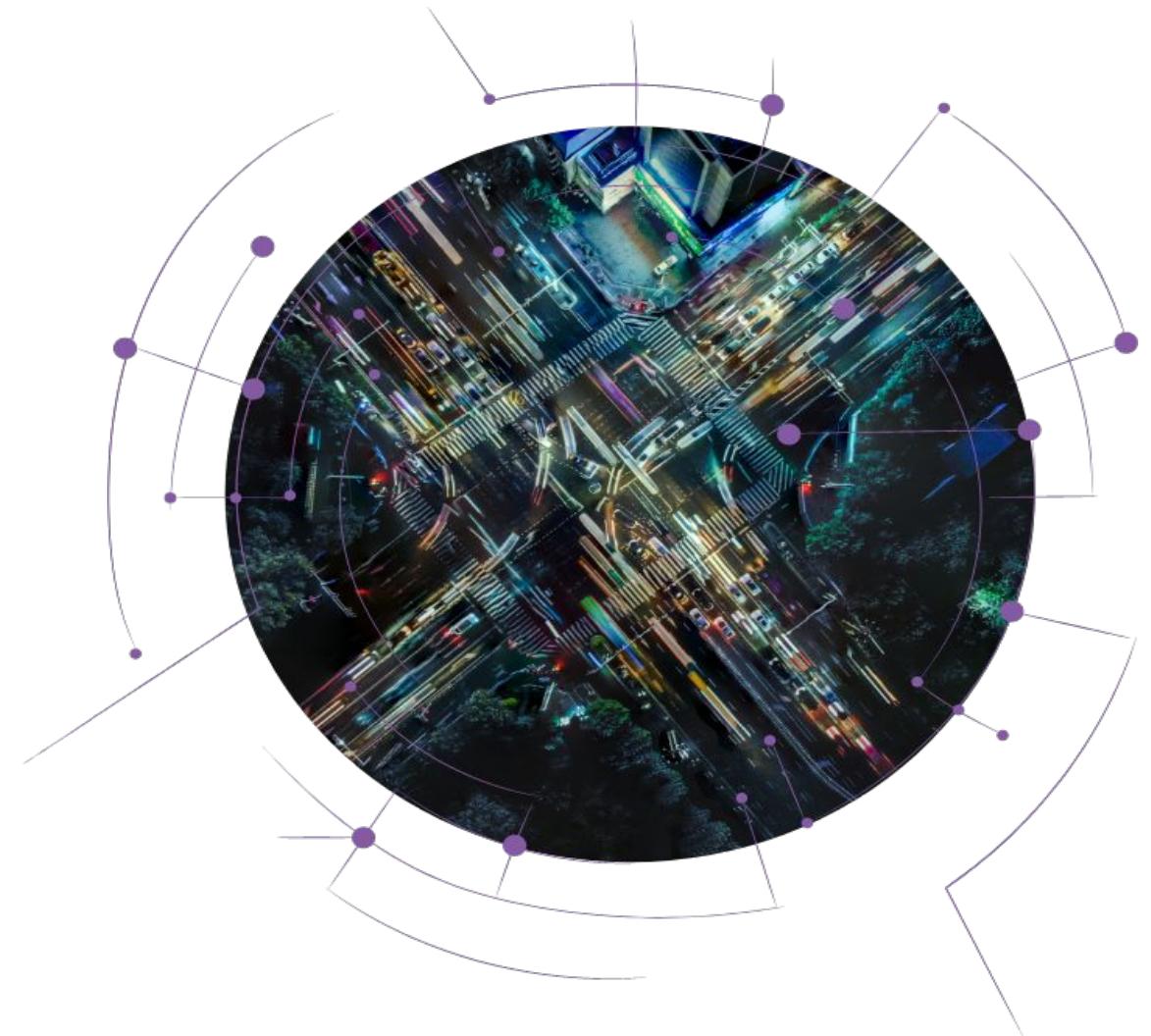


DATA SOCIETY:

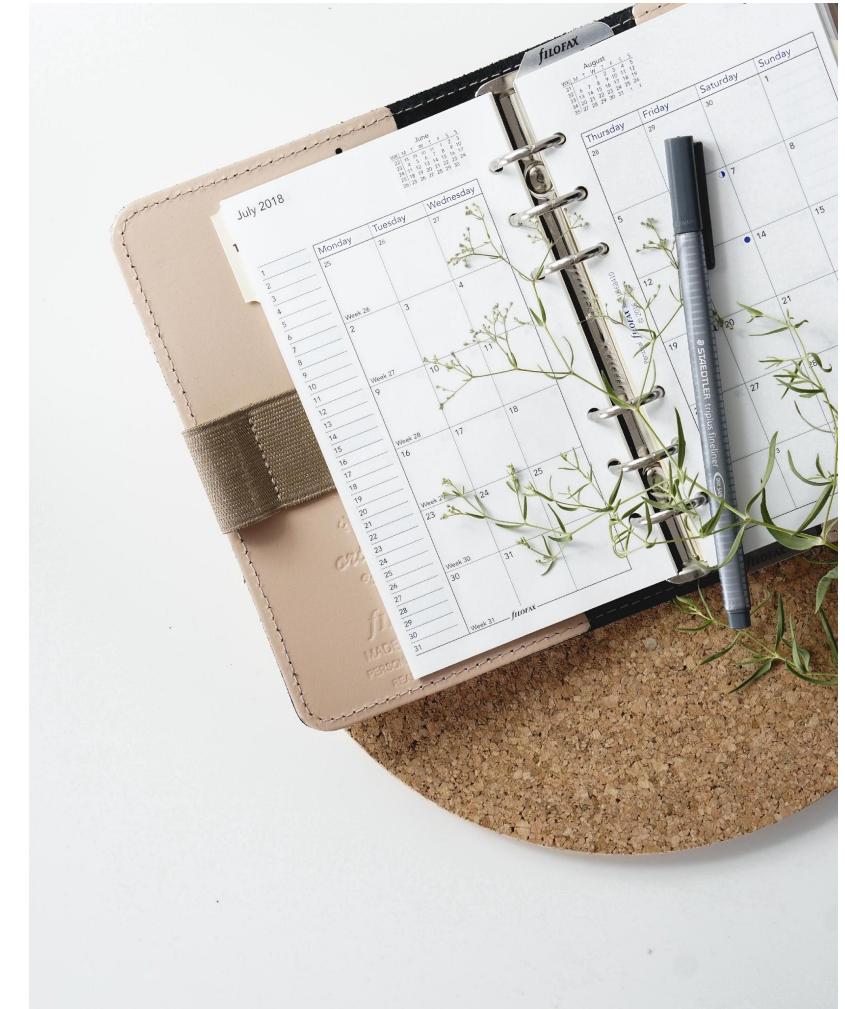
Power BI Intermediate

Day 1



Welcome!

- Hours and breaks
 - 4 days
 - 11 am - 2 pm
 - 1-2 short breaks each class
- Materials
 - PDF slide decks, Datasets, lab and exercise files



Best practices for virtual classes

- Find a quiet place, free of as many distractions as possible. Headphones are recommended.
- Remove or silence alerts from cell phones, e-mail pop-ups, etc.
- Participate in activities and ask questions. This will be interactive!



Who we are

Data Society's mission is to integrate Big Data and machine learning best practices across entire teams and empower professionals to identify new insights.

We provide:

- High-quality data science training programs
- Customized executive workshops
- Custom software solutions and consulting services

Since 2014, we've worked with thousands of professionals to make their data work for them.



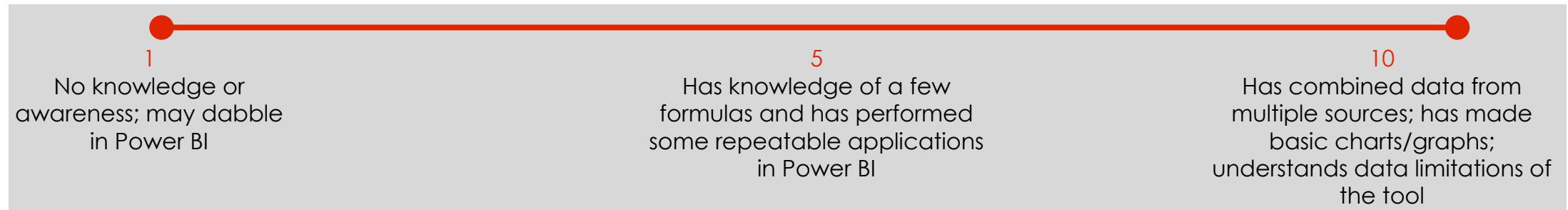
Pre-Work for PowerBI

If needed, review the following:

- [Overview of Power BI Desktop](#) - 4 minutes
- [Getting started with Power BI Desktop](#) - 9 minutes
- [Connect to data sources in Power BI Desktop](#) - 8 minutes

Chat question

- What do you rate your current Power BI literacy level on a scale of 1-10?
- Share your answers in the chat.



Setting Expectations

- This course is for users who have a pre-existing working knowledge of Power BI with analytical experience using Power BI Desktop and service
- Each lesson will be accompanied by a Lab and knowledge checks
- This is **not** a DAX centric course but is designed for Business Intelligence report development
- Students are advised to explore the additional resources to further skill development beyond this course

Agenda

Day 1

- Power BI Overview
- Getting data from different sources
- Clean Transform & Load Data

Day 2

- Design a Data Model
- Create Model Calculations using DAX

Day 3

- CALCULATE Function and Modifiers
- Time Intelligence DAX Patterns

Day 4

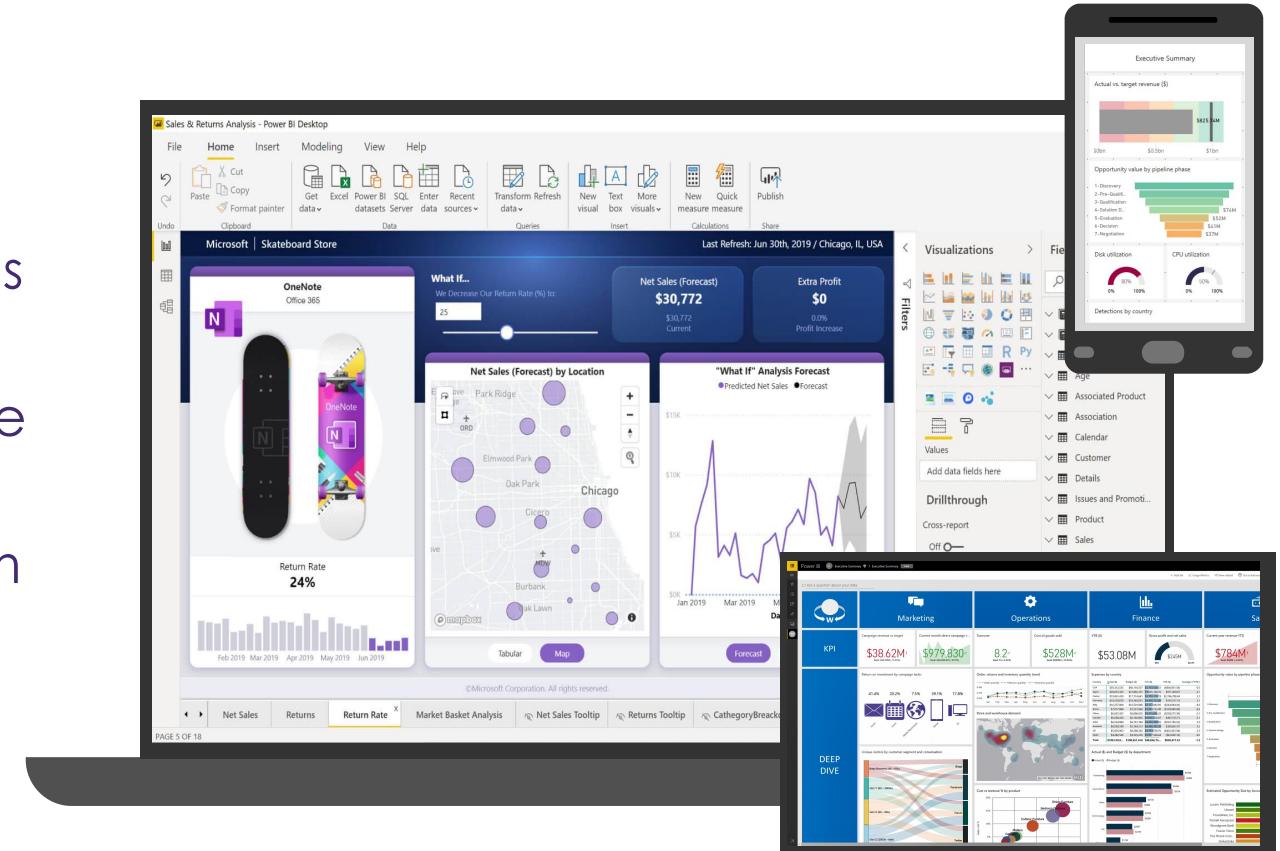
- Advanced Report Design Principles
- Row-Level Security
- Power BI App Deployment

Agenda

- Power BI Overview
- Getting data from different sources
- Clean Transform & Load Data

Power BI Overview

- Connect to all your data and get a consolidated view across your business through a single pane of glass
- Create ad-hoc analysis, live dashboards and interactive reports that are easy to consume on the web and across mobile devices
- Build smart apps by infusing insights from your data and drive action with the power of Microsoft Power Platform

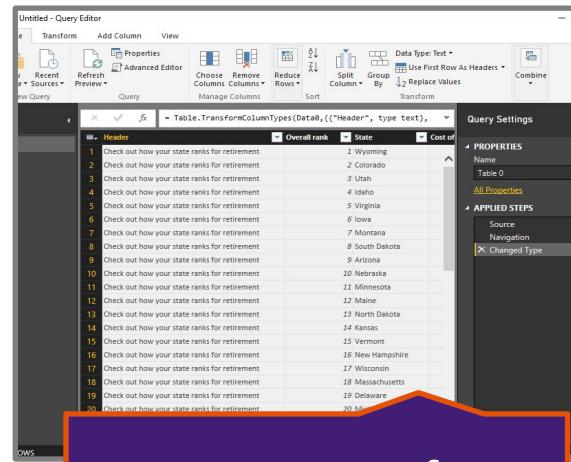


NEW Analytics Stack Microsoft Fabric

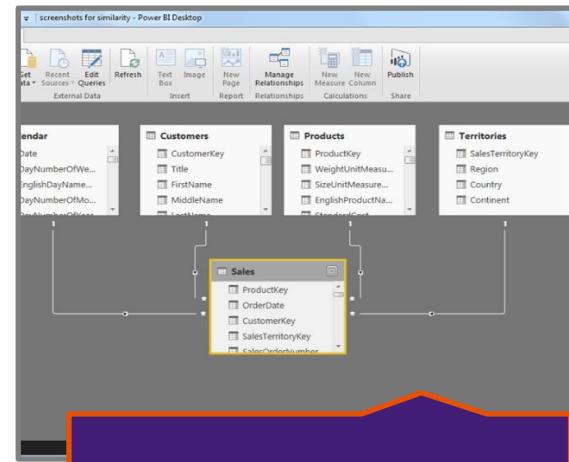


4 Layers of Power BI

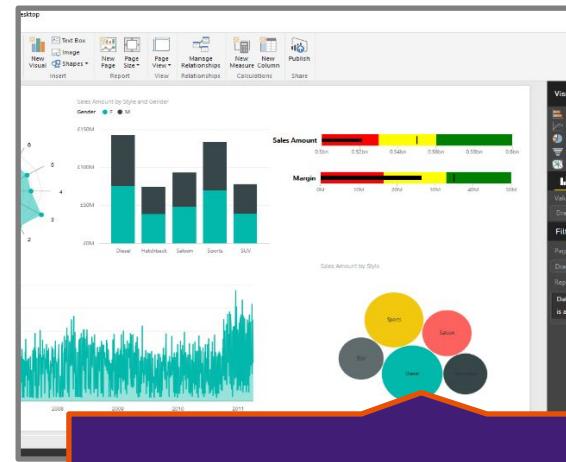
Model Developer



Extract, Transform
and Load (ETL)

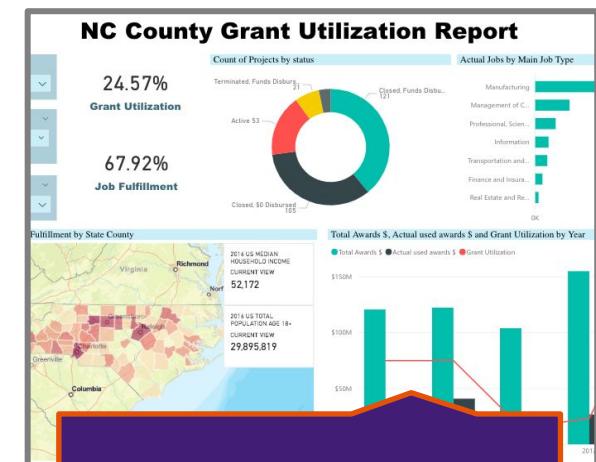


Data Modeling



Report Design

Report Developer



Web Portal

Introduction to the Data Sets



For our live exercises, we will be using the SuperStore data set. This is a retail business data set that provides us with both facts and dimension tables allowing us to analyze key transactional information regarding the business performance.

Fact Tables

Consumer
Corporate
Home Office
Sales Forecast

Dimension Tables

Category
Sub-Category
Product
Sales Rep
Sales Rep Profile



For the Lab exercises, we will be using the Power Cycle data set. This is a Global fictitious Cycle Retailer and your Job will be to use the data set to complete the lab based on the instructions provided in the lab resource materials.

Fact Tables (Sales Table)

Australia
Canada
France
Germany
UK
US
Power Cycle Target

Dimension Tables

Product Category
Product Sub-Category
Product Sales Region
Salesperson

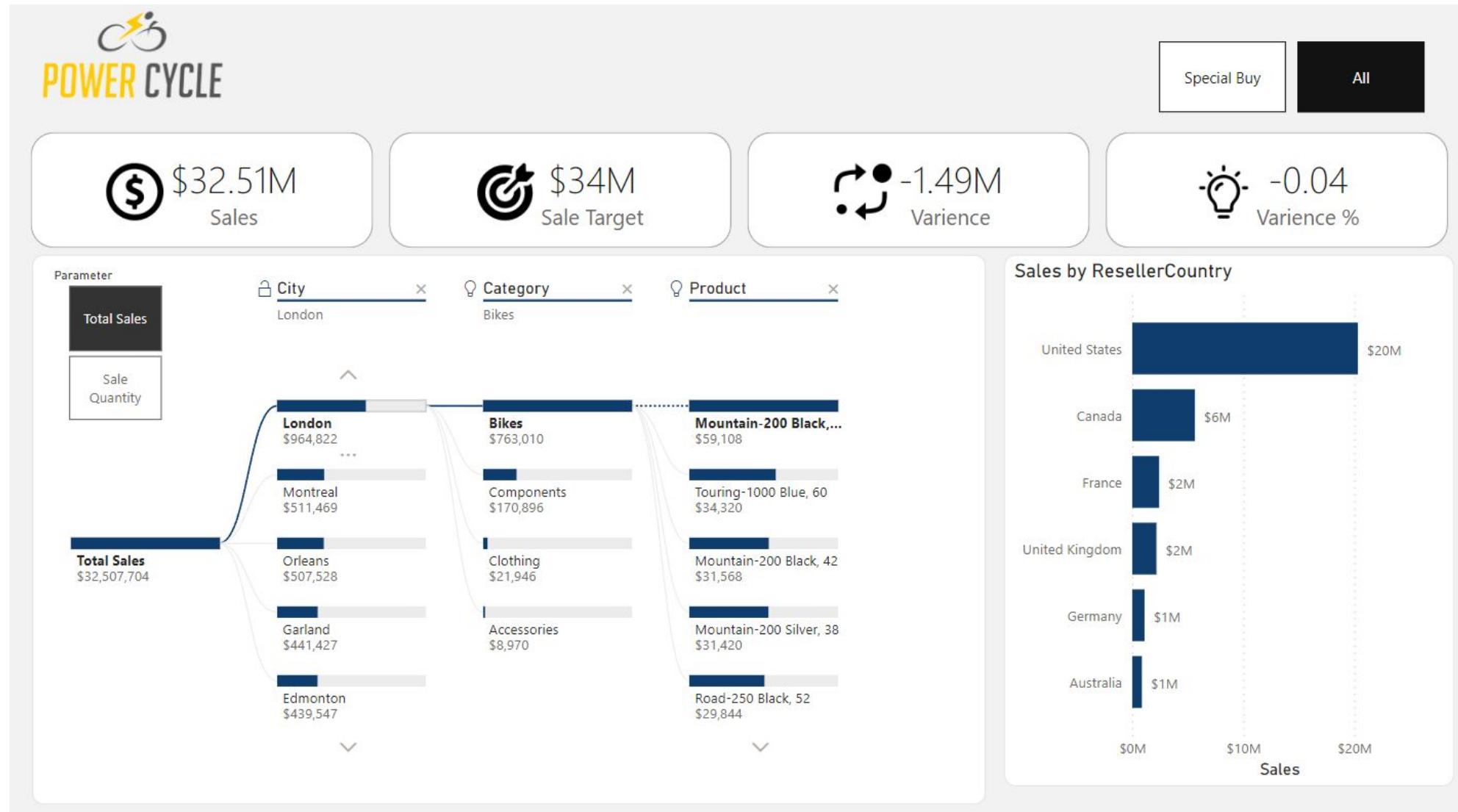
Introduction to the Lab Exercises



1. The data set is from a fictitious cycling retailer with an international footprint. Your goal as the analyst is to use Power BI and DAX calculations to answer key questions about the business including sales trends, top performing resellers, actual sales vs. budget and segmentation.
2. There will be a total of 6 Lab to be completed. Each Lab will build upon the previous one
3. All you've been given is a collection of raw csv files containing POS sales records, along with details about the company's products, customers, locations and employees.

