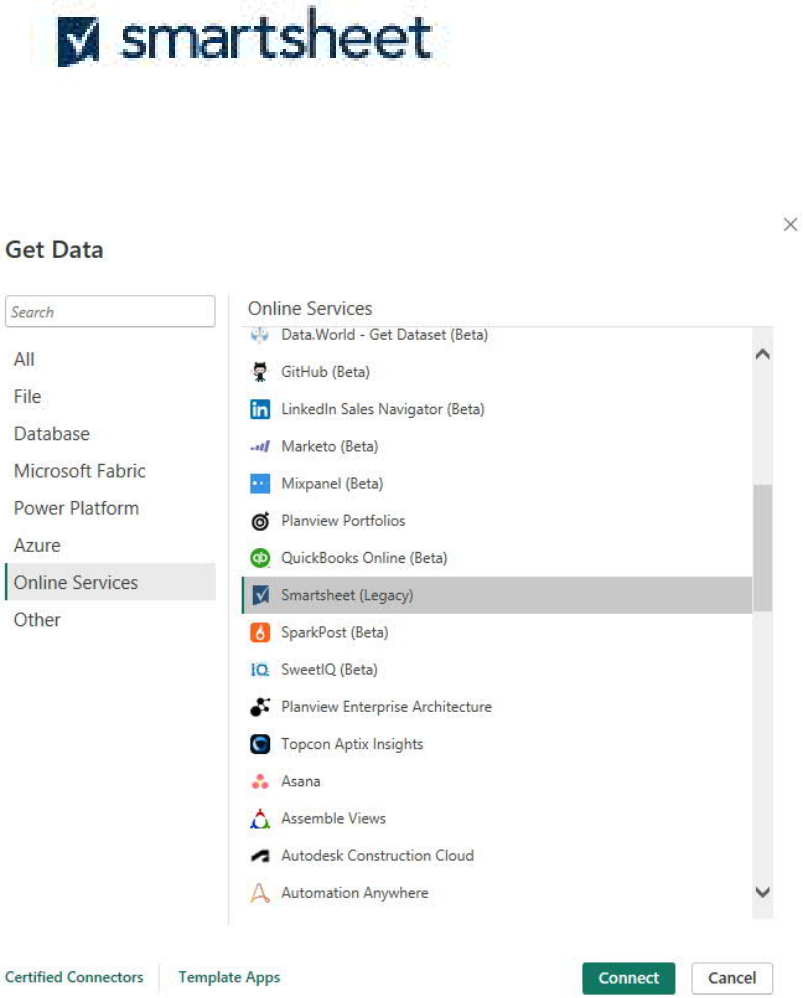
- North Carolina
- Florida
- Virginia

North Carolina Lien and Legal Report

The North Carolina Lien and Legal Report is published weekly and subscribers receive email notification when the report has been posted to their member account (login required to access the report). Bankruptcy records are very current and many times our members see a bankruptcy listing in the legal report or a NACM Credit Report before receiving a notification from the bankruptcy court in the mail. Other records come from multiple sources including physical trips to counties to review and key in data in person. These records may not be from the prior week but in every possible way we make the effort to post public records as quickly as possible.

The sample report for the North Carolina Lien and Legal Report is available for download.

Report Area	Details	Monthly Price



Connect Excel to Smartsheet via Power Query

Asked 10 years, 4 months ago Modified 9 years, 1 month ago Viewed 3k times

0

I have been trying to pull data from one of my Sheets in Smartsheet using Power Query without no luck. I generated an API token from Smartsheet that I used in the Power Query Wizard but still failed.

smartsheet-api powerquery

Share Improve this question Follow

asked Dec 22, 2014 at 12:01
Ivan Oboth
41 • 1 • 6

Add a comment

1 Answer

Sorted by: Highest score (default)

2

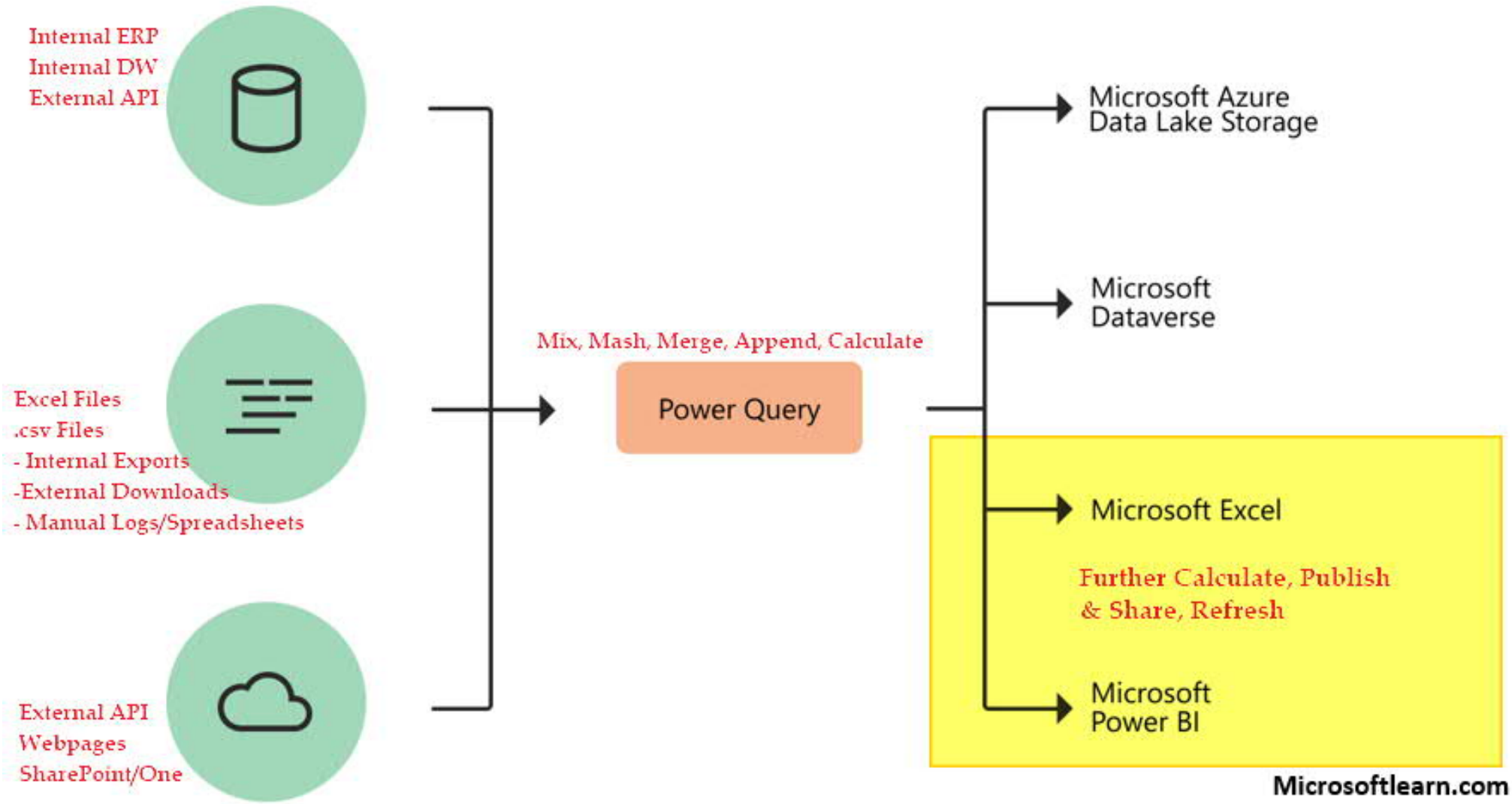
1. In power query editor I basically hardcoded the authorization header + API token:

```
= Web.Contents(
    "https://api.smartsheet.com/1.1/sheet/4693183612381060",
    [
        Headers = [
            #"Authorization" = "Bearer YOUR_API_TOKEN"
        ]
    ]
)
```

Connecting to Extra Data Sources

Useful External Data Sources for Credit Managers

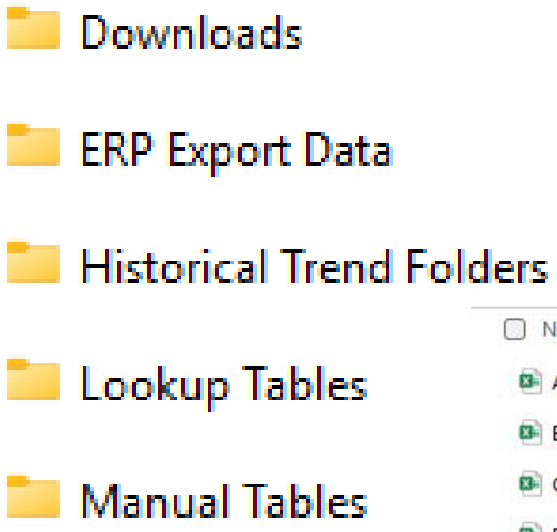
- Credit Risk Monitor
- Moody's Boost Report
- Legal Review
- Public Information Websites
- WSJ Prime Rate
- Other 3rd Party Downloads

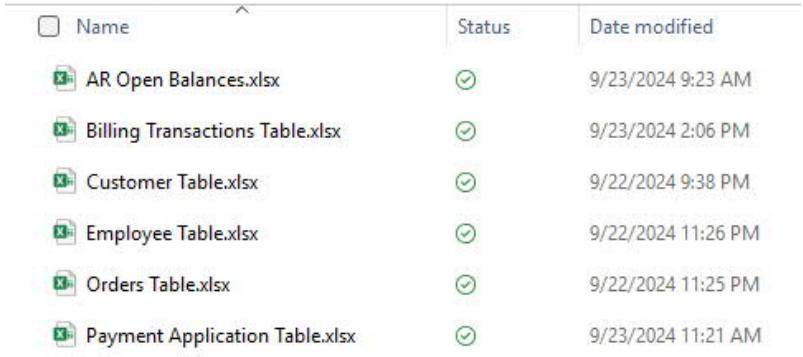

Internal ERP
Internal DW
External API
Excel Files
.csv Files
- Internal Exports
-External Downloads
- Manual Logs/Spreadsheets
External API
Webpages
SharePoint/One
Mix, Mash, Merge, Append, Calculate
Power Query
Microsoft Azure
Data Lake Storage
Microsoft
Dataverse
Microsoft Excel
Further Calculate, Publish
& Share, Refresh
Microsoft
Power BI
Microsoftlearn.com

Organizing Data Sources

Create folders for Data Sources:

- Look Up Tables
  - Department Hierachy
- 3rd Party Downloads
- ERP Exports
  - Historical Trend Reports
  - Daily Reports
- Manual Data Sources





Why use Power Query?

1. Efficiency- Reduce redundancy, build reports once and refresh
2. Speed – Data refresh reports, Data cleaning and transforming
3. Consistency – Increase accuracy with stored calculations
4. Preservation - Documentation of steps as audit trail.
5. Duplication - Automation of complicated reports others can update
6. Enhancement – Combine data from multiple sources and added calculations.

#	Existing Challenges with Data	How Power Query Helps
1	Inconsistent Data Formats	Automatically detect and standardize data formats (e.g., dates, numbers, text).
2	Manual Data Cleaning	Automates repetitive data cleaning tasks like removing duplicates, changing case, or splitting columns.
3	Combining Data from Multiple Sources	Easily merges data from different files, databases, and web services into a single unified table.
4	Large and Unmanageable Datasets	Handles large datasets more efficiently by querying only relevant data and allowing load on demand.
5	Data Entry Errors (typos, extra spaces, missing values)	Provides tools to correct data entry errors (e.g., trim spaces, fill missing data) automatically.
6	Difficulty Refreshing Updated Data	Allows users to refresh datasets with one click when the underlying data source is updated.

#	Existing Challenges with Data	How Power Query Helps
7	Complex Transformations Requiring Multiple Steps	Streamlines complex transformations using an intuitive, step-by-step interface, which is repeatable and modifiable.
8	Combining Multiple Worksheets into One	Power Query can easily append multiple worksheets into a single table.
9	Data from Non-Excel Formats (e.g., CSV, JSON, XML)	Easily imports and converts data from various file formats without requiring manual reformatting.
10	Pivot Table Source Data Management	Cleans and reshapes data to provide well-structured, reliable sources for pivot tables.
11	Difficulty Tracking Data Changes	Keeps a record of all transformations, making it easy to review and adjust as needed.
12	Complex Filtering and Sorting	Provides advanced filtering, sorting, and custom column creation without needing complex formulas.

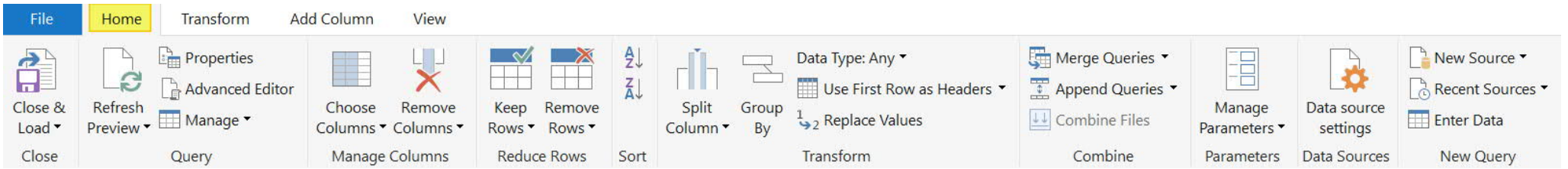
Power Query really shines when you need to combine multiple tables together from various sources and of various types repeatedly and add your own customization and calculations.

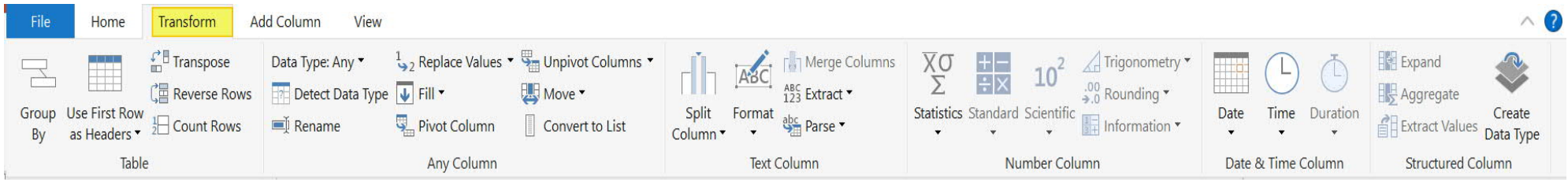
Connecting to Data Sources

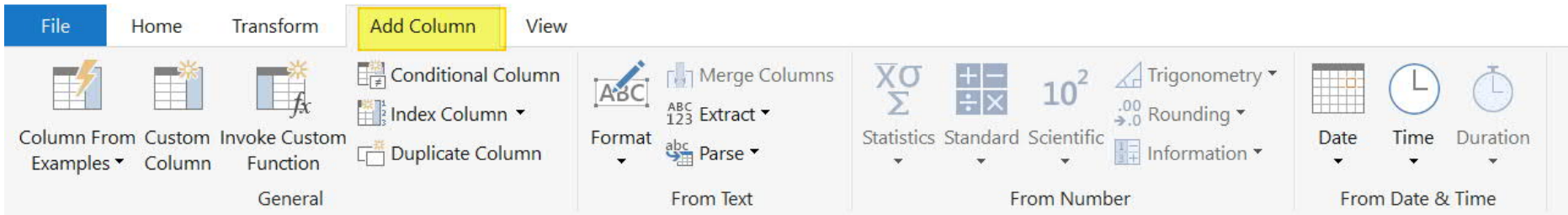
Supported Data Sources

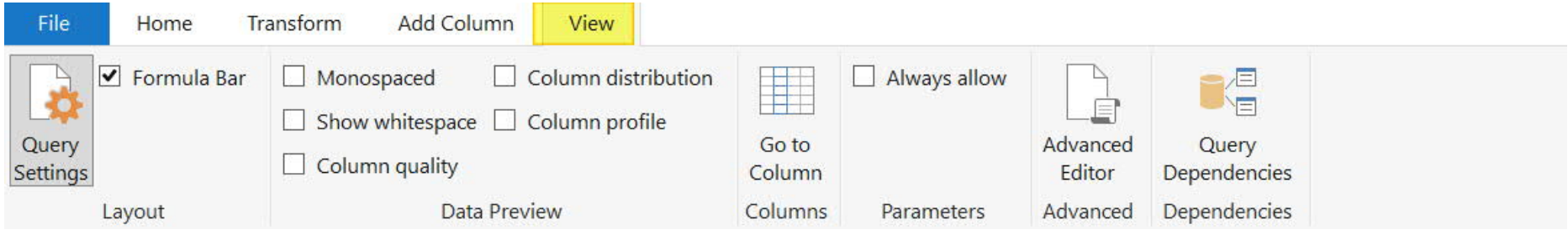
- Excel Tables
- Excel files, CSVs and text files
- Databases (SQL Server, Oracle)
- Online services (SharePoint, web API's, SmartSheet)
- Folders
- SharePoint Folders

According to Microsoft, Data preparation can consume up to 80% of business users time, delaying analysis and decision-making tasks. Several issues contribute to this condition, Power Query helps with many of them.

