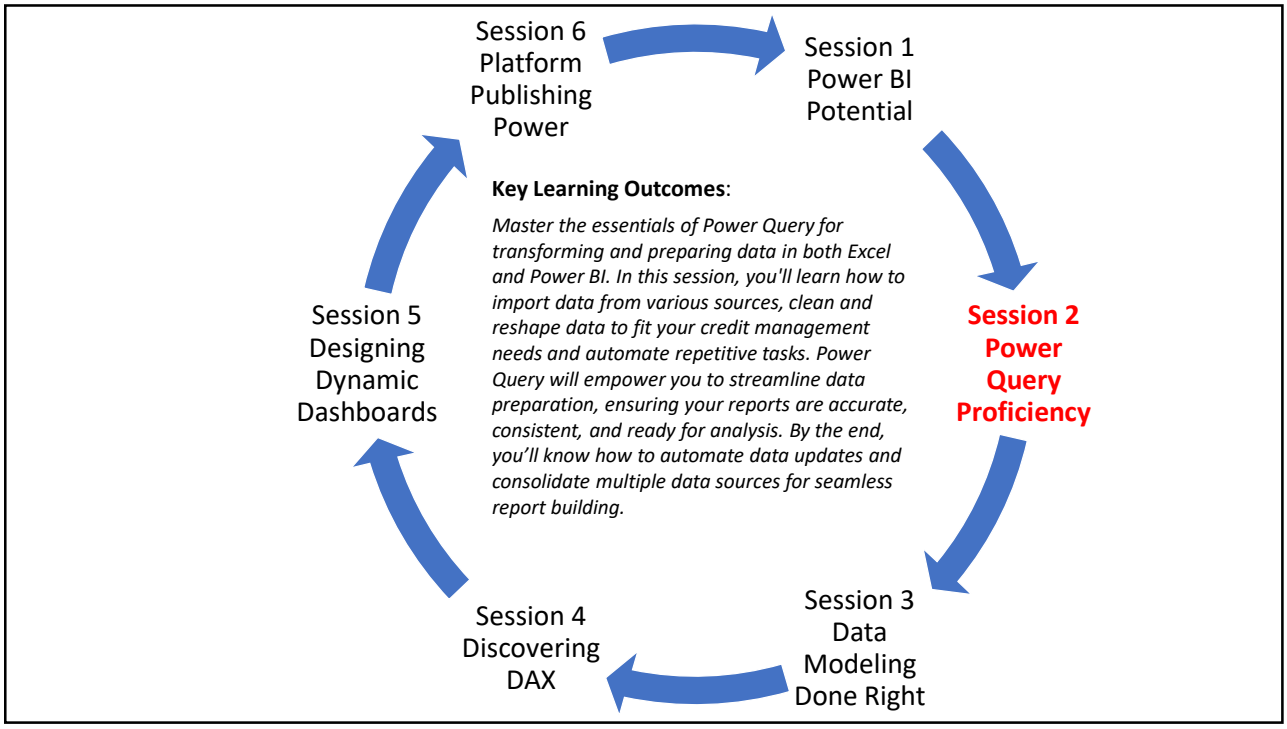
NACM'S 129TH CREDIT CONGRESS & EXPO Cleveland MAY 18-21, 2025

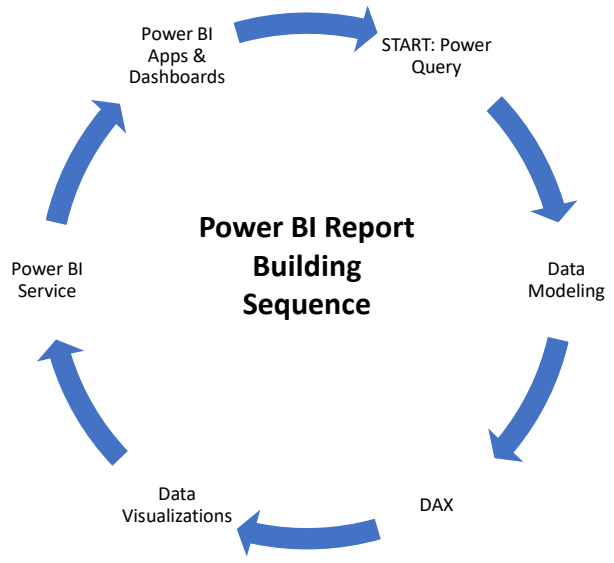
Power Query Proficiency

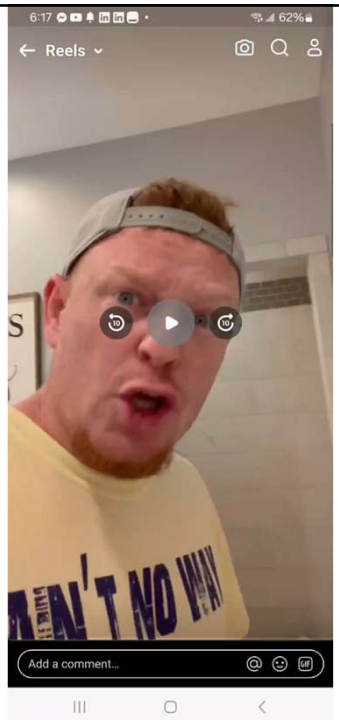
Presented by: Rebekyah Brewer

Date: May 20, 2025

Session: #37043




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


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

Basic Data Concepts

Before diving into Power Query, a beginner should have a good grasp of:

- **What is Data?** – Understanding structured vs. unstructured data.
- **Types of Data Sources** – Files (Excel, CSV, JSON), Databases (SQL, Access), Web APIs, etc.
- **Data Transformation Basics** – What it means to clean, filter, merge, and transform data.
- **Relational Data Concepts** – How tables relate to each other using keys (useful for joins and merges).
- Understanding **real-world data issues** like missing values, duplicates, incorrect formatting, and inconsistencies.

Prerequisites – Individual

Excel Basics (Highly Recommended)

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 **named ranges and structured references**.
- Experience working with v-lookups

Prerequisites – Individual

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, Many-to-One).
- **Loading data into Power BI** (vs. keeping transformations in Power Query)

Basic SQL Knowledge (Optional not Required but Helpful)

Power Query allows importing data from databases, so knowing some basic SQL queries can be beneficial:

- SELECT statements.
- Filtering using WHERE.
- JOIN operations (for merging data in Power Query).