**This data is adapted from Microsoft Adventure works data pack and is for informational purposes only . These samples are provided "as is" without warranty of any kind. The example companies, organizations, products, domain names, email addresses, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, person, place, or event is intended or should be inferred.*

Final Report



Employee RLS Page

Amy Alberts
DESKTOP-2ASG5KB\shannonl

Salesperson

- Amy Alberts
- Brian Welker
- David Campbell
- Garrett Vargas
- Jae Pak
- Jillian Carson
- José Saraiva
- Linda Mitchell
- Lynn Tsotfias
- Michael Blythe
- Pamela Anzman-Wolfe
- Rachel Valdez
- Ranjit Varkey Chudukatil
- Shu Ito
- Stephen Jiang
- Syed Abbas
- Tete Mensa-Annan
- Tsvi Reiter

Sum of Quantity by Year and Month Short Name

Month Short Name	Sum of Quantity
Aug 2018	~300
Nov 2018	~50
Feb 2019	~100
May 2019	~150
Jun 2019	~180
Jul 2019	~10
Aug 2019	~450
Sep 2019	~120
Oct 2019	~10
Nov 2019	~150
Dec 2019	~550
Jan 2020	~10
Feb 2020	~150
Mar 2020	~100
Apr 2020	~10
May 2020	~10

ResellerCountry State Category Subcategory Product Sum of Total Sales

ResellerCountry	State	Category	Subcategory	Product	Sum of Total Sales
France	Essonne	Bikes	Road Bikes	Road-250 Black, 44	\$1,466
France	Essonne	Bikes	Road Bikes	Road-250 Black, 48	\$1,309
France	Hauts de Seine	Bikes	Mountain Bikes	Mountain-200 Black, 38	\$3,687
France	Hauts de Seine	Bikes	Mountain Bikes	Mountain-200 Black, 42	\$2,458
France	Hauts de Seine	Bikes	Mountain Bikes	Mountain-200 Silver, 42	\$2,486
France	Hauts de Seine	Clothing	Gloves	Full-Finger Gloves, L	\$69
France	Hauts de Seine	Clothing	Gloves	Full-Finger Gloves, S	\$23
France	Nord	Accessories	Bike Racks	Hitch Rack - 4-Bike	\$648
France	Nord	Accessories	Bottles and Cages	Water Bottle - 30 oz.	\$18
France	Nord	Accessories	Cleaners	Bike Wash - Dissolver	\$25
Total					\$712,681

Lesson 1

Getting data from different sources

After completing this module, students will be able to:

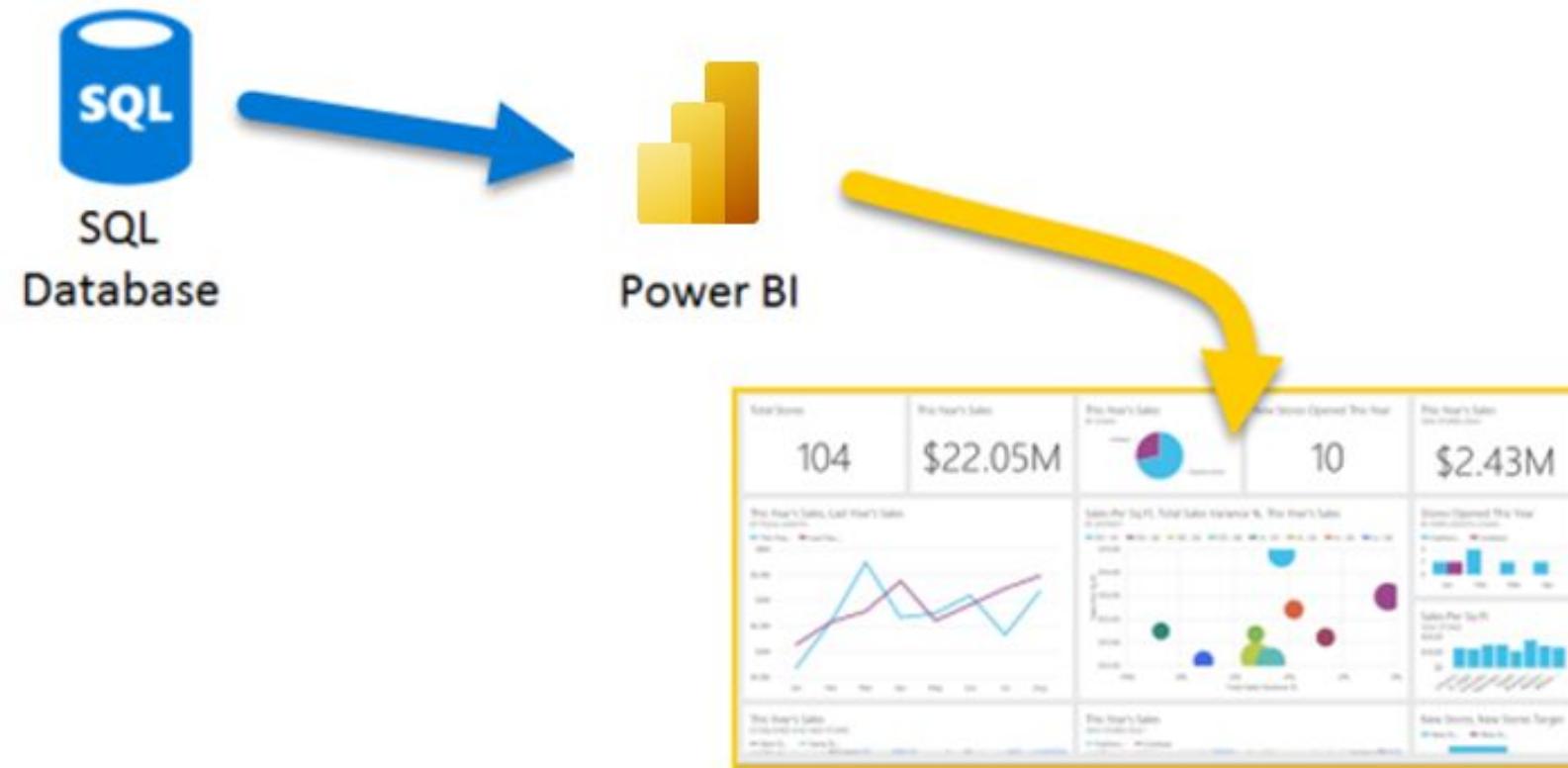
- Identify and retrieve data from different data sources
- Understand the connection methods and their performance implications
- Connecting to SharePoint and Power BI services
- Understanding storage modes

Source Types

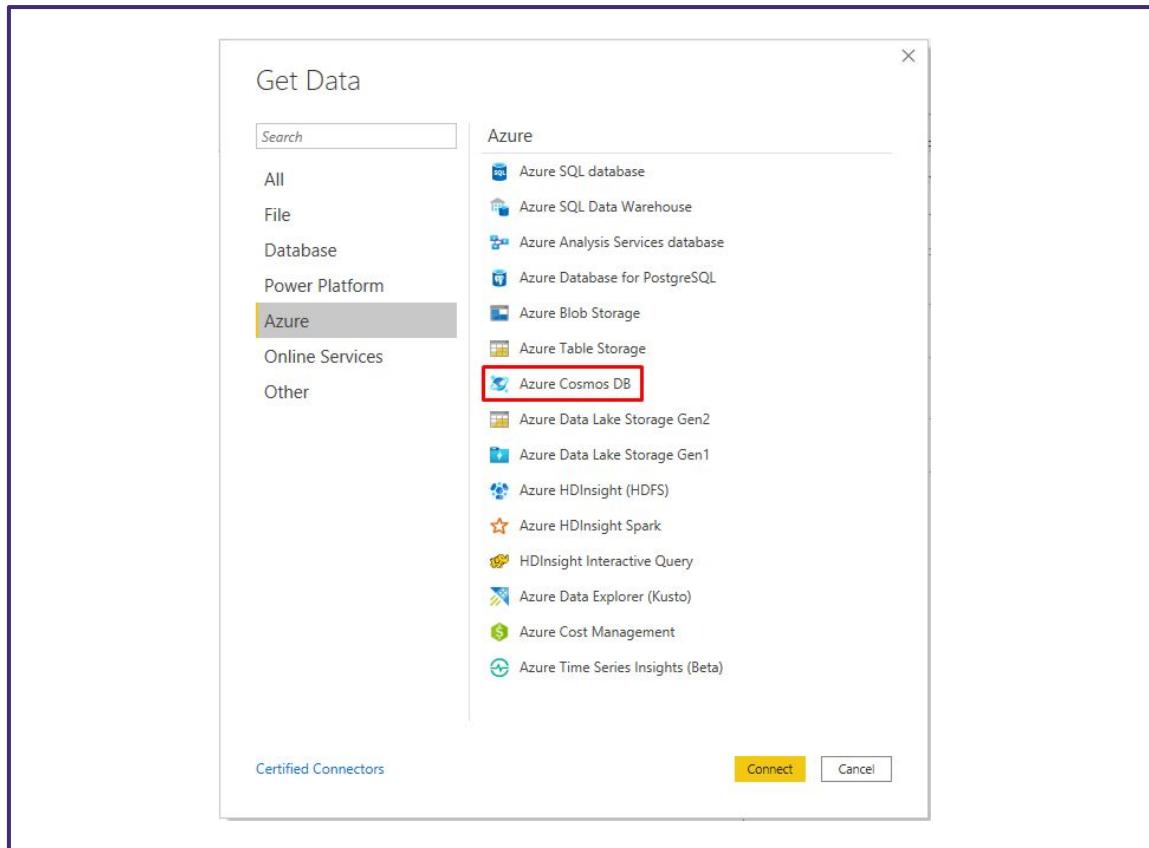
- Flat File
- Relational Data Sources
- Non-SQL Data (unstructured)
- Online Services
- API's



Get data from relational data sources



Unstructured Data Source



The screenshot shows the 'Module 3 - Power Query Editor'. The ribbon tabs are Home, Transform, Add Column, View, Tools, and Help. The 'File' tab is active, showing the 'Close & Apply' icon highlighted with a red box. The 'Queries [9]' pane on the left lists: Category, Sales, Territory, Order, Budget, Country, New Country, Product, and Employee Data (highlighted with a yellow box). The main area displays a table with columns: Department (A_B), Extension (1²3), and Position Title (A_C). The data rows are:

	A _B Department	1 ² 3 Extension	A _C Position Title
1	MARKETING		425 Marketing Advisor
2	MARKETING		206 Marketing Advisor
3	MARKETING		207 Brand Manager
4	MARKETING		349 Senior Brand Manager
5	MARKETING		425 Marketing - Coordina...
6	MARKETING		210 Marketing - Coordina...
7	MARKETING		208 Marketing Consultatant
8	MARKETING		249 Marketing Consultatant
9	OPERATIONS		425 Supervisor

Getting Data From SharePoint

The screenshot shows the Power BI desktop interface. On the left, a navigation pane lists 'Anonymous', 'Windows', and 'Microsoft account'. A red box highlights the 'Sign in' button in the Microsoft account section. The main area displays a 'SharePoint' connection dialog with the URL 'https://themeasuredproduct.sharepoint.com/'. A red box highlights the 'Sign in' button. To the right, the 'Navigator' window is open, showing a hierarchical list of SharePoint items under the root site. The 'BudgetRequests' item is selected and highlighted with a red box. The 'BudgetRequests' table is displayed in the preview pane, showing five rows of data.

FileSystemObjectType	Id	ServerRedirectedEmbedUri	ServerRedirectedEmbed
0	1		null
0	2		null
0	3		null
0	4		null
0	5		null

Get data from Analysis Services

An analytical data engine that lets you digest data from multiple data sources and create calculations on the fly.

SQL Server Analysis Services database

Server ⓘ

asazure://westus.asazure.windows.net/azureanalysisservicesiketest

Database (optional)

adventureworks

Import

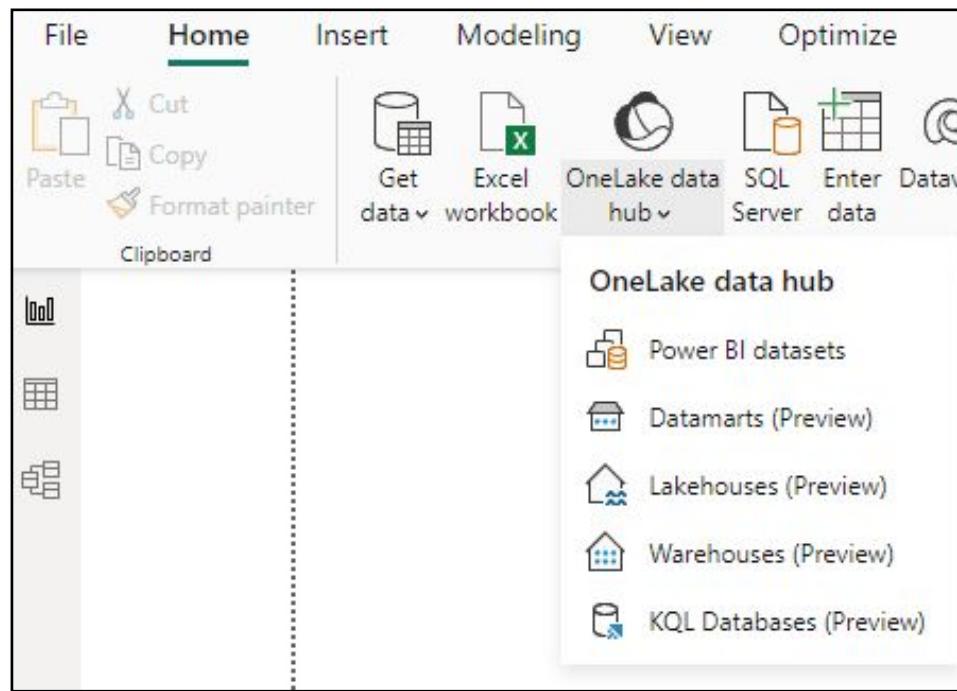
Connect live

▷ MDX or DAX query (optional)

OK

Cancel

Power BI Published Dataset



This screenshot shows a modal dialog box titled 'OneLake data hub'. The header includes the title and a message: 'Discover data from across your org and use it to create reports'. Below the header are three filter buttons: 'All' (selected), 'My data', and 'Endorsed in your org'. To the right are two search/filter buttons: 'Filter by keyword' and 'Filter(1)'. The main area is a table listing datasets. The columns are: Name, Refreshed, Location, Endorsement, and Sensitivity. Each dataset row includes a small preview icon, the dataset name, its last refresh time, its location, its endorsement status (e.g., 'Promoted'), and its sensitivity level. At the bottom right of the dialog are 'Connect' and 'Cancel' buttons.

Name	Refreshed	Location	Endorsement	Sensitivity
Lab 6 -Advanced Visualization...	9/17/23, 10:56:37 PM	Power BI Intermediate Tr...	Promoted	-
Lab 7 -RLS	8/17/23, 3:04:07 PM	Power BI Intermediate Tr...	-	-
Purview Hub	9/17/23, 11:27:12 PM	Admin monitoring	-	-
Sales Report	9/12/23, 2:50:06 PM	Smart Dolphins Demo	-	-
CRM Analytics for Dynamics 3...	9/5/23, 11:52:58 PM	Agile CRM Analytics for ...	-	-
VAS_API 8.7.23	8/11/23, 5:26:57 PM	My Workspace	-	-
NC Grants Lab 5 Complete	7/14/23, 4:07:22 PM	Power BI Training	-	-

Web Services & API's

The screenshot illustrates the Power BI Data Flow interface for transforming data from a web service.