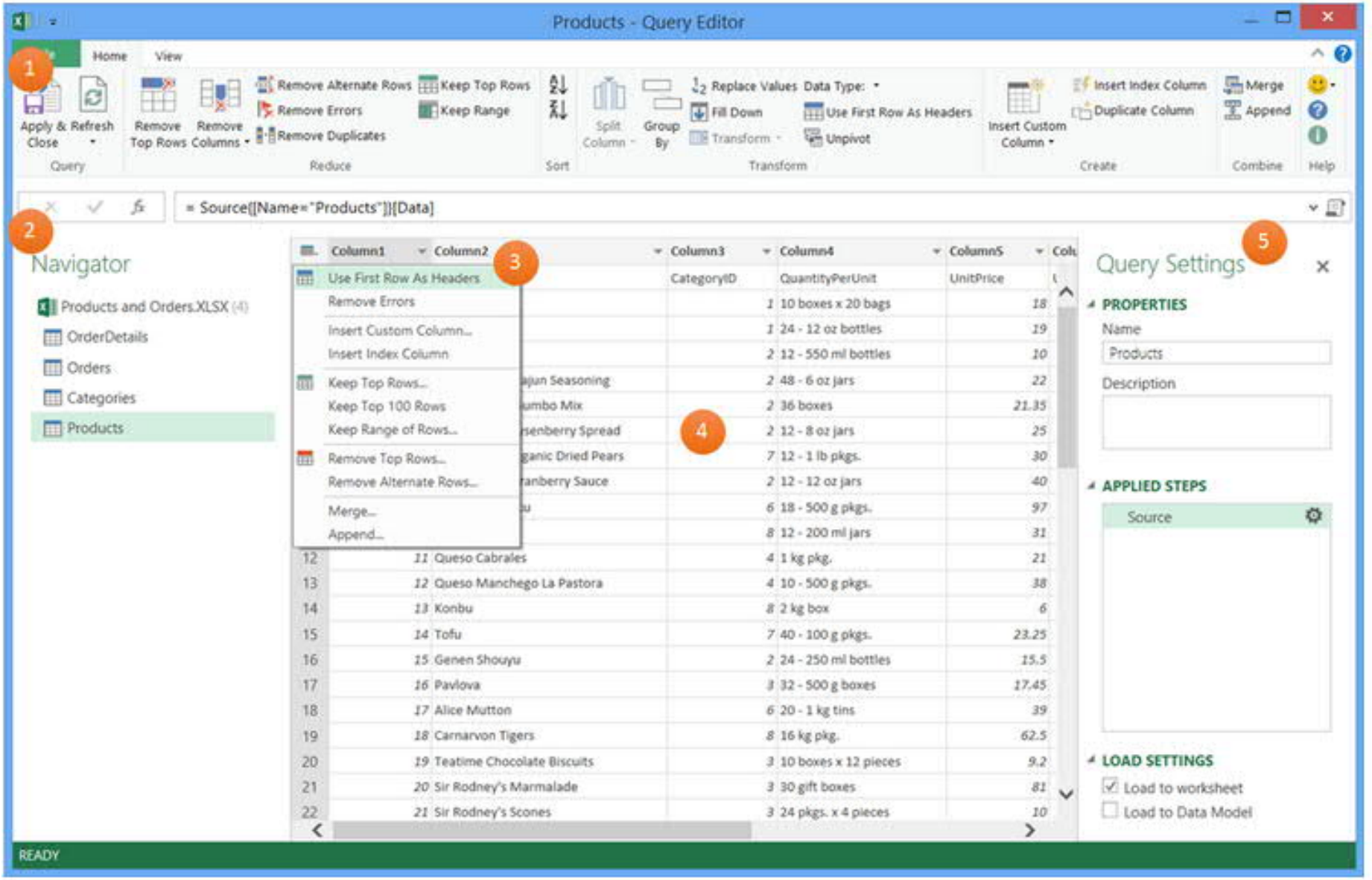

File
Home
Transform
Add Column
View
Close & Load
Refresh Preview
Properties
Advanced Editor
Manage
Choose Columns
Remove Columns
Keep Rows
Remove Rows
Split Column
Group By
Data Type: Any
Use First Row as Headers
Replace Values
Merge Queries
Append Queries
Combine Files
Manage Parameters
Data source settings
New Source
Recent Sources
Enter Data
Close
Query
Manage Columns
Reduce Rows
Sort
Transform
Combine
Parameters
Data Sources
New Query


File
Home
Transform
Add Column
View
Group By
Use First Row as Headers
Transpose
Reverse Rows
Count Rows
Data Type: Any
Detect Data Type
Rename
Replace Values
Fill
Pivot Column
Unpivot Columns
Move
Convert to List
Split Column
Format
Merge Columns
Extract
Parse
Statistics
Standard
Scientific
Trigonometry
Rounding
Information
Date
Time
Duration
Expand
Aggregate
Extract Values
Create Data Type
Table
Any Column
Text Column
Number Column
Date & Time Column
Structured Column


File
Home
Transform
Add Column
View
Column From Examples
Custom Column
Invoke Custom Function
Conditional Column
Index Column
Duplicate Column
Format
Merge Columns
Extract
Parse
Statistics
Standard
Scientific
Trigonometry
Rounding
Information
Date
Time
Duration
General
From Text
From Number
From Date & Time

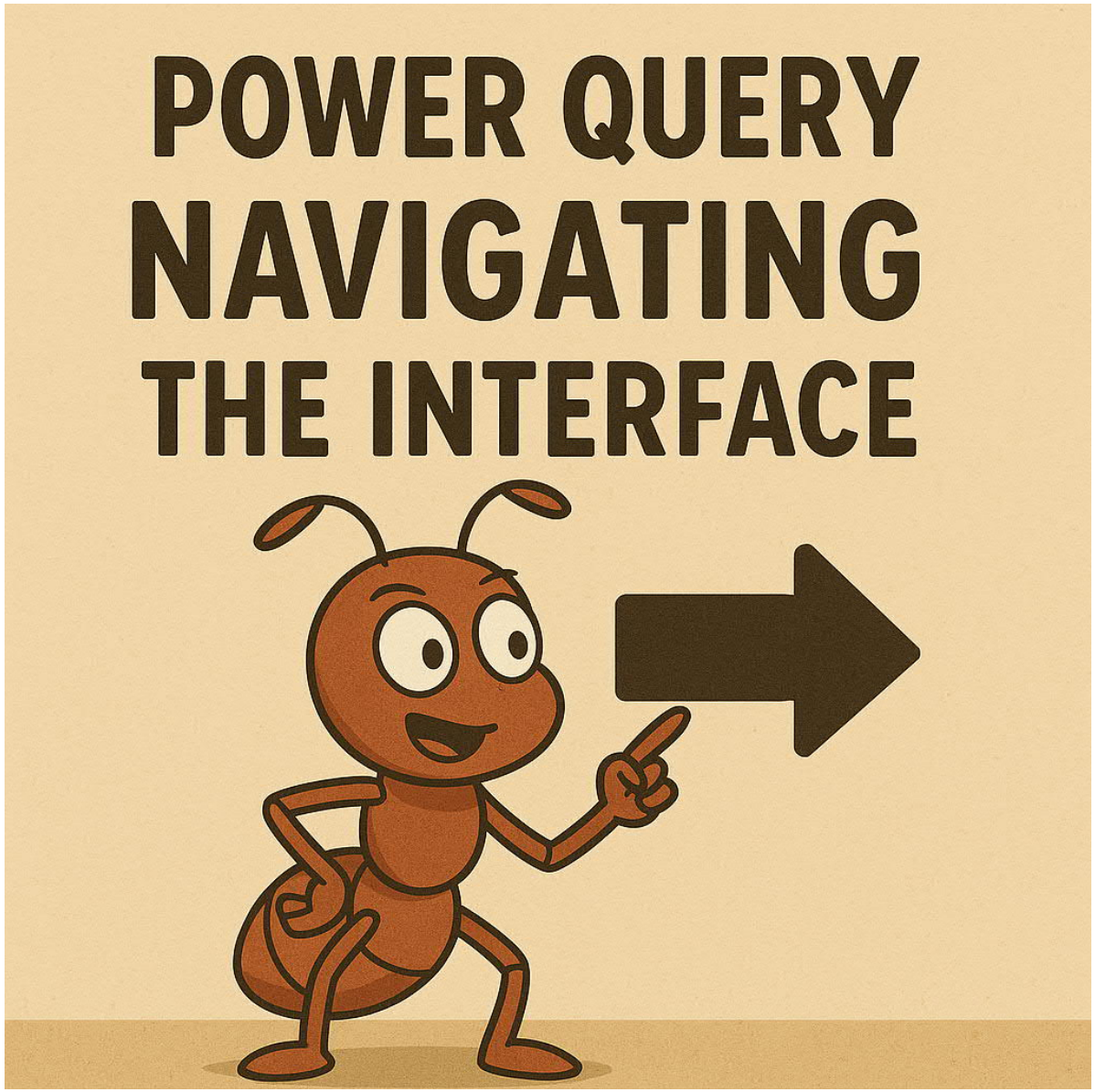

File
Home
Transform
Add Column
View
Query Settings
Formula Bar
Monospaced
Column distribution
Show whitespace
Column profile
Column quality
Go to Column
Always allow
Advanced Editor
Query Dependencies
Layout
Data Preview
Columns
Parameters
Advanced
Dependencies

Query Editor



Benefits of Learning Power Query in Excel:

1. Excel is familiar territory for all of us
2. Increased opportunities for internal use leads to wider application
3. No SQL Coding Required, Easy drag and drop interface, Formula like approach
4. Increased usage cases leads to increased experience
5. Small quick wins will encourage motivation to learn more.
6. Logical transition to Power PI
7. Low- Risk Learning – No impact on Raw Data

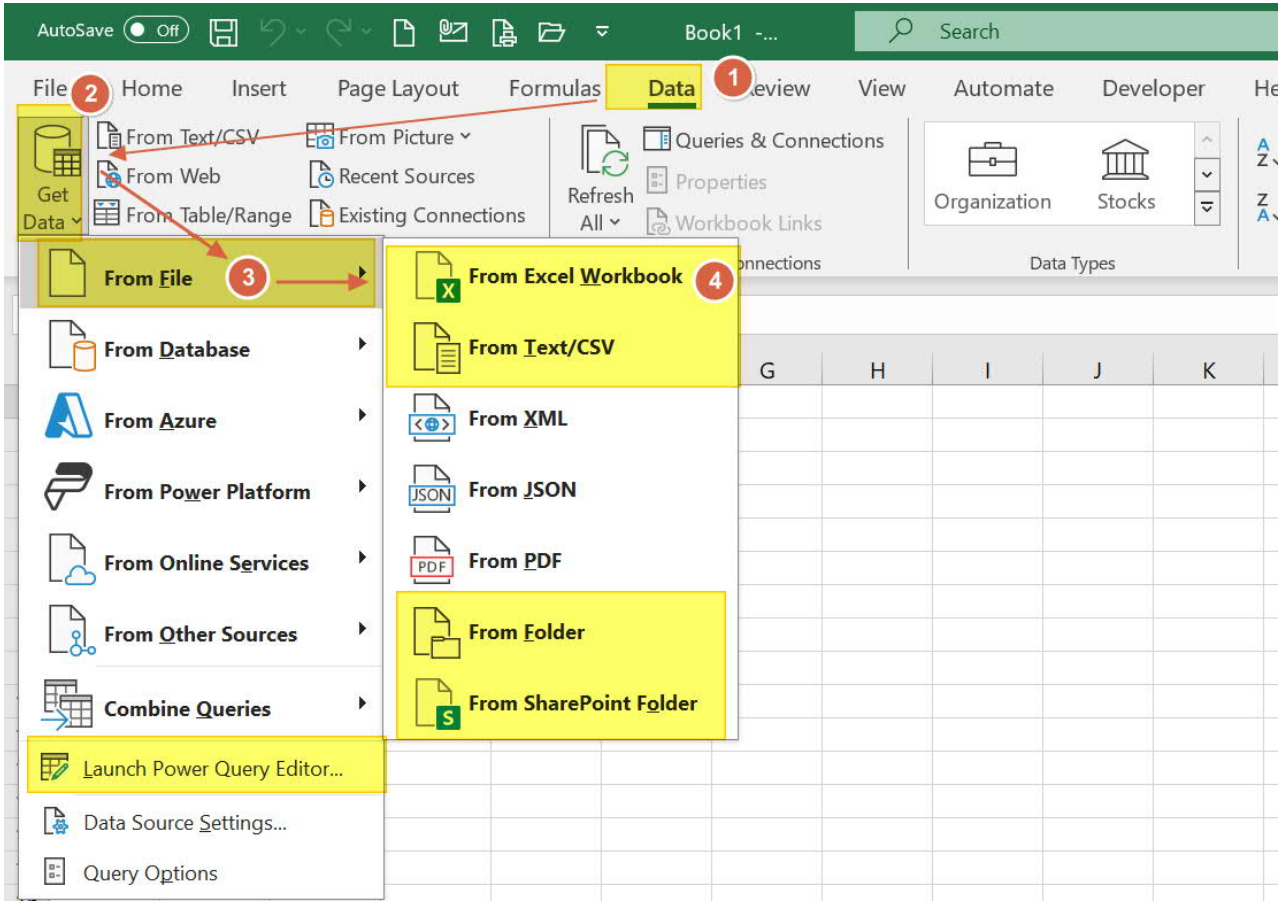

POWER QUERY
NAVIGATING
THE INTERFACE


Let’s start on familiar territory with some spreadsheets!!

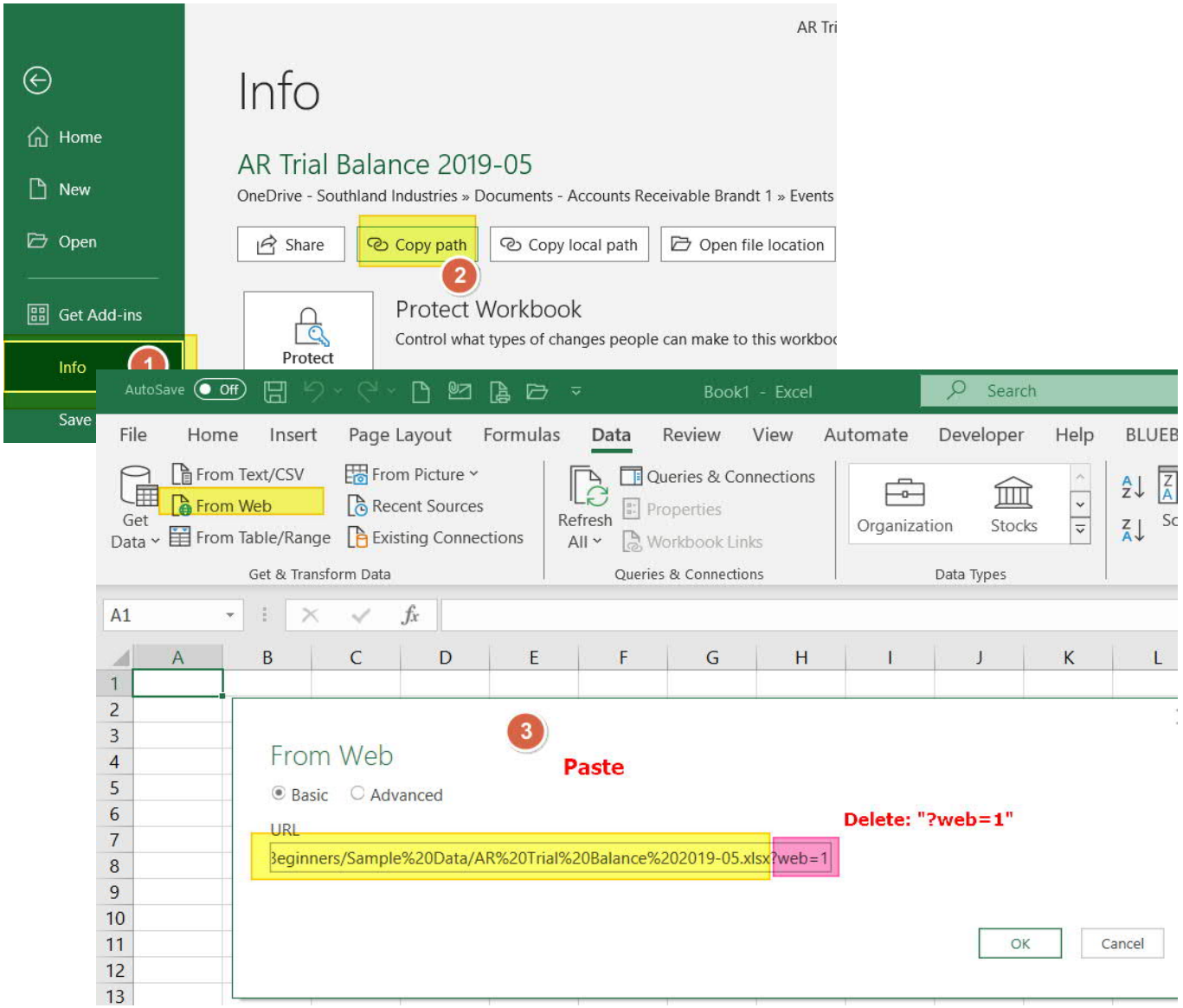
1. Load Excel SharePoint Data Sources as Web Links

1. **OPEN** Power BI
    1. **ADD** New Data Source from **Web**
    2. **PASTE** file path into from Web Url.

Accessing Power Query in Excel



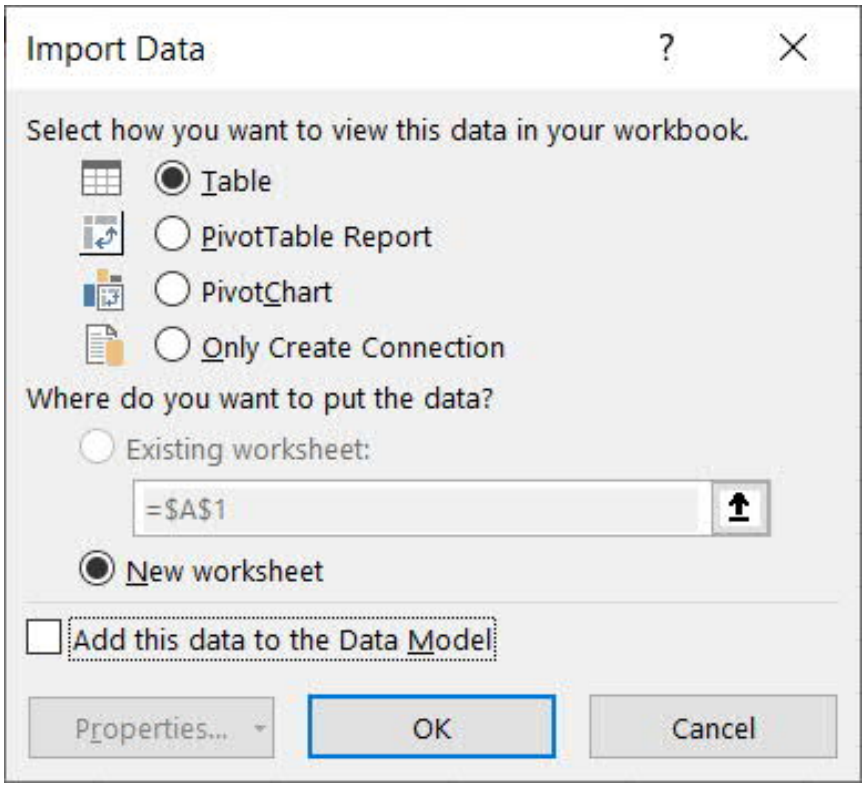
Loading Power Quiery in Exce;

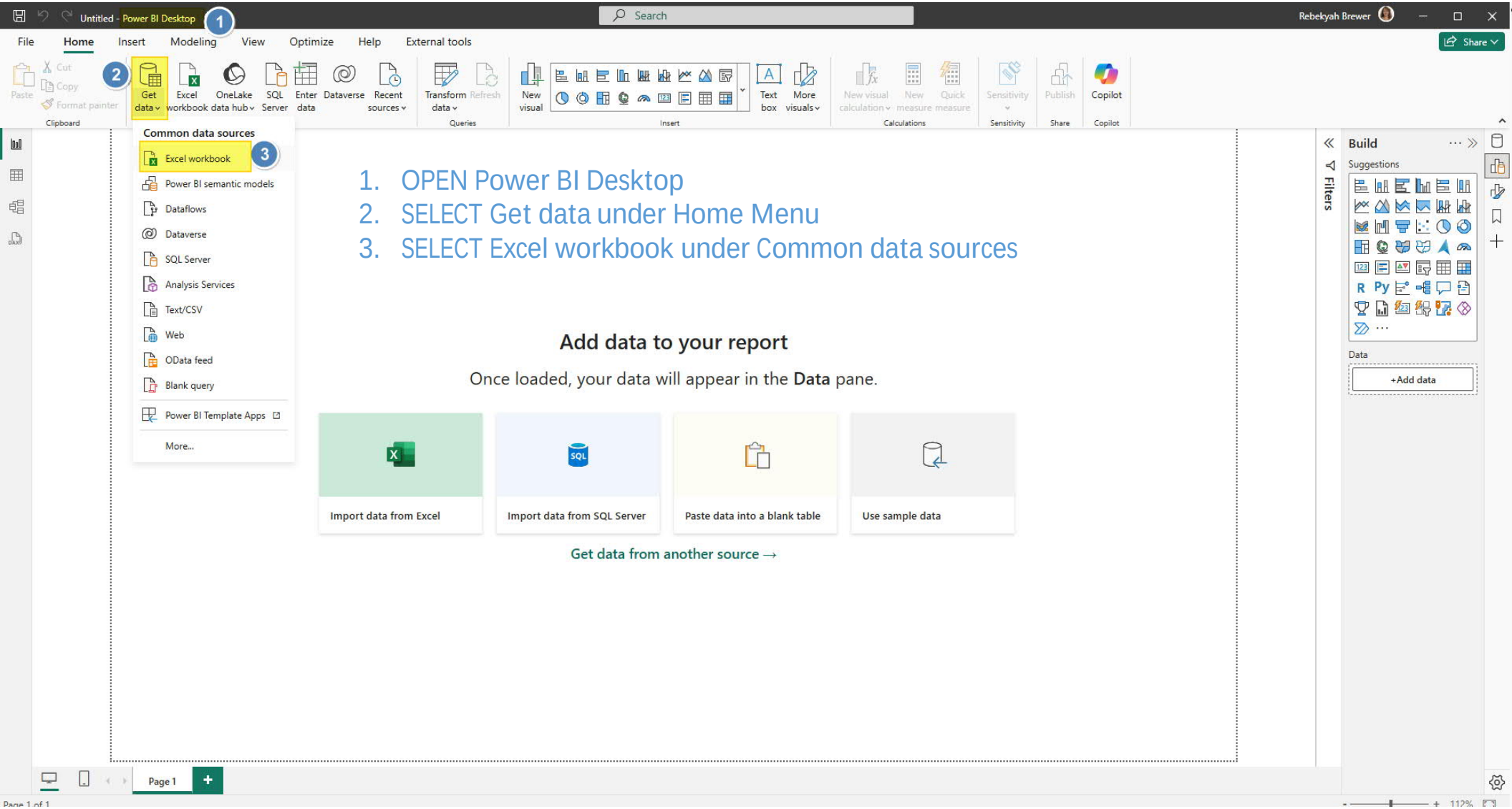


Data Load Options

Where can you load?

