Advanced Power BI

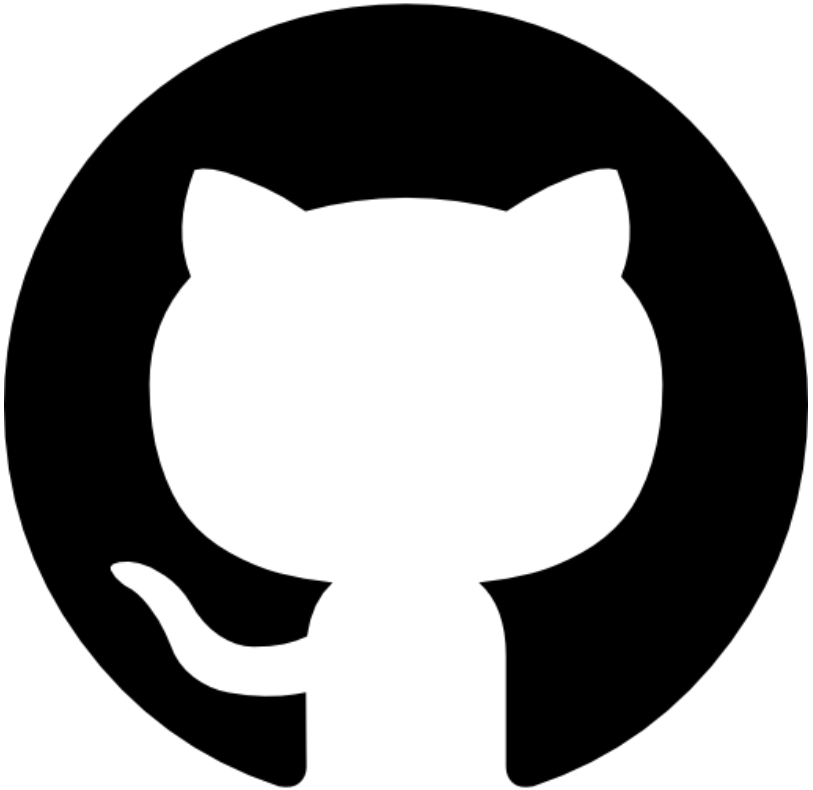
Overview

1. *Day One*
    a. Session 1: Introduction
    b. Session 2: Core ETL with Power BI
    c. Session 3: Advanced ETL with Power BI
    d. Session 4: Core Visualizations
    e. Session 5: Advanced Visualizations
    f. Session 6: Daily Capstone
2. *Day Two*
    a. Session 7: Recap and Introduction to Data Modelling
    b. Session 8: Core DAX
    c. Session 9: Advanced DAX
    d. Session 10: Advanced Filters
    e. Session 11: Advanced Report Design
    f. Session 12: Daily Capstone
3. *Day Three*
    a. Session 13: Recap and Introduction to the Data Lifecycle
    b. Session 14: Sharing with Power BI
    c. Session 15: Deployment Pipelines
    d. Session 16: OLS vs RLS
    e. Session 17: Capstone Work
    f. Session 18: Capstone Presentations & Final Recap

Course Resources

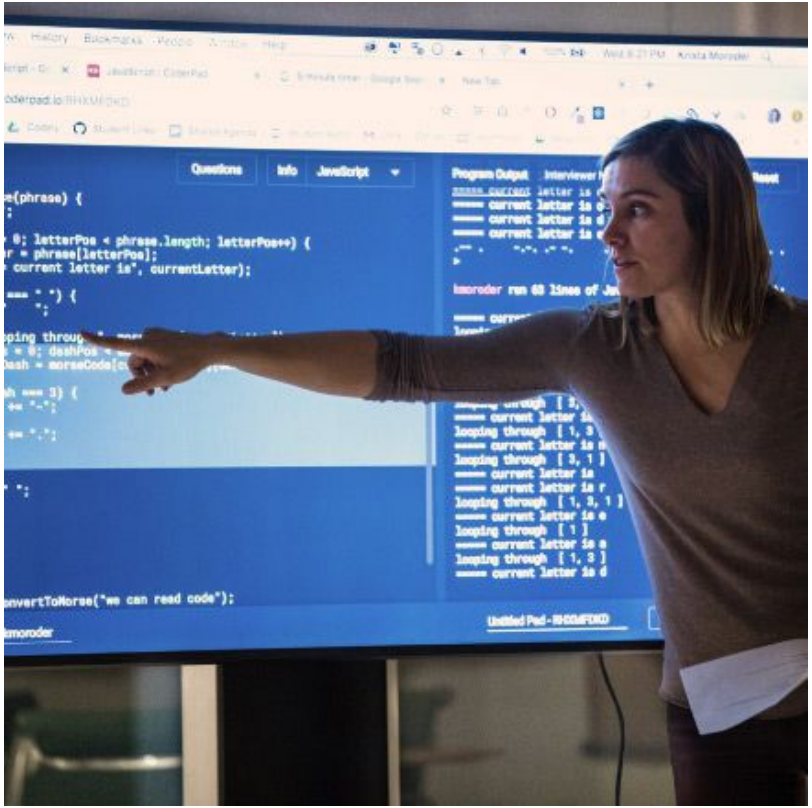
All of the csv files, pbix files, instructions and supporting material can be found in the GitHub repository linked below:

https://github.com/dave-melillo/advanced_powerbi/tree/main



Session Structure

- Theory
- Instructor Do
- Student Do
- Daily Capstone
- Course Capstone

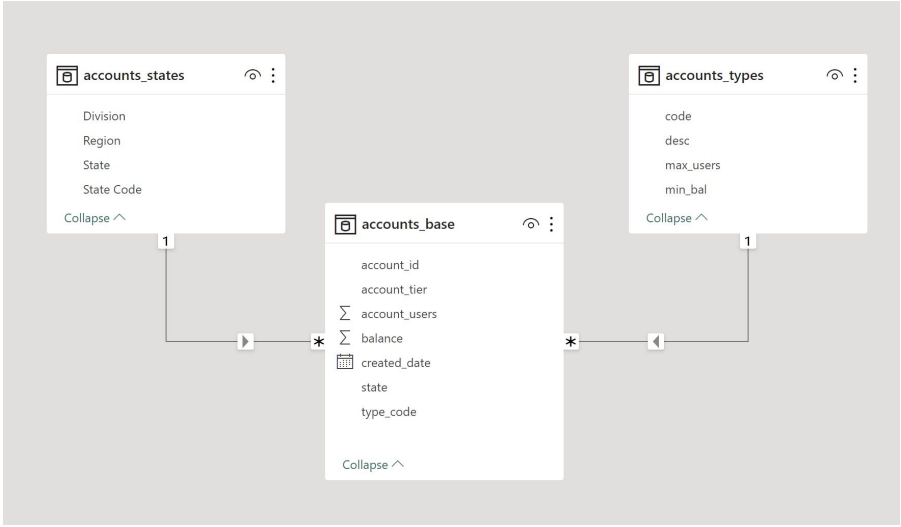


Scoping & Designing Effective BI Projects

Problem Statement: FinServ Inc, a fictitious company that offers insurance, financial and investment products to the public, needs you to do some basic analysis on a list of customer accounts which includes accounts balances, account users and other details.