From Web dialog (top):

- Basic (selected) Advanced
- URL: `http://api.openmetrolinx.com/OpenDataAPI/api/v1/Gtfs/Feed/VehiclePosition`
- OK Cancel

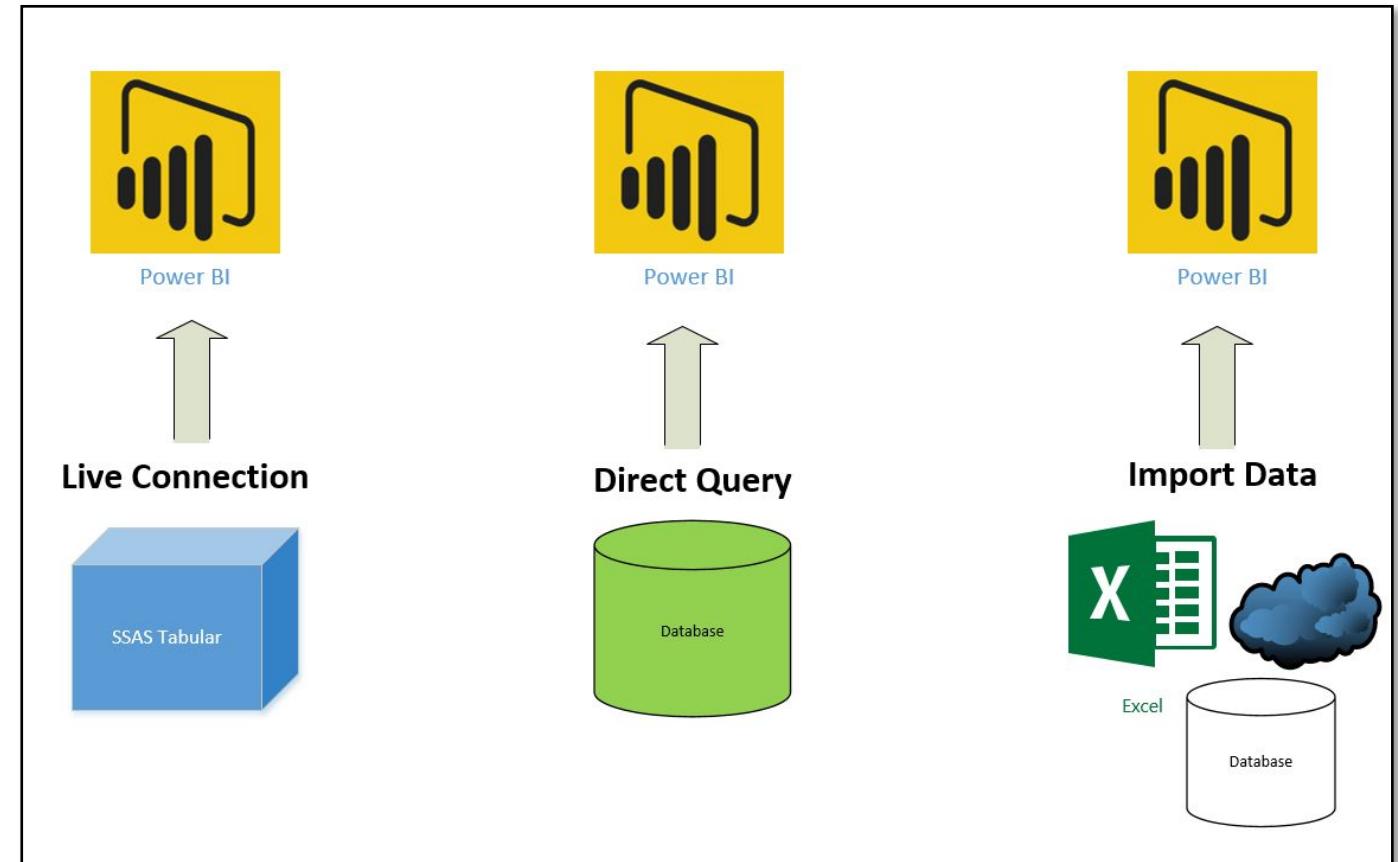
Data Flow Editor (bottom):

- Query: `= Json.Document(Web.Contents("http://api.openmetrolinx.com/OpenDataAPI/api/v1/Gtfs/Feed/VehiclePosition?key=30020173"))`
- Header: Record
- Entity: List
- Table View (Preview):
 - Columns: trip.trip_id, entity.vehicle.trip.route_id, entity.vehicle.trip.direction_id, entity.vehicle.trip.start_time, entity.vehicle.trip.start_date, entity.vehicle.trip.schedule_relationship, entity.vehicle.position
 - Row 1: 100%, 0%, 0% (Valid, Error, Empty), 100%, 0%, 0% (Valid, Error, Empty)
 - Row 2: 09231223-41, 35 distinct, 3 unique, 2 distinct, 0 unique, 0, 6:30:00 AM, 20230918, SCHEDULED
 - Row 3: 09231223-41, 35 distinct, 3 unique, 2 distinct, 0 unique, 1, 6:55:00 AM, 20230918, SCHEDULED
 - Row 4: 09231223-56, 35 distinct, 3 unique, 2 distinct, 0 unique, 0, 7:00:00 AM, 20230918, SCHEDULED
 - Row 5: 09231223-41, 35 distinct, 3 unique, 2 distinct, 0 unique, 0, 7:00:00 AM, 20230918, SCHEDULED
 - Row 6: 09231223-56, 35 distinct, 3 unique, 2 distinct, 0 unique, 0, 7:10:00 AM, 20230918, SCHEDULED
 - Row 7: 09231223-33, 35 distinct, 3 unique, 2 distinct, 0 unique, 0, 7:25:00 AM, 20230918, SCHEDULED
 - Row 8: 09231223-56, 35 distinct, 3 unique, 2 distinct, 0 unique, 1, 7:25:00 AM, 20230918, SCHEDULED
 - Row 9: 09231223-41, 35 distinct, 3 unique, 2 distinct, 0 unique, 1, 7:25:00 AM, 20230918, SCHEDULED
- Properties (right):
 - Name: VehiclePosition?key=30020173
 - All Properties
- Applied Steps (right):
 - Source
 - Converted to Table
 - Expanded header
 - Expanded entity
 - Expanded entity1
 - Expanded entity.vehicle
 - Expanded entity.vehicle.trip
 - Expanded entity.vehicle.vehicle
 - Expanded entity.vehicle.position
 - Changed Type

Storage Modes

The three different types of storage modes you can choose from:

- Import
- DirectQuery (Small data size)
- Dual (Composite)



Implications of using DirectQuery

Benefits

- Frequently changing data
- Need near real-time
- Large data volumes
- Multi-dimensional data

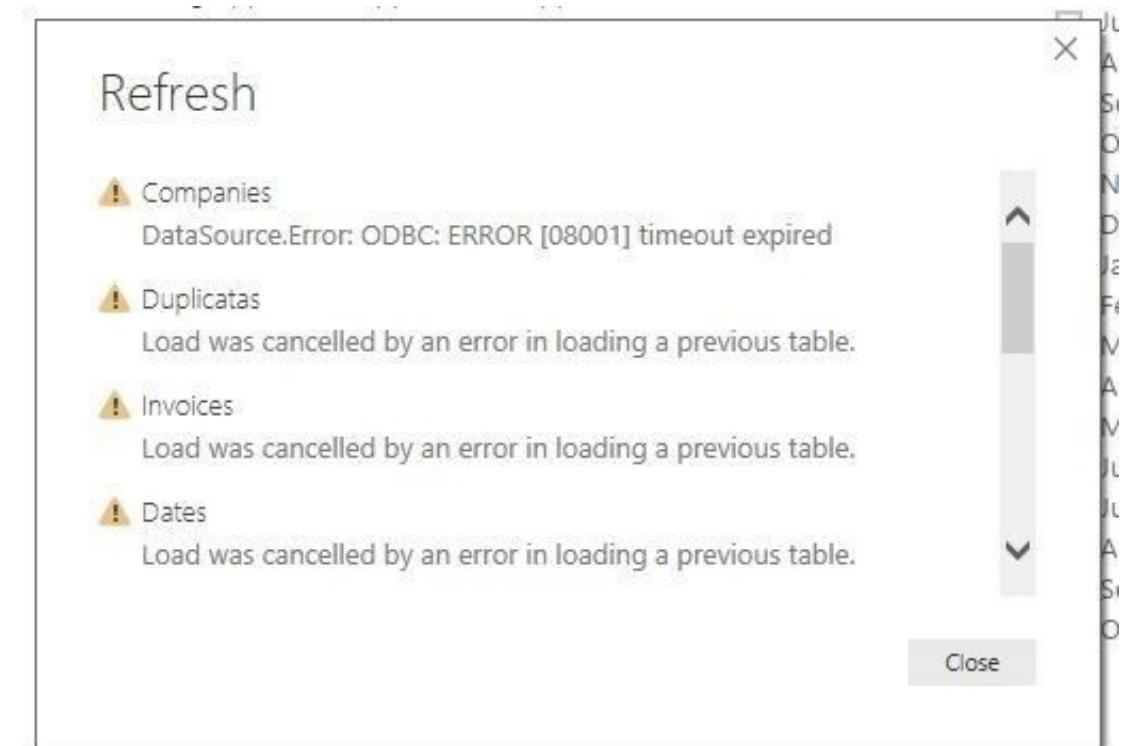
Limitations

- Performance: Dependent on the underlying data source
- Security: Understand how data moves between source and destination
- Modeling: Some modeling capabilities are limited or aren't supported
- Transformation: Some data transformation techniques are limited

Dealing with Import Errors

When loading your data, you may encounter the following errors:

- *Query Timeout*
- *Couldn't find data formatted as a table*
- *Could not find file*
- *Data type errors*



Chat Question

What are some common data import errors do you encounter when creating reports?



Optimize Query Performance



Performance in Power Query depends heavily on the performance at the data source.

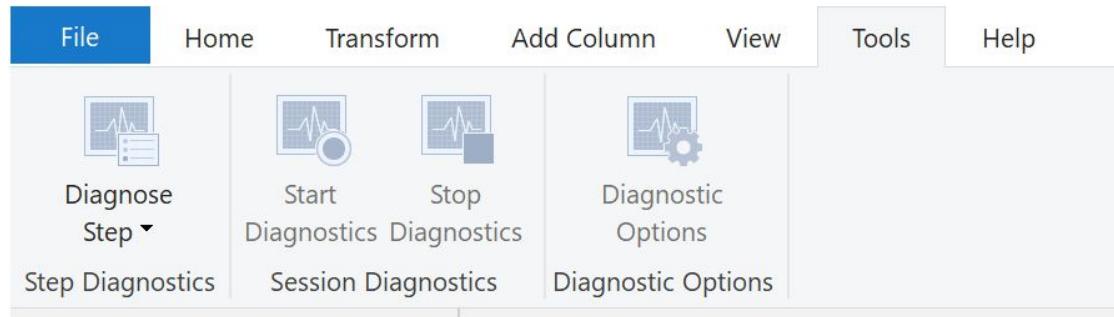


Follow performance tuning guidelines of the source product.



Some performance tuning can be done in Power BI.

Fix Performance Issues



A ^B Id	A ^B Query	A ^B Step	A ^B Exclusive Duration	A ^B Category	A ^B Data Source Kind
1.1	Product	Changed Type	0.00:00:00.0457545	Evaluator	null
1.2	Product	Source	0.00:00:01.9567741	Evaluator	null
Diagnostics_2020-0...					

Query Folding

The figure consists of three screenshots of the Microsoft Power Query Editor interface, illustrating the process of generating native SQL queries.

- Screenshot 1:** Shows the 'Applied Steps' pane on the left. A context menu is open over the 'Changed Type' step, with the 'View Native Query' option highlighted by a red box.
- Screenshot 2:** Shows the 'Query Settings' dialog box. A context menu is open over the 'Renamed Columns' step, with the 'View Native Query' option highlighted by a red box.
- Screenshot 3:** Shows the 'Native Query' dialog box, which displays the generated SQL query. A red arrow points from the second screenshot to this one, indicating the final output of the query folding process.

Native Query

```
select [$Outer].[StateID] as [StateID],  
[$Outer].[StateCode] as [State Code],  
[$Outer].[StateName] as [State],  
[$Outer].[TimeZone] as [Time Zone],  
[$Inner].[RegionName] as [Region]  
from  
(  
    select [...].[StateID] as [StateID],  
    [...].[StateCode] as [StateCode],  
    [...].[StateName] as [StateName],  
    [...].[RegionID] as [RegionID2],  
    [...].[TimeZone] as [TimeZone]  
    from [dbo].[State] as [...]  
) as [$Outer]  
left outer join [dbo].[Region] as [$Inner] on ($Outer].[RegionID2] = [$Inner].[RegionID]
```

OK

The process that lets Power Query generate a single query statement to retrieve and transform source data.

Knowledge Check 1



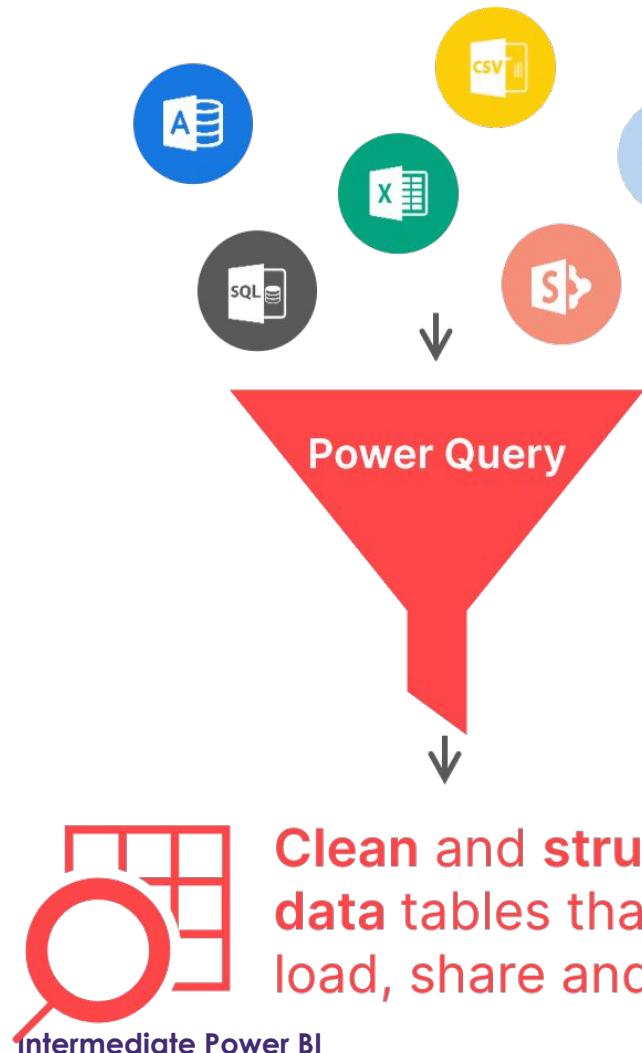
Lesson 2

Clean Transform, Load the Data

After completing this module, students will be able to:

- Apply data shape transformations
- Enhance the structure of the data
- Profile and examine the data

Power Query Editor



Before

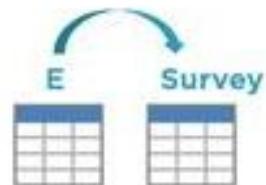
	Date	Customer	Prod 1	Kg 1	Bag 1	Prod 2	Kg 2	Bag 2
1	1/1/2019	A	Rice	10	1	Potato	100	10
2	1/2/2019	A	Potato	20	2	Rice	200	20
3	1/3/2019	B	Tomarto	30	3	Beans	300	30
4	1/4/2019	C	Beans	40	4	Beans	400	40

After

	Date	Customer	Product	Kg	Bag
1	2019-01-01	A	Potato	100	10
2	2019-01-01	A	Rice	10	1
3	2019-01-02	A	Potato	20	2
4	2019-01-02	A	Rice	200	20
5	2019-01-03	B	Beans	300	30
6	2019-01-03	B	Tomarto	30	3
7	2019-01-04	C	Beans	440	44

ETL Tools & Utility

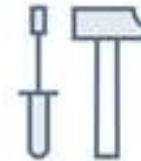
Data Transformation



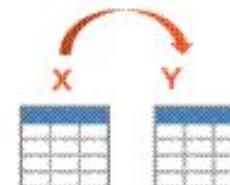
Rename objects



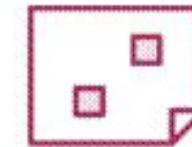
Combine queries



Fixing metadata



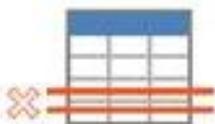
Object Names



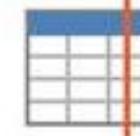
Hidden Objects



Sort Order



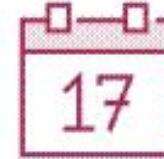
Filtering rows



Eliminating columns



Adding columns



Data Types & Categories



Format



Summarization

Power BI Transform ribbon

- File, Home, Transform (selected), Add Column, View, Tools, Help
- Table, Any Column, Text Column, Number Column, Date & Time Column, Scripts
- Icons: Transpose, Reverse Rows, Group By, Use First Row as Headers, Count Rows, Detect Data Type, Fill, Rename, Pivot Column, Convert to List, Move, Unpivot Columns, Split Column, Format, Merge Columns, ABC 123 Extract, abc Parse, Statistics, Standard, Scientific, Trigonometry, Rounding, Date, Time, Duration, Run R script, Run Python script.

Power Query Editor ribbon

- File, Home, Transform, Add Column, View, Tools, Help
- General, From Text, From Number, From Date & Time, AI Insights
- Icons: Column From Examples, Custom Column, Invoke Custom Function, Conditional Column, Index Column, Duplicate Column, Merge Columns, ABC 123 Extract, abc Parse, Statistics, Standard, Scientific, Trigonometry, Rounding, Date, Time, Duration, Text Analytics, Vision, Azure Machine Learning, AI Insights.

Power Query Interface

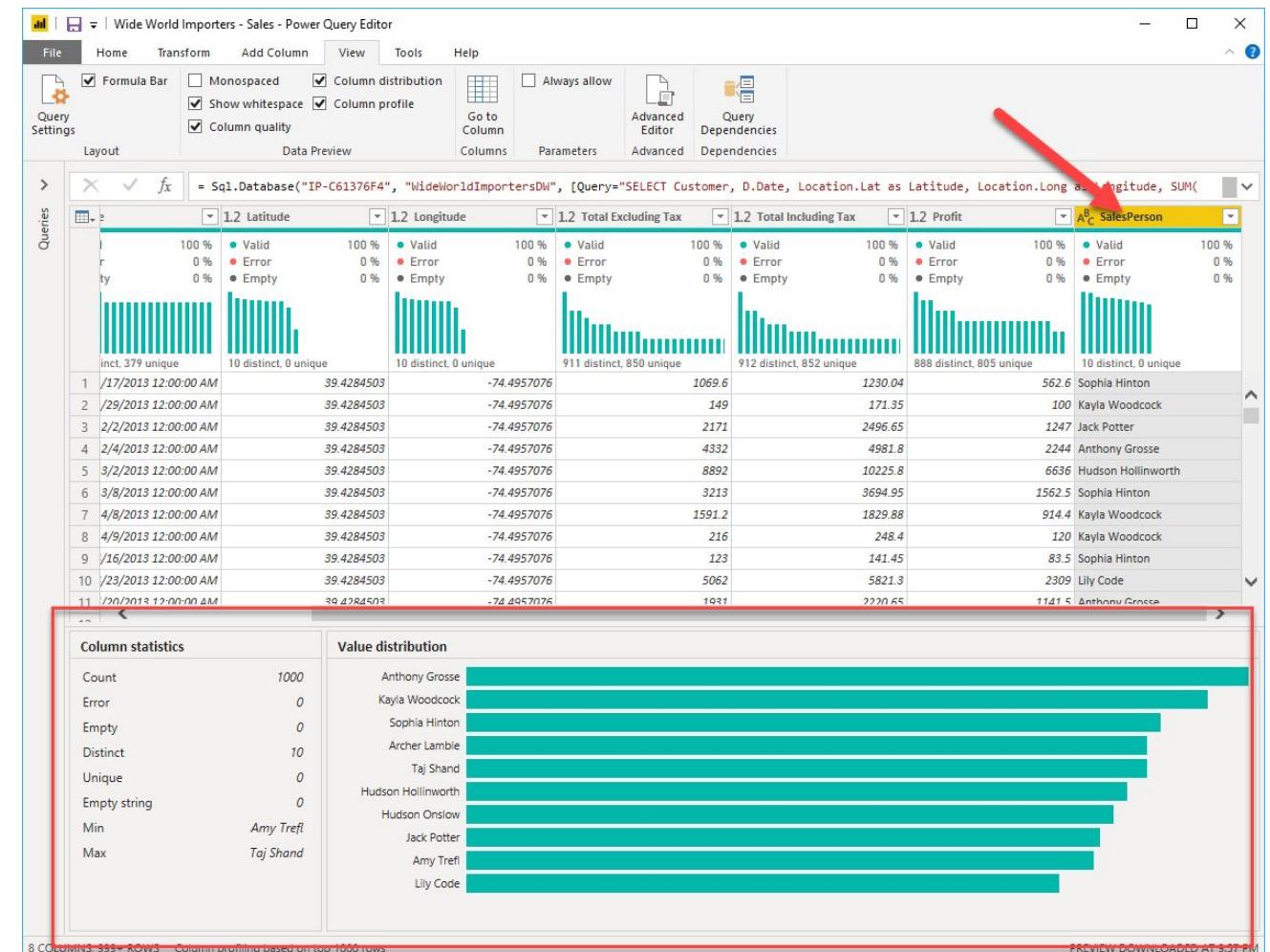
The screenshot illustrates the Power Query interface with several key components highlighted by red boxes and arrows:

- Left Pane (Queries List):** A sidebar on the left lists "Queries [13]" under categories like "Transform File from Country Files [1]", "Other Queries [9]", and "Targets". A red box and arrow point to this area, with the text: "Queries are listed and available for selection, viewing, and shaping".
- Ribbon:** The top bar contains the ribbon with tabs like File, Home, Transform, Add Column, View, Tools, and Help. A red box and arrow point to the ribbon area, with the text: "In the ribbon, buttons are active to interact with the data in the query".
- Center Pane:** The main workspace displays a table with columns EmployeeID, Date, and Target. The table shows 18 distinct EmployeeIDs, 54 distinct dates, and 58 distinct Targets. Data summary statistics are provided for each column. A red box and arrow point to this area, with the text: "In the center pane, data from the selected query is displayed and available for shaping".
- Query Settings Window:** A floating window titled "Query Settings" lists "Properties" (Name: Targets) and "Applied Steps". The "Applied Steps" section includes steps like Source, Navigation, Changed Type, Removed Top Rows, Promoted Headers, Changed Type1, Unpivoted Other Columns, Changed Type2, and Renamed Columns. A red box and arrow point to this window, with the text: "The Query Settings window appears, listing the query's properties and applied steps".