- Excel Tables
- Power Pivot Data Model
- Power BI
- Connection Only



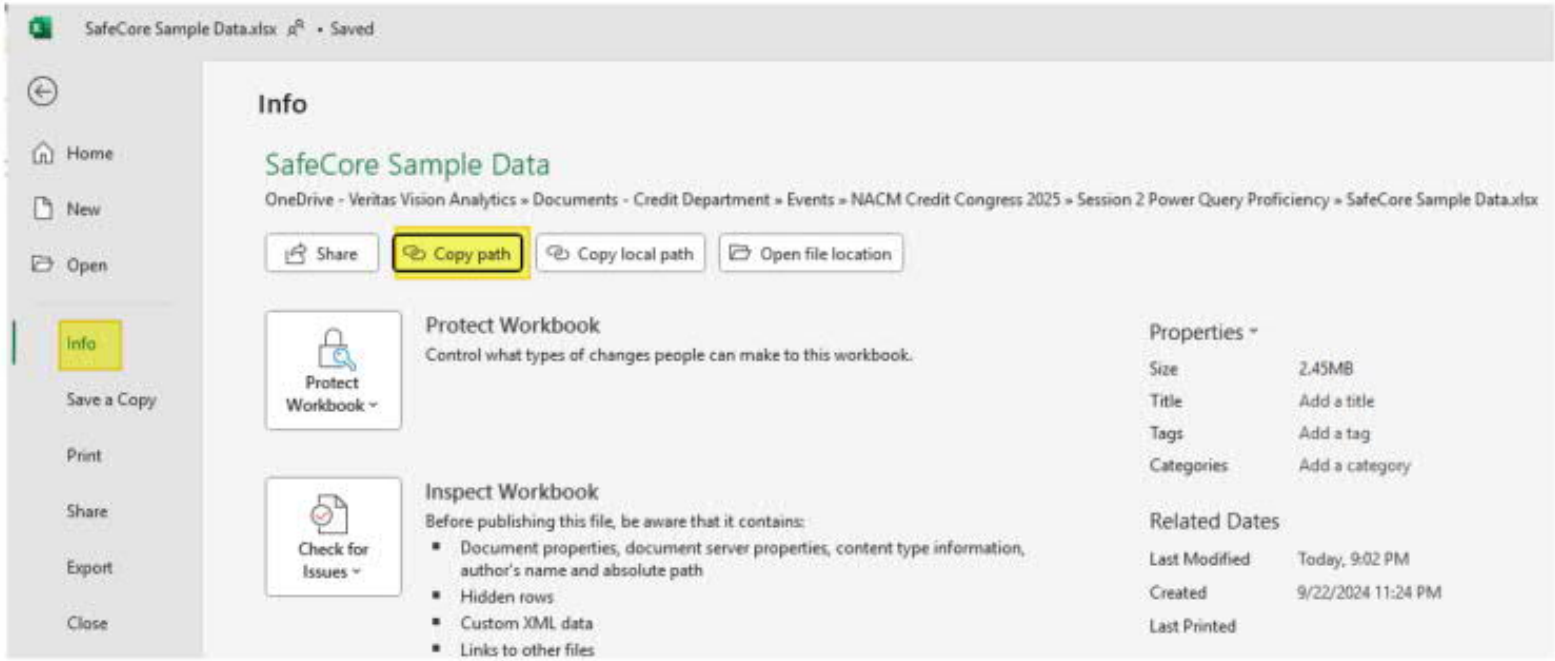

Untitled - Power BI Desktop
1
Search
Rebekyah Brewer
File
Home
Insert
Modeling
View
Optimize
Help
External tools
Share
Paste
Cut
Copy
Format painter
Clipboard
2
Get data
Excel workbook
OneLake data hub
SQL Server
Enter data
Dataverse
Recent sources
Queries
Transform data
Refresh
New visual
Text box
More visuals
Insert
New visual calculation
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
Copilot
Common data sources
3
Excel workbook
Power BI semantic models
Dataflows
Dataverse
SQL Server
Analysis Services
Text/CSV
Web
OData feed
Blank query
Power BI Template Apps
More...
1. OPEN Power BI Desktop
2. SELECT Get data under Home Menu
3. SELECT Excel workbook under Common data sources
Add data to your report
Once loaded, your data will appear in the Data pane.
Import data from Excel
Import data from SQL Server
Paste data into a blank table
Use sample data
Get data from another source →
Filters
Build
Suggestions
Data
+Add data
Page 1
Page 1 of 1
112%

1. Load Excel SharePoint Data Sources as Web Links

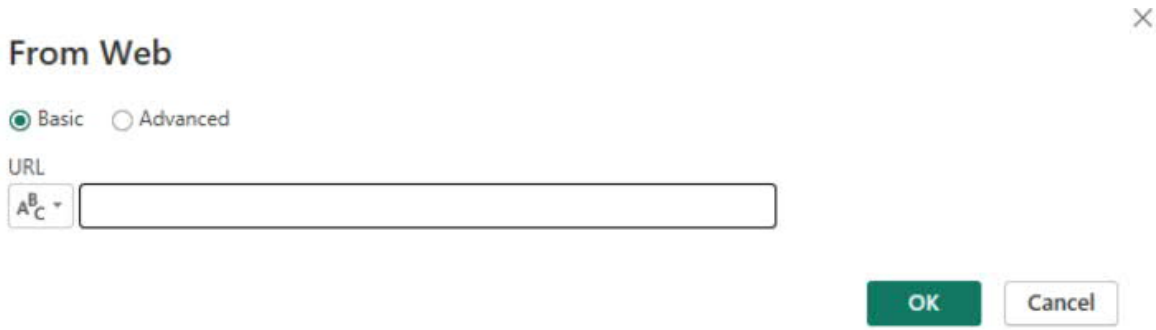
1. **OPEN** Power BI
    1. **ADD** New Data Source from **Web**
    2. **PASTE** file path into from Web Url.

Get Weblink for an Excel File on SharePoint

1. **OPEN** Excel Data Source file in Excel to get web address.
   1. **CLICK** File
   2. **SELECT** 'Info",
   3. **CLICK** 'Copy path' button to copy file path.



4. **PASTE** file path into from Web Url.

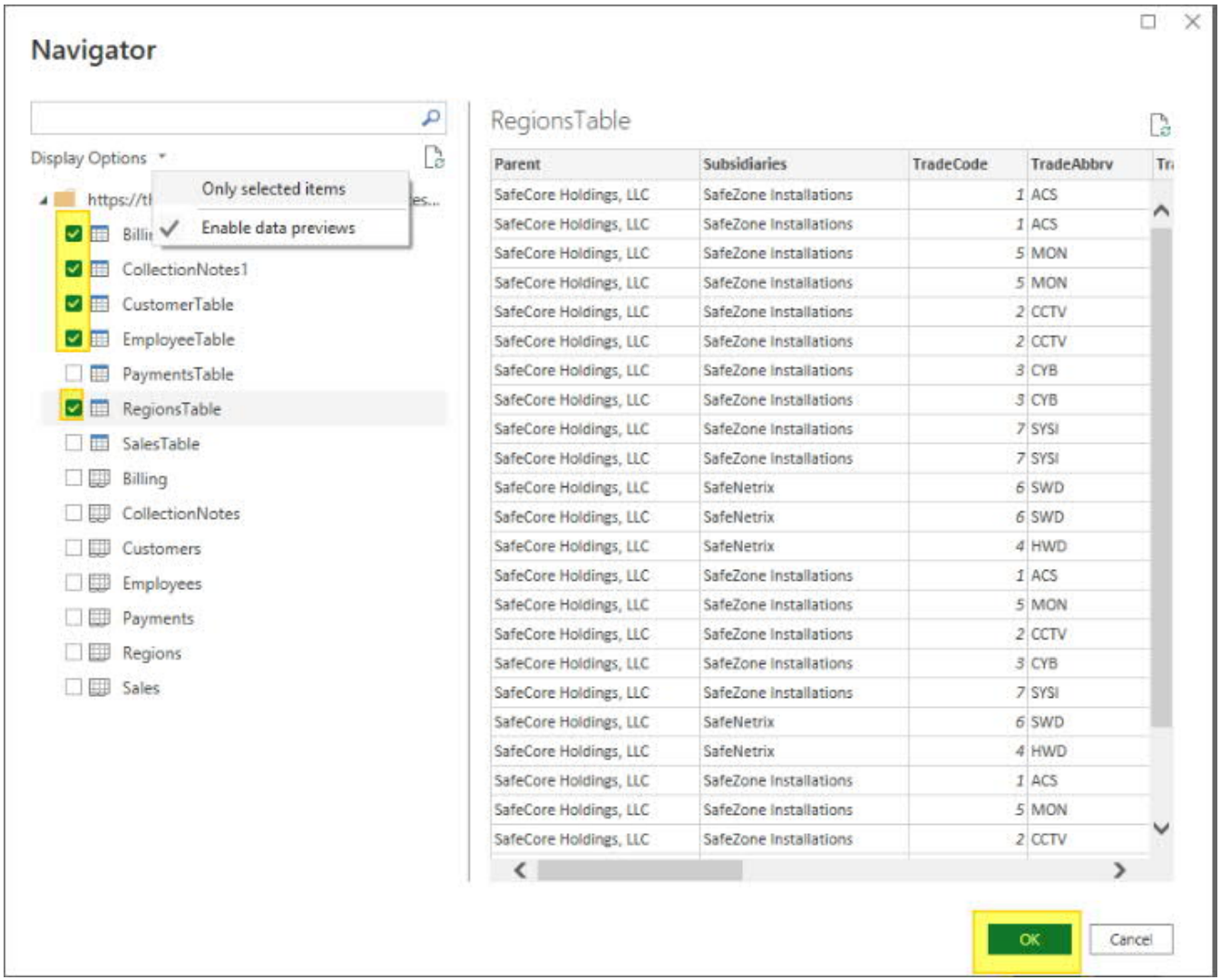


5. **DELETE** everything after .xlsx. Delete ?web=1

https://thedigitalbrewery.sharepoint.com/sites/CreditDepartment/Shared%20Documents/Events/NACM%20Credit%20Congress%202025/Session%202%20Power%20Query%20Proficiency/SafeCore%20Sample%20Data.xlsx?web=1

6. **CLICK** Ok

SELECT Tables to Load



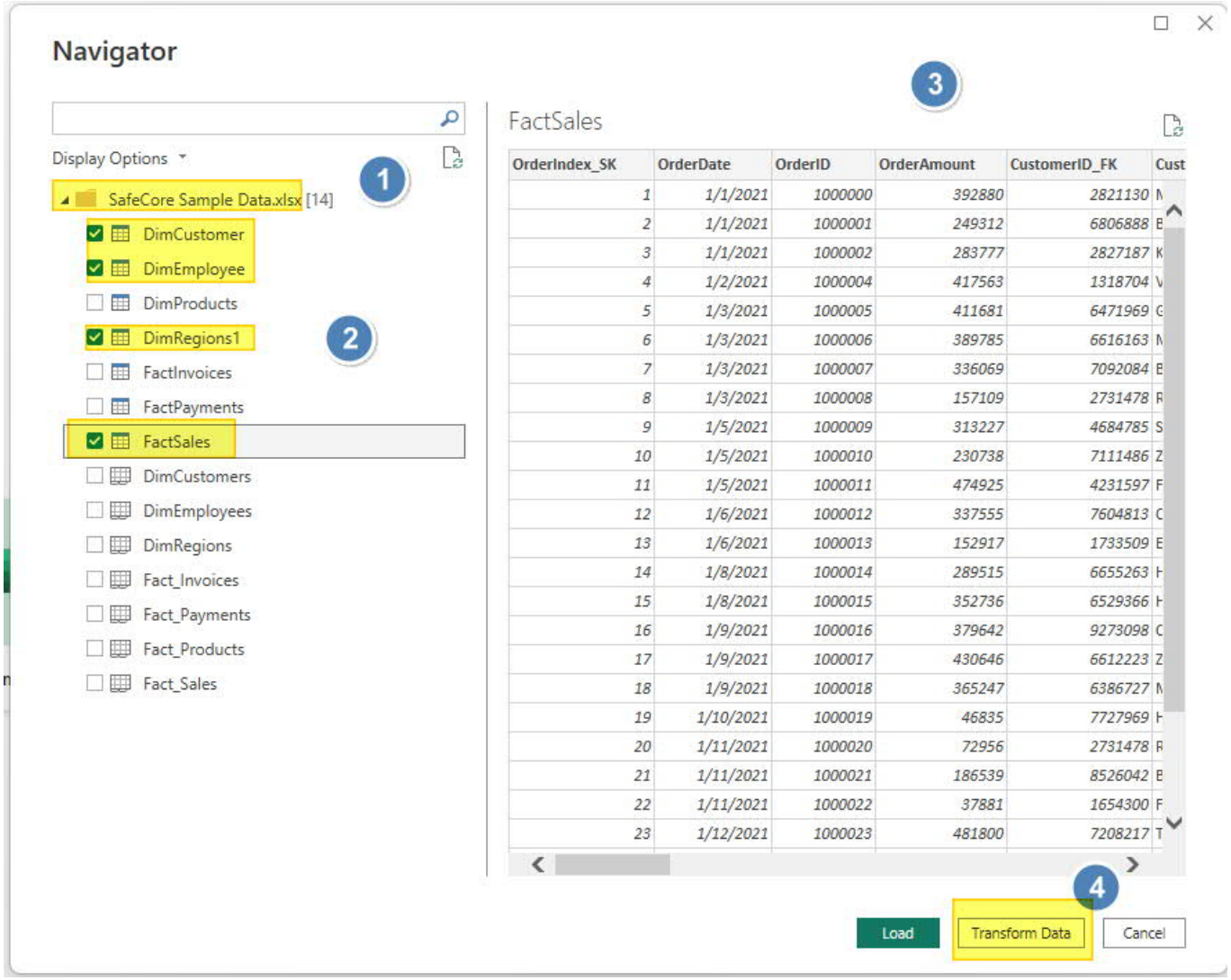
CLICK Ok

1. OPEN Example File:
   SafeCore Sample Date.xlsx

2. SELECT Tables to import:
   - DimCustomer
   - DimEmployee
   - DimRegions
   - FactSales

3. PREVIEW Sample Data

4. CLICK Transform Data to open Power Query



Step 1

Create Group Folders to organize your data sources:

- Datasources
- Staging Table (Optional)
- DataModel

Step 2

- Move your date source files to your Datasource folder
- Disable load for each

Step 3

- Reference each of your tables to create copies. (Right Click choose reference)
- Move to your DataModel folder
- Rename for user friendliness.

Step 4

- Set your data types.
- Rename your Columns to make them user friendly in natural language.
- Remove Extra Columns.
- Perform additional transformations as needed.

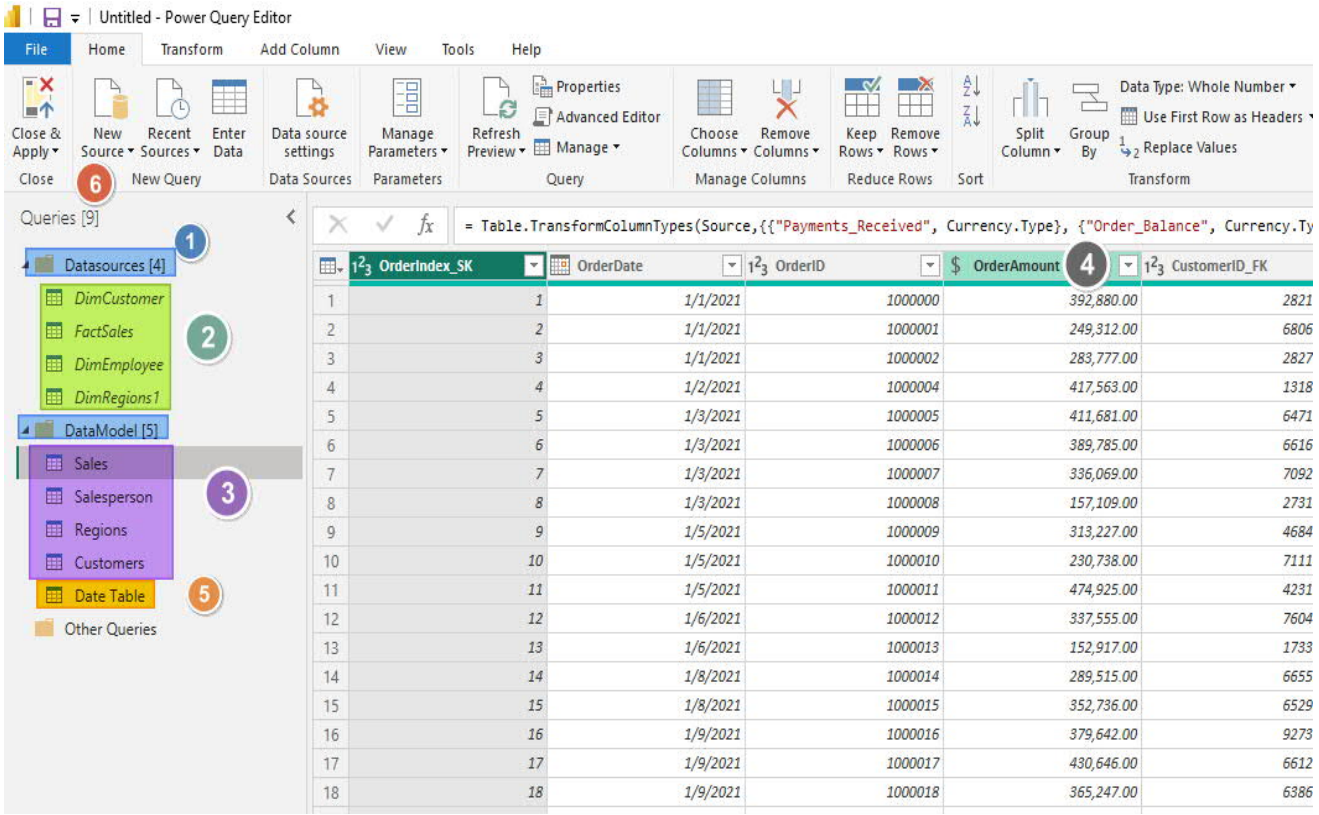
Step 5

- Create Your Date Table

(Use .txt file/Select Blank Query/Advanced Editor)

Step 6

- Click Close and Apply



2. Rename Data Source Files

Renaming your imported files in Power Query with prefixes like src *and later creating references renamed with stg* follows a well-structured, professional data modeling practice rooted in the principles of layered query architecture. This brings multiple benefits, particularly when working on scalable, maintainable, and collaborative Power BI or Excel data models.

Renaming helps with clarity of purpose and source. It becomes instantly obvious to you and others what a query is doing and what the source is just by looking at its name.
ie.
src_xls for Excel data pulls
src_web for Websites data pulls
src_sql for SQL data pulls
src_ERP for ERP data pulls

There is no rulebook for naming sequence.