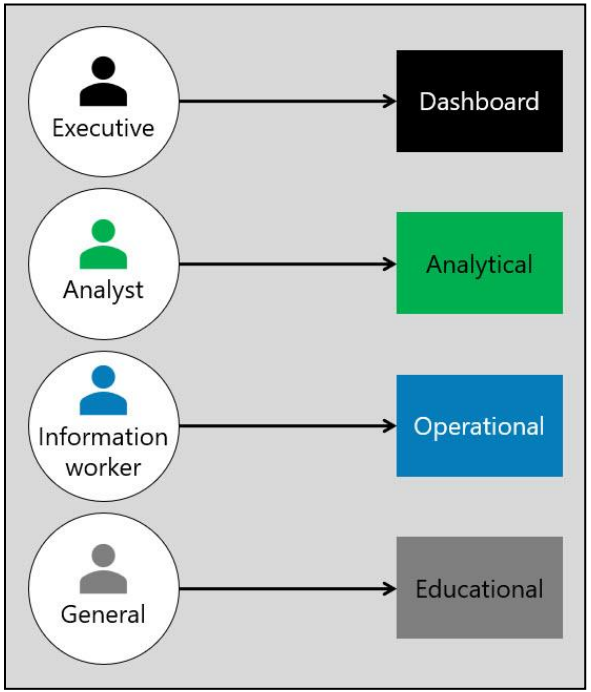
The three files we were given are:

- **accounts_base.csv**
  - A summary of customer accounts with descriptive details such as account balance, creation date, customer tier, and more.
- **accounts_states.csv**
  - Descriptive information about each state in the USA, such as what region they belong to. This will be helpful to summarize accounts by different dimensions.
- **accounts_types.csv**
  - Descriptive information about each account product offered by FinServ Inc.



Scoping & Designing Effective BI Projects

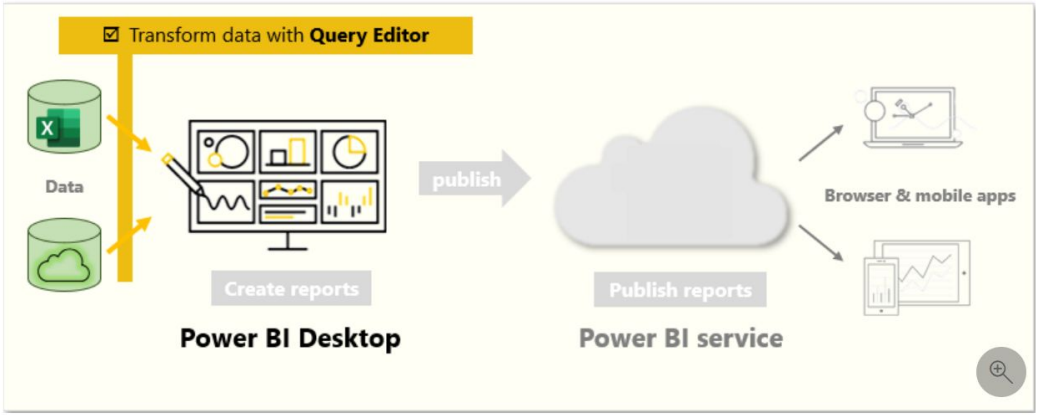
- Identify Audience
- Determine Report Types
- Define User Interface Requirements
- Define User Experience Requirements



Instructor Do: Demonstrate Power BI linear workflow; Ingest, Transform, Visualize

Core ETL with Power BI

- Identify column headers and names
- Promote headers
- Rename columns
- Remove columns
- Rename a query
- Replace values
- Replace values
- Remove duplicates
- Change the column data type
- Parse Text
- Concatenate Text
- Add Conditional Columns

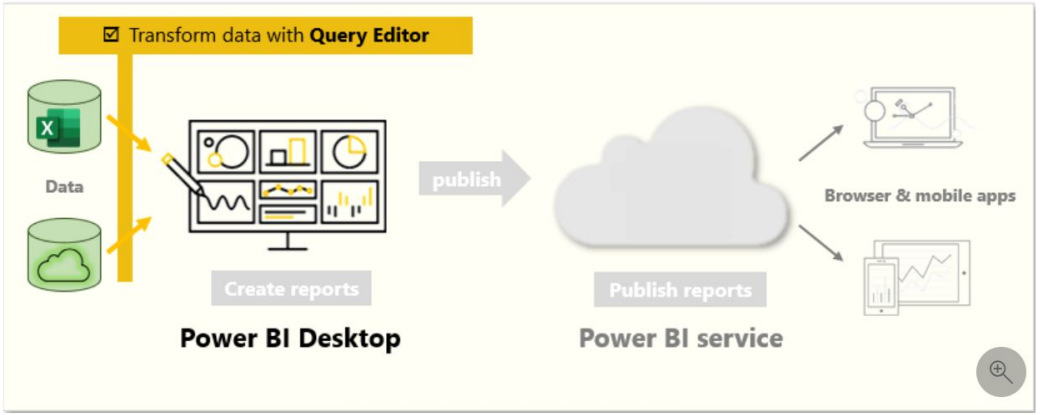


Instructor Do: Demonstrate loading from various sources and basic transformations with Power Query Editor (i.e change data types, parse text, remove characters,concatenation…)

Student Do: Load Data from CSV source and apply basic transformations with Power Query Editor.

Advanced ETL with Power BI

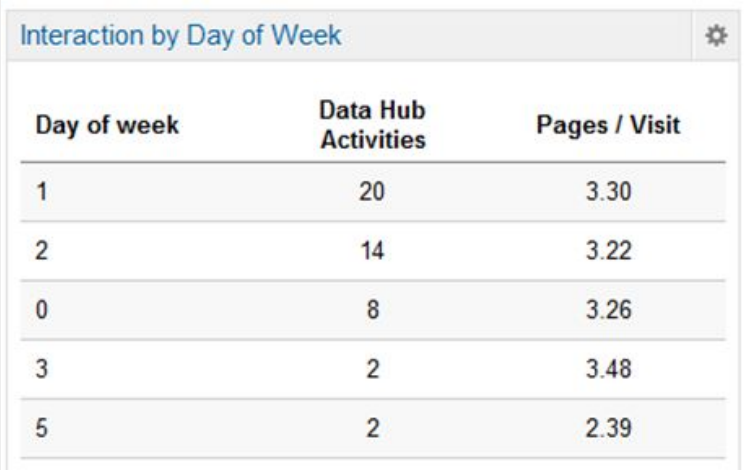
- Merge
- Append
- Group By



Core Visualizations

Tabular Format

Tabular format is best used when exact quantities of numbers must be known. Numbers are presented in rows and columns, and may contain summary information, as in PivotTables. This format is not conducive to finding trends and comparing sets of data because it is hard to analyze sets of numbers and the presentation becomes unwieldy with larger datasets.

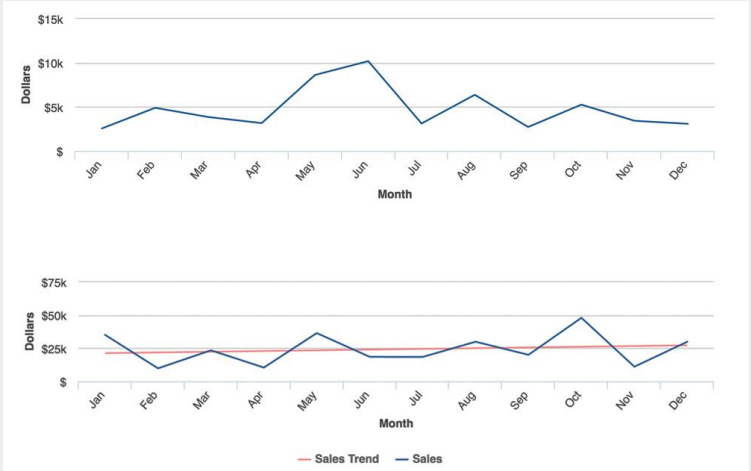


Interaction by Day of Week

Day of week	Data Hub Activities	Pages / Visit
1	20	3.30
2	14	3.22
0	8	3.26
3	2	3.48
5	2	2.39