Find data anomalies and data statistics

- Data profiling entails studying data nuances, detecting anomalies, exploring underlying structures, and querying statistics like row counts and value distributions.
- This is crucial for simplifying data interaction and streamlining the development of report elements on the frontend.

Note: Data Preview is based on the first 1,000 rows



Combine multiple queries into one

Append

Concatenate rows from three or more tables into a single table.

Two tables Three or more tables

Available tables

Production Suppliers
Sales Customers
HR Employees

Add >>

Tables to append

Production Suppliers
Sales Customers
HR Employees

OK

Merge

Select a table and matching columns to create a merged table.

Sales Orders

orderid	custid	empid	orderdate	requireddate	shippeddate	shipperid	freight	shipname
10248	85	5	7/4/2014	8/1/2014	7/16/2014	3	32.38	Ship to 85-B
10249	79	6	7/5/2014	8/16/2014	7/10/2014	1	11.61	Ship to 79-C
10250	34	4	7/8/2014	8/5/2014	7/12/2014	2	65.83	Destination SCQ
10251	84	3	7/8/2014	8/5/2014	7/15/2014	1	41.34	Ship to 84-A

Sales OrderDetails

orderid	productid	unitprice	qty	discount
10248	11	14.00	12	0
10248	42	9.80	10	0
10248	72	34.80	5	0
10249	14	18.60	9	0
10249	51	42.40	40	0

Join Kind

Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

Fuzzy matching options

✓ The selection matches 830 of 830 rows from the first table.

OK Cancel

Splitting Columns

The screenshot illustrates the process of splitting a single column into multiple columns in Power Query. On the left, a table shows a single column named "Reseller CityState". This column contains city and state names separated by a hyphen. A red box highlights this column. A large green arrow points from the original table to the result on the right. The result shows three new columns: "Reseller", "Reseller CityState.1", and "Reseller CityState.2". The "Reseller" column lists various resellers. The "Reseller CityState.1" column contains the first part of the city-state pair (e.g., Rhodes, Newcastle). The "Reseller CityState.2" column contains the second part (e.g., New South Wales). Each of these three columns has its own summary statistics and a bar chart.

Reseller	Reseller CityState.1	Reseller CityState.2
Nationwide Supply	Rhodes	New South Wales
Popular Bike Lines	Newcastle	New South Wales
Twin Cycles	Darlinghurst	New South Wales
Bike Part Wholesalers	Sydney	New South Wales
Eastside Cycle Shop	Melbourne	Victoria
Second Bike Shop	Matraville	New South Wales
Seaside Bike Works	North Sydney	New South Wales

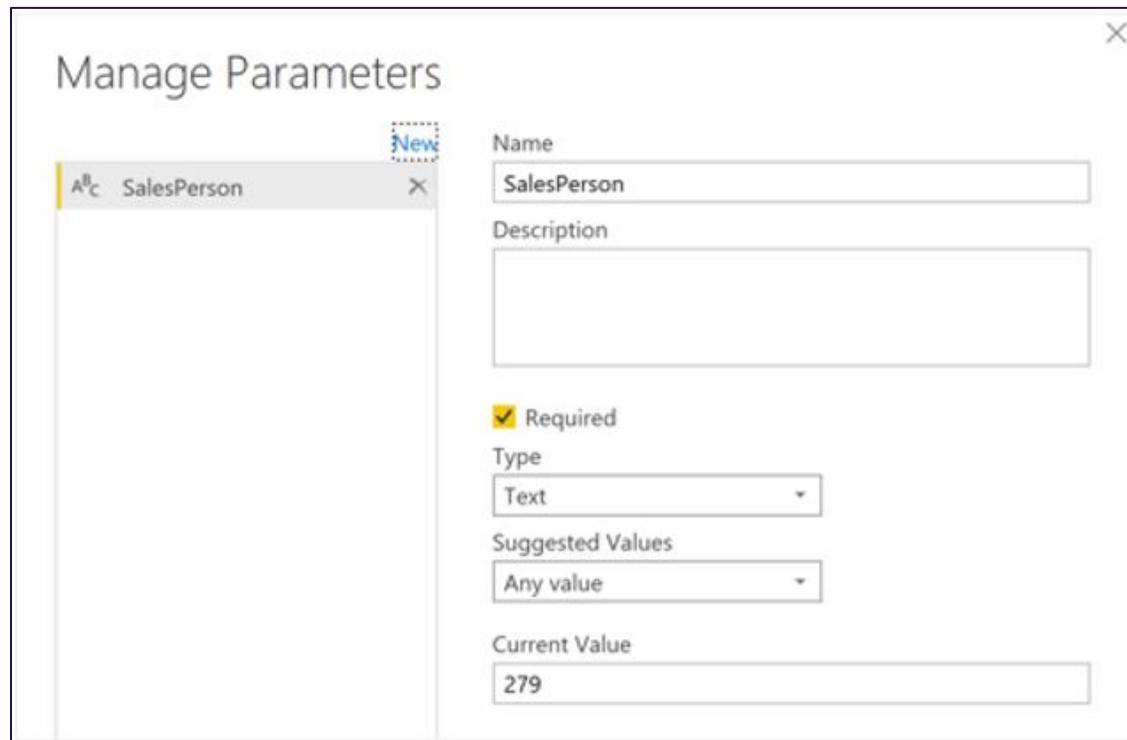
To split columns in Power Query, first, select the column you want to divide. Then, use the "Split Column" function and specify whether you want to split by a delimiter (like a comma or space) or by a fixed number of characters.

Unpivot columns or pivot

	Category Name	Subcategory Name
1	Bikes	Mountain Bikes
2	Bikes	Road Bikes
3	Bikes	Touring Bikes
4	Clothing	Bib-Shorts
5	Clothing	Caps
6	Clothing	Gloves
7	Clothing	Jerseys
8	Clothing	Shorts
9	Clothing	Socks
10	Clothing	Tights
11	Clothing	Vests
12	Accessories	Bike Racks
13	Accessories	Bike Stands
14	Accessories	Bottles and Cages

	1.2 Bikes	1.2 Components	1.2 Clothing	1.2 Accessories
1	3	14	8	12

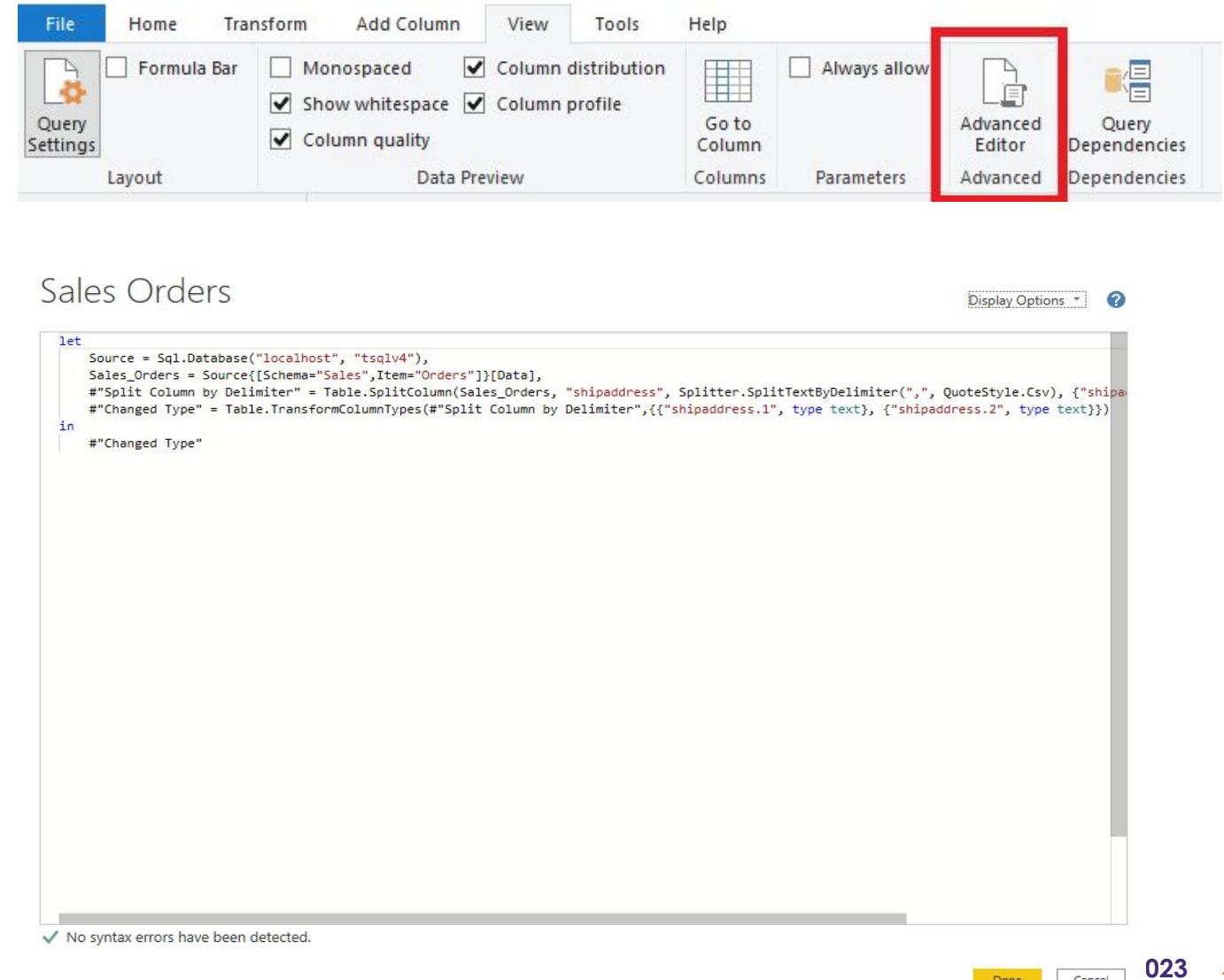
Dynamic reports with parameters



	A ^B C SalesOrderNumber	1 ² 3 SalesOrderID	1 ² 3 SalesPersonID	
1	SO43659		43659	279
2	SO43660		43660	279
3	SO43681		43681	279
4	SO43684		43684	279
5	SO43685		43685	279
6	SO43694		43694	279
7	SO43695		43695	279
8	SO43696		43696	279

Use Advanced Editor to modify M code

- When shaping data in Power Query, each action creates a step in the process that can be reordered, deleted, or modified.
- While the graphical interface is commonly used, Power Query operates with the underlying M language.

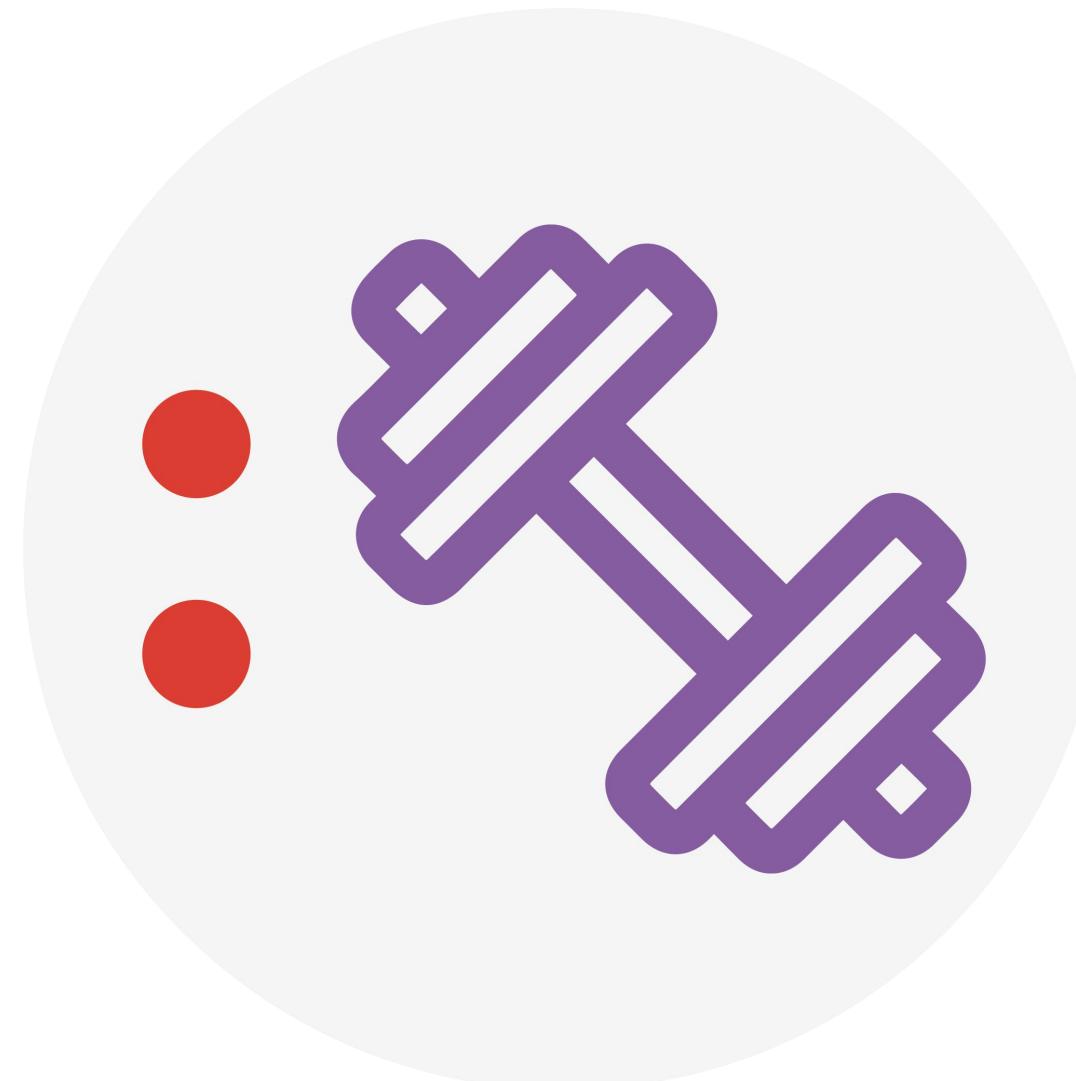


Data reduction best practices

- ✓ Remove unnecessary columns
- ✓ Remove unnecessary rows
- ✓ Group by and summarize
- ✓ Optimize column data types
- ✓ Preference for custom columns
- ✓ Disable Power Query load
- ✓ Disable auto date/time
- ✓ Switch to Mixed mode (Composite model)



Live Demo



Power BI Options & Settings

Options

GLOBAL

- Data Load
- Power Query Editor
- DirectQuery
- R scripting
- Python scripting
- Security**
- Privacy
- Regional Settings
- Updates
- Usage Data
- Diagnostics
- Preview features**
- Auto recovery
- Report settings

CURRENT FILE

- Data Load**
- Regional Settings**
- Privacy
- Auto recovery
- Published dataset settings
- Query reduction
- Report settings

Preview Features

- Shape map visual [Learn more](#)
- Spanish language support for Q&A [Learn more](#)
- Q&A for live connected Analysis Services databases [Learn more](#)
- Connect to external datasets shared with me [Learn more](#) | [Share feedback](#)
- Modern visual tooltips [Learn more](#) | [Share feedback](#)
- Sparklines [Learn more](#)
- Metrics visual [Learn more](#)
- Quick measure suggestions [Learn more](#) | [Share feedback](#)
- Field parameters [Learn more](#)
- Enhanced row-level security editor [Learn more](#)
- On-object interaction [Learn more](#) | [Share feedback](#)
- Enable setting sensitivity label on exported PDF [Learn more](#)
- Dynamic format string for measures [Learn more](#)
- Save to OneDrive and SharePoint [Learn more](#)
 - Share to OneDrive and SharePoint [Learn more](#)
- Power BI Project (.pbip) save option [Learn more](#)
- New card visual [Learn more](#)

Data Load

Type Detection

- Detect column types and headers for unstructured sources

Relationships

- Import relationships from data sources on first load (i)
 - Update or delete relationships when refreshing data (i)
 - Autodetect new relationships after data is loaded (i)
- [Learn more](#)

Time intelligence

- Auto date/time (i) [Learn more](#)

Background Data

- Allow data previews to download in the background

Parallel loading of tables (i)

Maximum number of concurrent jobs [Learn more](#)

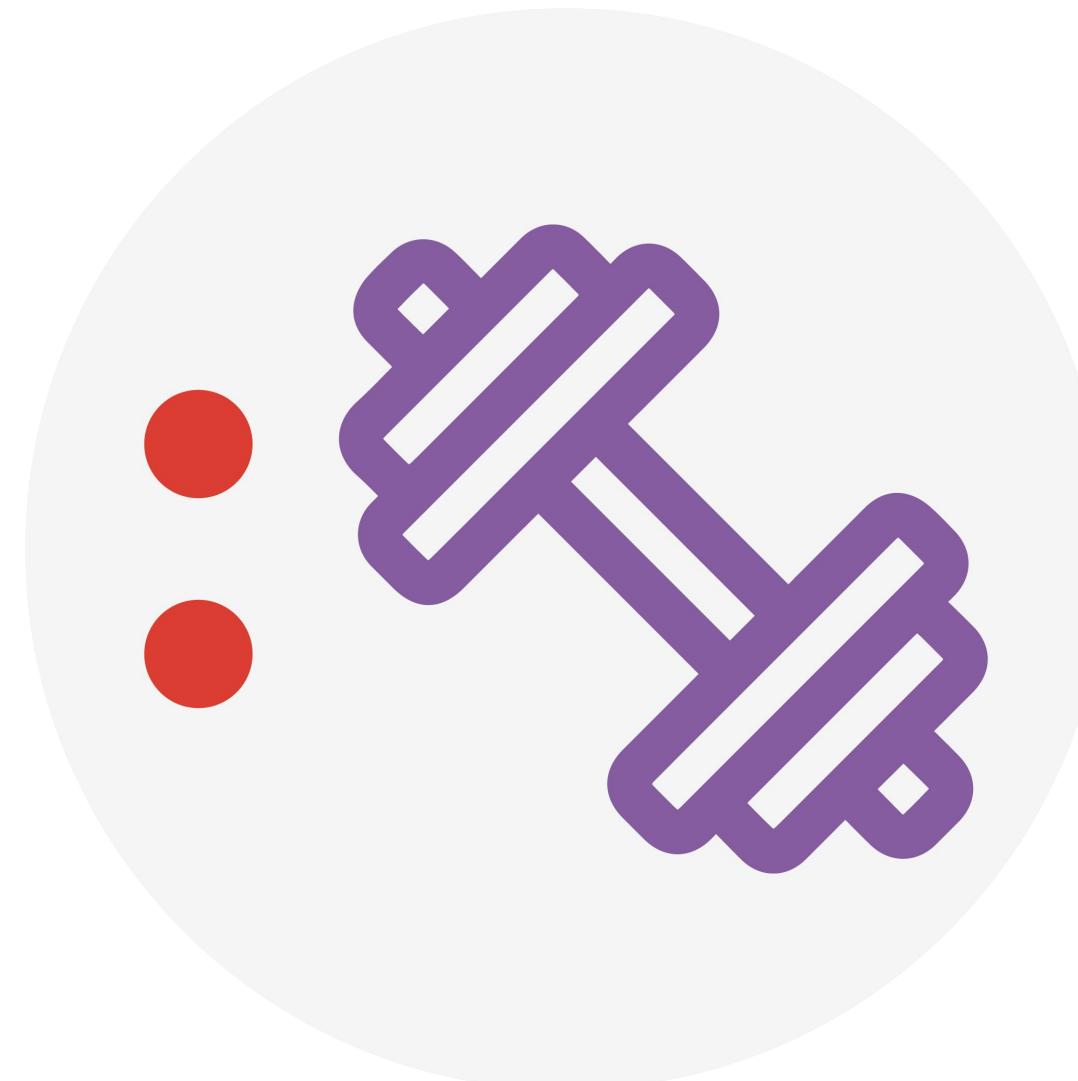
- Default
- One (disable parallel loading)
- Custom

Regional Settings

Locale for import

Locale determines the regional settings used to interpret numbers, dates, and time in imported text for this file.