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

Computer Capabilities & Performance Considerations

Power Query processes data transformations, and performance can be impacted by your system specs.

- **RAM** – 8GB minimum; 16GB+ recommended for handling large datasets.
- **Processor** – Intel i5/i7 or AMD Ryzen 5/7 or higher for better performance.
- **Storage (SSD Recommended)** – Faster SSD drives improve data processing speed compared to HDD.
- **Internet Speed** – If working with cloud data, a stable internet connection is necessary.

Power BI Performance Tips:

- Avoid importing large unnecessary datasets; use filtering & query folding.
- Use **DirectQuery** mode instead of Import mode for large databases (if supported).
- Reduce steps in Power Query to improve load speed
- Use Incremental refreshes or expand data once you publish in Power BI Service

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 (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.

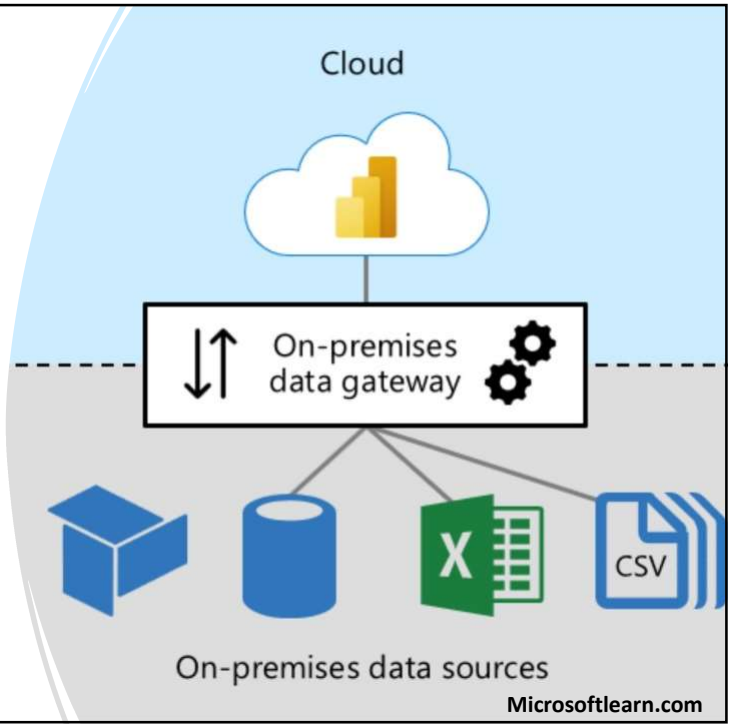
Databases (SQL, MySQL, etc.) – Requires database credentials and proper configuration.

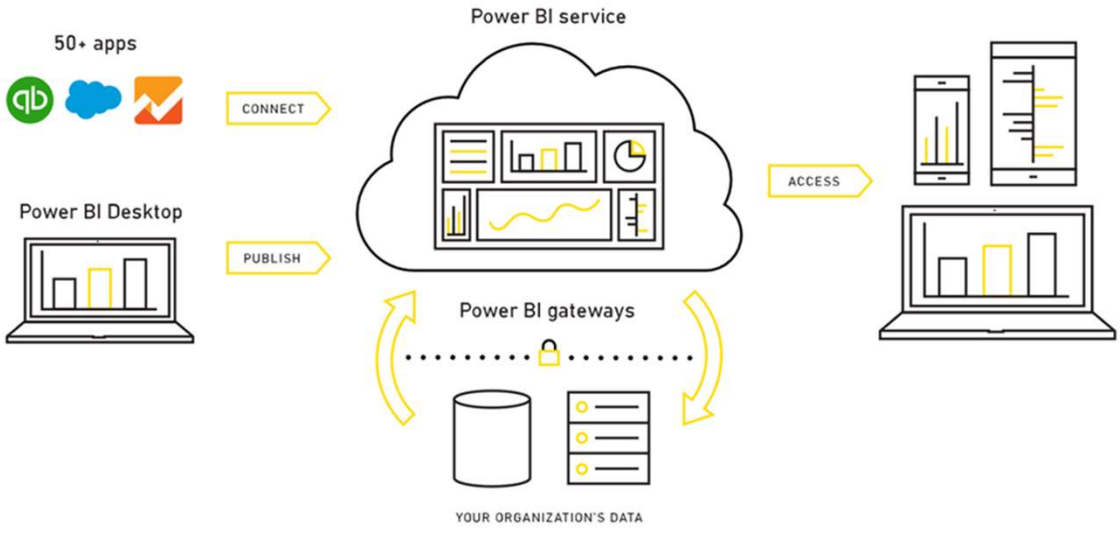
Webpages – Already in the cloud, easy to connect to.