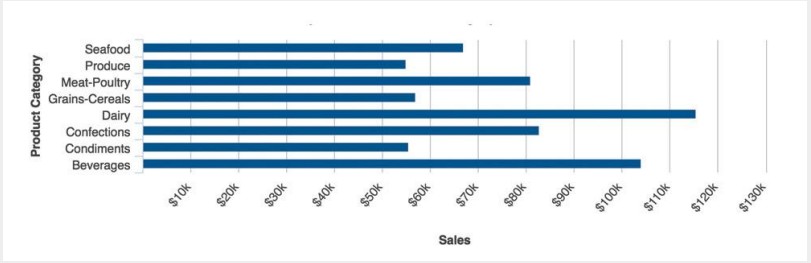
Line Charts

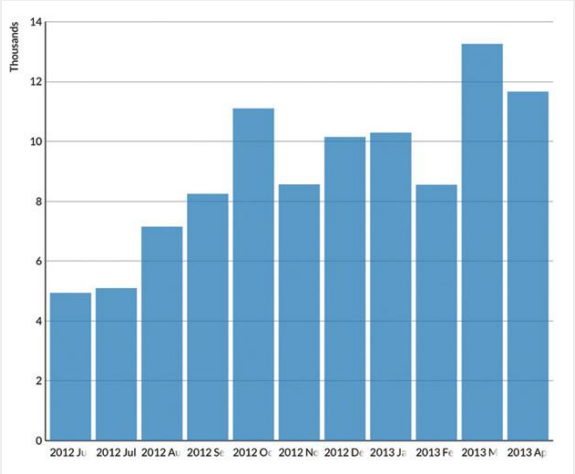
Line charts are best used when trying to visualize continuous data over time. Line charts are set against a common scale and are ideal for showing trends in data. You might also add a trend line or a goal line to illustrate performance in a certain period against a set benchmark.



Bar Charts

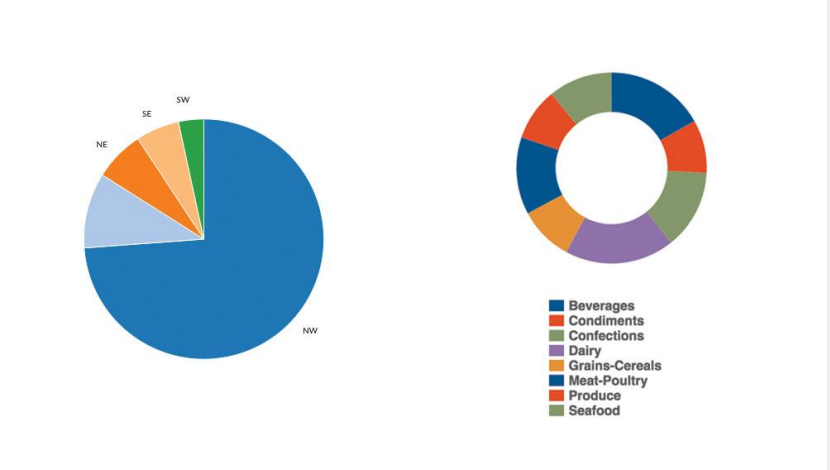
Bar charts are best used when showing comparisons between categories. Typically, the bars are proportional to the values they represent and can be plotted either horizontally or vertically. One axis of the chart shows the specific categories being compared, and the other axis represents discrete values. Bar charts are ideal when you're working with limited space.





Pie Charts

Pie charts are best used to compare parts to the whole. Pie charts make it easy for an audience to understand the relative importance of values, but when there are more than five sections, it can become difficult to compare the results. The difference between the sections can become too narrow to effectively interpret. Alternate visual styles include the exploded pie wedge chart, for emphasizing important data, and the donut pie chart, to support information by inserting a design element in the center of the pie.



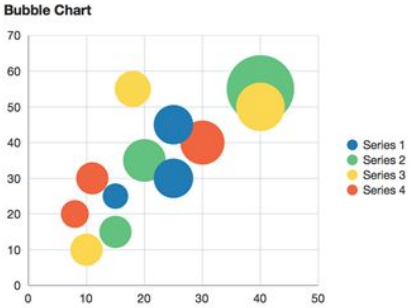
KPI/Card Charts

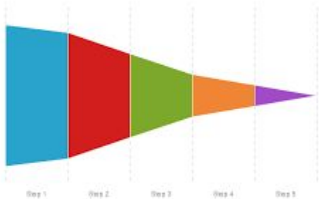
A Key Performance Indicator (KPI) chart is a visual representation that showcases key metrics or performance indicators to monitor the health, progress, and success of a business, project, or process. KPIs are specific measurements that reflect important aspects of an organization's goals or objectives. KPI charts are designed to provide a clear and concise overview of these metrics over time or in relation to certain benchmarks.

Advanced Visualizations

Advanced Visualizations

- **Bubble Charts/Scatter Plots:** Bubble charts and scatter plots are used to show multiple dimensions of data by comparing entities in terms of their relative values, positions, and sizes. They are valuable for visualizing relationships and identifying patterns in data points.
- **Funnel Charts:** Funnel charts display data in a funnel-like structure, illustrating a process or sequence of steps, often used for sales and marketing to visualize conversion rates at each stage.
- **Waterfall Charts:** Waterfall charts represent incremental changes in data, typically used to show how an initial value is affected by a series of positive and negative values. They are useful for understanding the cumulative impact of various factors on a total.
- **Matrices and Hierarchies:** Matrices and hierarchies are used to display data in a structured grid or tree-like format, making it easier to compare and analyze data in a tabular or hierarchical manner. They are suitable for multi-dimensional data.
- **Maps**: Maps are best for showing a geographical representation of data. They can display data by location, helping to identify regional trends or disparities. Common uses include plotting sales data, population statistics, or any location-based information.
- **Custom Visuals with R & Python**: Custom visuals allow you to incorporate additional data visualization libraries or create custom charts using R or Python code. This flexibility is handy when standard Power BI visuals don't meet your specific requirements.





Day 1 Capstone

In this capstone project, you will work with Pokémon trading card sales data to build a Power BI report.

The project is designed to reinforce the concepts and skills learned on the first day of the course, covering data processing, cleaning, transformation, ETL, and the creation of both basic and advanced visualizations.

Recap & Introduction to Data Modelling

Entity Relationship Diagrams

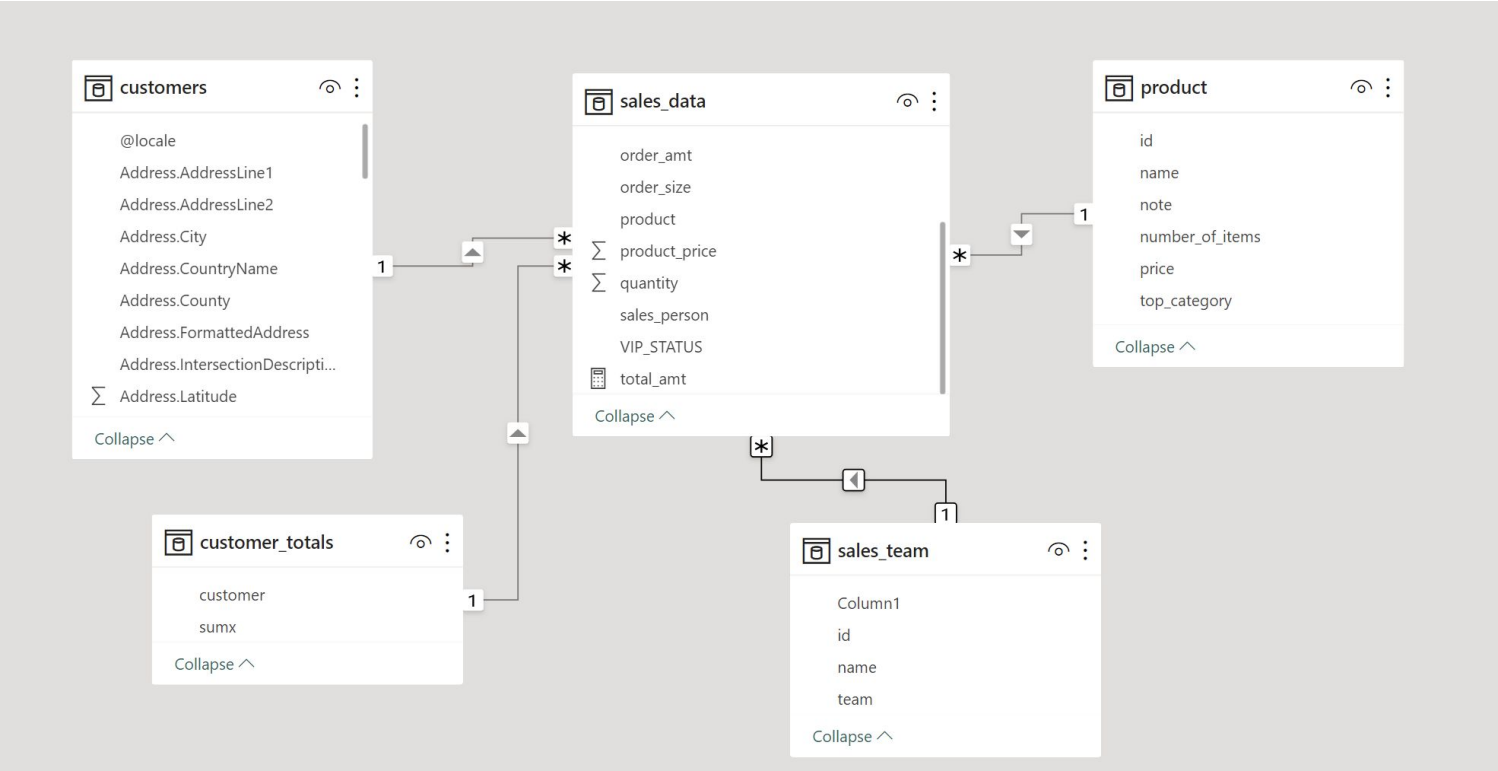
What is an ERD?