Power Cycle Lab Exercise 1



Knowledge Check 2



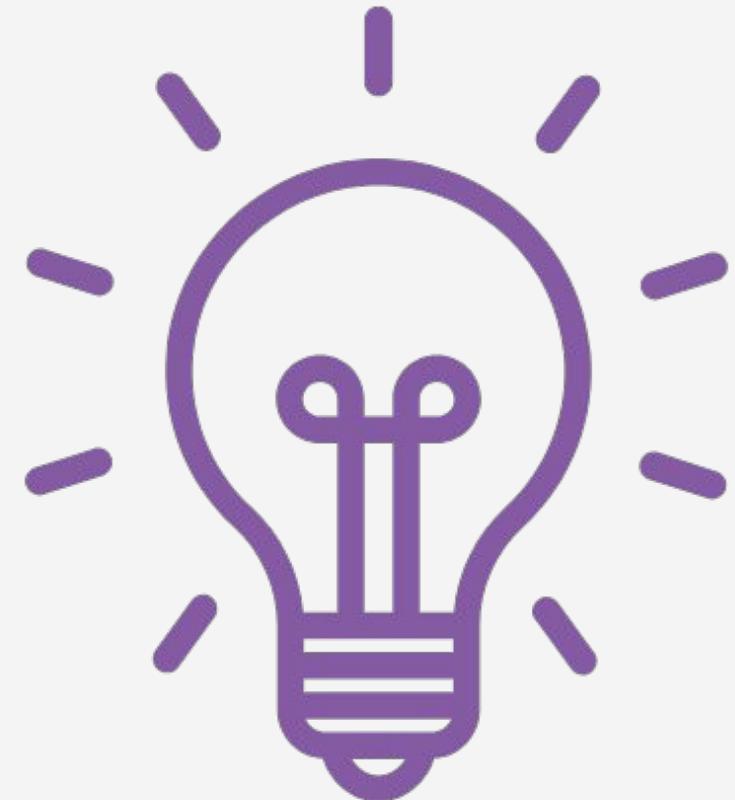
Summary and Q&A

Today, we learned how to :

- Get data from different sources
- Clean Transform & Load Data

In the next session, we will learn how to:

- Design a Data Model
- Create Model Calculations using DAX



DATA^SOCIETY:

Power BI Intermediate Day 2



Warm up

- Take a few minutes and draft a "tweet" of less than 280 characters that summarizes what you learned in the previous sessions.
- Share it in the chat box (and on Twitter, too, if you like!)



Agenda

- Design a Data Model
- Create Model Calculations using DAX

Lesson 3

Design a Data Model

After completing this module, students will be able to:

- Define relationships and their cardinality
- Implement Dimensions and Hierarchies
- Understanding filter propagation

STAR SCHEMA

- **Fact tables** contain observational or event data values: sales orders, product counts, prices, transactional dates and times, and quantities.
- **Dimension tables** contain the details about the data in fact tables: products, locations, employees, and order types. These tables are connected to the fact table through key columns. Dimension tables are used to filter and group the data in fact tables



Snowflake Schema

Snowflake schema allows relationships between dimensions and differs from the star schema in that it has more hierarchies between tables.

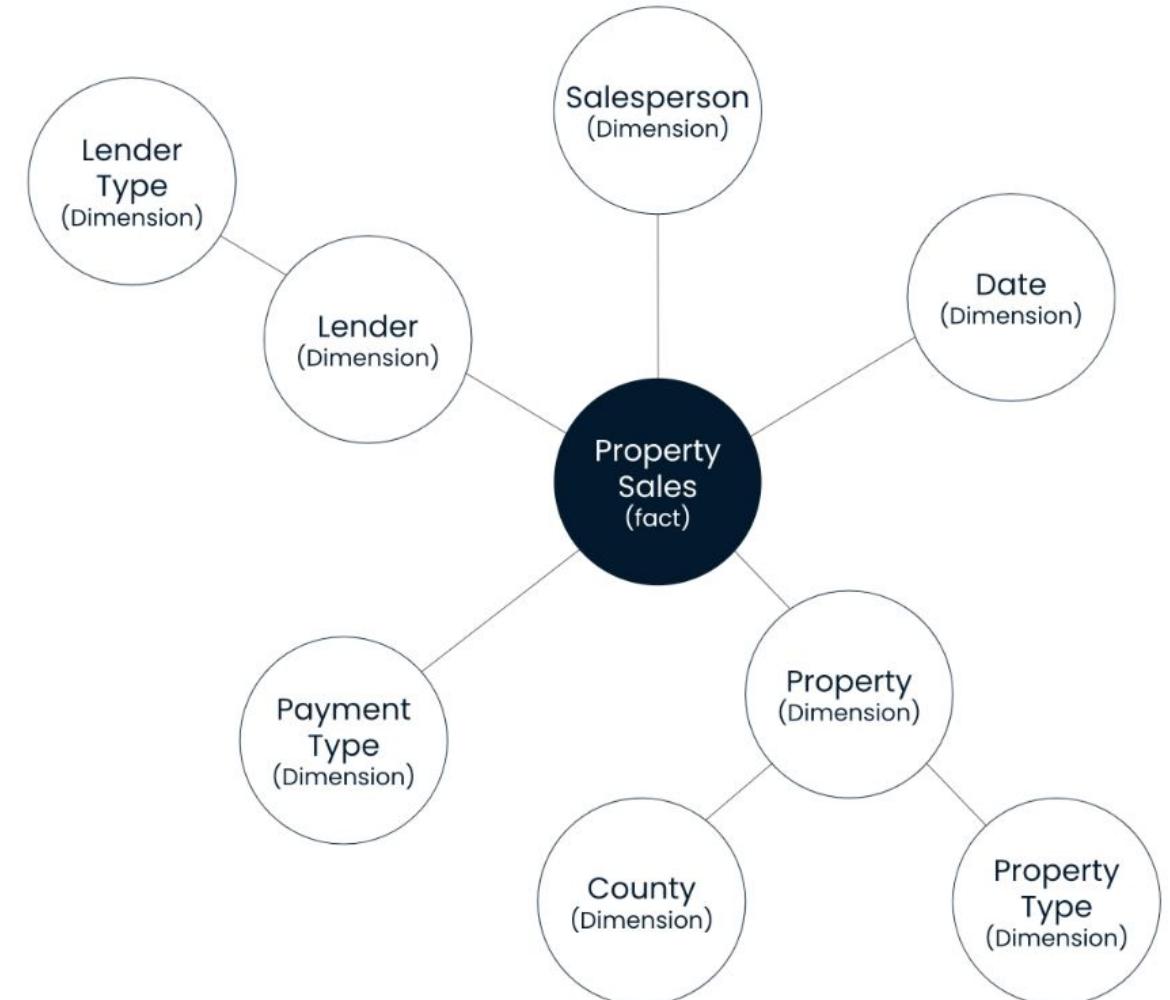


Table Normalization

	SalesOrderNumber	OrderDate	ProductKey	ResellerKey	SalesAmount
1	SO69561	2020-05-31	594	546	226.00
2	SO69560	2020-05-30	513	100	218.45
3	SO69560	2020-05-30	594	100	113.00
4	SO69539	2020-05-28	243	529	858.90
5	SO69539	2020-05-28	378	529	1466.01
6	SO69541	2020-05-28	594	661	113.00
7	SO69542	2020-05-28	243	317	1717.80
8	SO69544	2020-05-28	243	666	3435.60
9	SO69545	2020-05-28	378	436	5864.04
10	SO69532	2020-05-27	594	312	113.00
11	SO69532	2020-05-27	513	312	436.90
12	SO69533	2020-05-27	594	476	226.00



	SalesOrderNumber	OrderDate	ProductKey	Product	Category	Color	Size	ResellerKey	SalesAmount
1	SO69561	2020-05-31	594	Mountain-500 Silver, 48	Bikes	Silver	48	546	226.00
2	SO69560	2020-05-30	513	ML Mountain Frame-W - Silver, 46	Components	Silver	46	100	218.45
3	SO69560	2020-05-30	594	Mountain-500 Silver, 48	Bikes	Silver	48	100	113.00
4	SO69539	2020-05-28	243	HL Road Frame - Red, 44	Components	Red	44	529	858.90
5	SO69539	2020-05-28	378	Road-250 Black, 52	Bikes	Black	52	529	1466.01
6	SO69541	2020-05-28	594	Mountain-500 Silver, 48	Bikes	Silver	48	661	113.00
7	SO69542	2020-05-28	243	HL Road Frame - Red, 44	Components	Red	44	317	1717.80
8	SO69544	2020-05-28	243	HL Road Frame - Red, 44	Components	Red	44	666	3435.60
9	SO69545	2020-05-28	378	Road-250 Black, 52	Bikes	Black	52	436	5864.04
10	SO69532	2020-05-27	594	Mountain-500 Silver, 48	Bikes	Silver	48	312	113.00
11	SO69532	2020-05-27	513	ML Mountain Frame-W - Silver, 46	Components	Silver	46	312	436.90
12	SO69533	2020-05-27	594	Mountain-500 Silver, 48	Bikes	Silver	48	476	226.00

COMPARE FACT AND DIMENSION TABLES

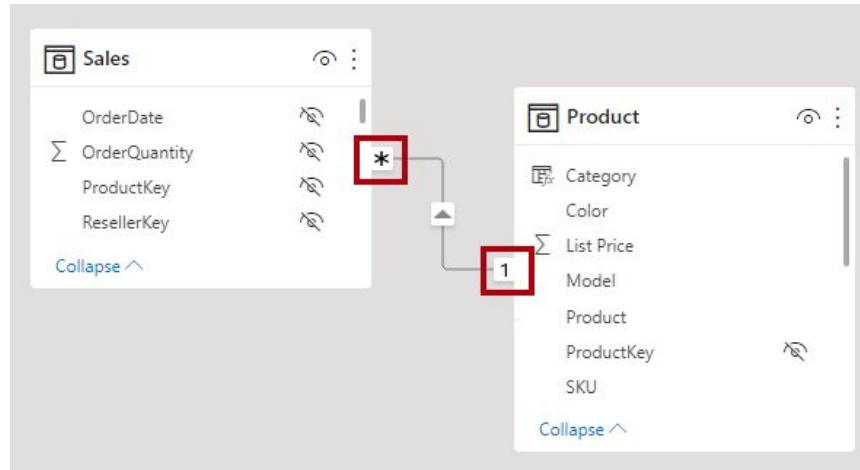
Characteristic	Dimension table	Fact table
Model purpose	Stores business entities	Stores events or observations
Table structure	Includes a key column and descriptive columns for filtering and grouping	Includes dimension key columns and numeric measure columns that can be summarized
Data volume	Typically, contains fewer rows (relative to fact tables)	Can contain numerous rows
Query purpose	To filter and group	To summarize

Relationship & Cardinality

Each model relationship is defined by a cardinality type. There are **four cardinality-type** options, representing the data characteristics of the "from" and "to" related columns. The "**one**" side means the column contains unique values; the "**many**" side means the column can contain duplicate values.

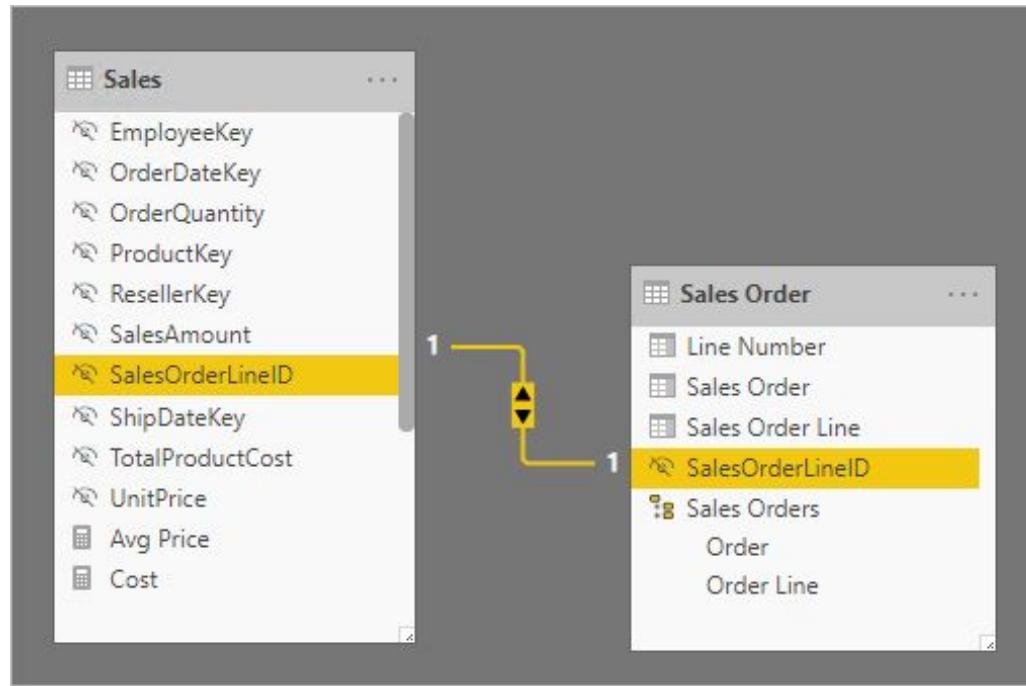
1. One-to-one (1:1)
2. One-to-many (1:*) Many-to-one (*:1)
3. Many-to-many (*:*)

One-to-many (1:*) Many-to-one (*:1)



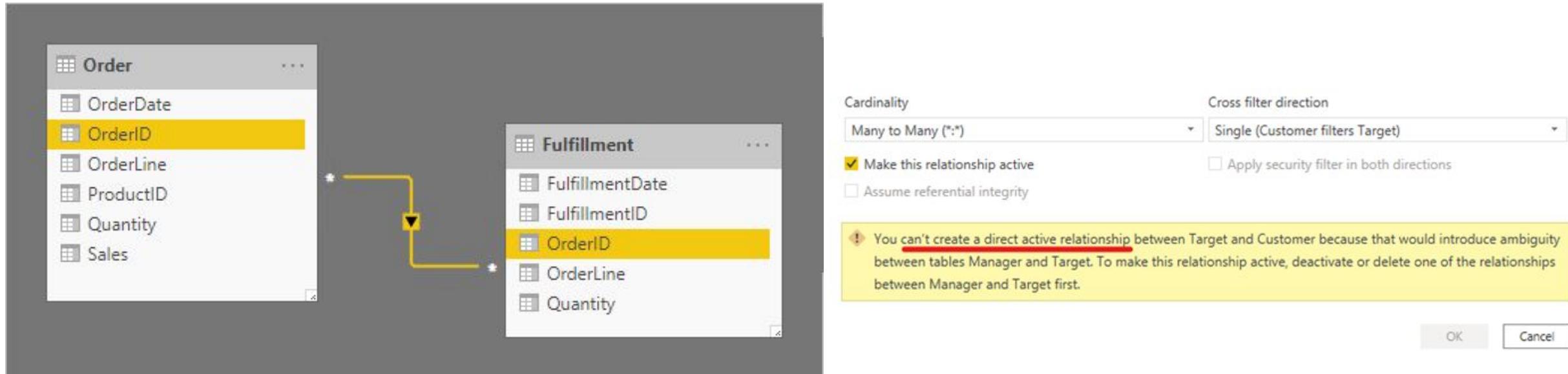
The one-to-many and many-to-one cardinality options are essentially the same, and they're also the most common cardinality types meaning there should be a single instance of each primary key while potentially accommodating multiple instances of each foreign key.

One-to-one (1:1)



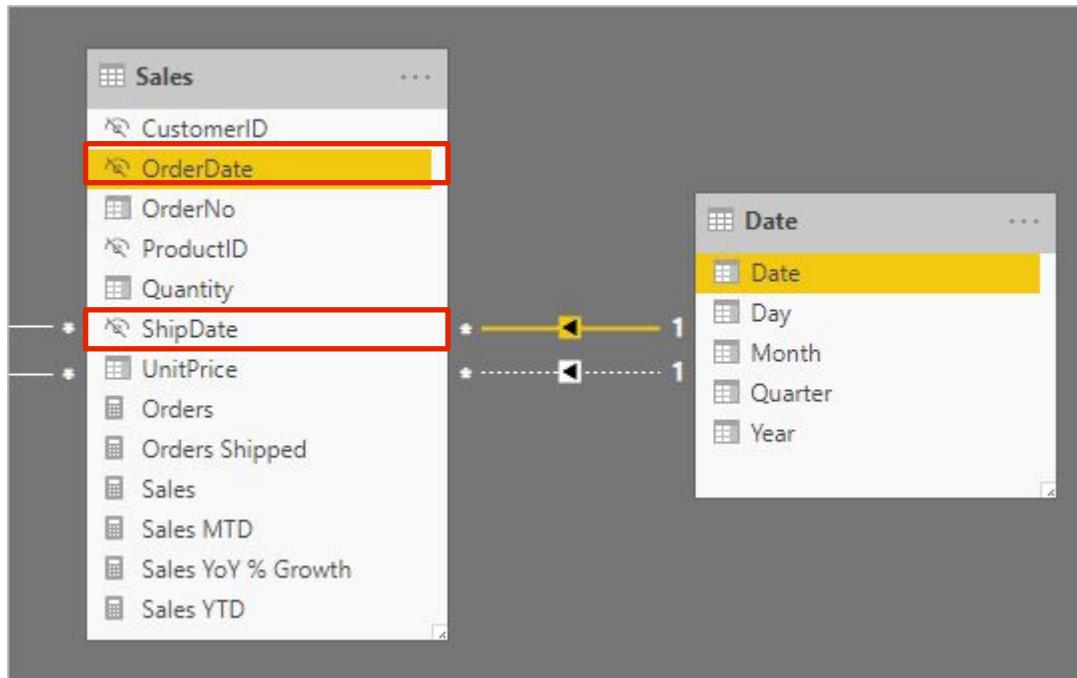
Maintaining one-to-one relationships may lead to redundant data storage since the tables involved often contain overlapping information. This redundancy can increase memory usage and make the data model more challenging to manage.