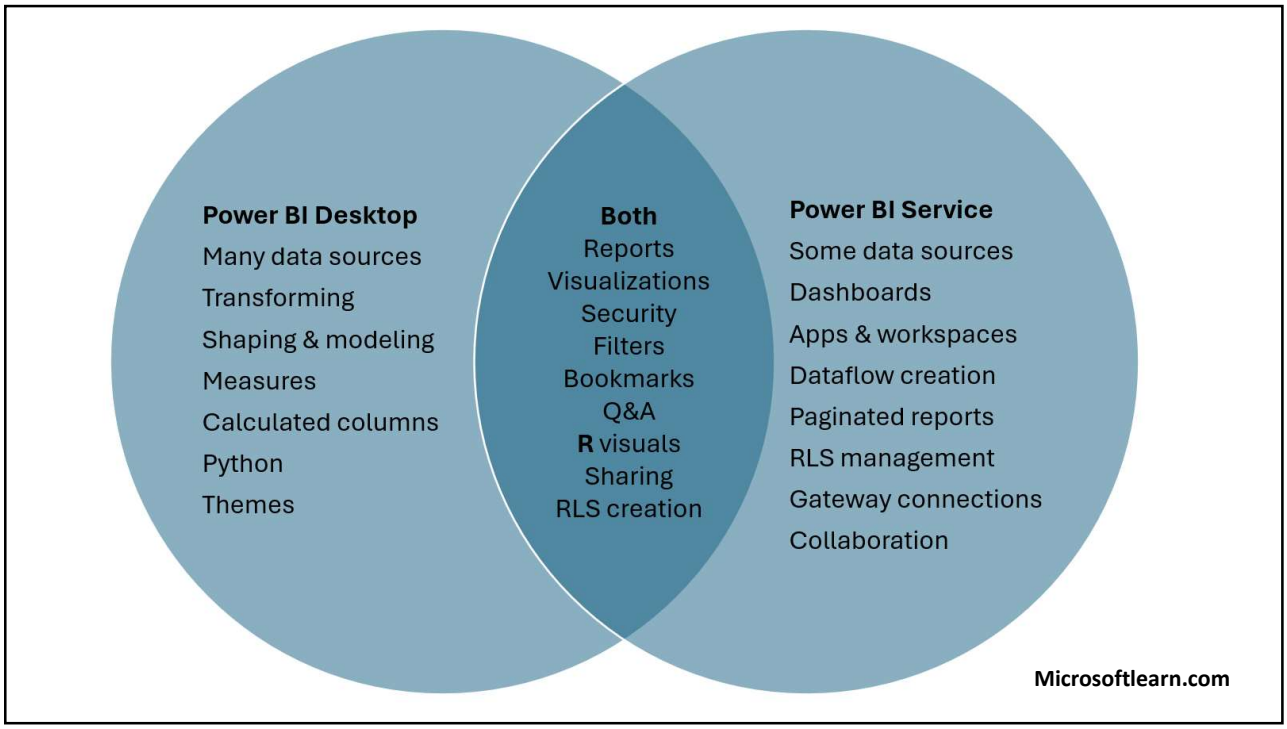
Understanding Data Connectivity & Storage Options cont'd.

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.







Prerequisites – Technical

Understanding Data Connectivity & Storage Options cont'd.

3. 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.

Prerequisites – Technical

Understanding Refresh & Automation

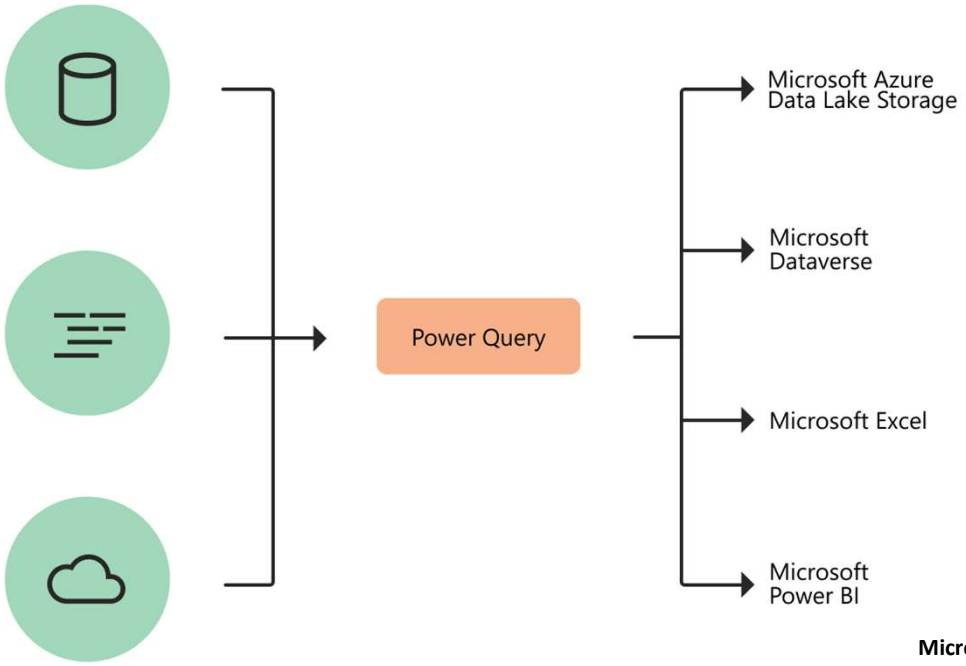
- **Manual Refresh** – Run refresh in Power BI Desktop.
- **Scheduled Refresh (Power BI Service)** – Requires a **gateway** for local files but not for cloud sources (SharePoint/OneDrive).
- **Incremental Refresh** – Available in Power BI Premium or PPU, useful for large datasets.

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.



Microsoftlearn.com

Currently, two Power Query experiences are available:

- **Power Query Online**—Found in integrations such as Power BI dataflows, Microsoft Power Platform dataflows, Azure Data Factory wrangling dataflows, and many more that provide the experience through an online webpage.
- **Power Query for Desktop**—Found in integrations such as Power Query for Excel and Power BI Desktop.

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.

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, and Power Query helps with many of them.