An Entity-Relationship Diagram (ERD) is a visual tool used in database design to help people understand how different pieces of information are related to each other. It's like creating a map of the important concepts and their connections within a database.

Resources

https://www.quickdatabasediagrams.com/
https://erdplus.com/
https://www.smartdraw.com/
https://www.lucidchart.com/pages/landing/er-diagram-software

Entity Relationship Diagrams



Facts and Dimensions

Facts:

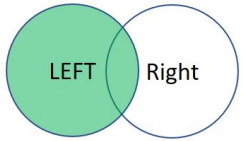
Facts are numeric data points or metrics that represent key performance indicators (KPIs) in data analysis.

Dimensions:

Dimensions provide descriptive context for facts and often have a hierarchical structure, such as time periods or categorical attributes like product categories.

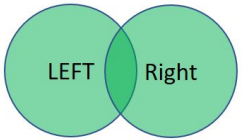
Joins

LEFT Outer



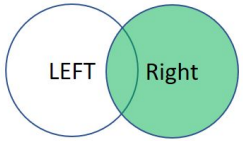
All rows from left and matching from right

Full Outer



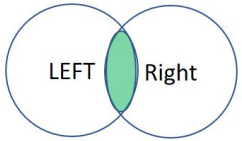
All rows from both: matching and not matching

RIGHT Outer



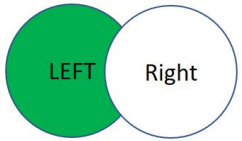
All rows from right and matching from left

Inner



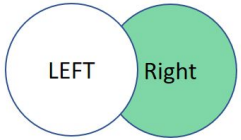
Only matching rows

Left Anti



Not matching rows from left

Right Anti



Not matching rows from right

S&P 500 Data

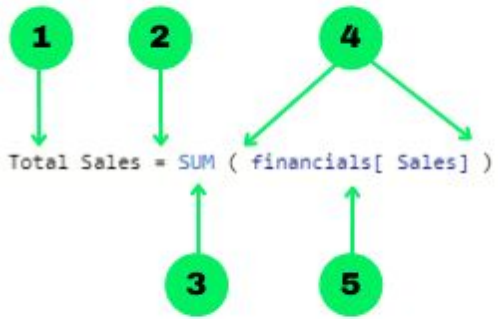
- Today we will be working with 5 years of stock data from S&P 500 companies. This data includes Open, High, Low, Close and Volume (OHLCV) metrics per day per company.
- We will use this data to analyze stock performance and build investing scenarios using DAX.

Core DAX

Writing DAX Formulas

Formulas are assembled by using:

- DAX functions
- DAX operators
- References to model objects
- Constant values, like the number 24 or the literal text "FY" (abbreviation for fiscal year)
- DAX variables
- Whitespace




DAX
FUNCTIONS
Calendar
Text
Aggregate
MIN
Date
Filter
Related
MAX
Counting
ALL

DAX Data Types

Whole number : Corresponds to a 64-bit integer in DAX. Values range from -263 to 263-1.

Decimal number : Translates to a 64-bit real in DAX. This type accommodates negative values ranging from -1.79 x 10^308 to -2.23 x 10^-308, zero (0), and positive values from 2.23 x 10^-308 to 1.79 x 10^308, limited to 17 decimal digits.

Boolean : Maps to the Boolean data type in DAX, representing either TRUE or FALSE.

Text : Corresponds to the String data type in DAX, representing a Unicode character string.

Date: Translates to the Date/time data type in DAX, with valid dates starting from March 1, 1900.

Currency: Corresponds to the Currency data type in DAX. This type covers a range from -9.22 x 10^14 to 9.22 x 10^14 and is limited to four decimal digits of fixed precision.

N/A : Represents the equivalent of a database (SQL) NULL and is referred to as BLANK in DAX.

BLANK Data Type:

The BLANK data type is a unique concept in DAX. It's used to denote both database NULL and empty cells in Excel, but it should not be confused with zero. Instead, think of it as the absence of a value.

Two DAX functions are closely related to the BLANK data type: the BLANK DAX function, which returns BLANK, and the ISBLANK DAX function, which assesses whether an expression evaluates to BLANK.

DAX Operators

Arithmetic Operators:

You can use arithmetic operators like addition (+), subtraction (-), multiplication (*), division (/), and exponentiation (^) to perform mathematical operations.

Comparison Operators:

Comparison operators allow you to compare values and return either TRUE or FALSE. They include operators like equal to (=), strict equal to (==), greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=), and not equal to (<>).

Text Concatenation Operator:

To combine two text values and create a single continuous text value, use the ampersand (&) character for text concatenation.

Logical Operators:

Logical operators are used to combine expressions that produce a single result. You can employ logical operators like AND (&&), OR (||), IN, and NOT for various logical conditions.

Operator Precedence:

In DAX, operator precedence determines the order in which operations are evaluated. This order is defined as follows:

- Exponentiation (^)
- Sign (-, as in -1)
- Multiplication and division (* and /)
- NOT
- Addition and subtraction (+ and -)
- Concatenation of text (&)
- Comparison operators (=, ==, <, >, <=, >=, <>)

Operator Type	Symbol	Example
Parenthesis	()	Determine precedence (5+2)*3=21 or 5+2*3 = 13
Arithmetic	+	4+2 = 6
	-	4 -2 = 2
	*	4*2 = 8
	/	4/2 = 2
Comparison	=	[Name] = “Jack”
	<>	[Name] <> ‘Jack’
	>	[Num] > 0
	>=	[Num] >= 0
	<	[Num] < 0
	<=	[Num] <= 0
Concatenation	&	“Jack is” & “17” will return “Jack is 17”
Logical	&&	AND
	||	OR
	!	NOT

DAX Variables

You can declare DAX variables in your formula expressions. When you declare at least one variable, a RETURN clause is used to define the expression, which then refers to the variables.