Many-to-many (*:*)



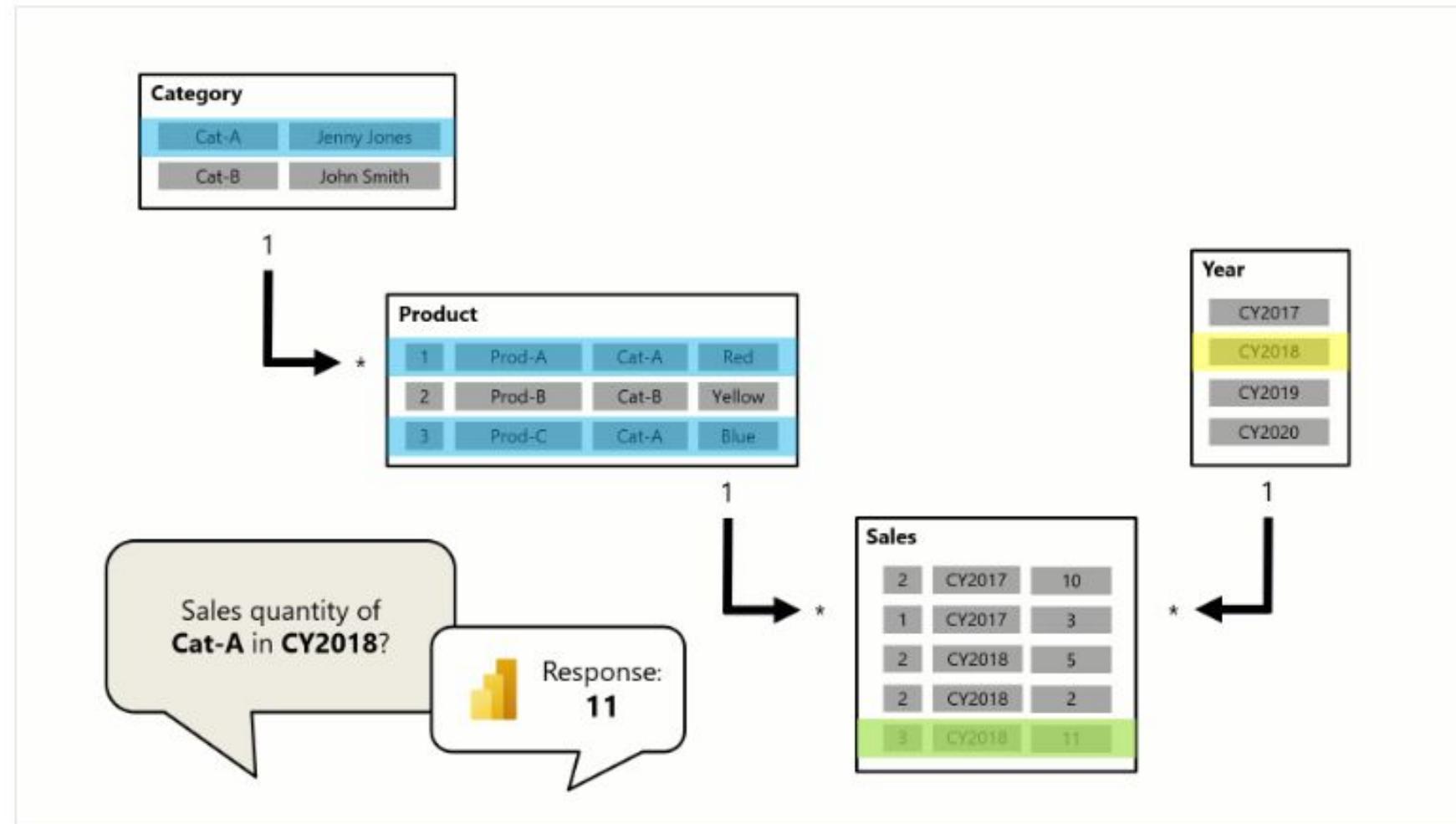
A many-to-many relationship means both columns can contain duplicate values. This cardinality type is infrequently used. It's typically useful when designing complex model requirements.

Inactive Relationships



- Inactive relationships in Power BI are relationships between tables that are temporarily turned off for filtering and calculations to avoid conflicts in data analysis.
- They can be activated as needed for specific tasks.

Filter Propagation



CROSS FILTER DIRECTION

Create relationship

Select tables and columns that are related.

Sales								
SalesOrderID	OrderDate	Sort_of_Sales	Freight	Freight_1	ProductID	OrderQty	SpecialOfferID	UnitPrice
52242	Sunday, July 7, 2013	Internet	0.1248	0.1248	870	1		10.99
52592	Sunday, July 14, 2013	Internet	0.1248	0.1248	870	1		10.99
52694	Tuesday, July 16, 2013	Internet	0.1248	0.1248	870	1		10.99

DatesTable				
DateCol	Year	Month	Week of Year	Day Name
Tuesday, May 31, 2011	2011	5	23	Tuesday
Tuesday, June 7, 2011	2011	6	24	Tuesday
Tuesday, June 14, 2011	2011	6	25	Tuesday

Cardinality Cross filter direction

Many to one (*:1)	Both
-------------------	------

Make this relationship active Apply security filter in both directions

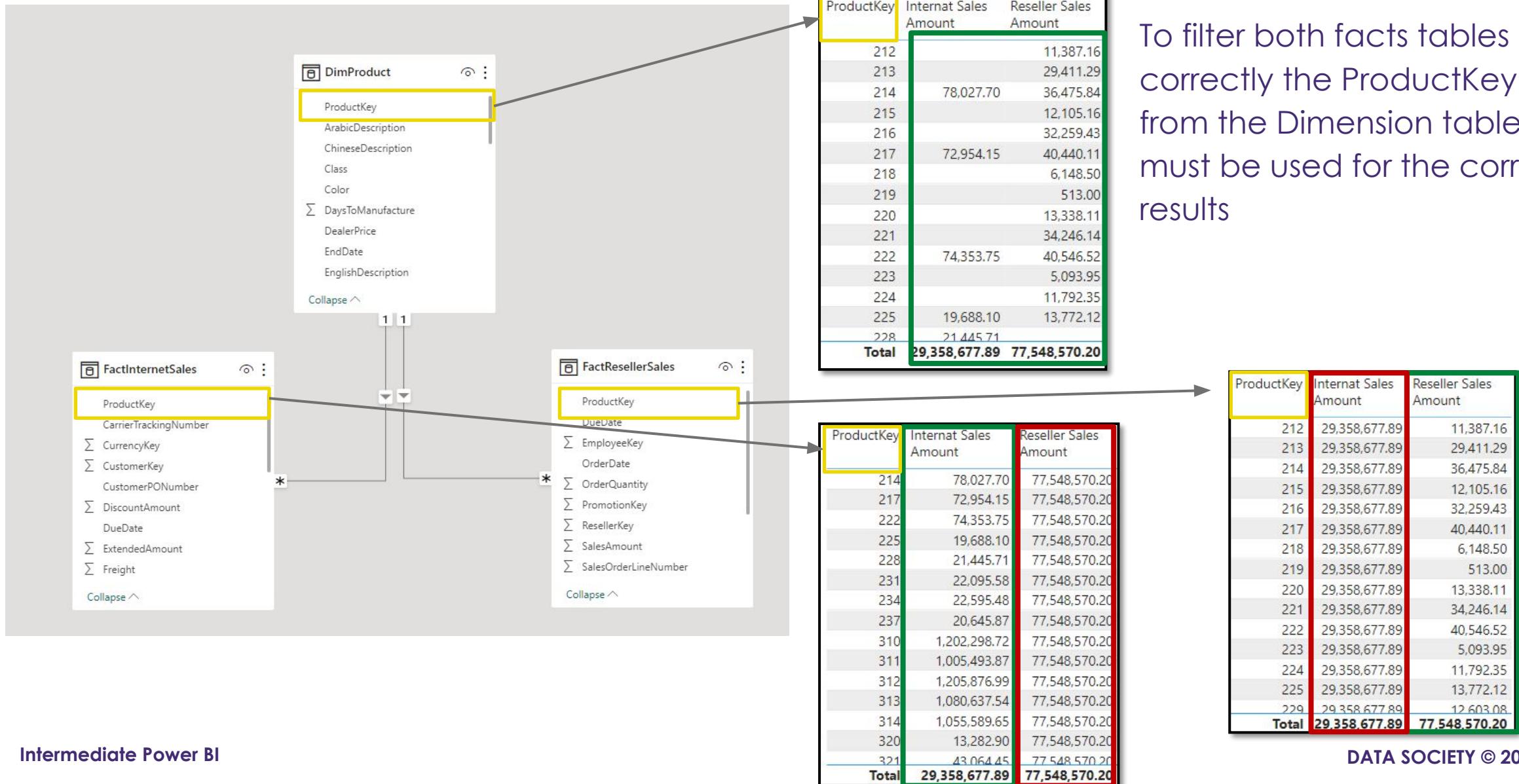
Assume referential integrity

OK Cancel

Cardinality type	Cross filter options
One-to-many (or Many-to-one)	Single Both
One-to-one	Both
Many-to-many	Single (Table1 to Table2) Single (Table2 to Table1) Both

Each model relationship is defined with a cross-filter direction. Your setting determines the direction(s) that filters will propagate. The possible cross-filter options are dependent on the cardinality type.

Filter Flow Example

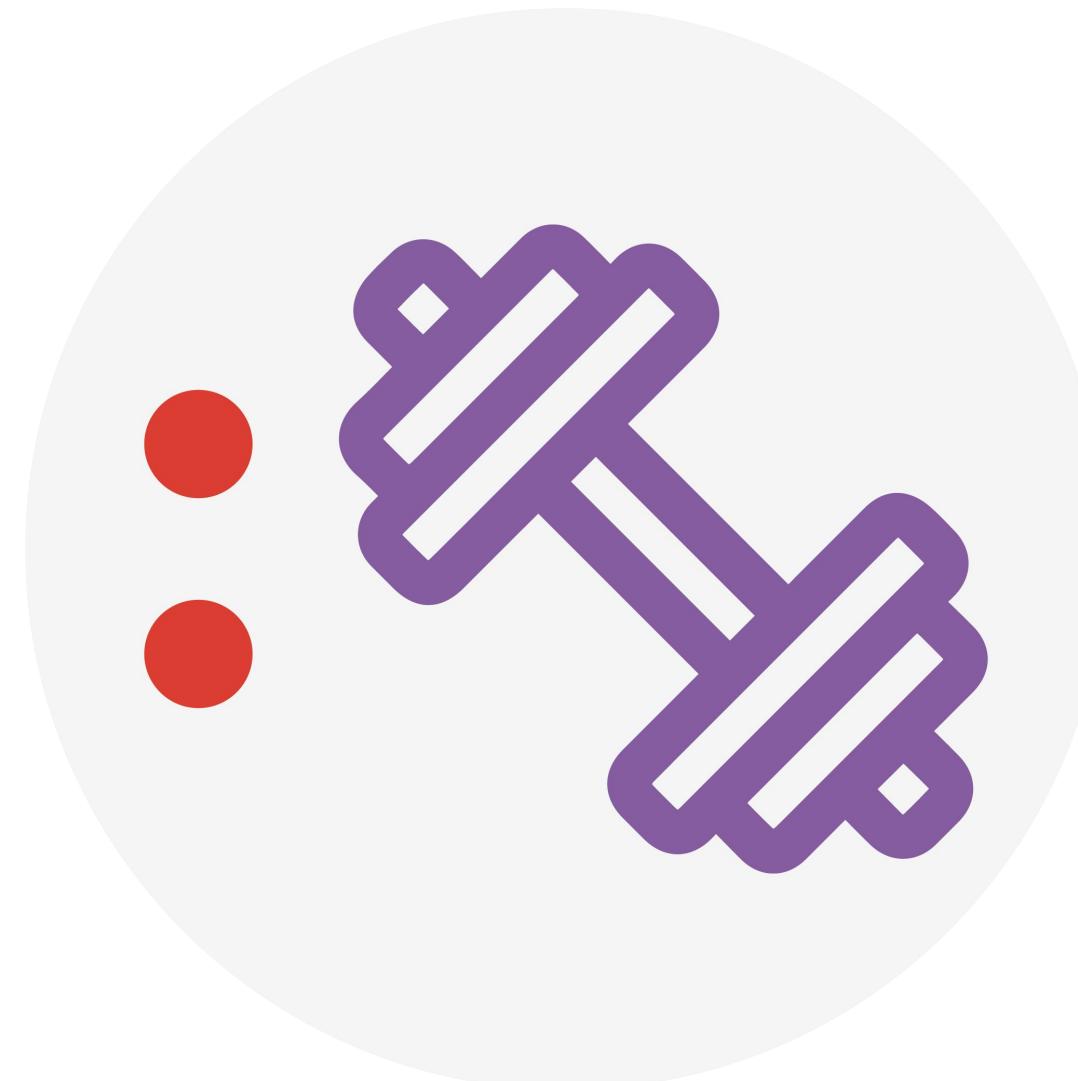


DATA MODEL BEST PRACTICES

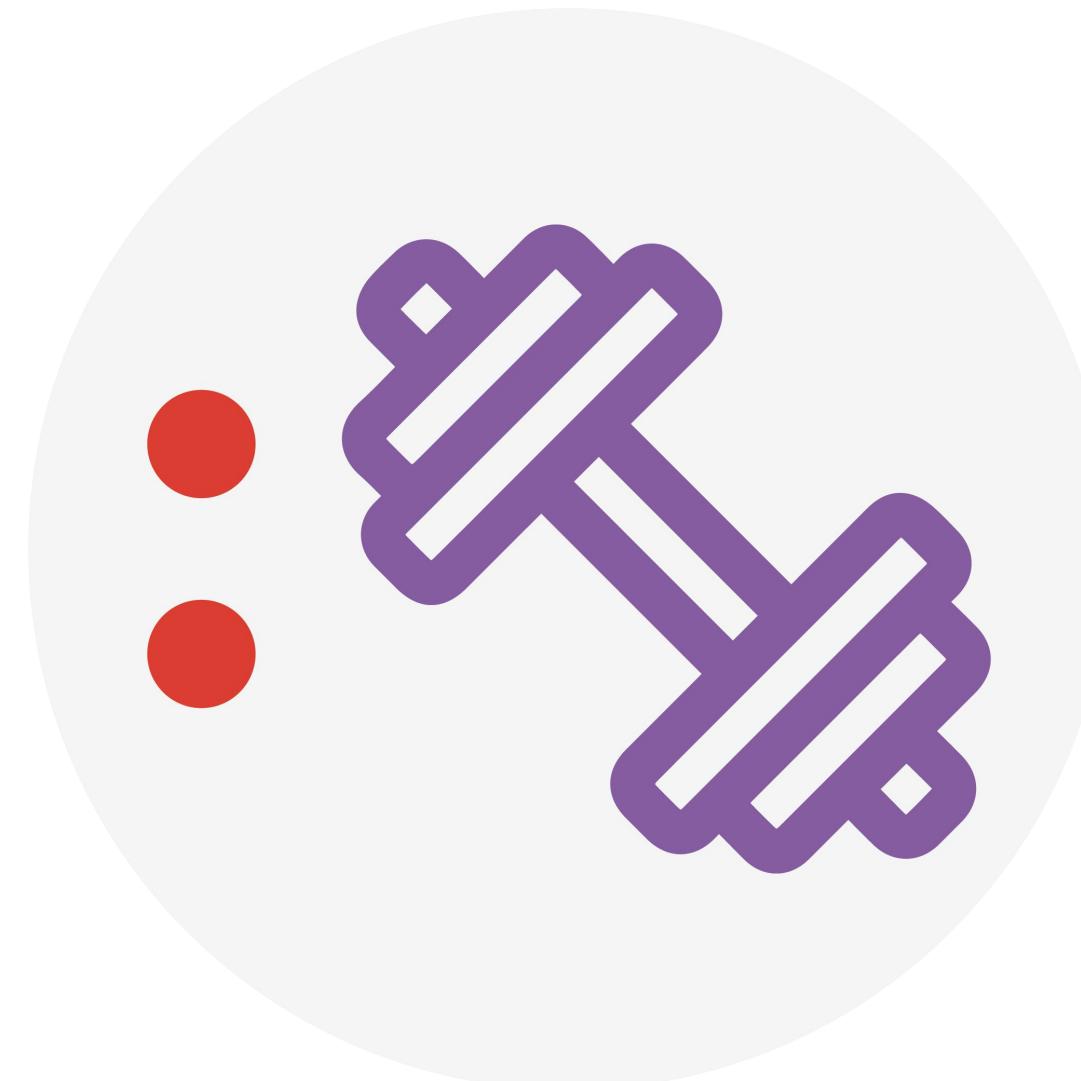
- ✓ Use a star schema with
- ✓ Contain relationships with one way filters (vs. bidirectional)
- ✓ Contain tables that each serve a specific purpose, including data (fact) tables and lookup (dim) tables
- ✓ Only include the data you need for analysis (no redundant or unnecessary records or fields)
- ✓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



Live Demo



Power Cycle Lab Exercise 2



Knowledge Check 3



Lesson 4

Create Model Calculations using DAX

After completing this module, students will be able to:

- Understand DAX functions
- Creating a calculated Data table using DAX
- Use DAX for formulas and expressions
- Create calculated tables and measures
- Build dynamic measures

DAX (Data Analysis Expressions)

DAX can be used to create three types of calculations:

- Calculated table
- Calculated column
- Measure
- DAX can also be used to define row-level security (RLS) rules



Common DAX Functions

Math & Stats

Common Examples:

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

- SUMX
- AVERAGEX
- MAXX/MINX
- COUNTX

Logic

Common Examples:

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

Text

Common Examples:

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

Filter

Common Examples:

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

Time Intelligence

Common Examples:

- TOTALMTD
- TOTALQTD
- TOTALYTD
- SAMEPERIODLASTYEAR
- PARALLELPERIOD
- DATESBETWEEN
- DATEADD
- DATESINPERIOD

Common DAX Operators

Arithmetic Operator	Meaning	Example
+	Addition	$2 + 7$
-	Subtraction	$5 - 3$
*	Multiplication	$2 * 6$
/	Division	$4 / 2$
\wedge	Exponent	$2 \wedge 5$

Comparison Operator	Meaning	Example
=	Equal to	[City] = "Boston"
>	Greater than	[Quantity] > 10
<	Less than	[Quantity] < 10
>=	Greater than or equal to	[Unit_Price] >= 2.5
<=	Less than or equal to	[Unit_Price] <= 2.5
<>	Not equal to	[Country] <> "Mexico"

Text/Logical Operator	Meaning	Example
&	Concatenates two values to produce one text string	[City] & " " & [State]
&&	Create an AND condition between two logical expressions	([State] = "MA") && ([Quantity] > 10)
(double pipe)	Create an OR condition between two logical expressions	([State] = "MA") ([State] = "CT")
IN	Creates a logical OR condition based on a given list (using curly brackets)	'Store Lookup'[State] IN { "MA", "CT", "NY" }

Standard Aggregation Measures

Typically, numeric columns are summarized using aggregation functions.