- *src_Excel_DimRegionsTable*
- *src_Excel_Collection Notes*
- *src_Web_DimCustomerTable*
- *src_ERP_DimEmployee Table*
- *src_DW_FactBillingTransactions*

Table	New Name
Regions	src_xls_DimRegionsTable
Collection Notes	src_xls_DimCollections
Customers	src_xls_DimCustomerTable
Employee	src_xls_DimEmployeeTable
Billing Transactions	src_xls_FactBillingTransactions

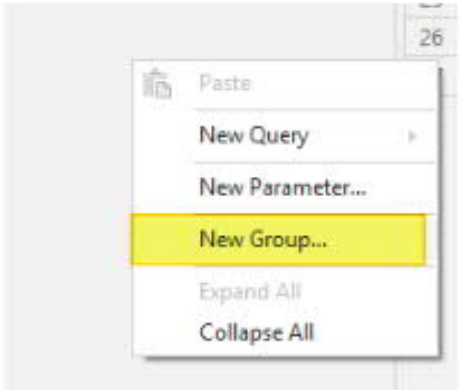
3. Create Groups to Organize your Queries by Folder

Groups can keep your queries organized as your tables increase and grow, making it easier to trace stages and solve problems. A few optional name ideas are:

- Data Sources
- Staging Tables
- Data Model
- Parameters

To Create a New Group:

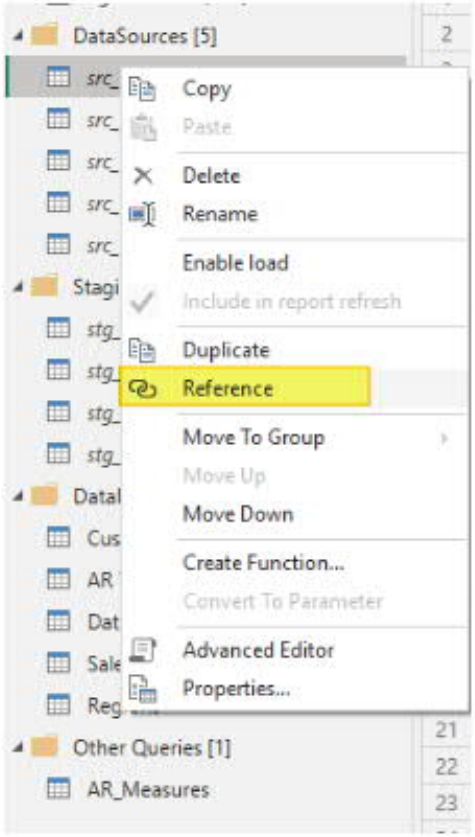
1. **RIGHT CLICK** on Queries field
2. **SELECT** 'New Group'
3. **NAME** Your Group



4. Create References for Your Tables before Transforming them.

1.. Right click on each table

14. **SELEC**T Reference to create a copy of the table.



5. Rename Reference Tables as Staging Tables as Needed

Table	New Name
src_xls_DimRegionsTable	stg_DimRegionsTable
src_xls_DimCollections	stg_DimCollections
src_xls_DimCustomerTable	stg_DimCustomerTable
src_xls_DimEmployeeTable	stg_DimEmployeeTable
src_xls_FactBillingTransactions	stg_FactBillingTransactions

Renaming your reference tables with prefix "stg_" for "Staging" helps separate your queries by layers and mimics the architecture used in data warehouse.

src_ (Source): This table or query is a direct pull from a raw source (e.g., Excel, CSV, Web, ERP, SQL). It should ideally remain unmodified.

Layer	Prefix	Description
Source	`src_`	Raw data import only. Minimal to no transformation.
Staging	`stg_`	Transformations applied: renaming columns, changing data types, filtering, etc.
Model	`dim_`, `fact_`	Tables used directly in the data model. Clean and optimized for DAX.

It encourages single responsibility per query, making debugging and maintenance easier.

1. **Avoids Breaking Changes**

When you reference a `src_` query and perform your transformations in a new `stg_` query:

- You can **repoint the source** (e.g., swap the Excel file for a SQL source) without affecting downstream logic.
- If a transformation needs to be adjusted, you **don't accidentally overwrite or corrupt** the source logic.

2. **Improves Reusability**

If multiple tables (e.g., fact and dimension tables) need to reference the same data:

- Keep `src_` raw and reusable.
- Use multiple `stg_` queries for different transformation pathways.

3. **Team Collaboration & Documentation**

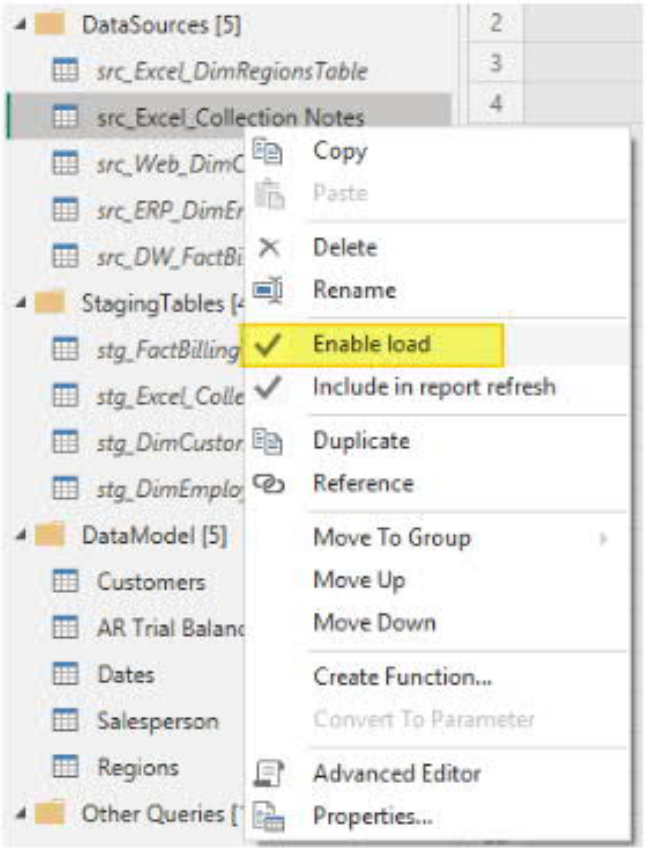
- Naming conventions provide a **shared language** for your team.
- Helps new developers or analysts understand data flow without needing deep walkthroughs.
- Aligns with standard ETL practices from tools like SSIS, Data Factory

4. **Aids in Performance Optimization**

- You can **disable load** for `src_` queries, and load only the `stg_` and final model tables.
- Reduces memory footprint and refresh time.

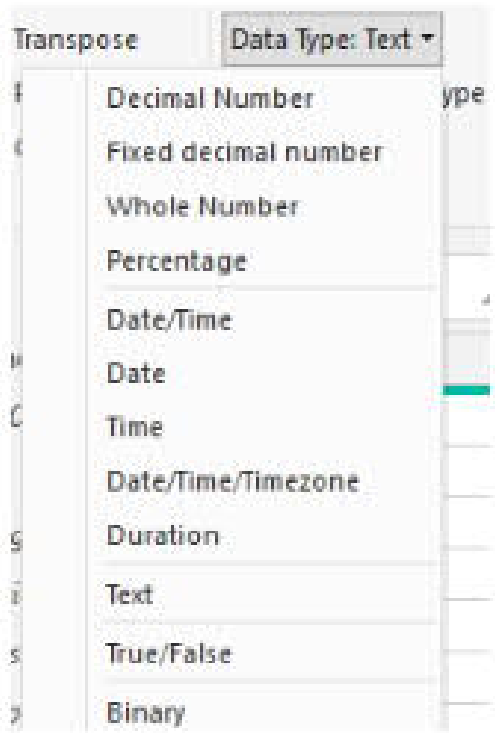
6. Disable Load of Data Source and Staging Tables.

1. **RIGHT CLICK** on Queries
2. **UNCHECK** 'Enable Load'



7. Format the Data Types on each of the Staging Tables

1. **SELECT** Columns that show data type errors
2. **SELECT** the correct Data Type under menu 'Transform' and 'Data Type'



8. Watch for Data Errors

DATA FORMATTING TIPS:

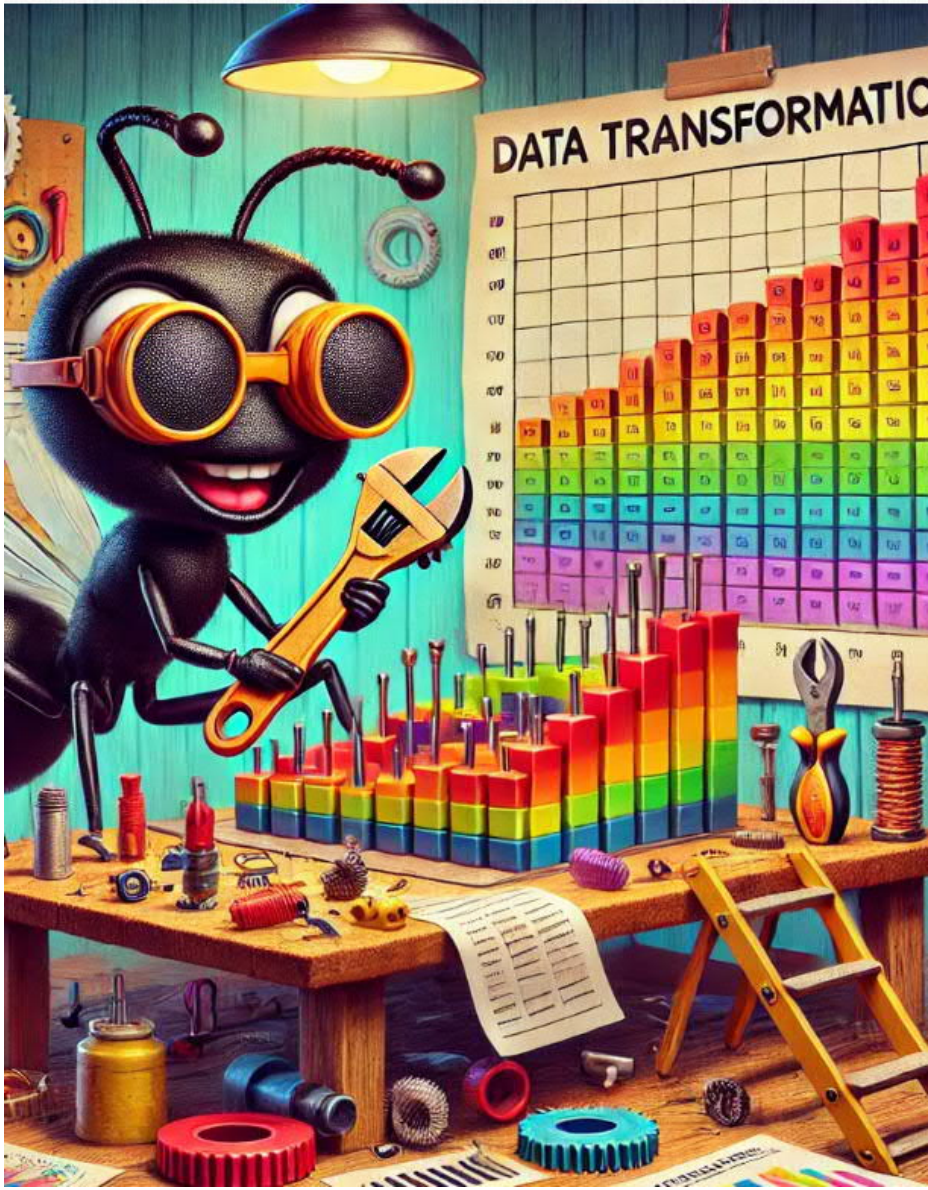
1. **SELECT** 'Fixed decimal number' for currency and other decimal numbers instead of "Decimal Number. This is optimization that rounds to 2 decimals and can reduce data siz
2. **REDUCE** Date & Time to Just Dates column if possible. Separate Date and Time by splitti columns if necessary for data optimization.
3. **REVIEW** for Errors.
    1. **SELECT** View menu
    2. **CLICK** 'Column profile', - Understand column characteristics and outliers. Provides detailed stats: min, max, count, average, text length, etc
    3. **CLICK** 'Column distribution' - Detect duplicates, data skew, or unexpected categorie Shows frequency distribution, distinct and unique counts.
    -**Distinct** - The count of all **different values**, regardless of how many times they occur. For values: `A, A, B, C` → **3** (`A`, `B`, `C`) Ask yourself: "How many types?"
    Distinct tells you how many different customers you're dealing with.
    - **Unique -** The count of values that appear **only once** in the dataset. For values: `A, B, C` → **2** (`B`, `C`) Ask yourself: "How many appear just once?"
    Unique helps spot **data anomalies**, duplicates, or entries that need validation — for example, if only one invoice number is used multiple times.
    4. **CLICK** Column quality - Quickly identify missing, invalid, or problematic data. Displa % of Valid, Error, and Empty values for each column
    5. **CLICK** Monospaced" - Easier to visually detect alignment and spacing problems. Displays text in a fixed-width font
    6. **CLICK** 'Show Whitespace' - Prevents join/match errors due to unseen text issues. Highlights invisible spaces (leading, trailing, internal) Use Trim & Clean.

Cleaning Data

1. Promoting Headers
2. Changing data types for consistency.
3. Removing Bottom Rows
4. Removing Columns
5. Removing Duplicate
6. Filtering blank rows and errors
7. Handling null values: Replace or remove
8. Replacing Values
9. Trimming, Extracting & Parsing
10. Reduce Data – Filtering Rows
11. Sorting Data

Shaping Data

1. Splitting and merging columns
    1. Splitting delimited text (e.g., names, addresses).
    2. Combining multiple columns into one.
2. Pivoting and unpivoting:
    1. Pivoting for summary views.
    2. Unpivoting for better analytics and modeling.
3. Duplicating Columns
4. Filtering rows based on conditions.
5. Adding Index Columns
6. Adding Conditional Columns
7. Adding Date Columns
8. Adding Columns by Example
9. Grouping Data for Aggregations
10. Creating Lists



Introduction to PowerQuery M Code

M Code, also known as the Power Query Formula Language, is a functional, case-sensitive language used in Power Query for data transformation within Microsoft Power BI, Excel, and other Microsoft products.

It is a mashup language to create queries that combine data similarto F# programming language.

M enables users to write custom scripts to import, modify, clean, and manipulate data efficiently by creating query steps that form a data transformation pipeline.

```
let
    grouped = Table.Group(
    Table.FromRecords({
        [CustomerID = 1, price = 20],
        [CustomerID = 2, price = 10],
        [CustomerID = 2, price = 20],
        [CustomerID = 1, price = 10],
        [CustomerID = 3, price = 20],
        [CustomerID = 3, price = 5]
    }), // Records
    "CustomerID",
    {"total", each List.Sum([price])} // aggreg
)
in
    grouped
```

M-Language Examples	
Today's Date	= Date.From(DateTime.LocalNow())
Date Add	= Date.AddDays([Document Date], 30)
IF Statements	= if 2 > 1 then 2 else 1 = if 1 = 1 then "yes" else "no"
Nested IF Statements	= if logical_test1 then value1_if_true else if logical_test2 then value2_if_true else value2_if_false
Complex IF Statements	= if logical_test1 then if logical_test2 then value2_if_true else value2_if_false else if logical_test3 then value3_if_true else if logical_test4 then value4_if_true else value4_if_false
AND Logic	= if logical_test1 and logical_test2 then value_if_true else value_if_false
OR Logic	= if logical_test1 or logical_test2 then value_if_true else value_if_false
Example: M Custom Column	= (([Amount]*.18)/365)*([Document Age]-35)

Parameters & Functions

A Power Query Function is a predefined or custom operation in Power Query that allows users to manipulate, transform, and cleanse data within Power BI, Excel, or other Microsoft data tools. Power Query functions are written using the M language, which is the scripting language behind Power Query.

Two Types

- Built In Functions
- Custom Functions

Purpose of Functions

Data Transformation - perform repetitive data transformations, such as filtering rows, changing data types.

Reusable Logic - Functions can be defined once and reused across different queries, promoting code reusability and consistency in data transformation processes.

Error Handling - Power Query functions include error-handling capabilities, allowing users to catch and manage errors gracefully during data transformation processes, which is crucial for ensuring data quality.

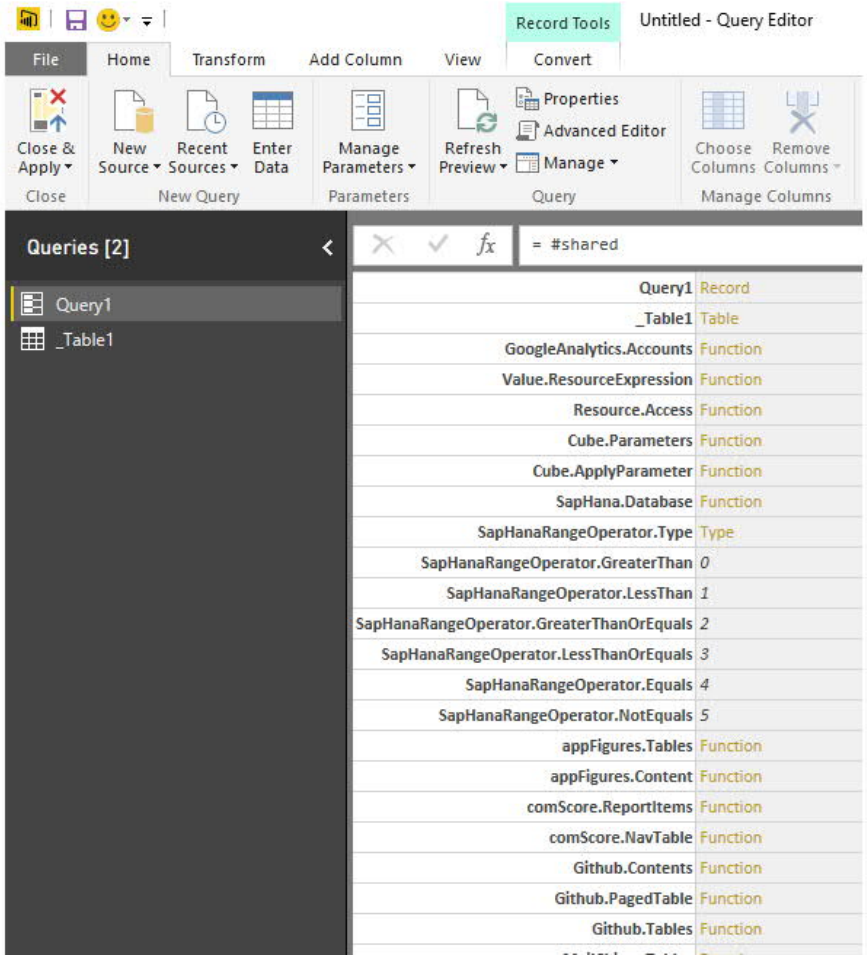
Functions

Built-in Functions - Power Query provides a wide range of built-in functions, categorized into different groups like text, date, time, number, table, and list functions. These functions help users carry out common tasks like text manipulation, date calculations, and numerical computations.

Examples of Built-in Functions

- Text.Upper(): Converts text to uppercase.
- DateTime.LocalNow(): Returns the current date and time.