We recommend that you use variables because they offer several benefits:

- Improving the readability and maintenance of your formulas.
- Improving performance because variables are evaluated once and only when or if they're needed.
- Allowing (at design time) straightforward testing of a complex formula by returning the variable of interest.

```
VAR SalesAmount =
    SUMX ( Sales, Sales[Quantity] * Sales[
VAR NumCustomer =
    DISTINCTCOUNT ( Sales[CustomerKey] )
RETURN
    DIVIDE ( SalesAmount, NumCustomer )
```

Advanced DAX

Iterator Functions

Data Analysis Expressions (DAX) provide a set of functions called iterator functions. Iterator functions are designed to go through each row of a specified table and evaluate a given expression for each row. They offer you the ability to precisely control how your model calculations summarize data.

You're probably already familiar with single-column summarization functions like SUM, COUNT, MIN, and MAX, which work on individual columns. Each of these functions has a counterpart iterator function that has an "X" suffix, such as SUMX, COUNTX, MINX, and MAXX. These iterator functions allow you to perform more advanced calculations.

In the case of iterator functions, you need to provide a table and an expression. The table can be a reference to a model table or an expression that generates a table. The expression should produce a single value or scalar. This combination of table and expression allows you to work with detailed data while summarizing it in a way that meets your specific needs.

- **SUMX**: Calculates the sum of a specified expression for each row in a table or a table expression.
- **COUNTX**: Counts the number of rows in a table or table expression where a specified condition is met.
- **AVERAGEX**: Calculates the average of a specified expression for each row in a table or table expression.
- **MINX**: Determines the minimum value of a specified expression for each row in a table or table expression.
- **MAXX**: Finds the maximum value of a specified expression for each row in a table or table expression.
- **MEDIANX**: Calculates the median value of a specified expression for each row in a table or table expression.
- **STDEVX**.P: Computes the population standard deviation of a specified expression for each row in a table or table expression.
- **STDEVX**.S: Calculates the sample standard deviation of a specified expression for each row in a table or table expression.
- **VARX.P**: Calculates the population variance of a specified expression for each row in a table or table expression.
- **VARX.S**: Computes the sample variance of a specified expression for each row in a table or table expression.

RANKX

RANKX

The RANKX DAX iterator function is used to rank values in a column based on the values of another column. It assigns a rank to each row in a table or table expression, indicating its position in the sorted order of values in the specified column. Here's how RANKX works:

Syntax

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

<table>: This is the table or table expression containing the rows you want to rank.

<expression>: This is the DAX expression that defines the values you want to rank. The expression is evaluated for each row in the table.

<value> (optional): You can specify a value that you want to find the rank for within the sorted values. If omitted, RANKX assigns ranks to all rows in the table based on the specified column.

<order> (optional): This argument defines the sort order. You can use 1 for ascending (default) or -1 for descending.

<ties> (optional): If there are ties (rows with the same values), you can specify how RANKX should handle them. You can use one of the following options: 0 (default) for average ranking, 1 for minimum ranking, and -1 for maximum ranking.

The CALCULATE Function & Filter Context

You can use the CALCULATE DAX function to modify filter context in your formulas. The syntax for the CALCULATE function is as follows:

Syntax

```
CALCULATE(<expression>, [[<filter1>], <filter2>]…)
```

The function requires passing in an expression that returns a scalar value and as many filters as you need. The expression can be a measure (which is a named expression) or any expression that can be evaluated in filter context.

Filters can be Boolean expressions or table expressions. It's also possible to pass in filter modification functions that provide additional control when you're modifying filter context.

When you have multiple filters, they're evaluated by using the AND logical operator, which means that all conditions must be TRUE at the same time.

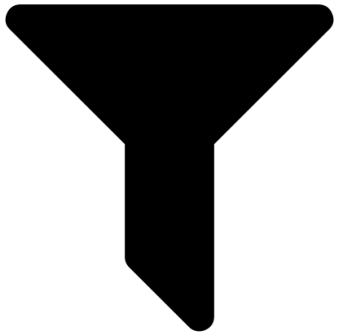
Advanced Filtering

Filters and Slicers

Filters and Slicers Overview:
Filters and slicers are the cornerstones of interactivity in Power BI. These tools are essential for refining and controlling the data presented in your visuals. Filters allow you to narrow down data based on specific criteria, while slicers provide user-friendly controls for end-users to interact with the data. Understanding the distinction between page-level filters, visual-level filters, and slicers is crucial.

Applying Basic Filters:
At its core, Power BI provides the ability to apply filters at different levels. **Page-level filters** affect all visuals on a given page. **Slicers**, which are user-friendly, customizable controls, allow end-users to dynamically filter data by making selections. We will explore how to create slicers for different data fields and enhance the user experience.

Interactive Filtering:
Filters and slicers in Power BI are interactive and flexible. They enable dynamic exploration of data. Participants will learn how to make selections, drill down into hierarchical data structures, and reset to default views, providing them with the tools to gain deeper insights and tailor their reports to specific requirements.

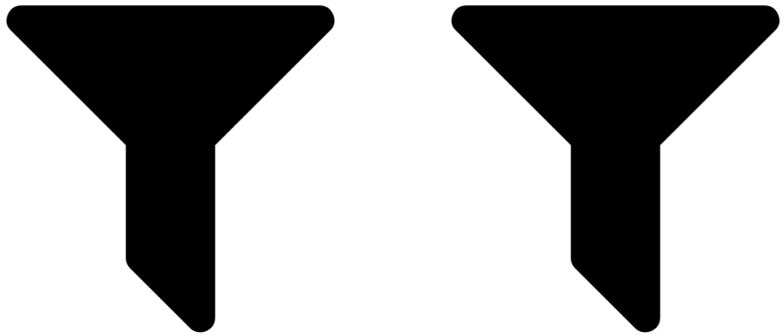


Advanced Filtering/Slicer Techniques

Visual-Level Filters:
Visual-level filters are the next step in enhancing your reporting. These filters allow you to exert granular control over individual visuals on a report page. By applying filters at the visual level, you can focus on specific aspects of your data within a single visual while keeping other visuals untouched. We'll cover scenarios where visual-level filters are particularly valuable.

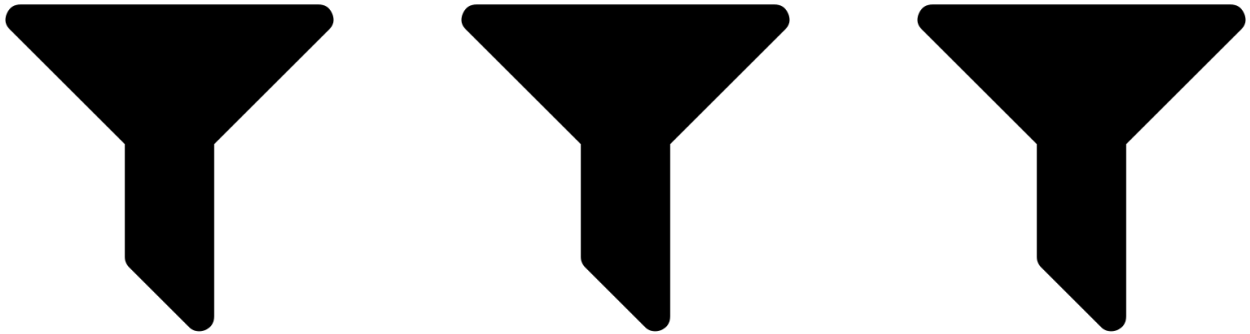
Drillthrough Filters:
Drillthrough is a powerful technique for detailed data exploration. We'll set up drillthrough pages and configure filters to enable this feature. This function empowers users to drill down into specific data points, revealing a deeper layer of information. This can be invaluable for analyzing hierarchies, examining outliers, or performing root-cause analysis.

Query Reduction Techniques:
Query reduction techniques are essential for optimizing performance. We will delve into the importance of query reduction in Power BI. Techniques like query folding, which minimizes data transfer between Power BI and the data source, improve report loading times and responsiveness. Understanding these techniques is crucial for ensuring smooth report performance.