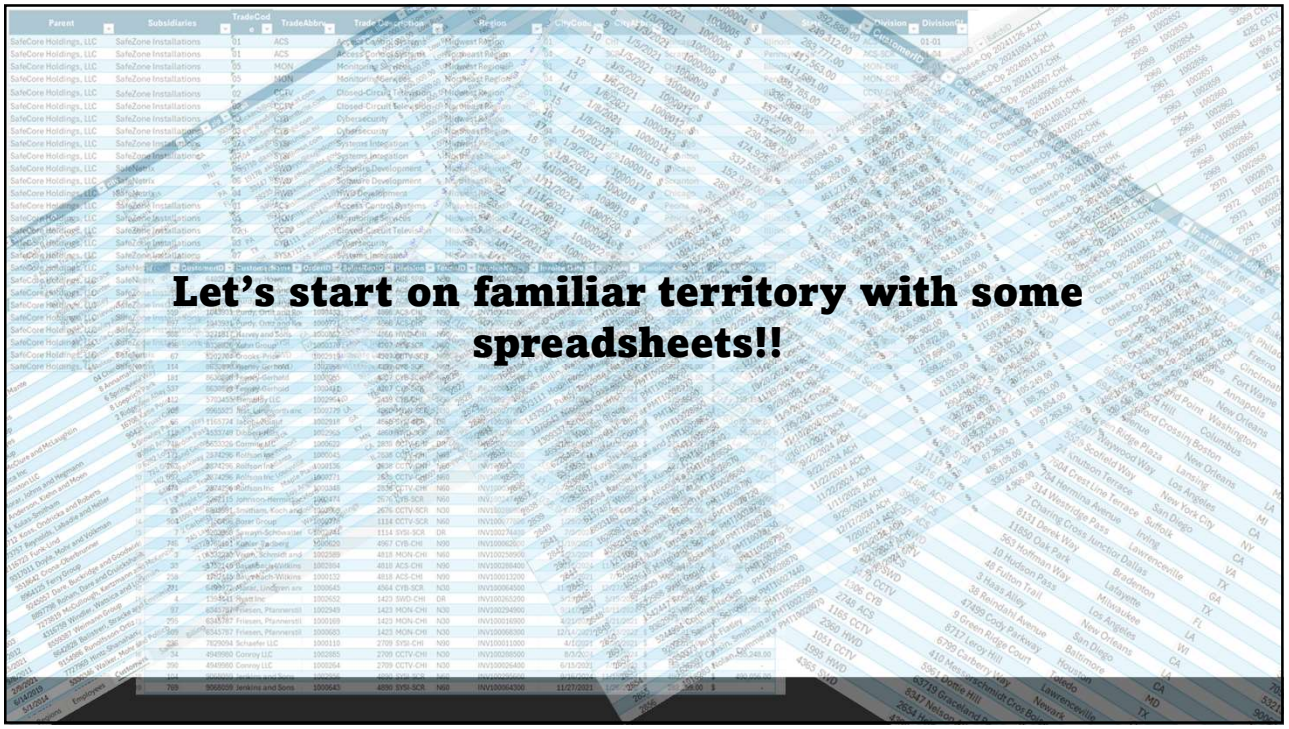
Benefits of Learning Power Query in Excel:

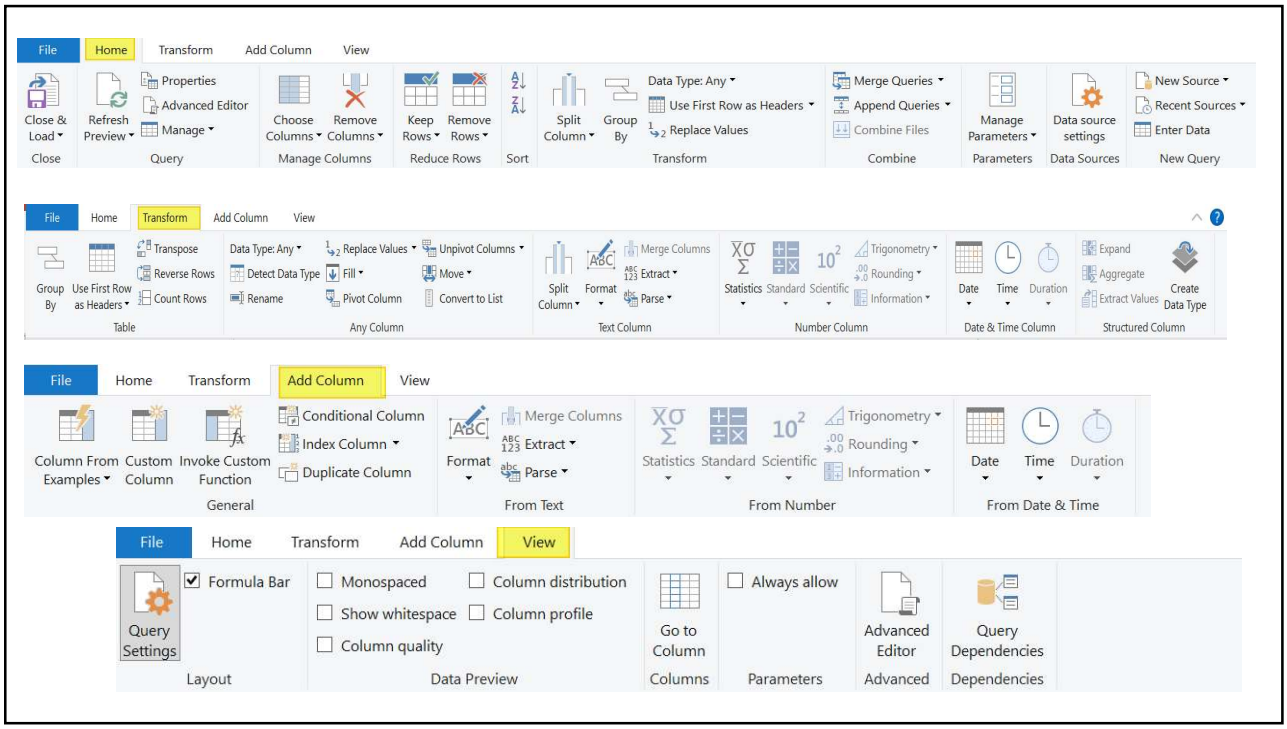
1. Excel is familiar territory for all of us
2. Increased opportunities for internal use leads to wider application
3. No SQL Coding Required, Easy drag and drop interface, Formula like approach
4. Increased usage cases leads to increased experience
5. Small quick wins will encourage motivation to learn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