Functions – Power Query Library of Functions



9. Create your final Data Model Tables by Referencing & Renaming

- Customers
- AR Trial Balance
- Salesperson
- Regions

Ideally, if you've staged your data properly, the Data Model group should contain clean, optimized, and report-ready tables.
This means:

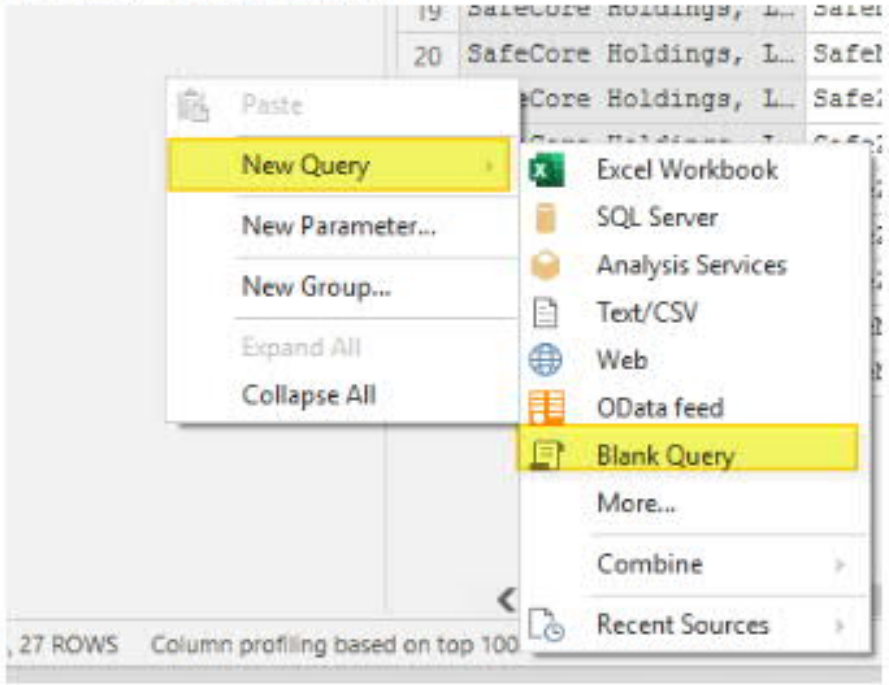
- Columns are renamed user-friendly (e.g., CustomerName instead of cust_nm)
- Unnecessary columns are removed (for size and clarity)
- Key fields are preserved for relationships
- Dimension and fact tables are shaped according to a star schema
- Data types are finalized
- Row-level filtering is done (e.g., removing future dates)

Changes You Might Still Make in Data Model Group
While the goal is minimal change, here are a few valid adjustments:

- **Add a calculated column** only if you missed something in staging
- **Change data types** that weren't detected properly
- **Rename columns** for clarity after seeing them in visuals
- **Filter rows** based on a reporting requirement that arises later (e.g., remove test customers)

10. Create Your Date Table

1. **SELECT** New Data Source from Menu
2. **SELECT** Blank Query
3. **SELECT** Home in menu
4. **OPEN** Advanced Editor
5. **COPY** and **PASTE** Calendar Code from file



Paste M Code in Advanced Editor

```
let
Source = List.Dates,
#"Invoked FunctionSource" = Source(#date(2015, 1, 1), Duration.Days(DateTime.Date(DateTime.FixedLocalNow()) - #date(2015,1,1)), #duration(1, 0, 0, 0)),
#"Table from List" = Table.FromList(#"Invoked FunctionSource", Splitter.SplitByNothing(), null, null, ExtraValues.Error),
#"Added Index" = Table.AddIndexColumn(#"Table from List", "Index", 1, 1),
#"Renamed Columns" = Table.RenameColumns(#"Added Index",{{"Column1", "Date"}}),
#"Added Custom" = Table.AddColumn(#"Renamed Columns", "Year", each Date.Year([Date])),
#"Added Custom1" = Table.AddColumn(#"Added Custom", "Month Number", each Date.Month([Date])),
#"Added Custom2" = Table.AddColumn(#"Added Custom1", "Day", each Date.Day([Date])),
#"Added Custom3" = Table.AddColumn(#"Added Custom2", "Day Name", each Date.ToText([Date],"ddd")),
#"Added Custom4" = Table.AddColumn(#"Added Custom3", "Month Name", each Date.ToText([Date],"MMM")),
#"Reordered Columns" = Table.ReorderColumns(#"Added Custom4",{"Date", "Index", "Year", "Month Number", "Month Name", "Day", "Day Name"}),
#"Added Custom5" = Table.AddColumn(#"Reordered Columns", "Quarter Number", each Date.QuarterOfYear([Date])),
#"Duplicated Column" = Table.DuplicateColumn(#"Added Custom5", "Year", "Copy of Year"),
#"Renamed Columns1" = Table.RenameColumns(#"Duplicated Column",{{"Copy of Year", "Short Year"}}),
#"Changed Type" = Table.TransformColumnTypes(#"Renamed Columns1",{{"Short Year", type text}}),
#"Split Column by Position" = Table.SplitColumn(#"Changed Type","Short Year",Splitter.SplitTextByRepeatedLengths(2),{"Short Year.1", "Short Year.2"}),
#"Changed Type1" = Table.TransformColumnTypes(#"Split Column by Position",{{"Short Year.1", Int64.Type}, {"Short Year.2", Int64.Type}}),
#"Removed Columns" = Table.RemoveColumns(#"Changed Type1",{"Short Year.1"}),
#"Renamed Columns2" = Table.RenameColumns(#"Removed Columns",{{"Short Year.2", "Short Year"}}),
#"Added Custom6" = Table.AddColumn(#"Renamed Columns2", "Quarter Year", each Number.ToText([Short Year]) & "Q" & Number.ToText([Quarter Number],"00")),
#"Reordered Columns1" = Table.ReorderColumns(#"Added Custom6",{"Index", "Date", "Day", "Day Name", "Month Number", "Month Name", "Quarter Number", "Quarter Year", "Short
Year", "Year"}),
#"Changed Type2" = Table.TransformColumnTypes(#"Reordered Columns1",{{"Date", type date}, {"Day", Int64.Type}, {"Index", Int64.Type}, {"Month Number", Int64.Type}, {"Quarter
Number", Int64.Type}, {"Month Name", type text}, {"Quarter Year", type text}, {"Year", Int64.Type}})
in
#"Changed Type2"
```

Best Practices for Query Design

1. Optimize query performance
2. Only load the data you need
3. Filter unnecessary rows & columns
4. Keep your steps together to reduce number of steps in queries
5. Watch your steps! Be careful in deleting previous steps
6. Name queries and steps for clarity.
7. Avoid changing column names
8. Safeguard your data sources
9. Avoiding resource-intensive actions (e.g., nested joins).
10. Organize Queries in Groups, Src, Stg, Final/Data Model
11. Documenting transformations using comments in M Code.
12. Keep Exports in Designated Places to Reuse, Refresh by Overwriting
13. Test thoroughly

11. Load Your Tables into Power BI

12. EXTRA CREDIT

1. Add Custom Columns to calculate Sales Tax, %
2. Add Custom Column for Bad Debt WO Date
3. Lien Deadline
4. Add Categories,
5. Add Prime Rate
6. Add Aging Buckets
7. Connect to SharePoint Folder

Connect to SharePoint Folders

Go to a Source File Applied Step

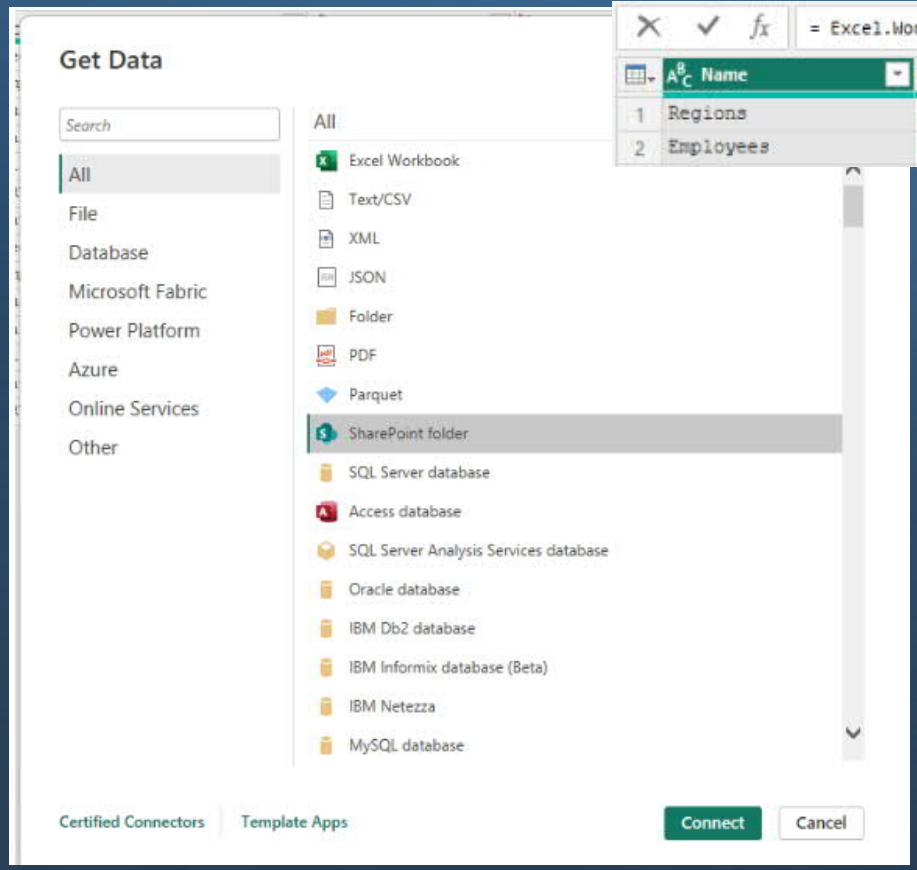
Copy the first part of web address for SharePoint Address

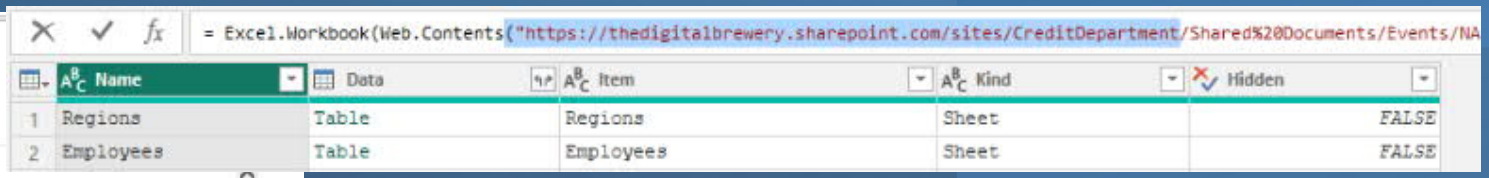
Connect to SharePoint Folders

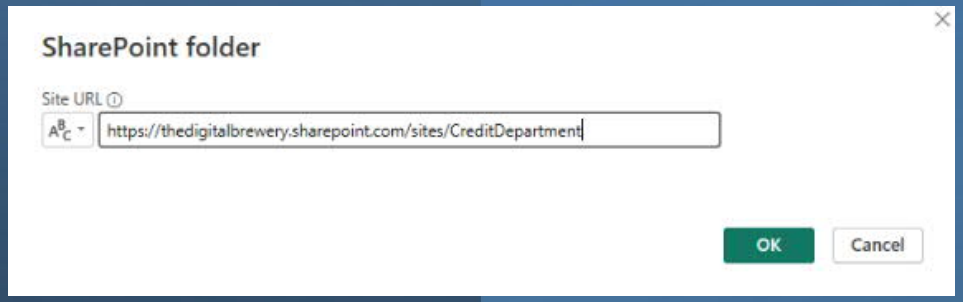
Go to a Source File Applied Step

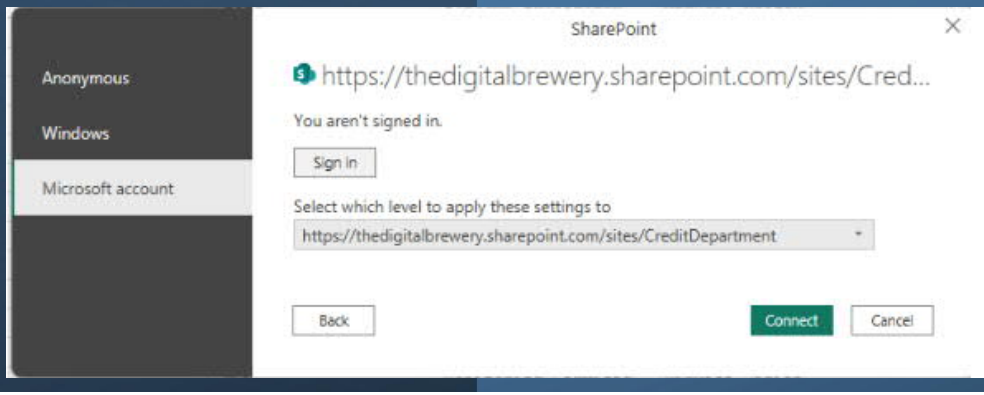
Copy the first part of web address for SharePoint Address

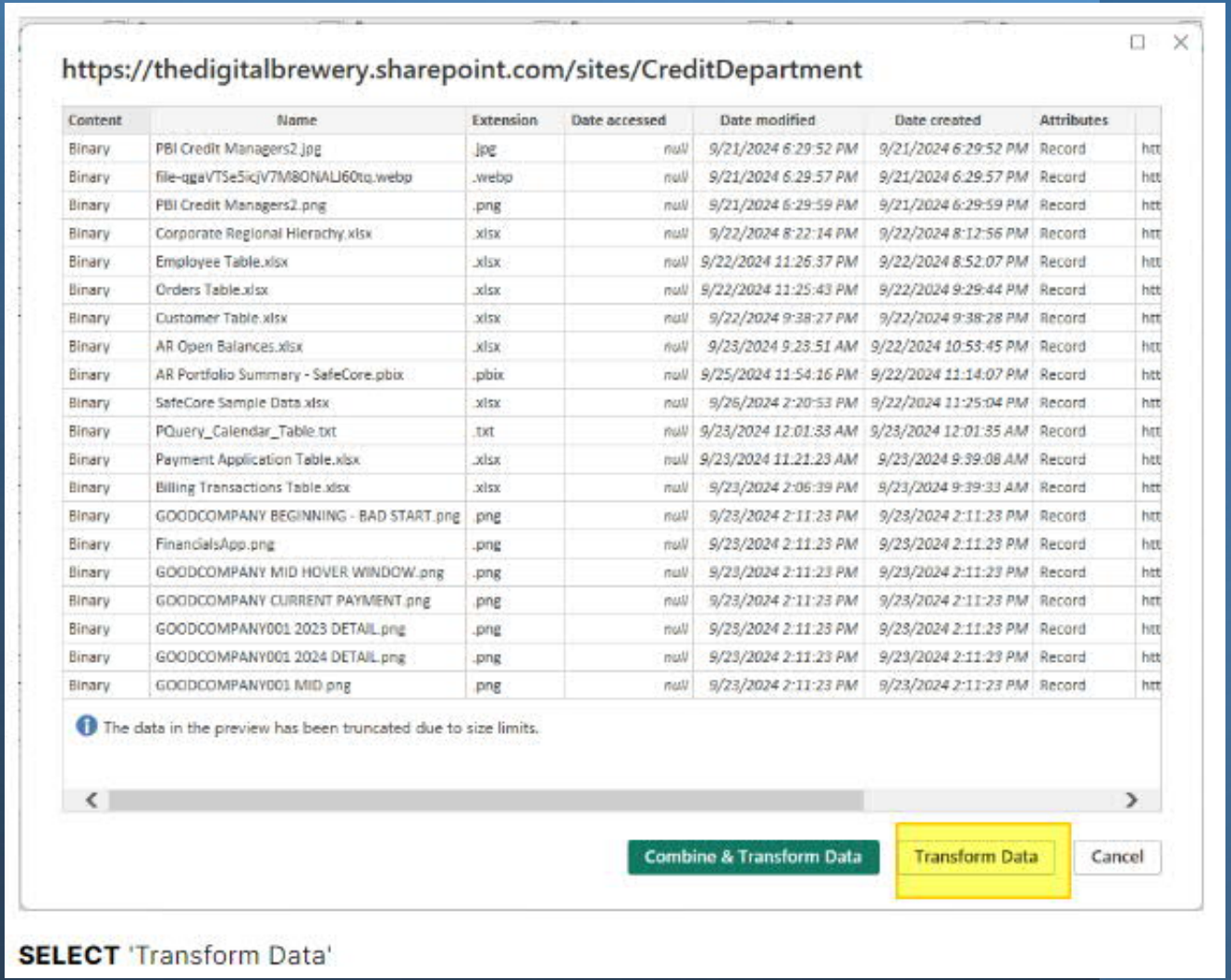
You can only connect to top level folders in SharePoint



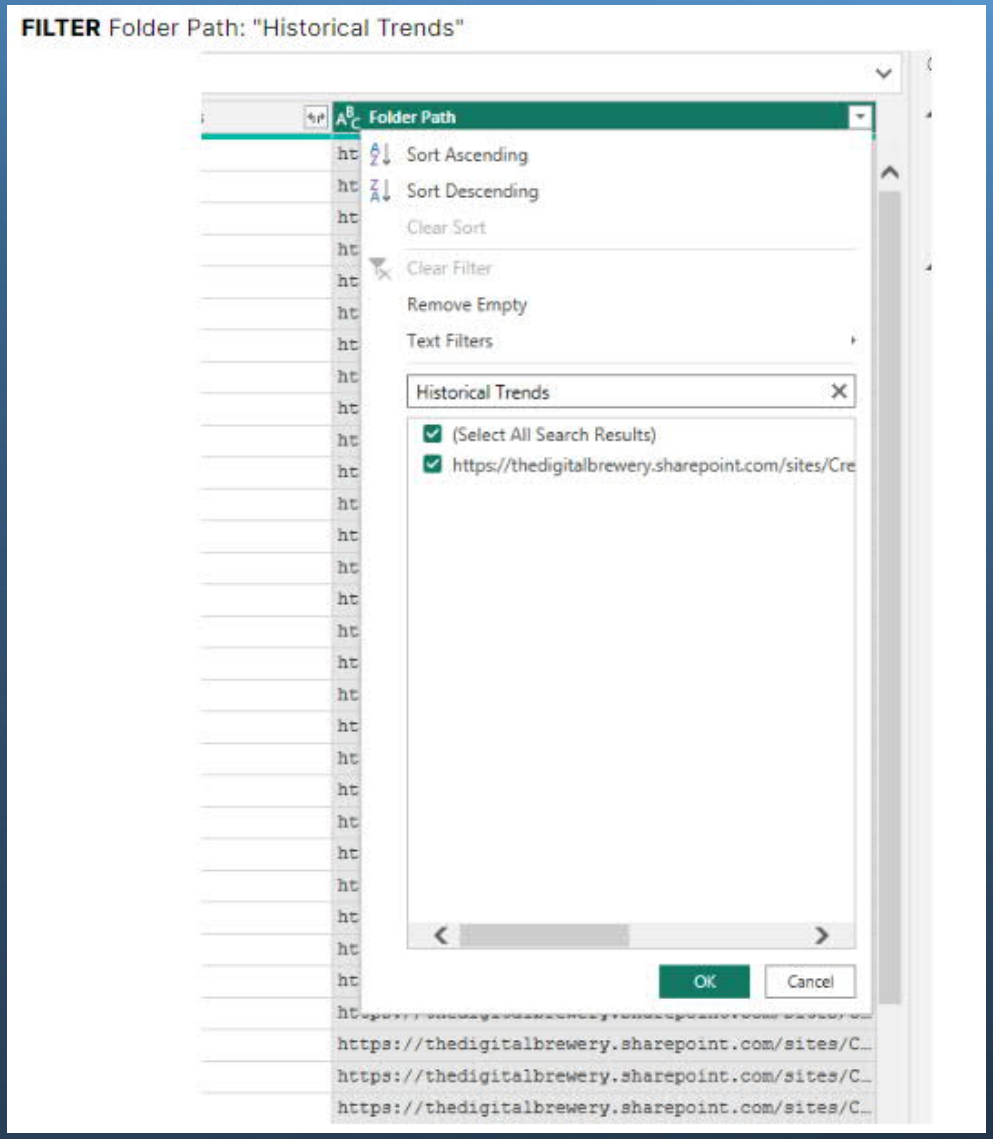








SELECT ‘Transform Data’



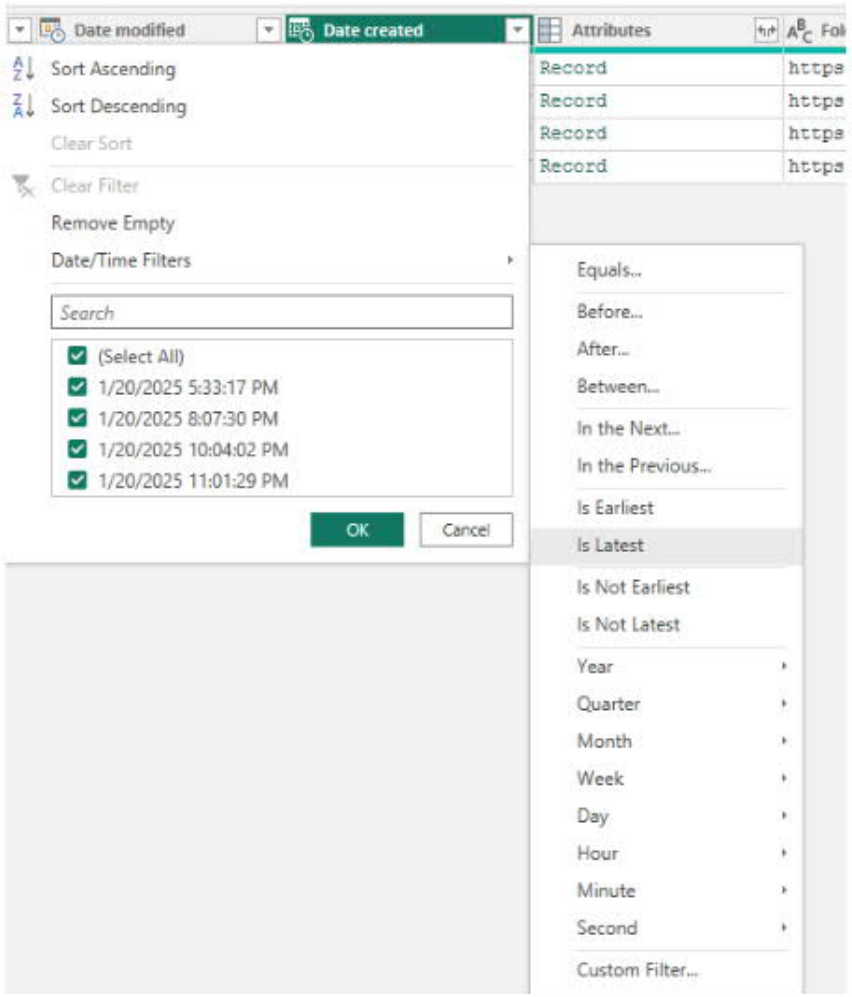
Search for your underlying Folder name.