Syncing Slicers

You can also sync two or more separate slicers. Syncing slicers is useful when working with composite models, as you might want to make the same selection across sources without relying on cross-source group relationships. To sync two or more separate slicers, you mark them as being part of a group.



Advanced Report Design

Basic Design Principles

Drillable visuals

The simplest way to show details is to use drillable visuals. The matrix visual is an excellent choice because it allows drilling on rows and/or columns. Therefore, by assigning hierarchies or multiple fields, report consumers can drill to the level of detail that they want. For example, the matrix rows could show years, and the report consumer could drill down to quarter, month, and date levels.

Visual tooltips

Many Power BI visuals include a Tooltips well. Fields that are added to this well is summarized in a tooltip. For example, a column chart visual that shows sales revenue by month could include order quantity as a tooltip. While the height of the column reveals the monthly revenue amount, when a report consumer hovers over the column, a tooltip describes the order quantity for that month.

Page tooltips

Page tooltips allow your report consumers to gain deeper insights quickly and efficiently from visuals. As its name suggests, a page tooltip is a report page that receives the filter context of the source element. In this way, they are closely related to drillthrough pages, with the difference being that the page tooltip overlays a small page of visuals.

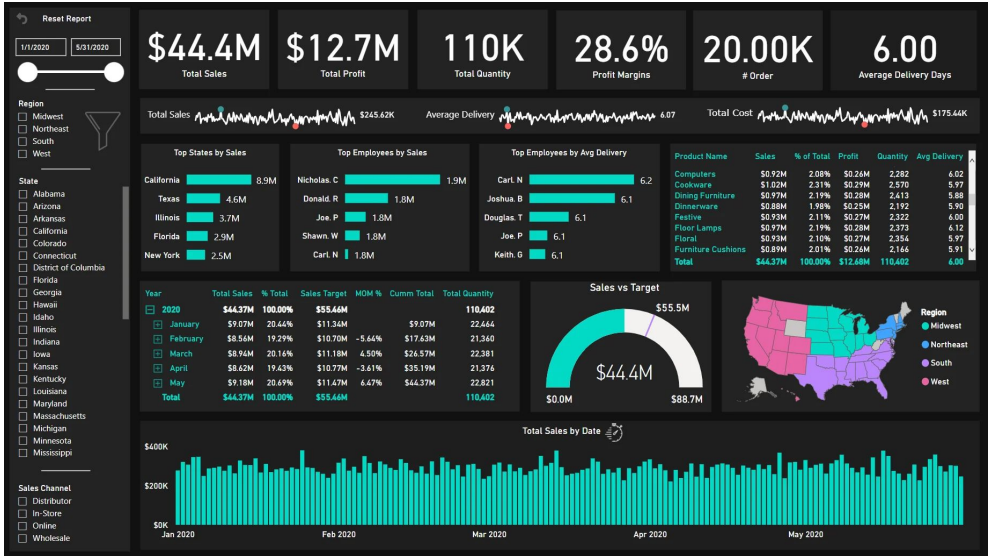
Conditional formatting

You can use conditional formatting to format table and matrix visuals. With this approach, you can apply rules to highlight specific cells by using:

- Background color
- Font color
- Data bars
- Icons

Overlaid analytics

You can also overlay analytics to highlight values. Certain visuals, like the line chart visual, allow you to add analytic options. Options include trend lines, constant lines, minimum or maximum lines, and many others.



Visual Headers

Toggle	Description	Icon	Considerations
Visual information	Notifies the user of a possible issue or an error with the visual		
Visual warning	NA	NA	
Visual error	NA	NA	
Drill on dropdown	Select the axis on which to apply the drill		Hierarchical data - Matrix
Drill up	Drill up one level in the hierarchy		Hierarchical data
Drill down	Drill down one field at a time, click the icon to turn the drill on		Hierarchical data
Show next level	Drill down all fields at once and go to the next level in the hierarchy		Hierarchical data
Expand to next level	Expand all fields at once and go one level down in the hierarchy		Hierarchical data
Pin	Pin visual to a dashboard		
Focus mode	Expand (pop out) the visual to see more detail		
See data layout	NA	NA	
More options	View the additional options available for the visual		
Filter	Displays the list of filters and slicers applied on the visual		
Help tooltip (aka Visual header tooltip in Service)	Show a text or report page tooltip to assist users		
Comment button	NA	NA	The toggle disappears if comments are disabled in the report settings
Personalize visual	Enter the personalization mode to modify the visual		The toggle disappears if personalization is disabled in the report settings
Copy	Copy the visual as an image to your clipboard		The icon greys out for unsupported visuals, e.g., Maps, Python & R visuals etc. The toggle disappears if the feature is disabled by the admin

Buttons

The **Back** action navigates to the previous page. It's useful in drillthrough scenarios, allowing the report consumer to quickly return to where they drilled from.

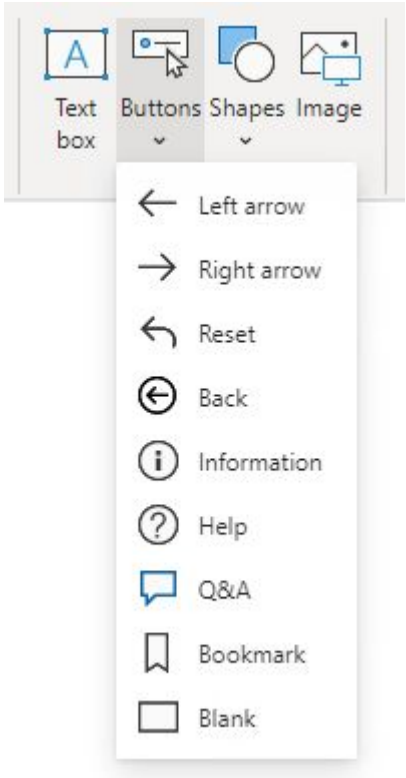
The **Bookmark** action activates a selected bookmark. Bookmarks can turn a Power BI report into a guided analytical experience, maximize available page real estate, and provide user-friendly interactions. Bookmarks are covered in the next unit.

The **Drillthrough** action is assigned a target drillthrough destination page. The button remains disabled until drillthrough becomes a valid action, which is the case when you are interacting with a visual that can navigate to the drillthrough page. When the Drillthrough action is selected, Power BI navigates to the drillthrough page, propagating appropriate filters.

The **Page** navigation action directs the report consumer to a specific report page. The page can be a specific page that is assigned at design time or a measure that returns the page name. Using a measure allows Power BI to dynamically determine the page based on the filter context.

The **Q&A** action opens a pop-up window that allows the report consumer to explore data by using intuitive, natural language capabilities and receive answers as data visualizations.

The **Web URL** action opens the URL by using the default web browser. Like the Page navigation action, this action can be a specific URL or one that is returned by a measure. The measure can produce a URL that appends filter context to the query string. For example, if the report consumer filters the page by a single customer, the measure can return a URL to an external system that includes the customer key in the query string.

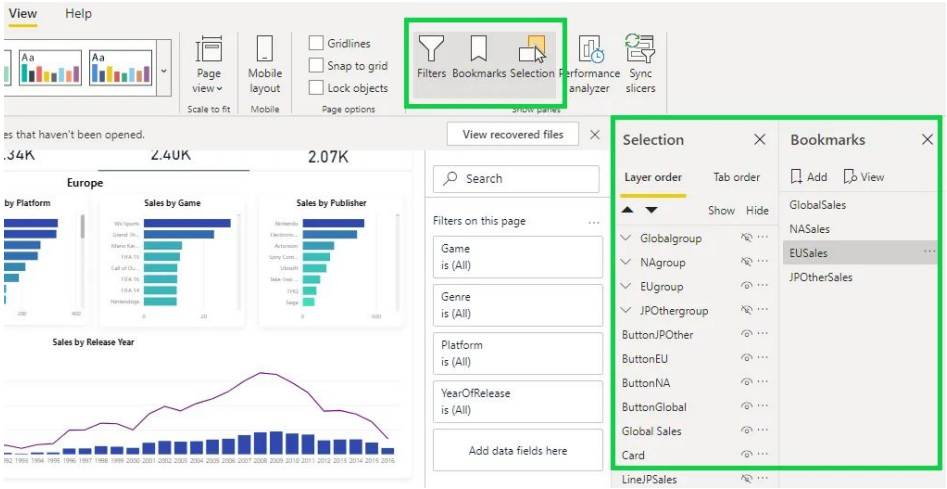


Bookmarks

The **Data** state captures anything that impacts the queries that Power BI sends to the dataset. For example, if a slicer is included in the scope of the bookmark, the Data state retains the applied slicer items when the bookmark was created (or updated). It will also capture sort order and the drill depth of a visual because the query is impacted.