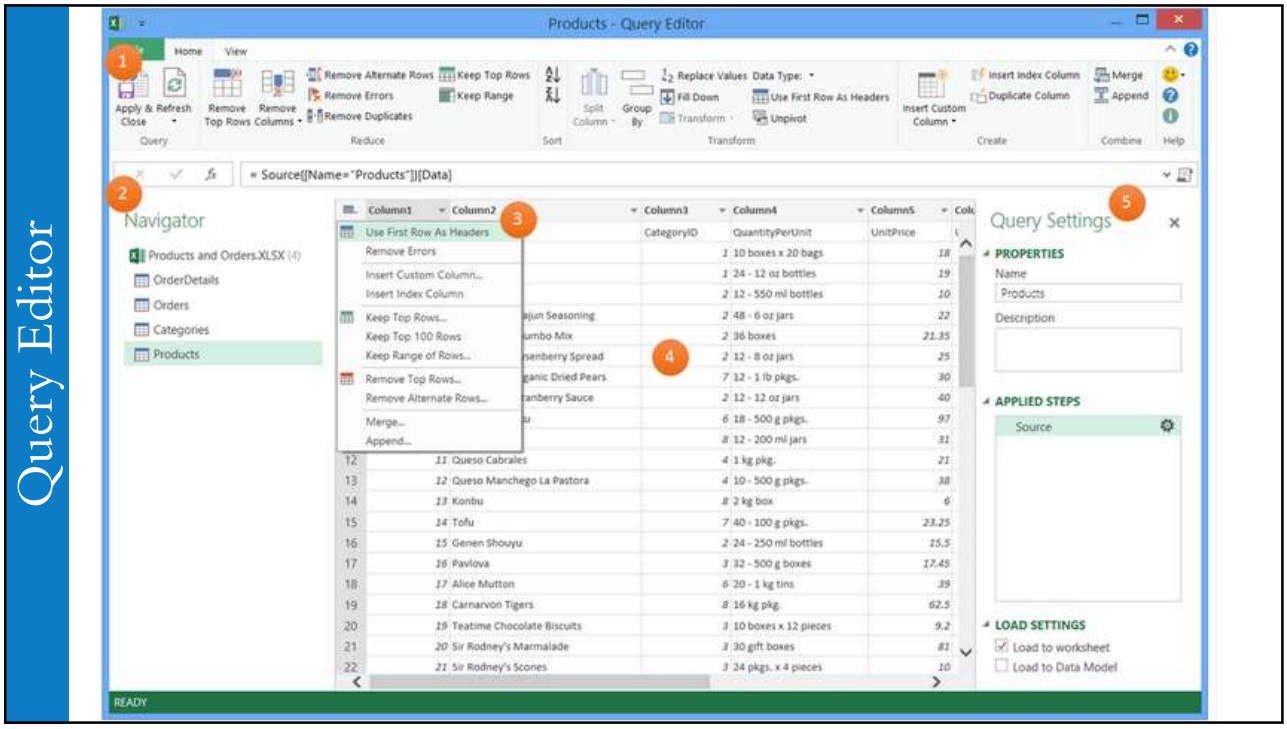
Navigating the Interface

1. **Overview of key components**
   - Query Navigator Pane
   - Data Preview Grid
   - Ribbon Tools for Transformation & Action
   - Applied Steps – Visual Record of transformation actions
   - Queries – Data Sources
2. **Where to access Power Query in Excel and Power BI**

.




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
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
Column From Examples
Custom Column
Invoke Custom Function
Conditional Column
Index Column
Duplicate Column
General
From Text
From Number
From Date & Time
Query Settings
Formula Bar
Monospaced
Show whitespace
Column quality
Column distribution
Column profile
Go to Column
Always allow
Advanced Editor
Query Dependencies
Layout
Data Preview
Columns
Parameters
Advanced
Dependencies

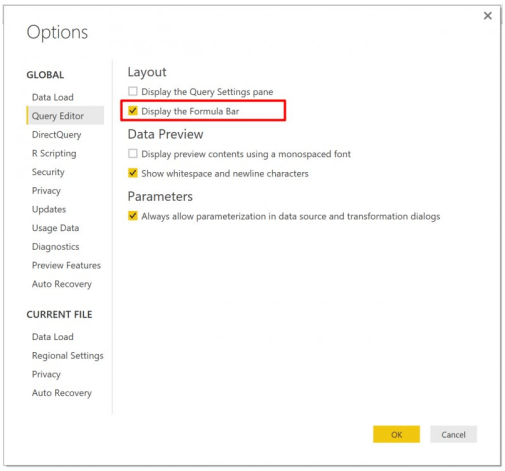

Query Editor
Products - Query Editor
Navigator
Products and Orders.XLSX (4)
OrderDetails
Orders
Categories
Products
Query Settings
PROPERTIES
Name
Products
Description
APPLIED STEPS
Source
LOAD SETTINGS
Load to worksheet
Load to Data Model
READY

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



Data Preparation & Loading

- Connecting Data Sources
- Exploring Data Load Options

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)
- Folders
- SharePoint Folders

Connecting to Data Sources

External Data Sources for Credit Managers

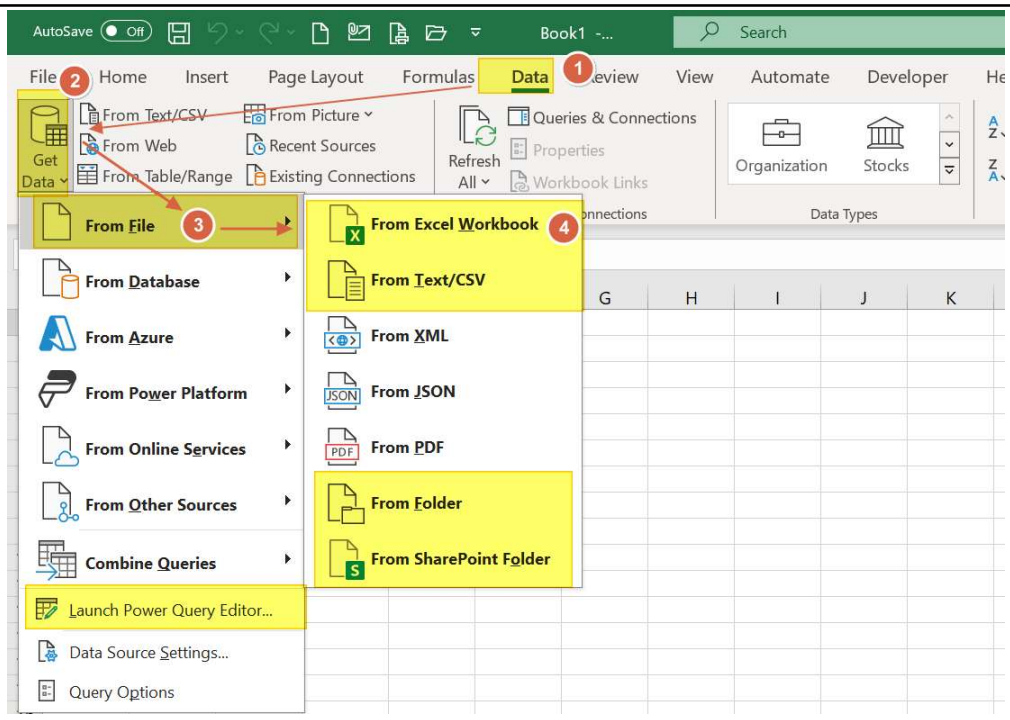
- Credit Risk Monitor
- Moody's Boost Report
- Legal Review
- Public Information Websites
- WSJ Prime Rate
- Other 3rd Party Downloads