Name the file folder something unique and identifiable.

Additional Option to Locate: Text Filter to File Contains: 'AR Trial Balance Historical'

FILTER Extension to .xlsx only:
.xlsx or .csv depending on your file type to error proof against a PDF or a wrong file being saved in this location.

FILTER Column 'Date Created' to "Is Latest" to grab only the most recent file (or)



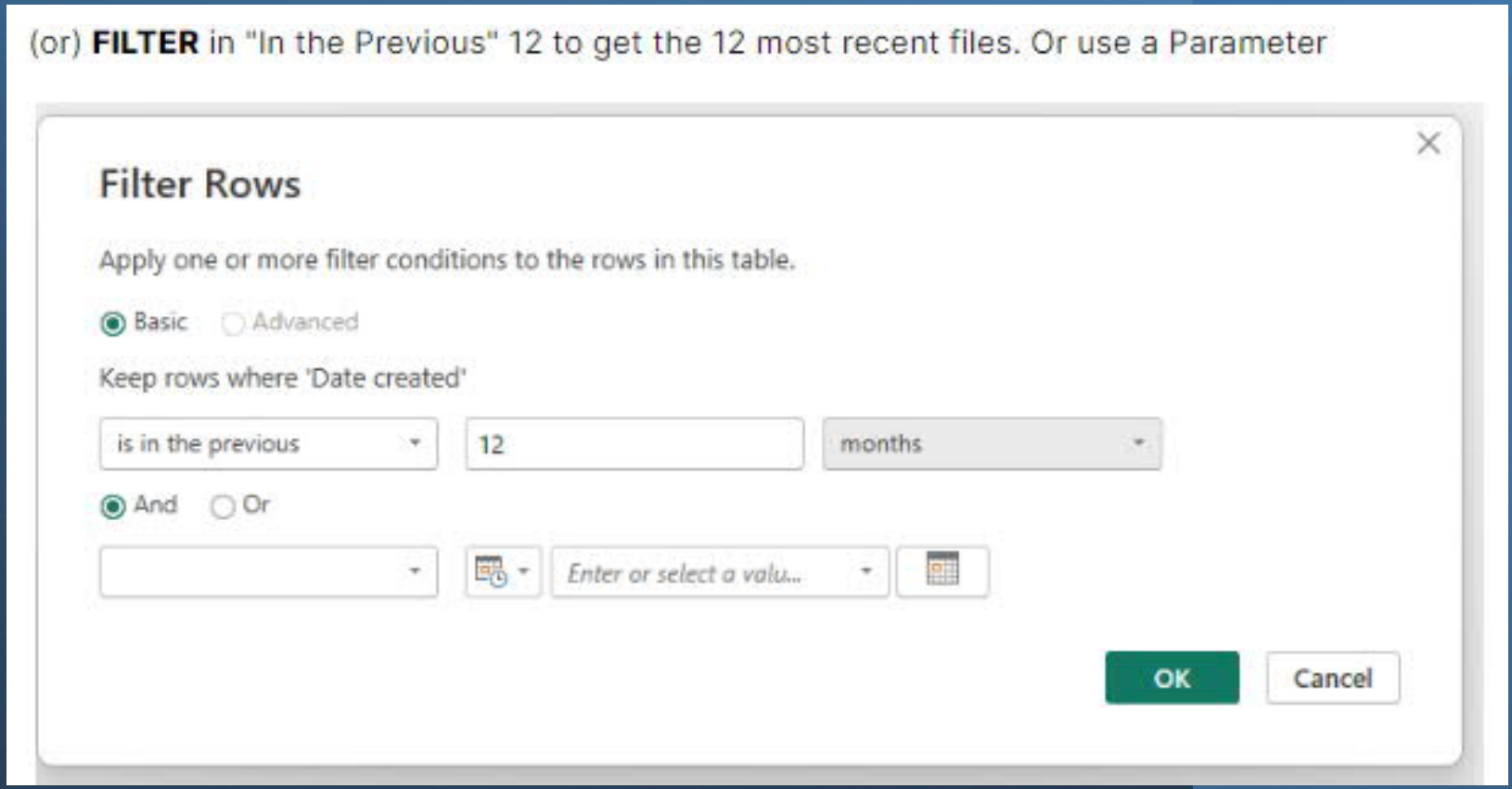
Filter Date Created Column for Latest File if you only want the most recent file pulled in.

Leave to append all files. Headers must be EXACT

Sheet name must not be unique

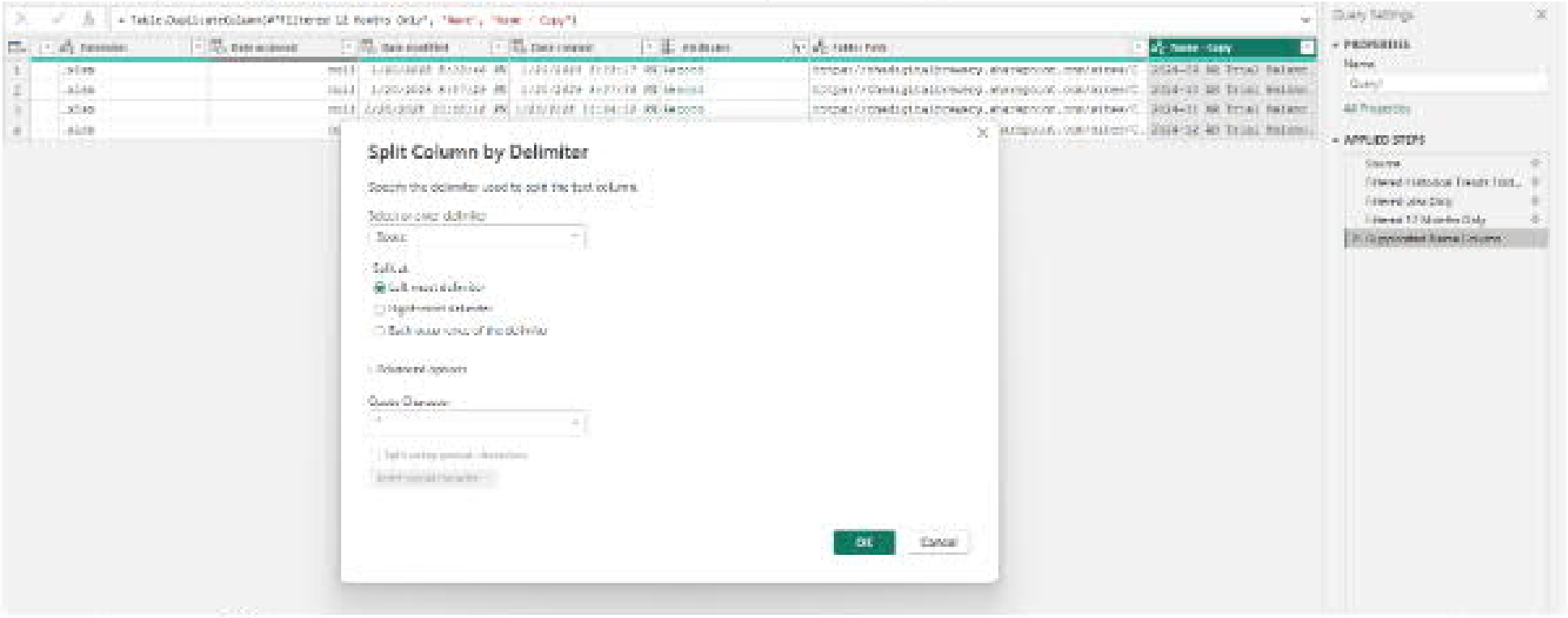
OPTIONAL FILTER Most recent files, or last 12 or 18 months of files to limit data

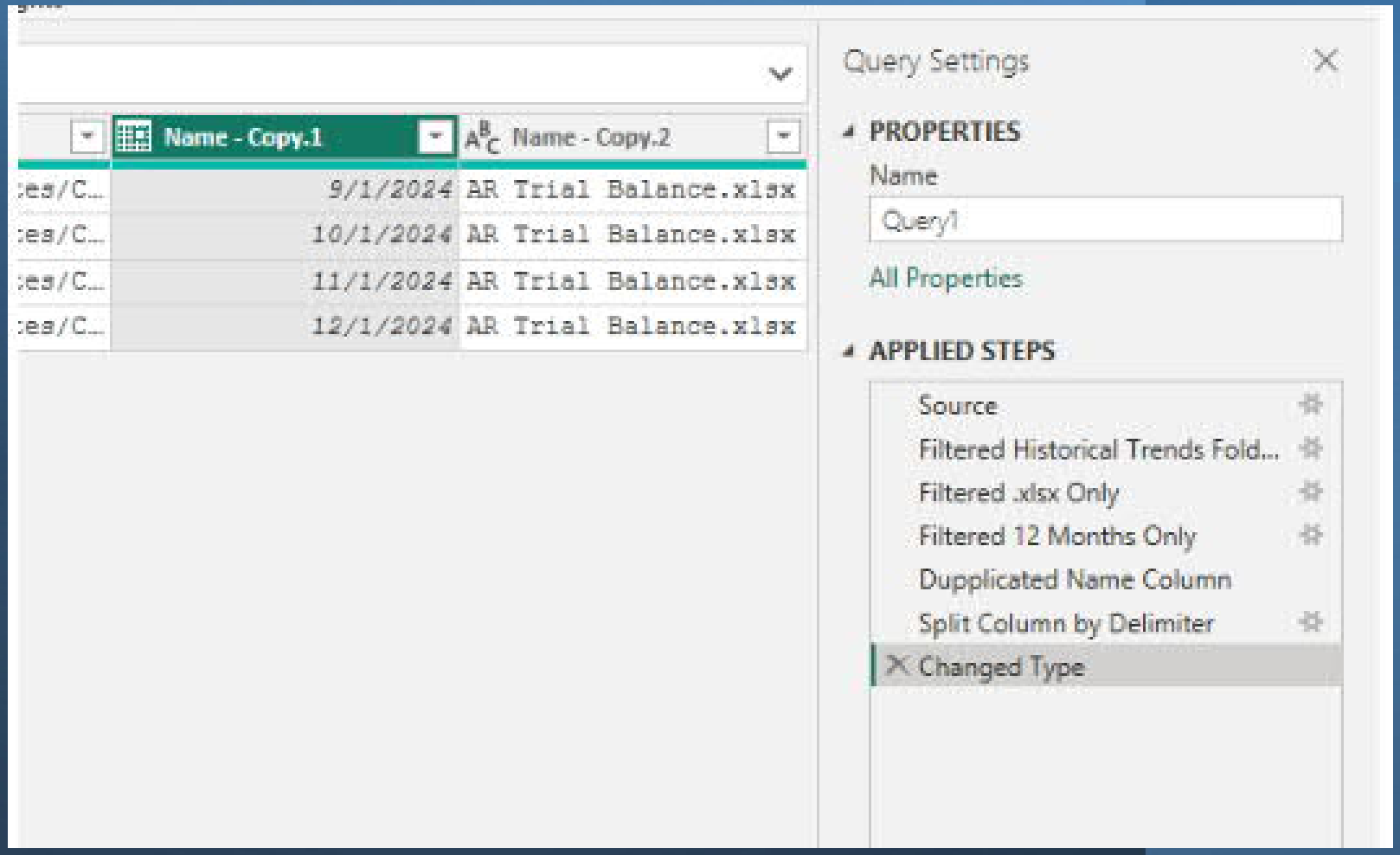
(or) **FILTER** in "In the Previous" 12 to get the 12 most recent files. Or use a Parameter



DUPLICATE File Name Column

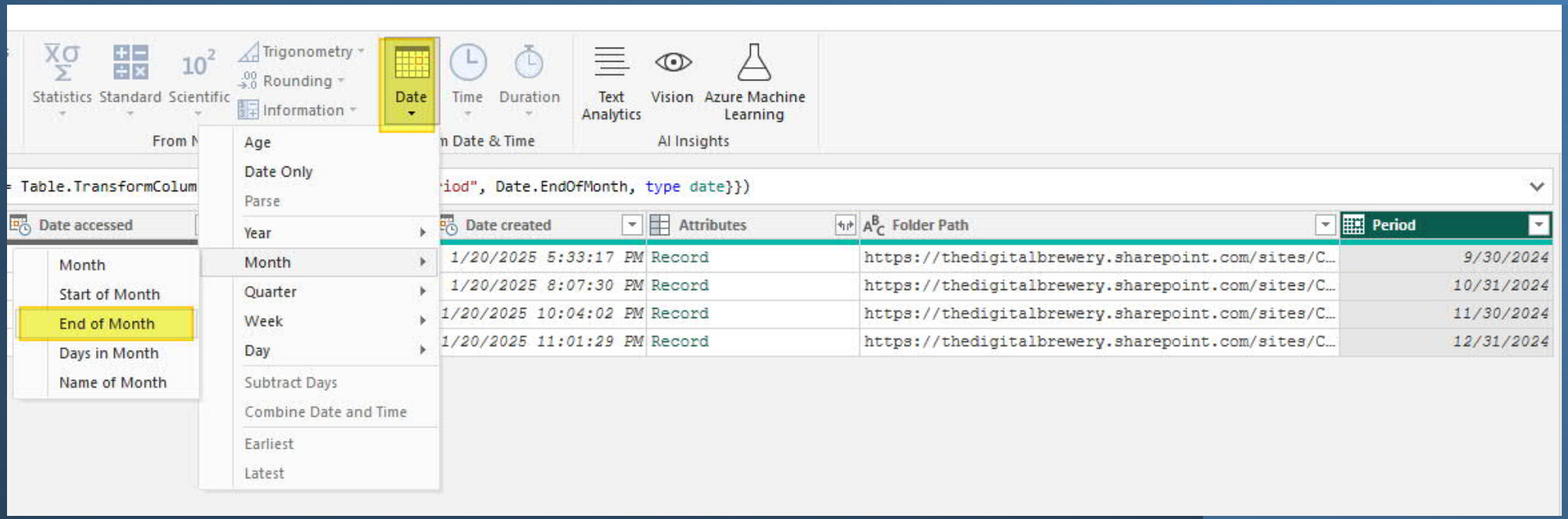
SELECT Split by Delimiter Left-most Center 'Space' to separate Period from File Name



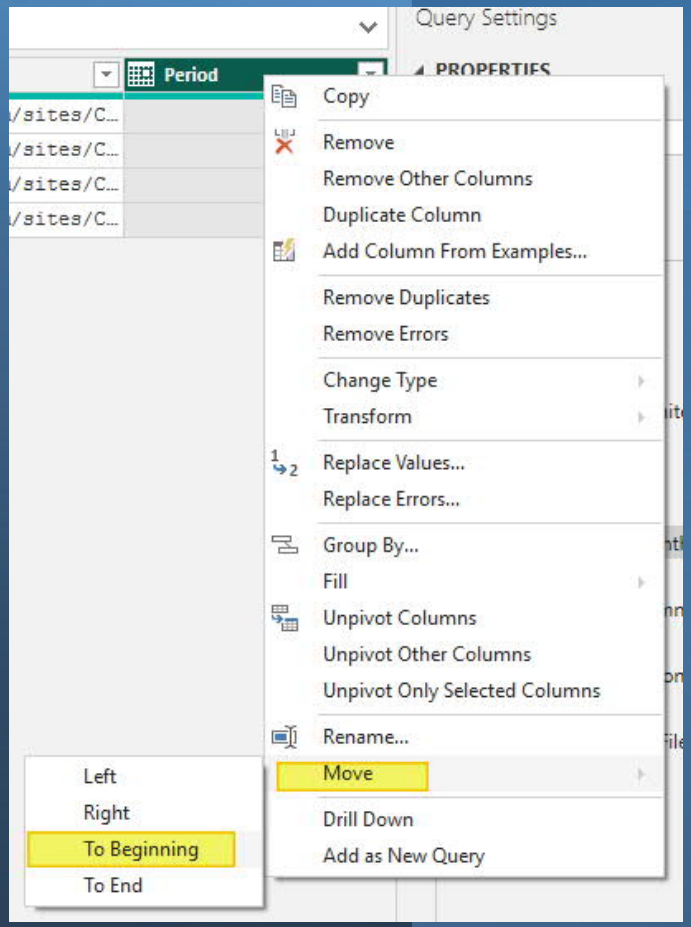

Name - Copy.1
Name - Copy.2
:es/C...
9/1/2024
AR Trial Balance.xlsx
:es/C...
10/1/2024
AR Trial Balance.xlsx
:es/C...
11/1/2024
AR Trial Balance.xlsx
:es/C...
12/1/2024
AR Trial Balance.xlsx
Query Settings
PROPERTIES
Name
Query1
All Properties
APPLIED STEPS
Source
Filtered Historical Trends Fold...
Filtered .xlsx Only
Filtered 12 Months Only
Dupplicated Name Column
Split Column by Delimiter
Changed Type

Select Period Column:

Transform > Date > Month >End of Month to change to Month-End Date

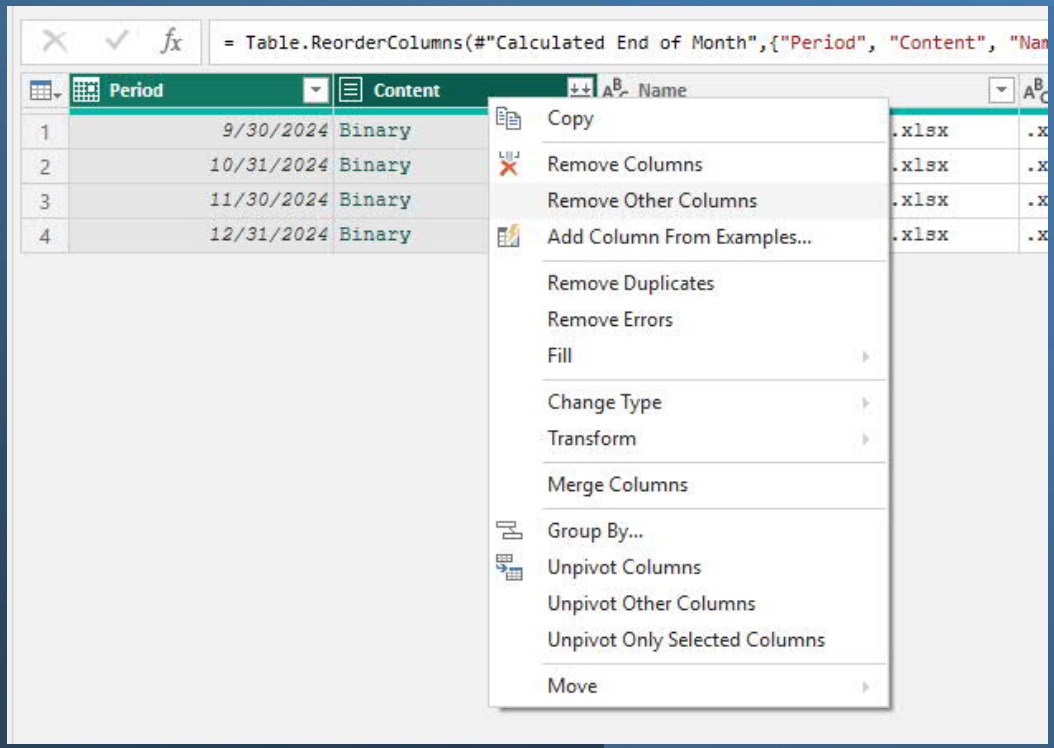


Select Period Column:
Right Click and Select:
‘Move’ and ‘To Beginning’