The **Display** state is related to the visibility of a report object. Objects consist of visuals and also elements like text boxes, buttons, shapes, and images. By using the Selection pane, you can hide or unhide objects and groups of objects. Additionally, you can swap visuals on a report page by creating bookmarks that capture hidden and unhidden objects.

The **Current** page state determines whether the bookmark will direct the report consumer to the bookmarked page or apply the current page. Disabling the Current page state is rare, but you should consider some creative use cases. For example, on a page tooltip, a bookmark can change the visuals without navigating from the page that the report consumer has selected.



Day 2 Capstone

In this capstone project, you will work with Netflix titles, user and behavior data to build a Power BI report.

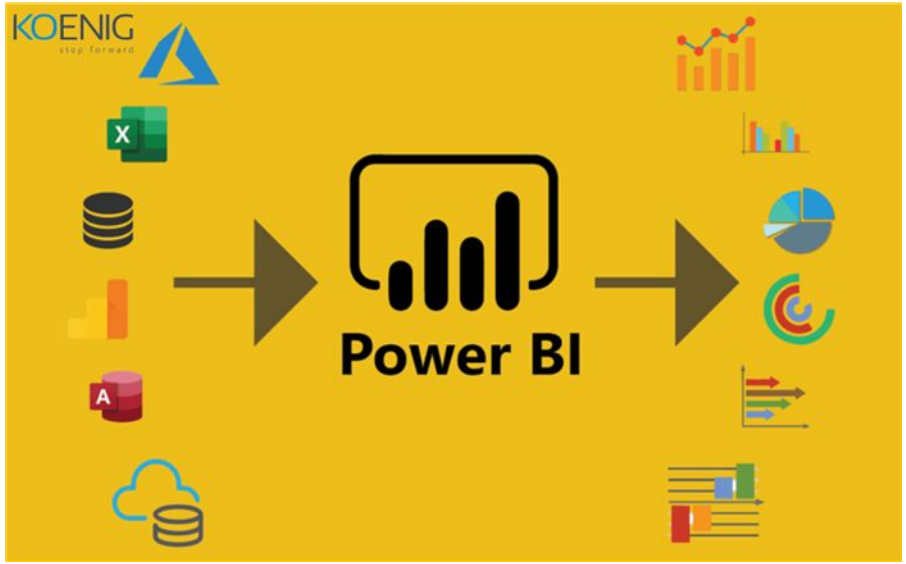
The project is designed to reinforce the concepts and skills learned on the second day of the course, covering DAX, filtering and advanced report design.

Recap and Introduction to the Data Lifecycle

Final Capstone

For your final capstone you will be expected to find a new data set and take your a Power BI project from creation to deployment.

Your instructor will support you along the way! Look to sites like https://www.kaggle.com/ for inspiration and resources.



Deployment Strategy, Source Control and Data Lifecycle Management

Application Lifecycle Management

Application Lifecycle Management (ALM) is a well-established practice in software development, which can also be effectively applied to Power BI. By implementing ALM principles in Power BI, you ensure consistency, data integrity, and an enhanced user experience in your reports.

Plan and Track
- Project planning and progress tracking

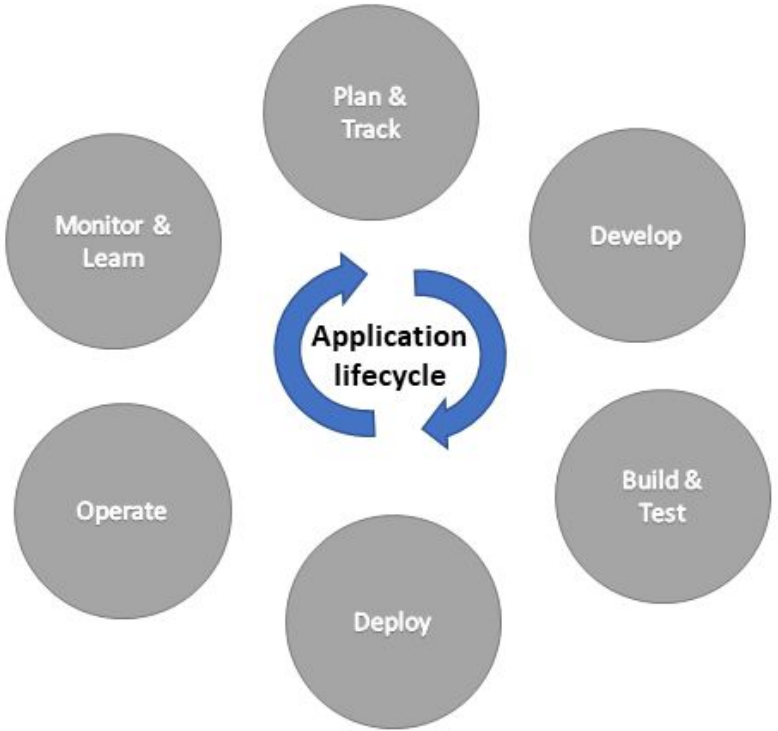
Develop
- Identifying and addressing current issues
- Application design and development
- Comprehensive testing

Build and Test
- Building and testing the application
- Ensuring robust code quality

Deploy
- Deploying the application to production environments

Operate
- Managing and operating the deployed application

Monitor and Learn
- Continuous monitoring and learning from user interactions



Choose an Effective Source Control Strategy for Power BI

OneDrive for Business or SharePoint:

- Ideal for organizations with users familiar with traditional folder structures.
- Suitable for .pbix file tracking.
- Enables version tracking and team collaboration.
- Manual conflict resolution may be needed if multiple users update the same file.

Git:

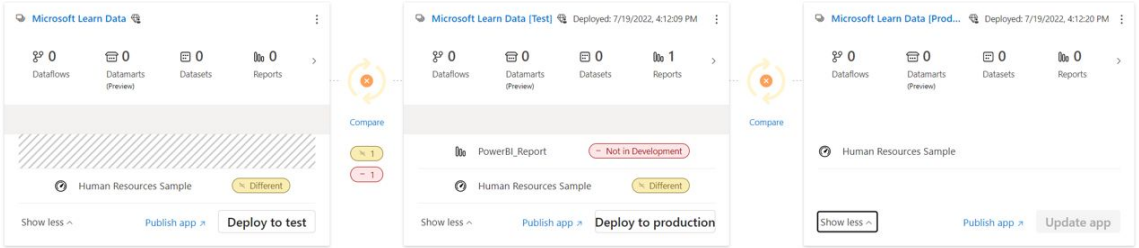
- Suitable for .bim files used in dataset development.
- Offers robust version control with branching, merging, and change tracking.
- Integrates with Git platforms like GitHub and Azure DevOps.
- Not ideal for tracking changes within .pbix files due to their binary nature.
- May increase repository size for large .pbix files, even with large-file storage (LFS) option.

Design an Efficient Deployment Strategy

In large organizations, dealing with numerous reports and datasets can be overwhelming. Application lifecycle management (ALM) recommends distinct environments for development, testing, and production. To streamline this process and save time, you need a well-thought-out deployment strategy.