Organizing Data Sources

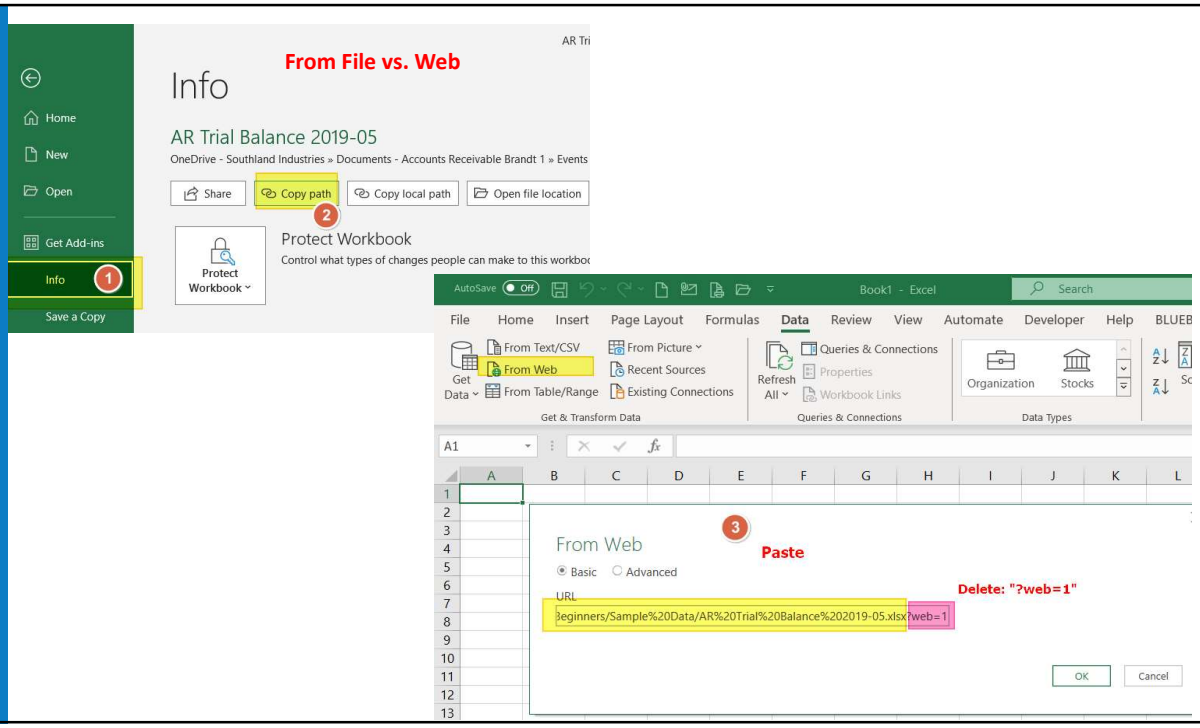
Create folders for Data Sources:

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

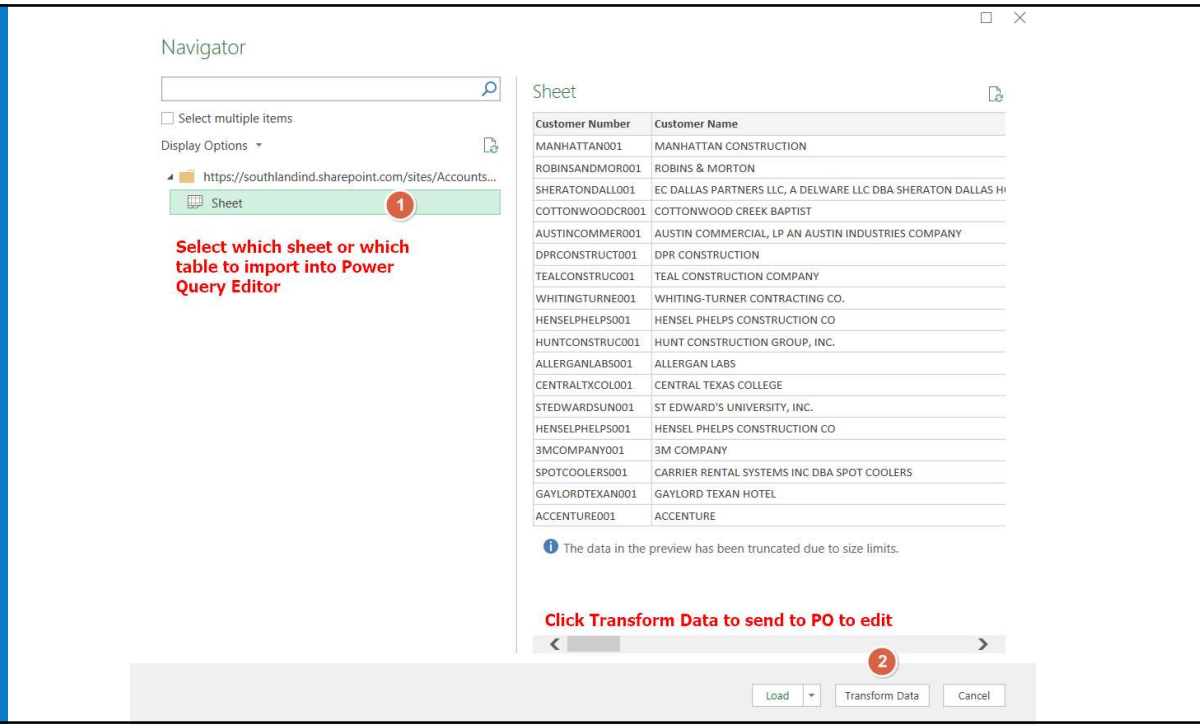
Loading Power Quiery



Excel – File vs. Web



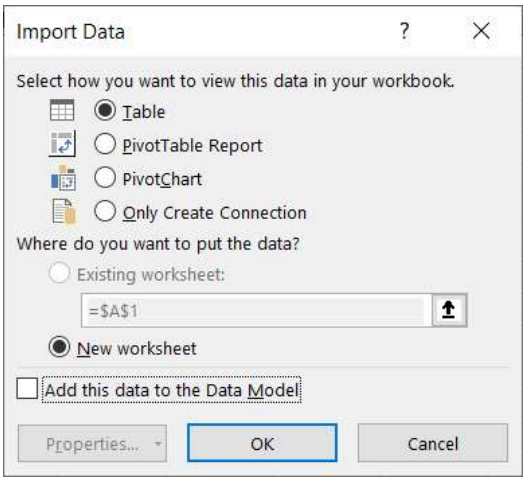
Loading Power Query



Data Load Options

Where can you load?