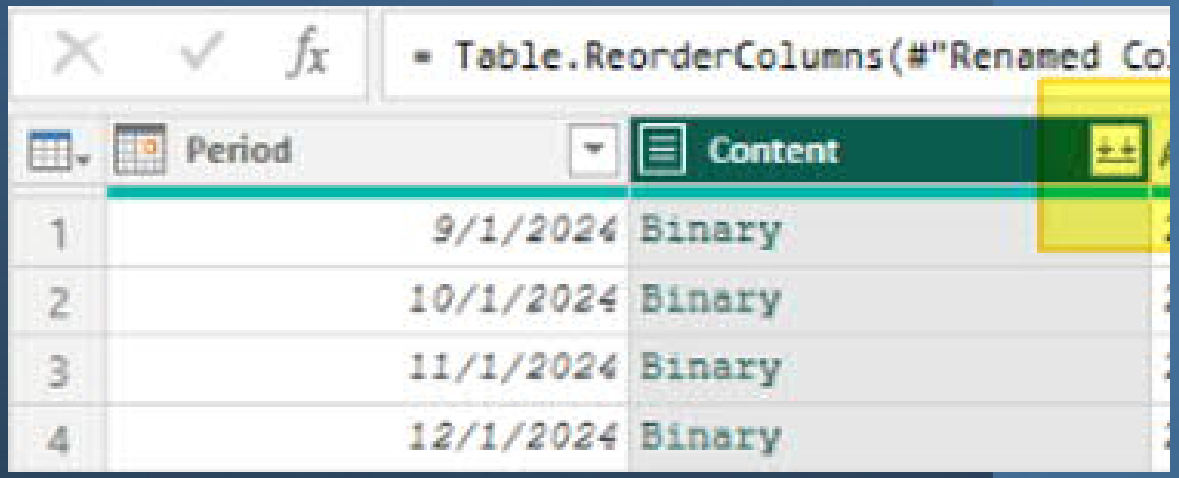


Keep ‘Period’ and ‘Content’

Remove Other Columns



CLICK the Double Down Arrows to Expand Binary Content

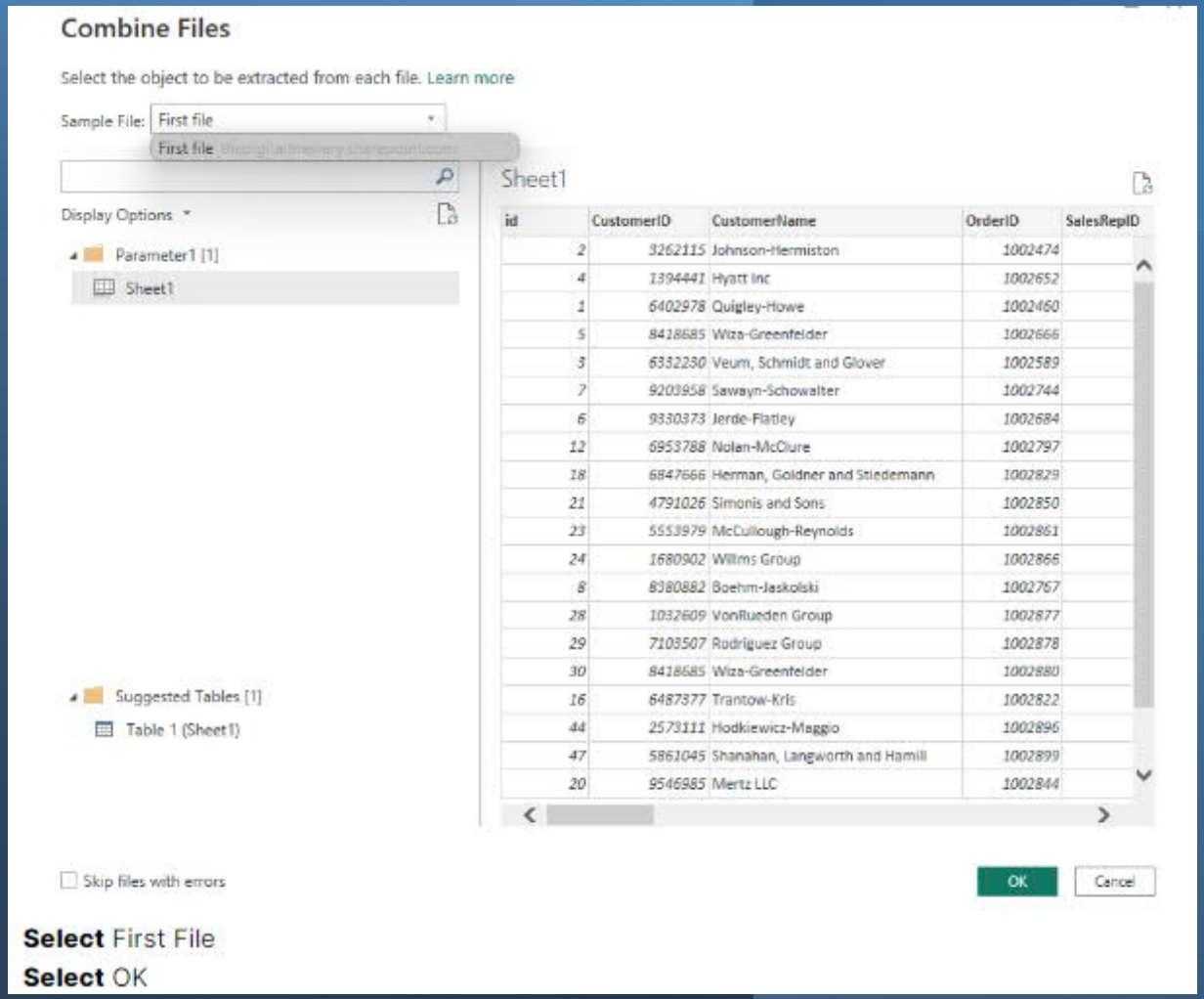


Leave First File as Example:

All Sheets must be Named the same!!

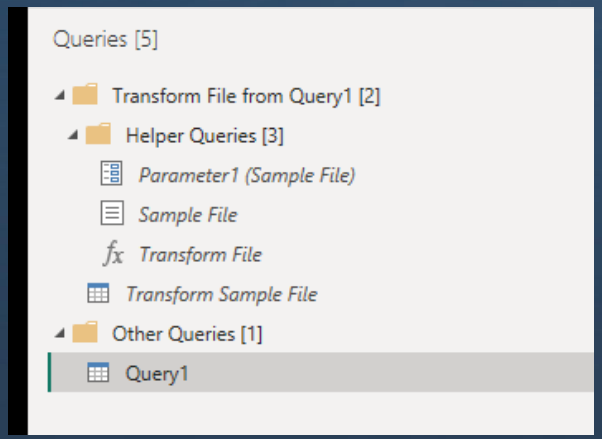
Default Sheet1 simplest

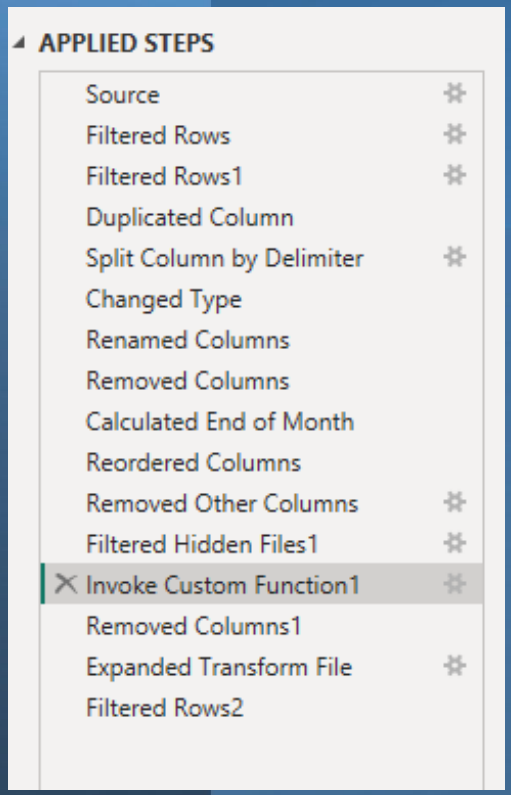
Click OK



Power Query will make the connection and expand the files.

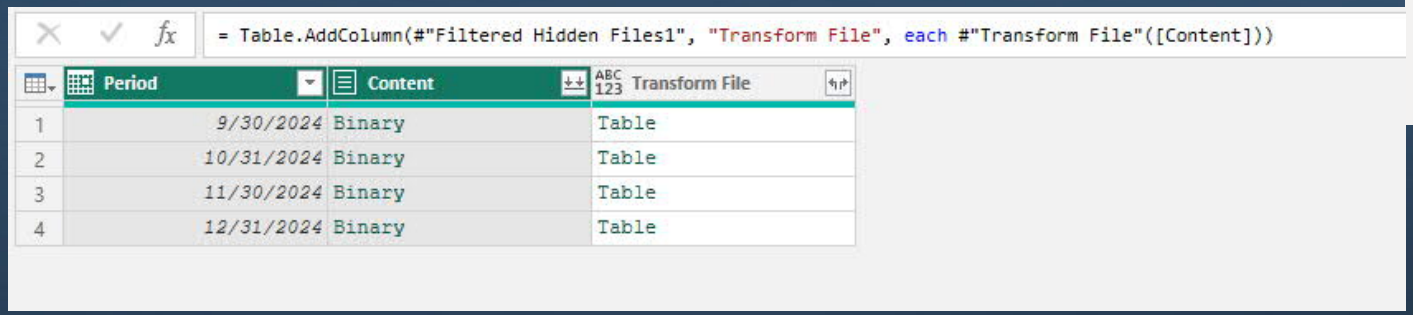
You will see several auto generated Applied Steps on Right and Queries created on left

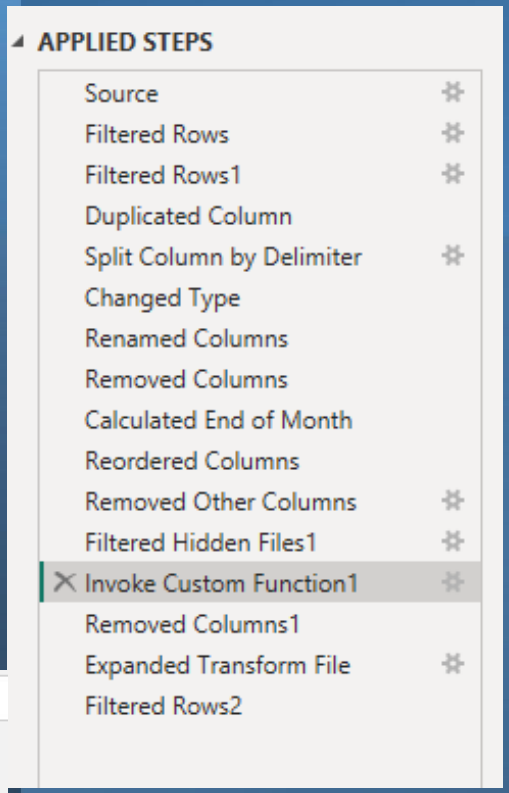




Select Step that says Invoke Custom Function 1 to return to stage where Table was created.

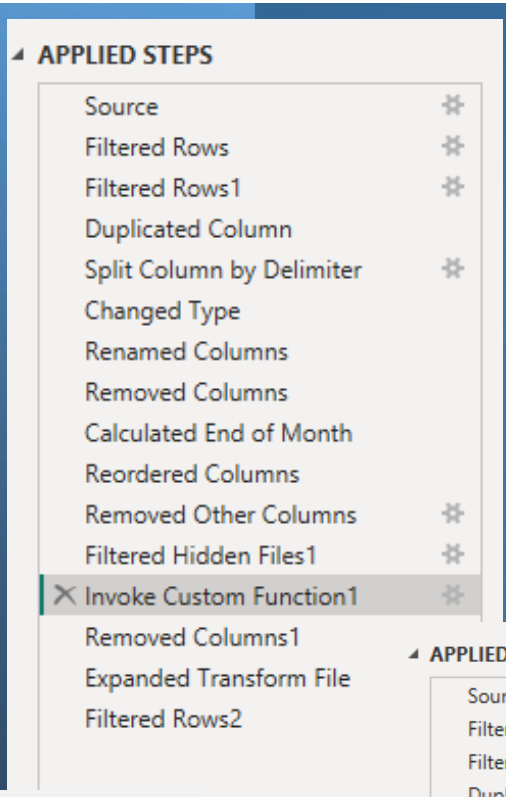
We need to insert our Period Date into Final output.

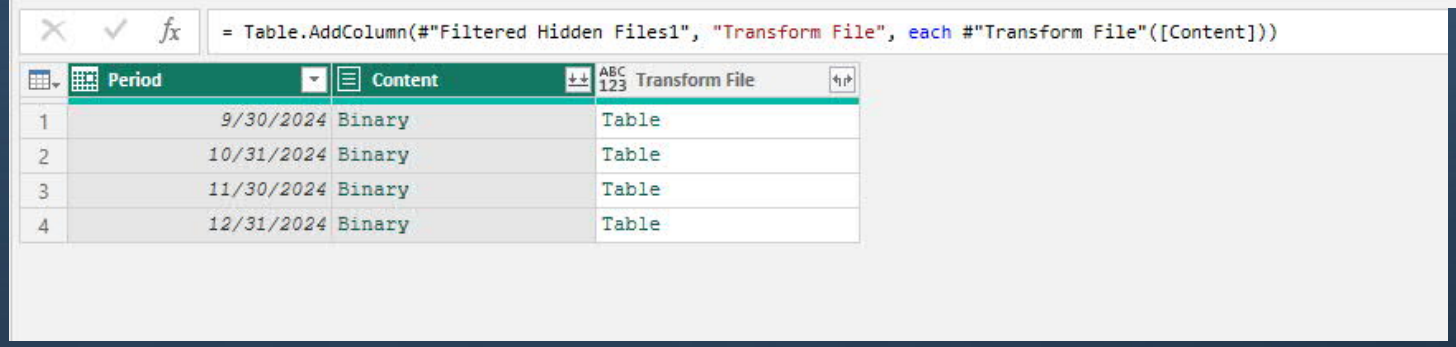


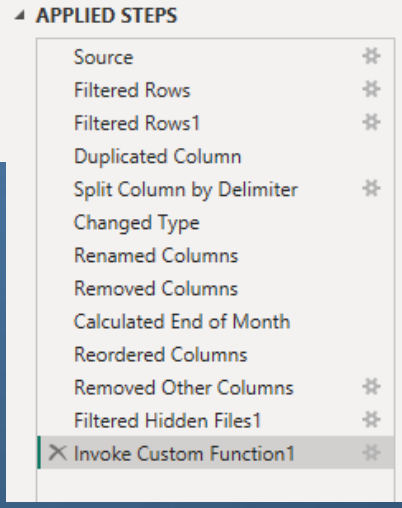


Select Invoke Custom Function1. Delete All Steps after this Applied Step.

Your Table should appear as below:

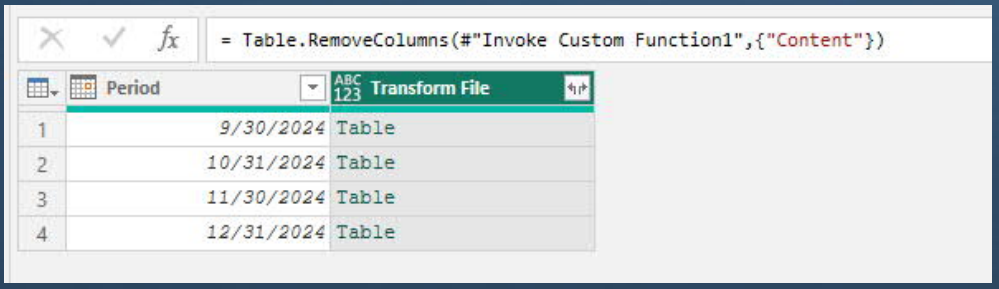


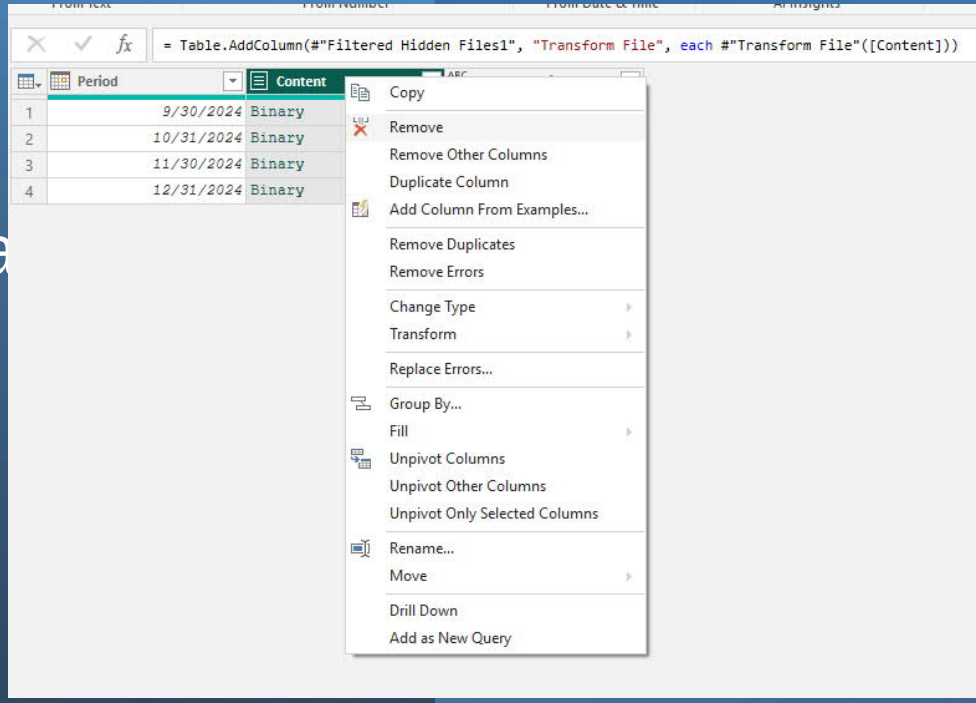




Delete the Content Column that has Binary rows.