Common examples of summarizing:

- Sum
- Count
- Average
- Min and Max

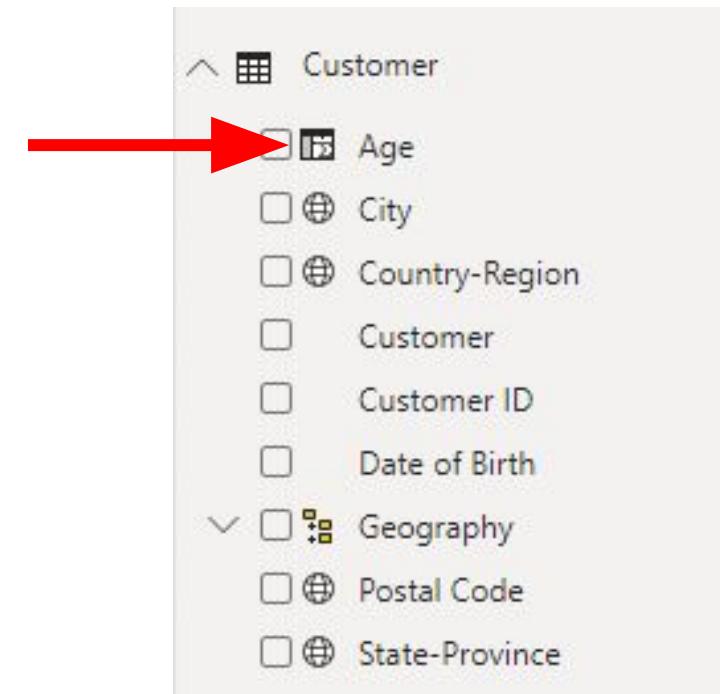
Column references always enclose the column name in square brackets

```
Total Sale = sum('Sales By Country Files'[Sales])
```

```
Sale Quantity = Average('Sales By Country Files'[Quantity])
```

DAX Calculated column

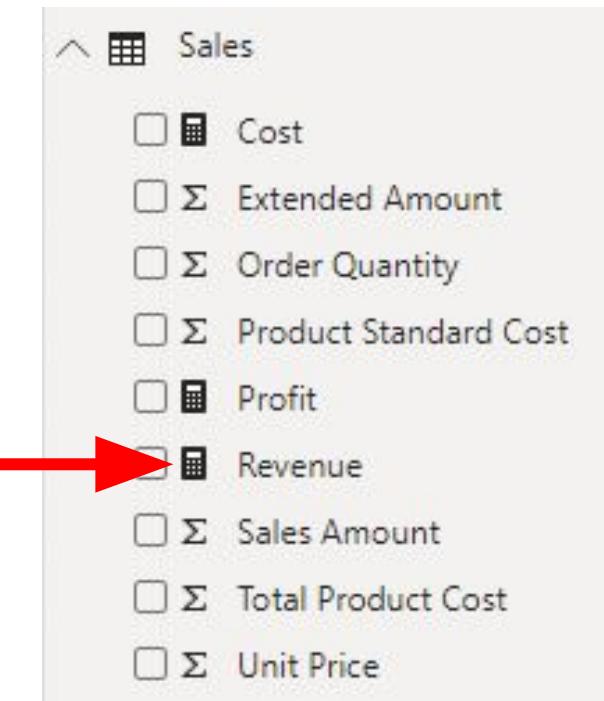
- A **Calculated column** adds a new column to a table
- A formula is evaluated for each row in the table
- The formula must return a single value
- It is only evaluated when the model is refreshed*
- It increases the storage size of the model*



*Except when added to a DirectQuery table

DAX Calculated measures

- A Measure summarizes model data
- The formula must return a single value
- It is evaluated at query-time
- Results are never stored in the model

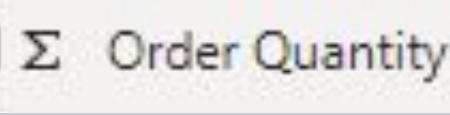
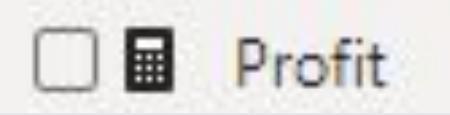


Calculated columns vs. measures

There are differences between calculated column and measure

	Calculated Columns	Measures
Purpose	Extend table	Summarize model data
Evaluation	Row context at data refresh-time	Filter context at query-time
Storage	Stores value in each table row	Never stores value
Visual use	Filter, group, or summarize	Designed to summarize

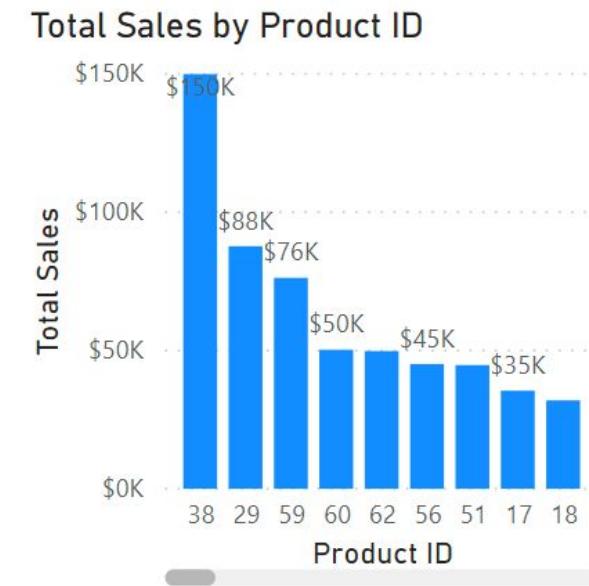
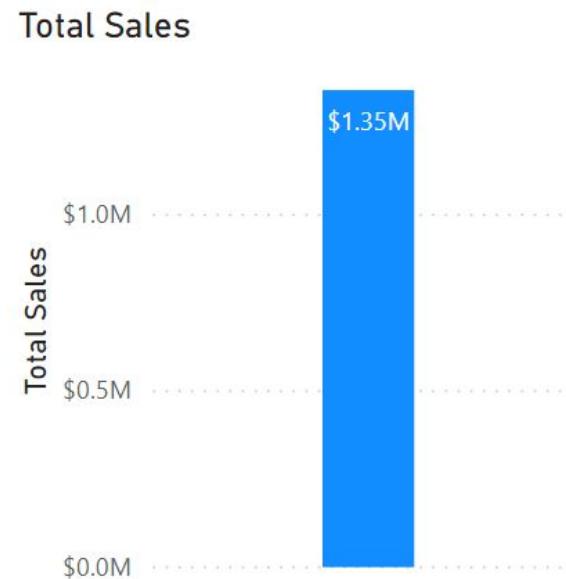
Implicit & Explicit Measures

	Benefits	Limitations
Implicit Measures 	<ul style="list-style-type: none">Accessible to non-technical users.Enables self-service analytics.Easy to modify and adjust.	<ul style="list-style-type: none">Limited flexibility for complex calculations.Limited control and customization.May lack granularity and detailed data analysis capabilities.
Explicit Measures 	<ul style="list-style-type: none">Greater flexibility and customization.Requires deeper understanding of DAX and the data model.Can be used in Analyze with Excel export option.	<ul style="list-style-type: none">Requires advanced technical knowledge of DAX.Time-consuming to create and manage complex calculations.Excessive use can impact report performance.

Understand context when using DAX

How context affects DAX measures is a difficult concept to comprehend. The ensuing visuals will demonstrate how context affects DAX measures so you can see how they interact together.

The following three visuals use the exact same DAX measure: Total Sales.



DAX Formatting

From

```
USSales = CALCULATE([Total Sales], FILTER(Reseller, Reseller[ResellerCountry] = "United States"))
```



To

```
USSales =  
CALCULATE (  
    [Total Sales],  
    FILTER ( Reseller, Reseller[ResellerCountry] = "United States" )  
)
```

Use **shift + enter** to split out and indent each component of your DAX formulas to make them more readable



<https://www.daxformatter.com/>

Formula Commenting

```
Sales Previous Year % Change =  
  
var CY = [Total Sales] // Current Year Sales ►  
  
var PY = [Sales Last Year] // Last Years Sales ►  
  
RETURN  
  
DIVIDE(CY-PY,PY)
```

Comment Type	Marker
Single Line Comment	-- or //
Multi Line Comment	/* ... */

Adding comments to your code can assist fellow users in understanding your script, and this practice proves especially beneficial for intricate queries featuring numerous lines and nested functions.

Using Variables

Variables can be defined in an expression to make writing DAX easier

Main benefits:

- Improves readability of a formula
- Improves performance when an expression is used multiple times
- Allows testing portions of a complex formula, by returning only a variable for review

```
VAR = <name> RETURN <result_expression>
```

Example:

```
VAR RevenuePriorYear = CALCULATE( [Revenue],  
SAMEPERIODLASTYEAR('Date'[Date]) )
```

RETURN

```
DIVIDE( [Revenue Measure] -RevenuePriorYear,  
RevenuePriorYear )
```

DAX Quick Measures

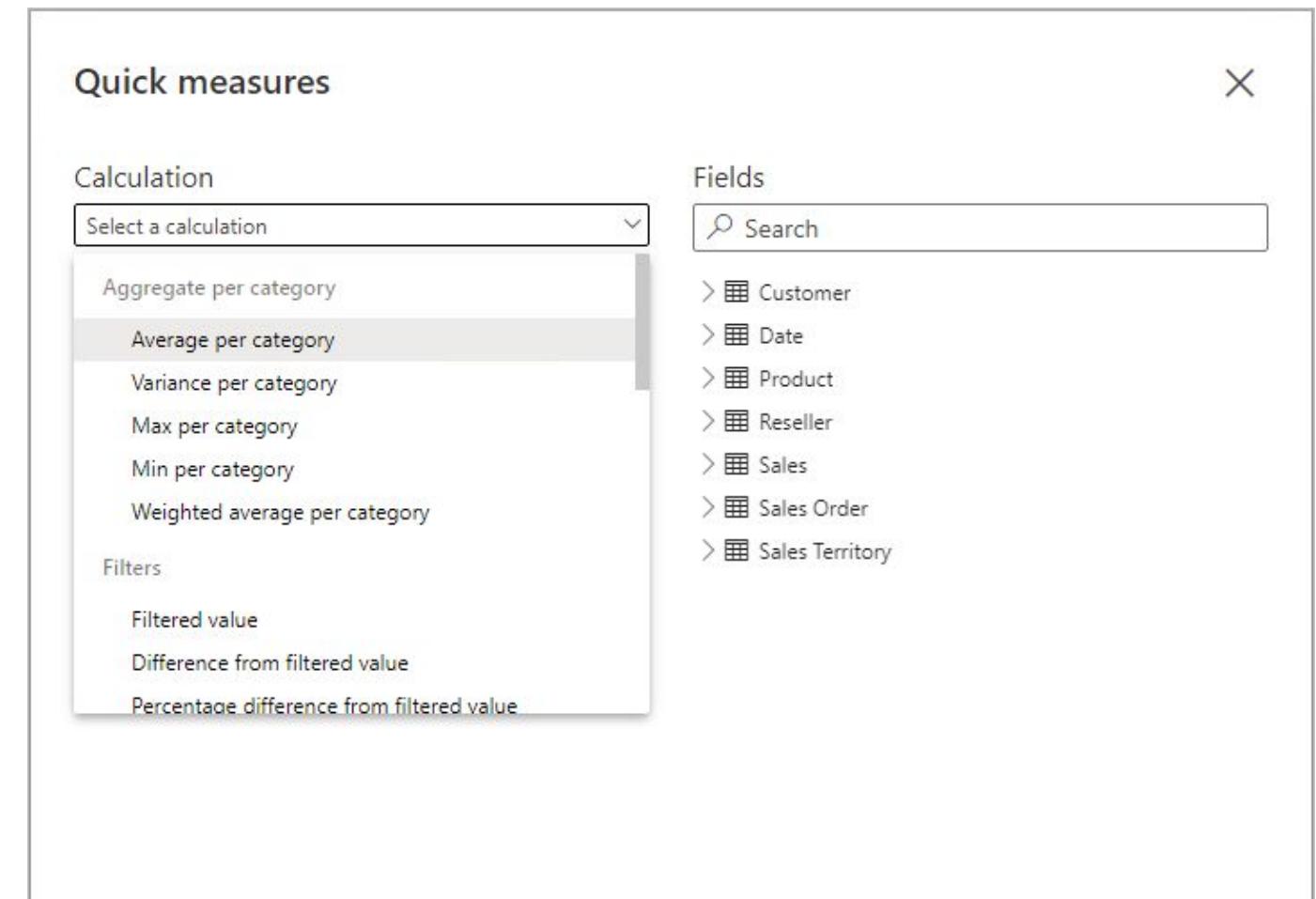
Power BI Desktop supports creating **Quick Measures**

Benefits:

- Quick and easy
- Common calculation templates are available
- Automatically creates the DAX expression
- No need to understand any DAX

Once created, they are like any other explicit measure

- To modify the measure, simply edit the formula



DAX Quick Suggestion AI generation (New)

Quick measure suggestions assist in creating DAX measures using natural language instead of templates or writing DAX from scratch.

Quick measures

Select a calculation to create a measure or describe the measure you need and we'll generate suggestions in DAX, which you can customize later.

Calculations Suggestions

Sales amount for California in 2020

Generate

Suggested measures

Total sales amount where state-province is California and year of date is 2020

Preview value

\$1,785,099.77

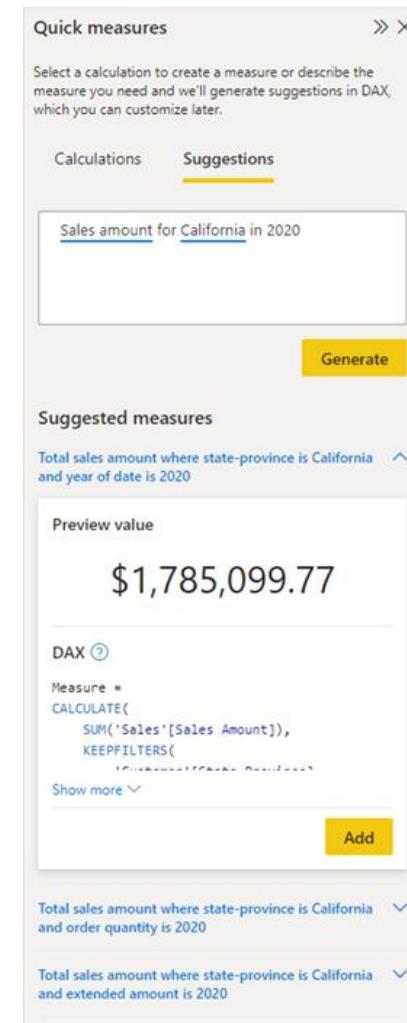
DAX ⓘ

Measure =
CALCULATE(
SUM('Sales'[Sales Amount]),
KEEPFILTERS()
Show more ▾

Add

Total sales amount where state-province is California and order quantity is 2020

Total sales amount where state-province is California and extended amount is 2020



Quick measures

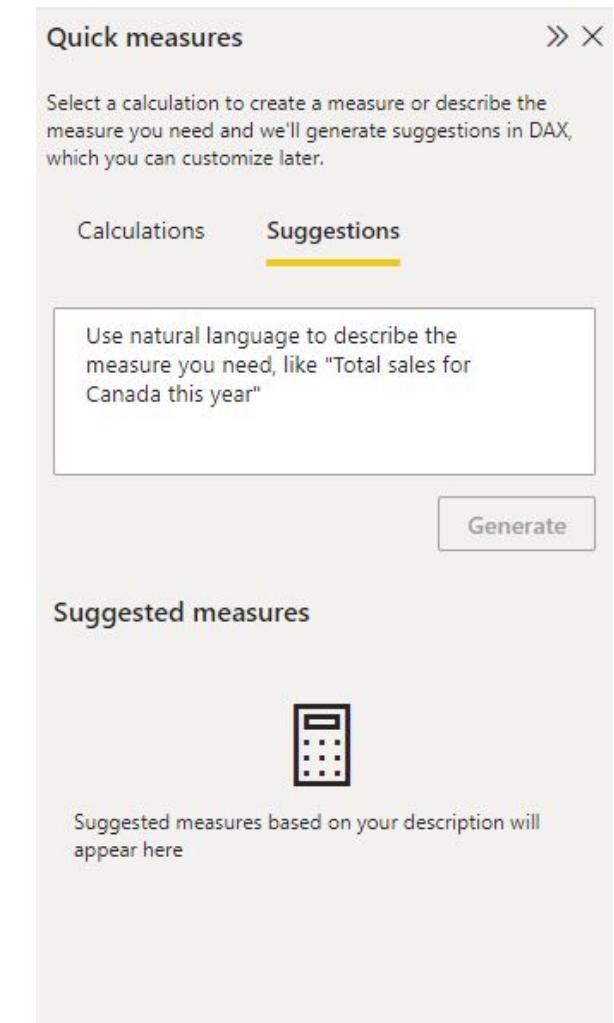
Select a calculation to create a measure or describe the measure you need and we'll generate suggestions in DAX, which you can customize later.

Calculations Suggestions

Use natural language to describe the measure you need, like "Total sales for Canada this year"

Generate

Suggested measures



Iterator functions

Iterator functions enumerate all rows of a given table and evaluate a given expression for each row

- They provide flexibility over how calculations summarize data
- All iterator functions require Table and Expression
- The expression must return scalar or single value
- Iterator functions are easily identified by the appended “X”

For example:

- Common aggregation: SUMX, AVERAGEX, COUNTX, MINX, and MAXX
- Special: CONCATENATEX, and RANKX

Iterator functions

Example

```
Measure =  
SUMX(  
    <Table>,  
    <Expression>  
)
```

```
Revenue =  
SUMX(  
    Sales,  
    Sales[Order Quantity] * Sales[Unit Price]  
)
```

Iterator functions example

Order Key	Order Quantity	Unit Price	
5	2	\$5.19	10.38
5	4	\$20.19	80.76
5	1	\$419.46	419.46
5	1	\$874.79	874.79
6	1	\$809.76	809.76
6	1	\$714.70	714.70
6	2	\$714.70	1,429.40
6	4	\$5.19	...

$\sum \quad 208,202.17$

Automatic Date Table

By default, Power BI automatically generates a concealed date table for any table that includes a Date or **DateTime** column on one side of a relationship. These automatically generated calendars encompass all dates up to the end of the year, irrespective of the actual date range within the table.

The screenshot shows the Power BI Data View. On the left, the Sales table is listed with various columns: CategoryID, Customer ID, Customer Name, Discount%, EmployeeID, LocationID, Order Date, Order ID, Product ID, Quantity, Row ID, Segment, and Ship Date. The 'Order Date' and 'Ship Date' columns are highlighted with red boxes. A red curly brace on the right side groups these two columns together, indicating they are related to the automatically generated date table on the right.

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

Automatically created a hidden date table in the background containing all these columns

PROS & CONS AUTOMATIC DATE TABLES

PROS



- Automatically generated
- No code or additional table required
- Built-in date Hierarchies aren't automatically generated

Cons



- Hidden background table that not be modified
- Inflates the Power BI file size due to every date field in your table having a auto-date table
- Not a centralized date dimension table

Date Dimension Table Requirements Checklist

- ✓ It should encompass all days corresponding to the years presented in your fact tables.
- ✓ It is necessary to designate at least one field as a Date or DateTime data type.
- ✓ Duplicate dates or datetime values are not permissible.
- ✓ If utilizing a time component within a date column, all times must be uniform (e.g., 12:00 PM).
- ✓ While not mandatory, it is recommended to label it as a date table for optimal use.

CALENDAR Function

- Returns a table with one column of all dates between start and end date
- Dates can also be referenced from date fields in other tables as seen in example 2

```
=CALENDAR(StartDate,EndDate)
```

Example 1: CALENDAR DATE(2020,01,01), DATE (2023,12,31)

Example 2:

```
CALENDAR  
DATE( YEAR ( MIN (Sales [Order Date] )), 1,1)  
DATE( YEAR ( MAX (Sales [Order Date] )), 12,31)
```



Date
1/1/2020 12:00:00 AM
1/2/2020 12:00:00 AM
1/3/2020 12:00:00 AM
1/4/2020 12:00:00 AM
1/5/2020 12:00:00 AM
1/6/2020 12:00:00 AM
1/7/2020 12:00:00 AM
1/8/2020 12:00:00 AM
1/9/2020 12:00:00 AM
1/10/2020 12:00:00 AM
1/11/2020 12:00:00 AM
1/12/2020 12:00:00 AM
1/13/2020 12:00:00 AM
1/14/2020 12:00:00 AM
1/15/2020 12:00:00 AM
1/16/2020 12:00:00 AM
1/17/2020 12:00:00 AM
1/18/2020 12:00:00 AM
1/19/2020 12:00:00 AM
1/20/2020 12:00:00 AM
1/21/2020 12:00:00 AM
1/22/2020 12:00:00 AM

CALENDARAUTO Function

- Returns a table with one column of dates based on a fiscal year end month
- Range of dates is calculated automatically based on data in the model

Example 1: CALENDARAUTO(4)

Example 2:

Calendar Table =

```
VAR MinYear = YEAR( MIN( 'Country'[Total Sales]))  
VAR MaxYear= YEAR( MAX('Country'[Total Sales]))  
  
RETURN  
  
FILTER(CALENDARAUTO(),  
YEAR( [Date] ) >= MinYear&&YEAR( [Date] ) <= MaxYear)
```



Date
5/1/2014 12:00:00 AM
5/2/2014 12:00:00 AM
5/3/2014 12:00:00 AM
5/4/2014 12:00:00 AM
5/5/2014 12:00:00 AM
5/6/2014 12:00:00 AM
5/7/2014 12:00:00 AM
5/8/2014 12:00:00 AM
5/9/2014 12:00:00 AM
5/10/2014 12:00:00 AM
5/11/2014 12:00:00 AM
5/12/2014 12:00:00 AM
5/13/2014 12:00:00 AM

The Date Dimension Table

The screenshot shows the Power BI Data View interface. On the left, the 'Sales' table is displayed with columns: OrderQty, ProductID, RevenueAvg, SalesOrderID, ShipDate, Sort_of_Sales, SpecialOfferID, StateProvinceID, and # TotalSales. The 'ShipDate' column is highlighted with a red border. On the right, the 'Order' table is shown with columns: CountryID, CustomerID, ListPrice, OrderDate, OrderQty, ProductID, SalesOrderID, Sort_of_Sales, and StateProvinceID. The 'OrderDate' column is also highlighted with a red border.