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

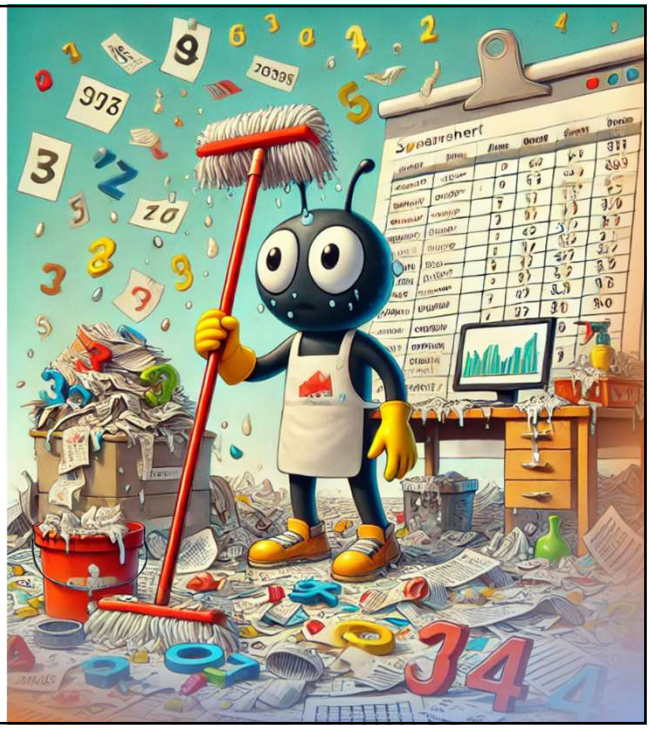


IV. Essential Data Transformation Techniques

- Cleaning Data
- Shaping Data

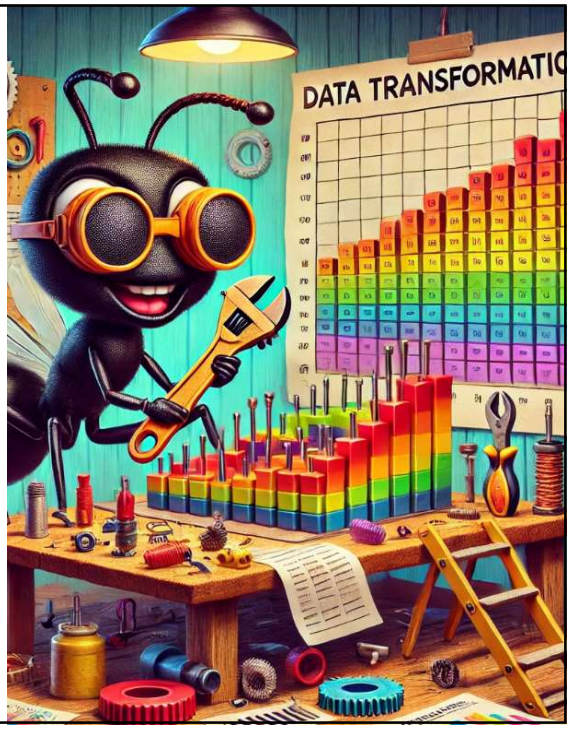
Cleaning Data

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



Advanced Power Query Techniques

- Combing Queries
- Introduction to M Code
- Creating Custom Columns
- Working with Advanced Editor
- Parameters & Templates
- Functions

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)