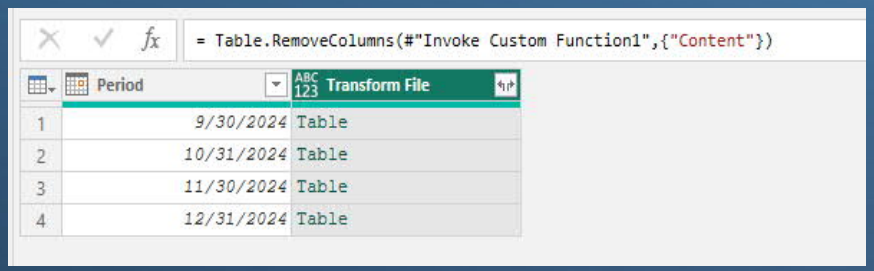
Your remaining table should appear a below:





CLICK the arrows on the ‘Transform File’ with tables as rows to expand

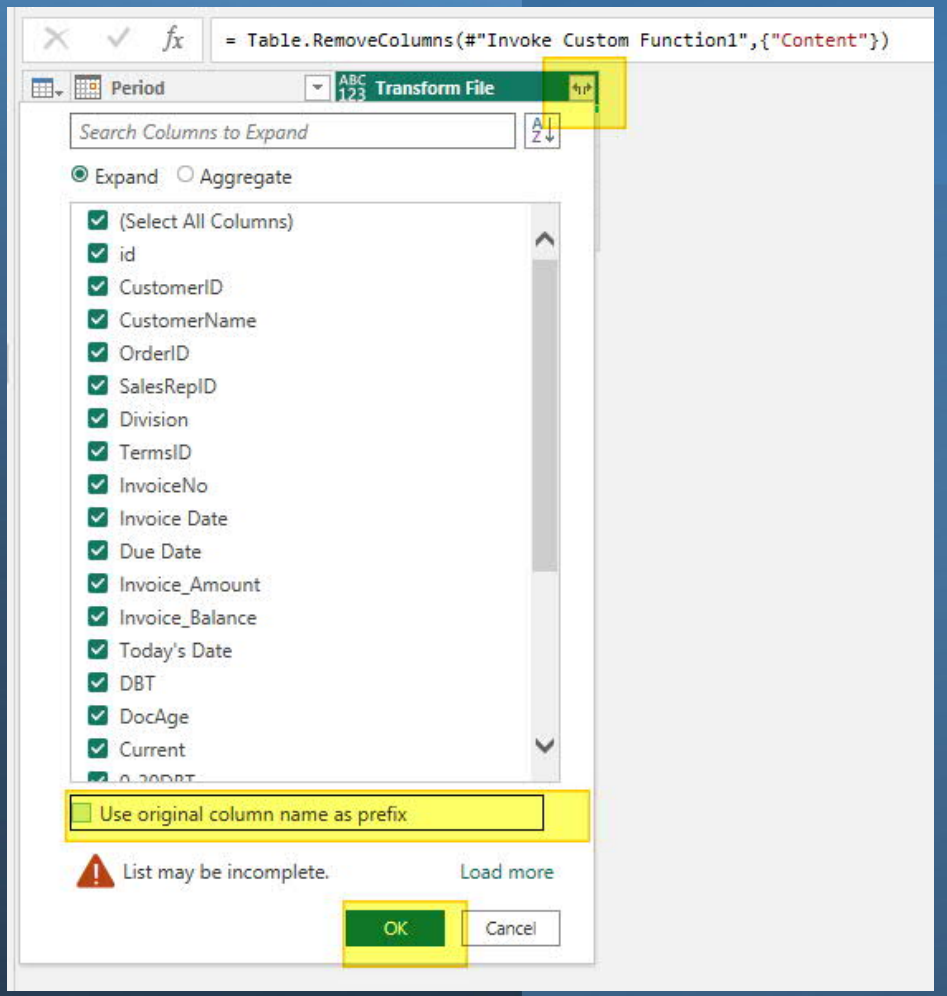
CLICK the arrows on the ‘Transform File’ with tables as rows to expand



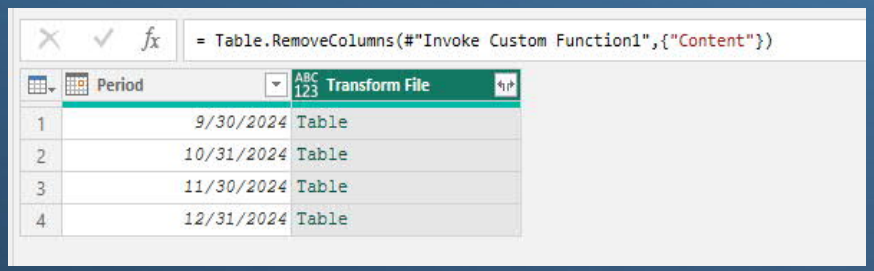
CHOOSE what columns in your spreadsheet you want to include.

UNCLICK the Use Original column name as prefix to shorten column name.

CLICK Ok when done.



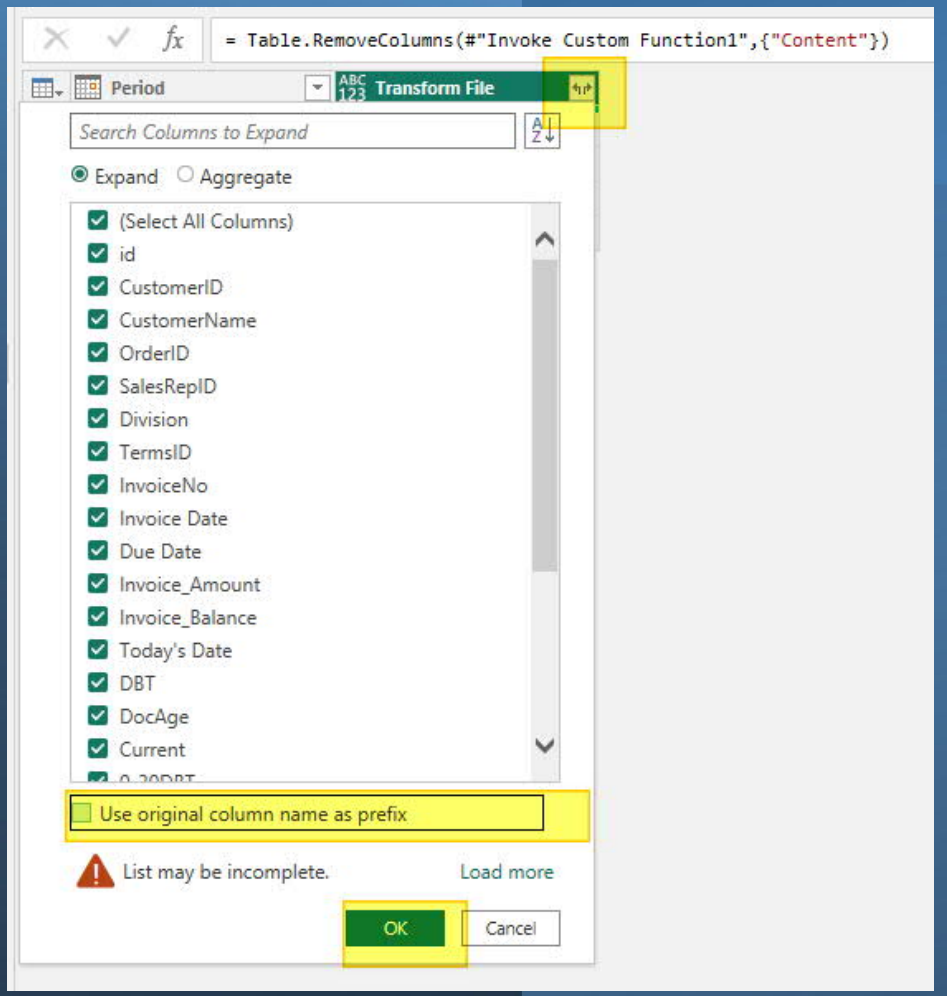
CLICK the arrows on the ‘Transform File’ with tables as rows to expand



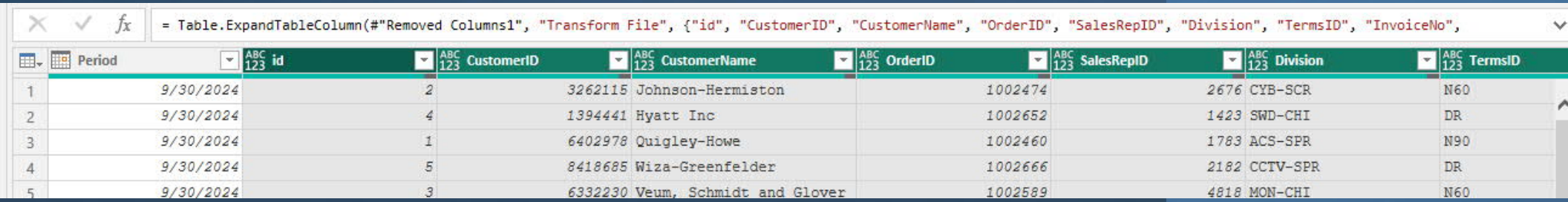
CHOOSE what columns in your spreadsheet you want to include.

UNCLICK the Use Original column name as prefix to shorten column name.

CLICK Ok when done.



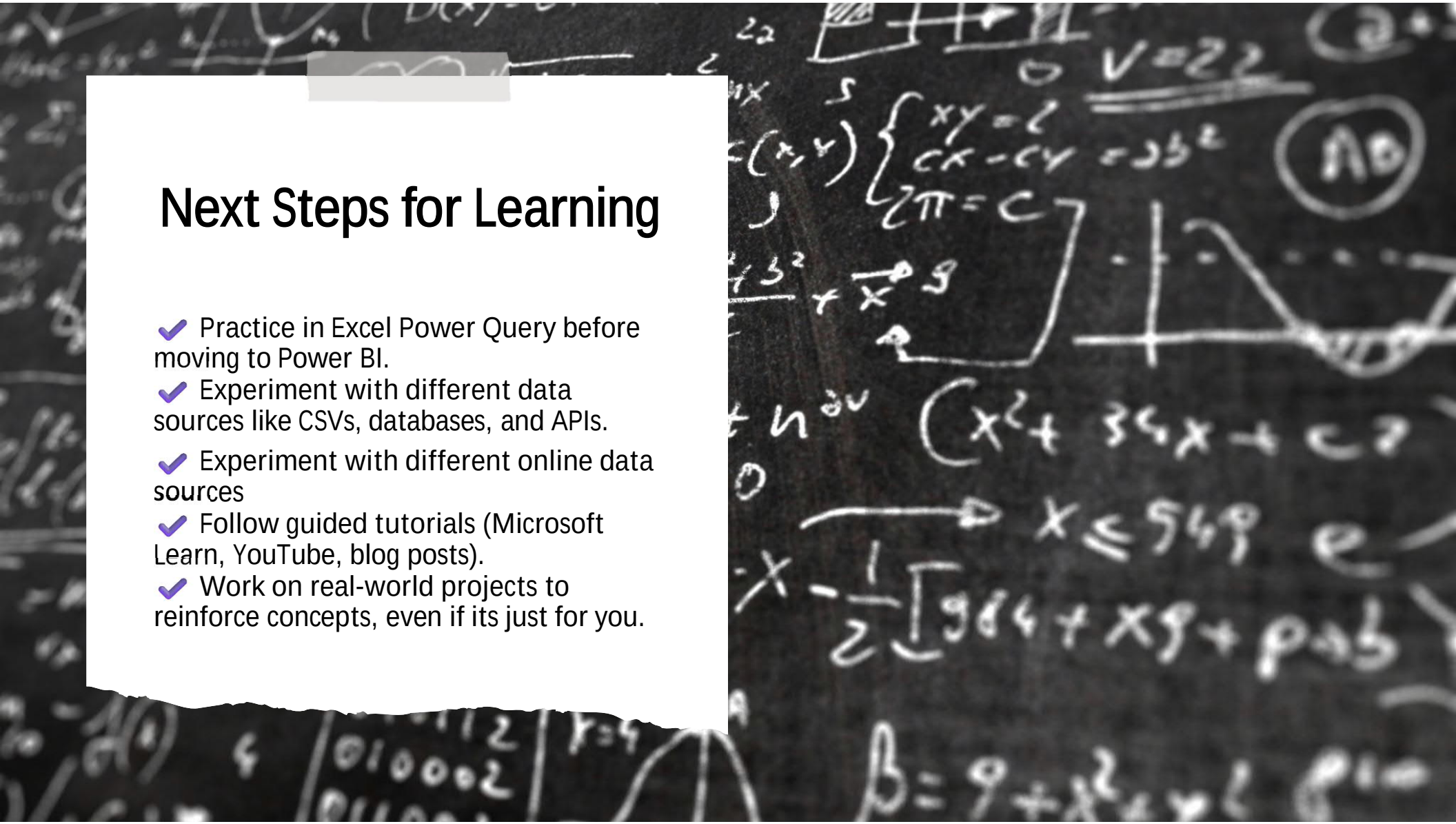
VERIFY All your Periods show in Filter.
UPDATE Your Data Type for each Column Type
RENAME as necessary.
NAME your Query
REFERENCE your Query as Staging or to DataSource for further edits.
Customizations. Ie. Top 10 Customers, Specific Division.



```
= Table.ExpandTableColumn(#"Removed Columns1", "Transform File", {"id", "CustomerID", "CustomerName", "OrderID", "SalesRepID", "Division", "TermsID", "InvoiceNo",
```

	Period	id	CustomerID	CustomerName	OrderID	SalesRepID	Division	TermsID
1	9/30/2024	2	3262115	Johnson-Hermiston	1002474	2676	CYB-SCR	N60
2	9/30/2024	4	1394441	Hyatt Inc	1002652	1423	SWD-CHI	DR
3	9/30/2024	1	6402978	Quigley-Howe	1002460	1783	ACS-SPR	N90
4	9/30/2024	5	8418685	Wiza-Greenfelder	1002666	2182	CCTV-SPR	DR
5	9/30/2024	3	6332230	Veum, Schmidt and Glover	1002589	4818	MON-CHI	N60

ADD new files to your folder as needed or scheduled.
REFRESH your data to see additional changes.

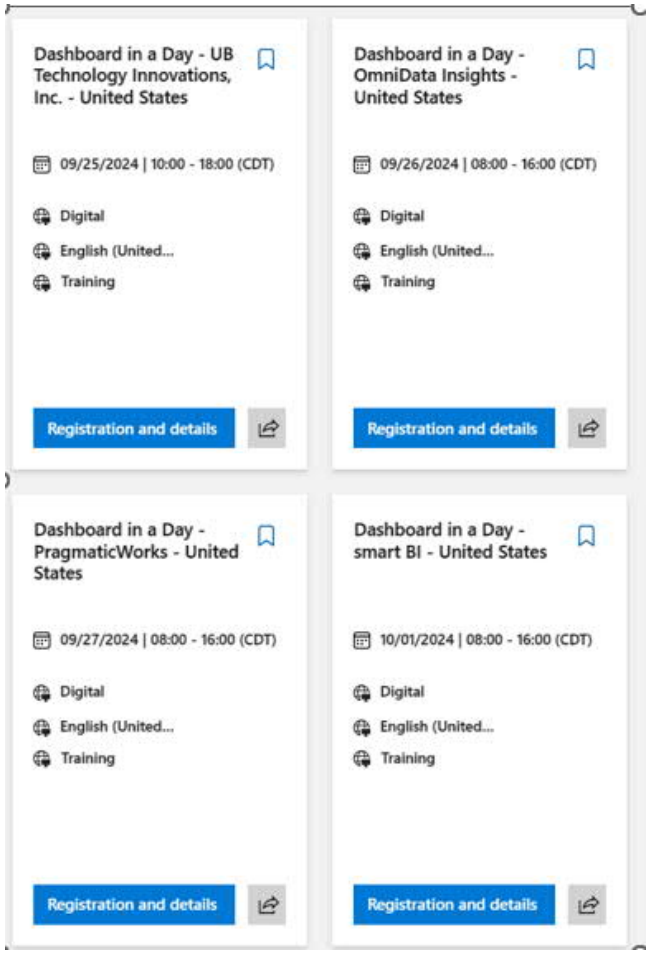


Next Steps for Learning

- ✔ Practice in Excel Power Query before moving to Power BI.
- ✔ Experiment with different data sources like CSVs, databases, and APIs.
- ✔ Experiment with different online data sources
- ✔ Follow guided tutorials (Microsoft Learn, YouTube, blog posts).
- ✔ Work on real-world projects to reinforce concepts, even if its just for you.

Attend a Free 1 Day Event Workshop:

Microsoft Dashboard in a Day



- Hands-On, Practical Learning Experience
- Rapid Skill Acquisition
- Guided Instruction from Experts
- Structured Learning Agenda
- Real-World Application of Skills
- Access to Workshop Materials & Resources
- Networking Opportunities
- Personalized Feedback & Support
- Boosts Confidence with Power BI
- Preparation for Advanced Learning
- Cost-Effective Training Option
- Immediate Insight into Power BI's Capabilities
- Exposure to Power BI Service Features

Pragmatic Works DAX Cheat Sheet for Beginners

Learn

700 XP

Get started building with Power BI

21 min • Module • 6 Units

Feedback

Beginner | Data Analyst | Business Analyst | Business User | Functional Consultant | Power BI

Learn about Power BI, the building blocks and flow of Power BI, and how to create compelling, interactive reports.

This module helps prepare you for Exam PL-200: Microsoft Power Platform Functional Consultant.

Learning objectives

In this module, you'll learn:

- How Power BI services and applications work together.
- Explore how Power BI can make your business more efficient.
- How to create compelling visuals and reports.

Start > ⊕ Add

Prerequisites

None

This module is part of these learning paths

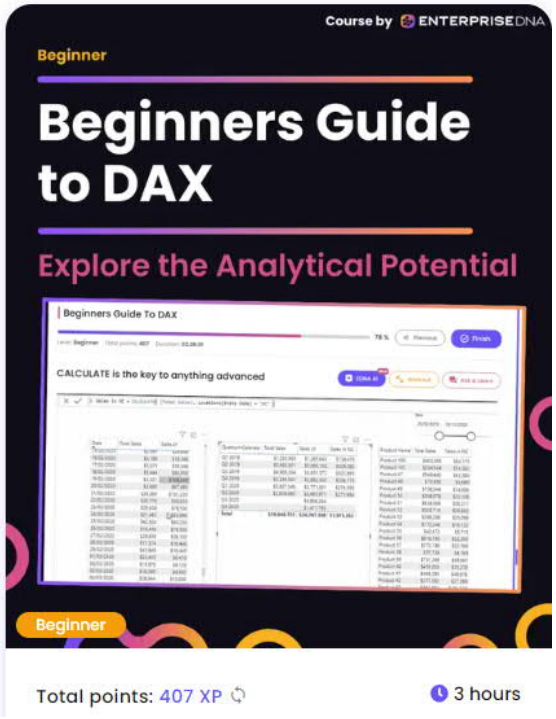
Create and use analytics reports with Power BI

Get started with Microsoft data analytics

Get started with Power BI

Click to start: Microsoft Learn

- Free Access to High-Quality Content
- Structured Learning Paths
- Hands-on Labs and Interactive Exercises
- Official and Up-to-Date Content
- Integration with Certifications
- Gamified Learning Experiences (Points, Badges)
- Self-Paced Learning
- Community and Q&A Integration
- Comprehensive Coverage of Power BI Features
- Scenario-Based Learning Modules



FREE COURSE - Ultimate Beginners Guide To Power BI - http://portal.enterprisedna.co/p/ultimate-beginners-guide-to-power-bi
FREE COURSE - Ultimate Beginners Guide To DAX - http://portal.enterprisedna.co/p/ultimate-beginners-guide-to-dax
FREE - 60 Page DAX Reference Guide Download - https://enterprisedna.co/dax-formula-reference-guide-download

Click to start: ENTERPRISE DNA

Sam McKay, CFA

- Some Free Courses else Paid Subscription
- Expert-Led Training & Courses
- Focus on Real-World Scenarios & Problem-Solving
  - Finance Focused
- Comprehensive Course Catalog
- Access to Learning Summits & Workshops
- Extensive Resource Library
  - Power BI .pbix file downloads
- Customized Learning Paths
- Innovative Data Challenges & Projects
- Supportive Community Forum
- Access to Power BI Showcases
- Focus on Advanced Analytics & AI Integration
- On-Demand, Self-Paced Learning
- Gamified Learning Experience (Points & Badges)
- Certification Programs
- Emphasis on Visualization & Design

Q&A and Closing

Questions?