Power BI Service Deployment Pipelines:

- Deployment pipelines in Power BI service create distinct workspaces for development, test, and production.
- They help identify discrepancies between workspaces quickly.
- Note that Power BI deployment pipelines require a Premium or Premium Per User workspace.



Sharing with Power BI

Band-Aid Sales Data

The marketing/sales team at Band-Aid ran a targeted campaign this summer at Target stores in California.

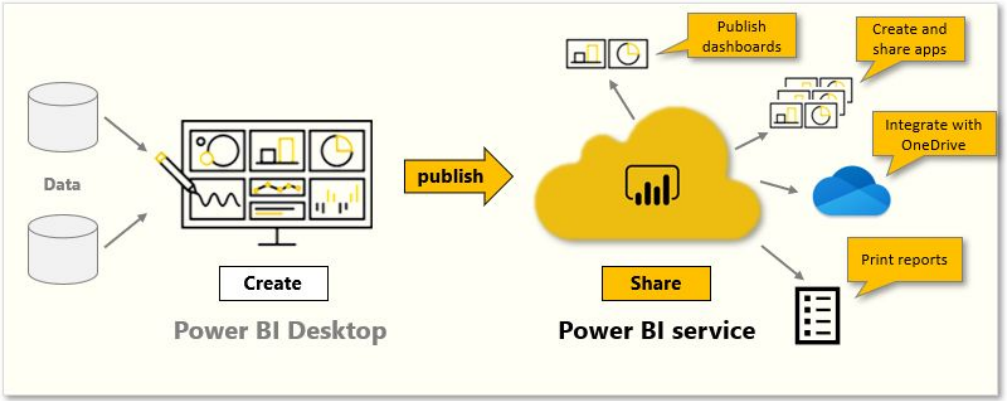
They want to analyze the sales data during this campaign and understand the most popular products, the most successful locations, the most effective sales teams and other general performance metrics of the campaign.

We are being asked to process several files and create a set of dashboards that will allow analysts and executives to easily identify trends.

Sharing with Power BI

Power BI offers various ways of sharing and collaborating with colleagues on your dashboards, reports, and datasets including:

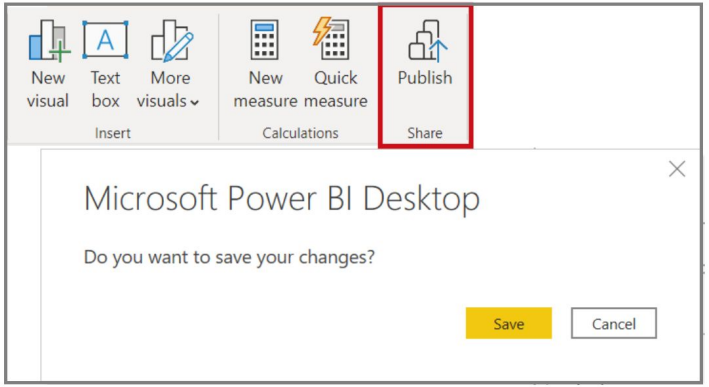
- Publish your reports from Power BI Desktop to the Power BI service.
- Create *Apps* that package a dashboard, report, and dataset to send to your colleagues, who can use the content pack as a starting point and further enhance it.
- Create *Groups*, which you can use as a security model to identify a subset of users who have access to dashboards, reports, and datasets that you create.
- Publish to the web so that live reports can be embedded in a webpage.



Publishing

Sharing your reports in the Power BI service is a quick and straightforward process.

1. Once you've finished creating your report, head to the Home tab and simply click the Publish button.
2. Power BI will bundle your report, complete with visualizations, queries, and custom measures, and swiftly transmit it to the Power BI service.
3. When the upload is finished, a notification will pop up, confirming the successful publication, and offering a direct link to access your report within the Power BI service.



Working with Reports in Power BI

Power BI Service is the cloud-based platform where your meticulously crafted reports come to life. Here, your data visualizations become interactive, shareable, and accessible to a broader audience. Let's explore the essential features and capabilities that make viewing reports in Power BI Service an indispensable experience:

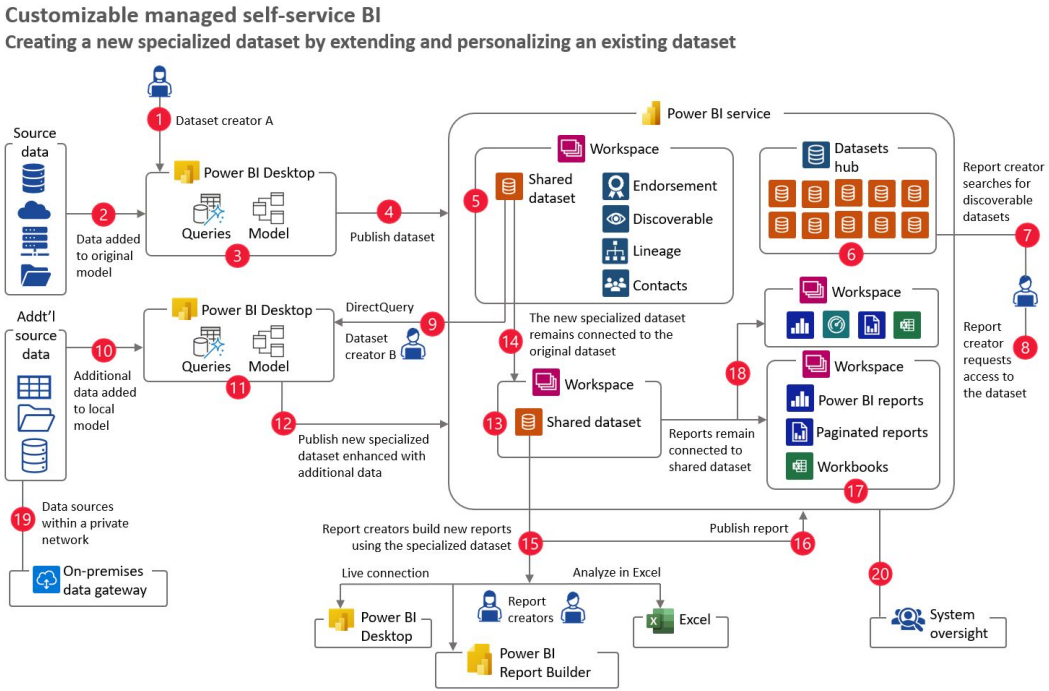
Viewing reports in Power BI Service is more than just static data. It's an interactive and collaborative experience that enables you to share, explore, and act on your data like never before. With a focus on accessibility, customization, and security, Power BI Service empowers you to harness the full potential of your data for better decision-making and business success.

Setting Up a Deployment Pipeline in Power BI Service

Creating a deployment pipeline in Power BI is a fundamental step in managing your reports and datasets efficiently. You can create a deployment pipeline through two different methods, provided you meet the necessary prerequisites:

Once you've successfully created a deployment pipeline, you'll be directed to a page where you can assign a workspace. Its important that this is a PRO level workspace and that you are an administrator in the workspace.

Please note that if you don't see the "Deployment Pipelines" tab in the left navigation panel or if there's no option to create deployment pipelines in your workspace, it's likely because you haven't met the prerequisites required for this feature.



Capstone

Implementing Row Level Security in Power BI

Row-level security (RLS) with Power BI can be used to restrict data access for given users. Filters restrict data access at the row level, and you can define filters within roles. In the Power BI service, users with access to a workspace have access to datasets in that workspace. RLS only restricts data access for users with Viewer permissions. It doesn't apply to Admins, Members, or Contributors.

You can configure RLS for data models imported into Power BI with Power BI Desktop. You can also configure RLS on datasets that are using DirectQuery, such as SQL Server. For Analysis Services or Azure Analysis Services lives connections, you configure Row-level security in the model, not in Power BI Desktop. The security option doesn't show up for live connection datasets.