Working with 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"
```

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.

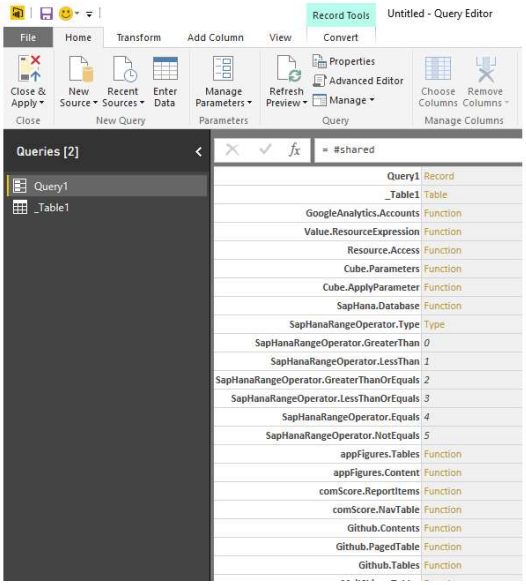
Functions

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

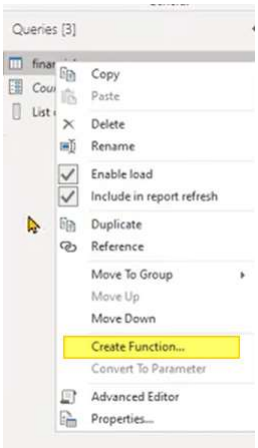
- Text.Upper(): Converts text to uppercase.
- DateTime.LocalNow(): Returns the current date and time.
- Table.Sort(): Sorts a table based on specified columns.

Functions – Power Query Library of Functions



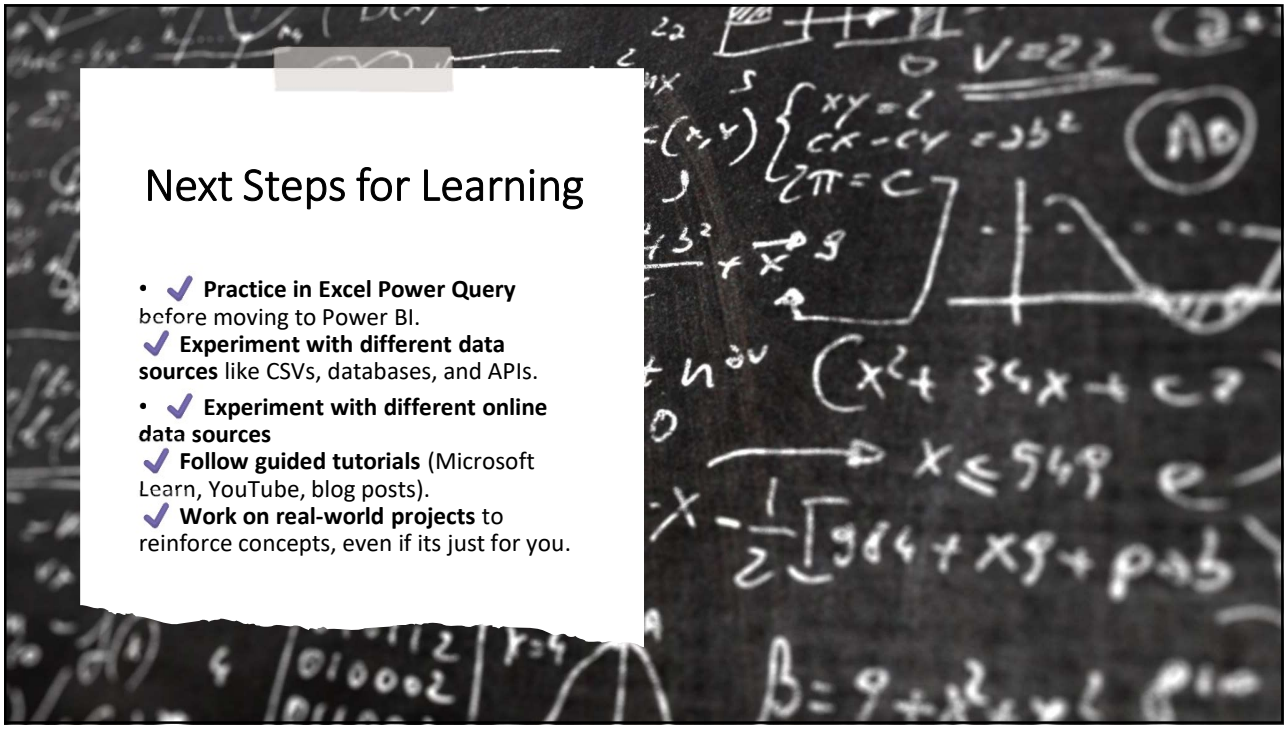
Functions

Create Custom Functions –



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

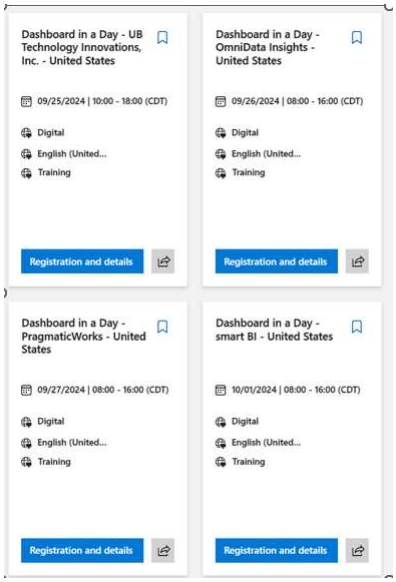


Next Steps for Learning

- ✔ **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.

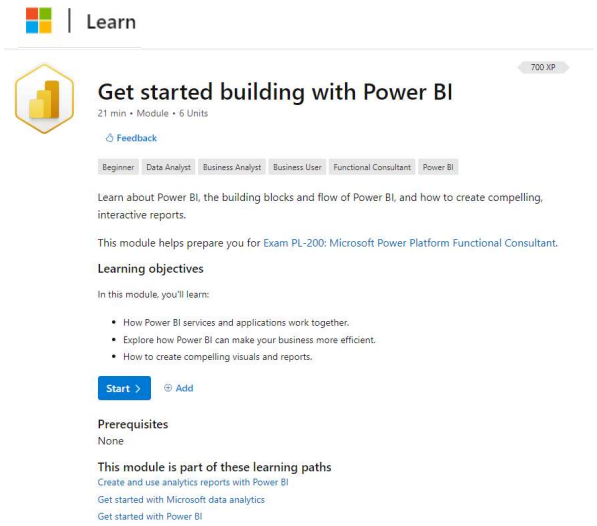
Attend a Free 1 Day Event Workshop:

Microsoft Dashboard in a Day



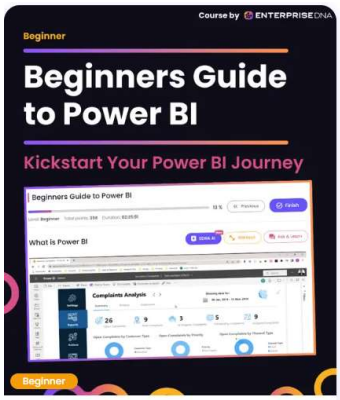
Pragmatic Works DAX Cheat Sheet for Beginners

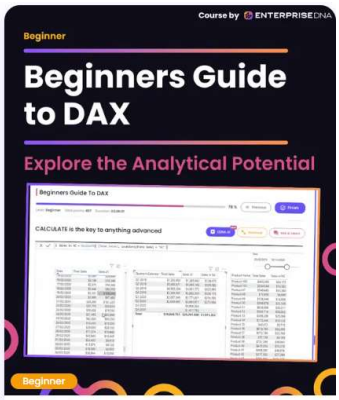
- Hands-On, Practical Learning Experience
- Rapid Skill Acquisition
- Guided Instruction from Experts
- Structured Learning Agenda
- Real-World Application of Skills
- Access to Workshop Materials & Resources
- Networking Opportunities
- Personalized Feedback & Support
- Boosts Confidence with Power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



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?