Date	Year	Month	Month Short Name	Month Long Name	Quater	Day of the week	WeekNum
7/1/2020	2020	7	Jul	July	Q3	Wed	27
7/2/2020	2020	7	Jul	July	Q3	Thu	27
7/3/2020	2020	7	Jul	July	Q3	Fri	27
7/4/2020	2020	7	Jul	July	Q3	Sat	27
7/5/2020	2020	7	Jul	July	Q3	Sun	27
7/6/2020	2020	7	Jul	July	Q3	Mon	28
7/7/2020	2020	7	Jul	July	Q3	Tue	28
7/8/2020	2020	7	Jul	July	Q3	Wed	28

Column Name	Fonction
DAX Defined Date	<code>Dates = CALENDAR(DATE(2011, 5, 31), DATE(2022, 12, 31))</code>
DAX Auto Date	<code>Dates = CALENDARAUTO()</code>
DAX Year	<code>Year = YEAR(Dates[Date])</code>
DAX Month	<code>MonthNum = MONTH(Dates[Date])</code>
DAX Week number	<code>WeekNum = WEEKNUM(Dates[Date])</code>
DAX Day of the week	<code>DayoftheWeek = FORMAT(Dates[Date], "DDDD")</code>

Formatting your date

By using the DAX **FORMAT** function to specify date/time formatting you can specify the format output you would like for your fields.

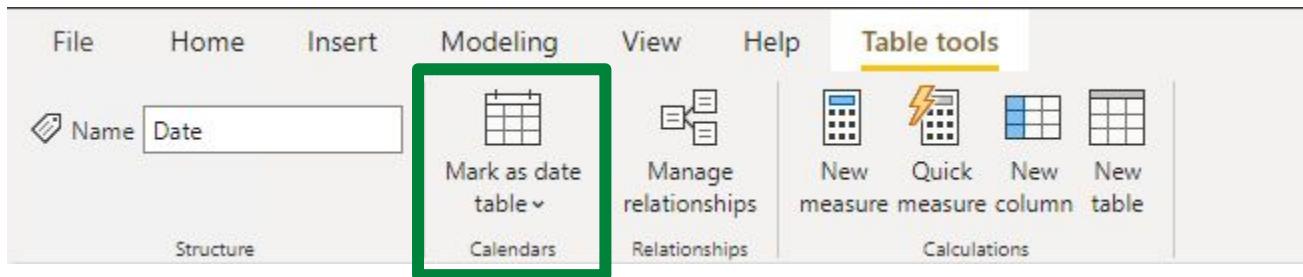
```
FORMAT(<value>, <format_string>[, <locale_name>])
```

```
Short Month Name = FORMAT('Table'[Date], "mmm")
```

Date	Short Month Name
5/1/2014 12:00:00 AM	May
5/2/2014 12:00:00 AM	May
5/3/2014 12:00:00 AM	May
5/4/2014 12:00:00 AM	May
5/5/2014 12:00:00 AM	May
5/6/2014 12:00:00 AM	May
5/7/2014 12:00:00 AM	May

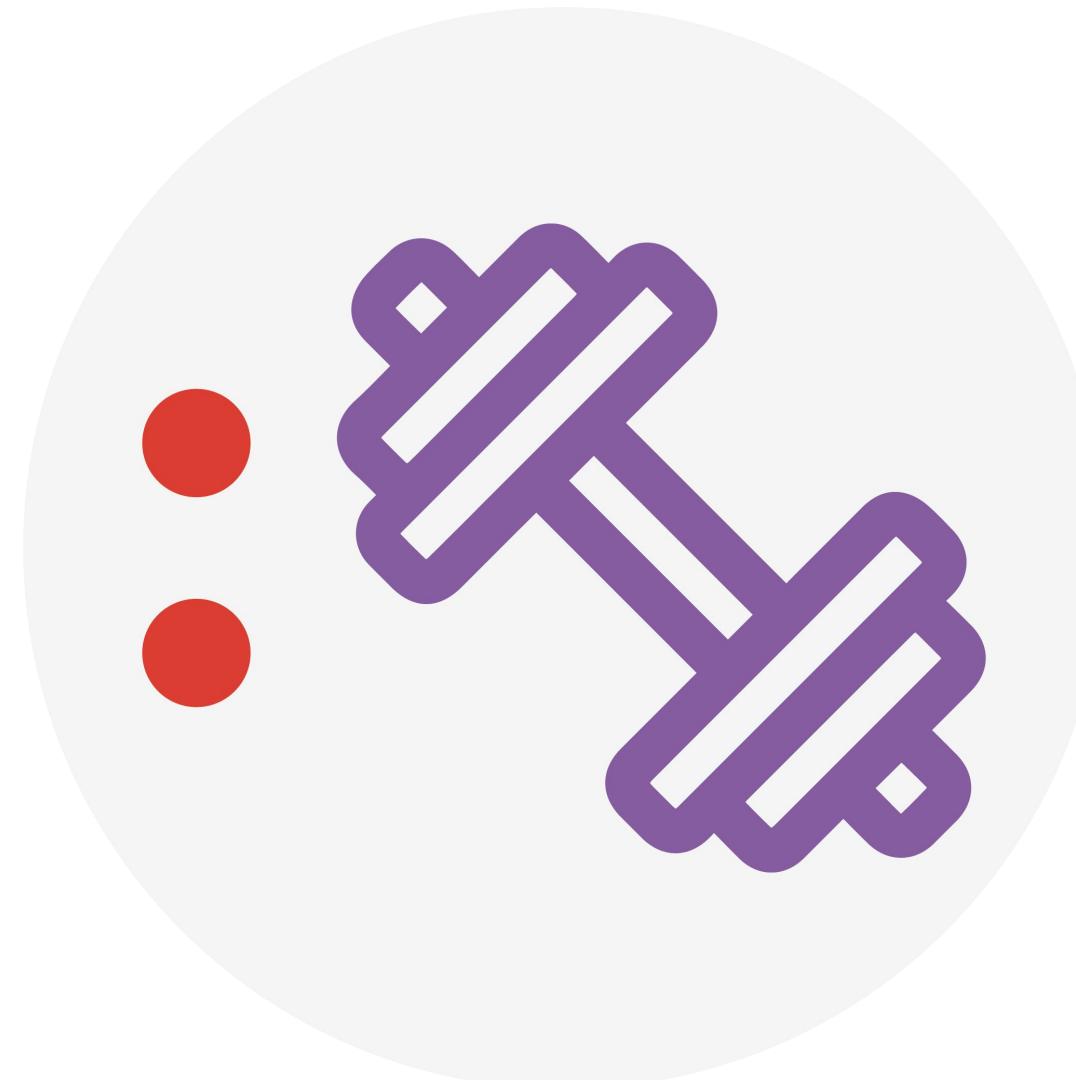
Code	Description
dddd	Display the full day name
ddd	Display the short day name
mmmm	Show full month name
mmm	Show abbreviated month name
mm	Display the month as a two digit number
q	Display the quarter of the year
yyyy	Show the year as a four digit number

Marking Date Dimension Table

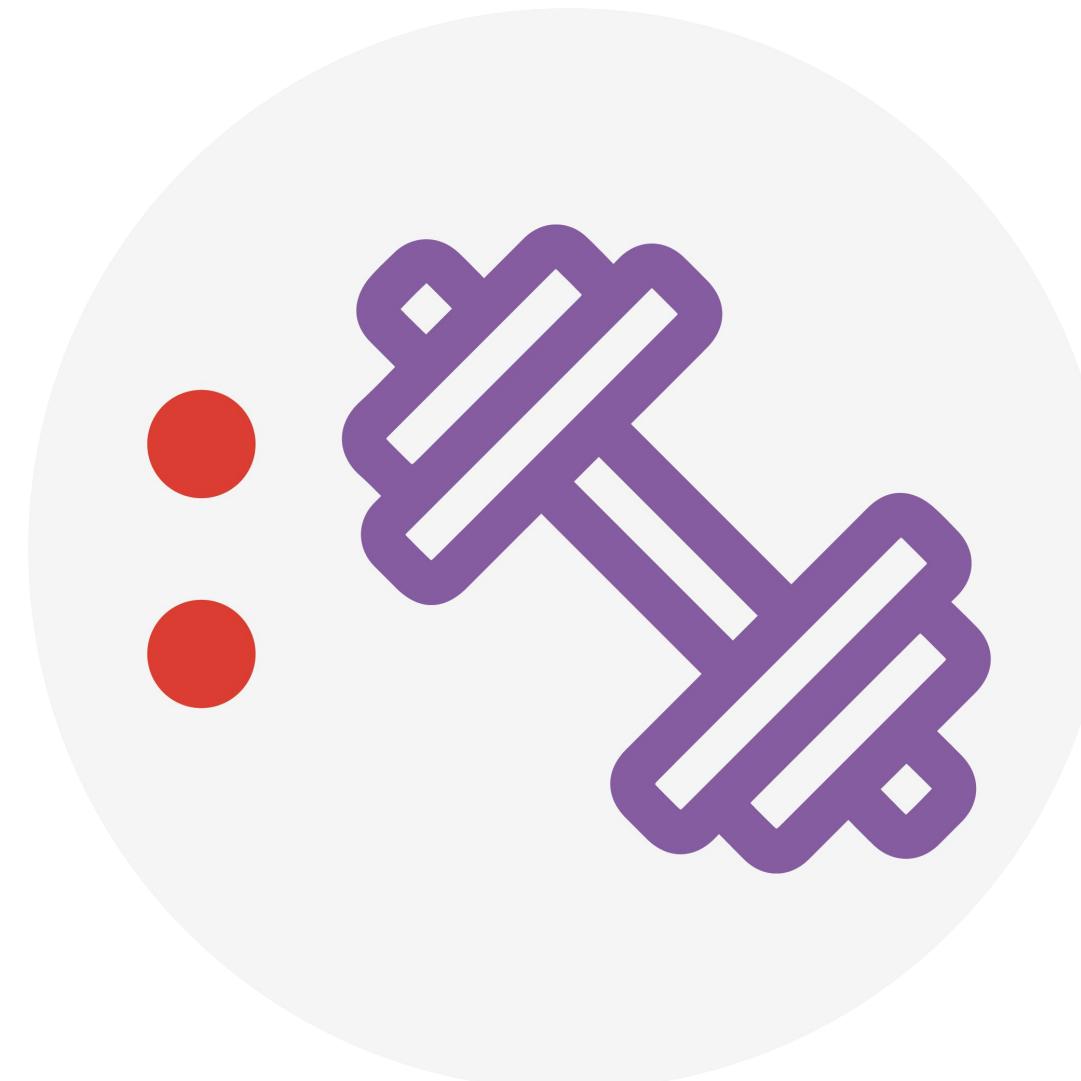


- To use the Time Intelligence functions, a table must be marked as a date table
 - It also allows the model tables to use a non-date type key
 - The date key can be in ISO format, for example, 20210630
- Multiple model tables can be marked as date tables

Live Demo



Power Cycle Lab Exercise 3



Knowledge Check 4



Summary and Q&A

Today, we learned how to:

- Design a Data Model
- Create Model Calculations using DAX

In the next session, we will learn about:

- CALCULATE Function and Modifiers
- Time Intelligence DAX Patterns



DATA SOCIETY:

Power BI Intermediate

Day 3



Recap polling questions



Agenda

- CALCULATE Function and Modifiers
- Time Intelligence DAX Patterns

Lesson 5

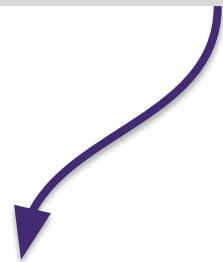
CALCULATE Function & Modifiers

After completing this module, you will be able to:

- Use the Calculate function
- Use filter modifier functions within Calculate
- Examine filter context

DAX Calculate Function

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

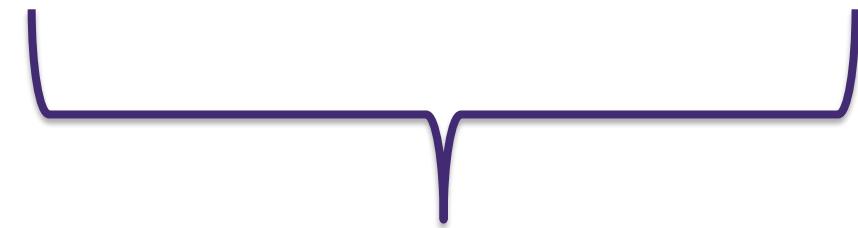


Name of an existing measure, or a DAX formula for a valid measure

Example

[Total Sales] existing measure

SUM(Regionsales[Sales]) a DAX formula



List of simple Boolean (True/False) filter expressions. Note you cannot create filters based on a measure

Example

Product[color] = "Blue"

Calendar[Year] > 2021

DAX Calculate Function

Boolean expression filters must evaluate as TRUE or FALSE

Each filter:

- Can reference only a single column
- Can not reference measures
- Can not use functions that scan or return table

1

```
Revenue Red =  
CALCULATE([Revenue], 'Product'[Color] = "Red")
```

2

```
Revenue Red =  
CALCULATE(  
    [Revenue],  
    FILTER(  
        'Product',  
        'Product'[Color] = "Red"  
    )  
)
```

Table Expression Filters

Table expression filters apply a table object as a filter

Each filter:

- Can be a reference to model table
- Can be a DAX function that returns a table object
- If columns are not in filter context, new filters are added to filter context
- If columns are already in filter context, existing filters are overwritten

```
Revenue High Margin Products =  
CALCULATE(  
    [Revenue],  
    FILTER(  
        'Product',  
        'Product'[List Price] > 'Product'[Standard Cost] * 2  
    )  
)
```

CALCULATE Filter Modifiers

When using CALCULATE, it is possible to pass in filter modification functions

Filter modifier functions include:

- **REMOVEFILTERS** – Remove filters from all tables, a single table or column(s)
- **ALL** – Remove filters from a single table or column(s)
- **ALLEXCEPT** – Remove filters from all columns of a single table, except those explicitly passed in
- **ALLNONBLANKROW** – From the parent table of a relationship, returns all rows but the blank row, or all distinct values of a column but the blank row, and disregards any context filters that might exist

CALCULATE Filter Modifier example

New filter added

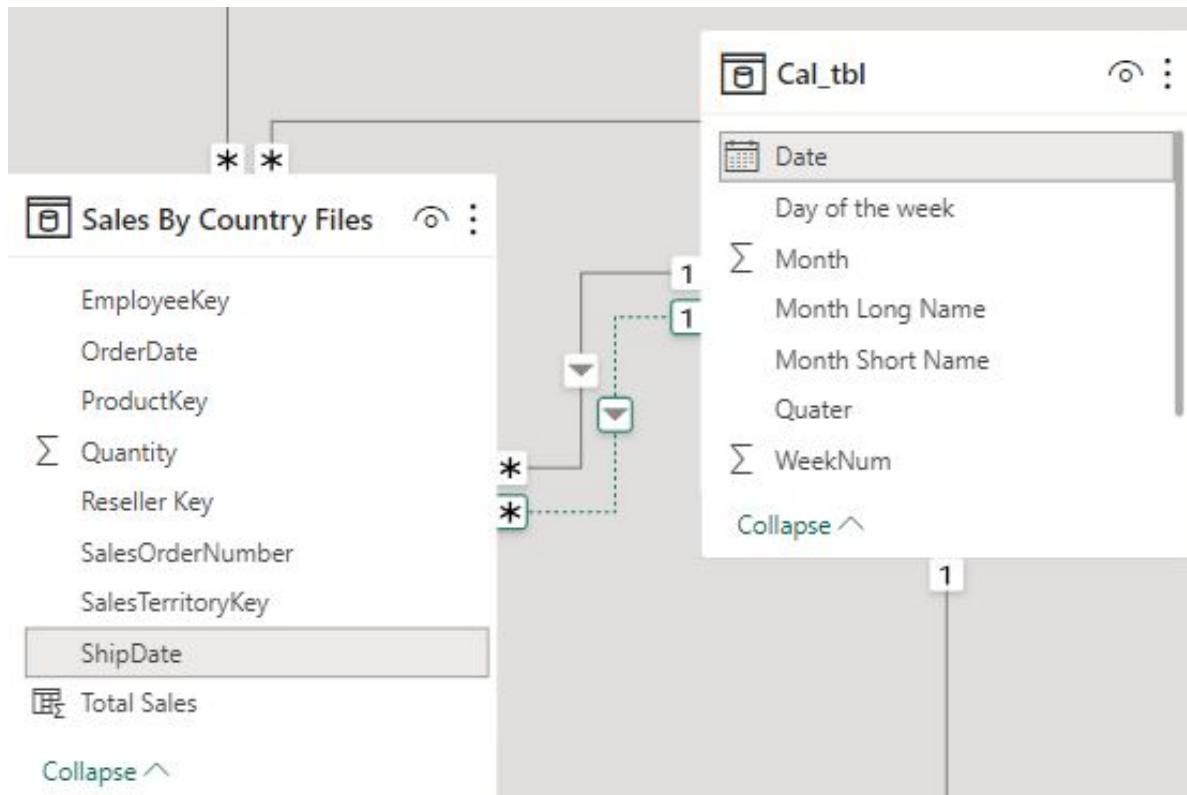
Region	Revenue	Revenue Red
Australia	\$10,655,335.96	\$2,681,324.79
Canada	\$16,355,770.46	\$3,573,412.99
Central	\$7,909,009.01	\$1,585,997.34
France	\$7,251,555.65	\$1,051,014.15
Germany	\$4,878,300.38	\$670,607.30
Northeast	\$6,939,374.48	\$1,876,016.33
Northwest	\$16,084,942.55	\$2,292,905.61
Southeast	\$7,879,655.07	\$1,457,221.07
Southwest	\$24,184,609.60	\$5,345,637.47
United Kingdom	\$7,670,721.04	\$1,063,753.75
Total	\$109,809,274.20	\$21,597,890.81

Existing filter overwritten

Color	Revenue	Revenue Red
Black	\$38,236,124.06	\$21,597,890.81
Blue	\$9,602,850.97	\$21,597,890.81
Grey		\$21,597,890.81
Multi	\$649,030.25	\$21,597,890.81
NA	\$1,099,303.91	\$21,597,890.81
Red	\$21,597,890.81	\$21,597,890.81
Silver	\$19,777,339.95	\$21,597,890.81
Silver/Black	\$147,483.91	\$21,597,890.81
White	\$29,745.13	\$21,597,890.81
Yellow	\$18,669,505.22	\$21,597,890.81
Total	\$109,809,274.20	\$21,597,890.81

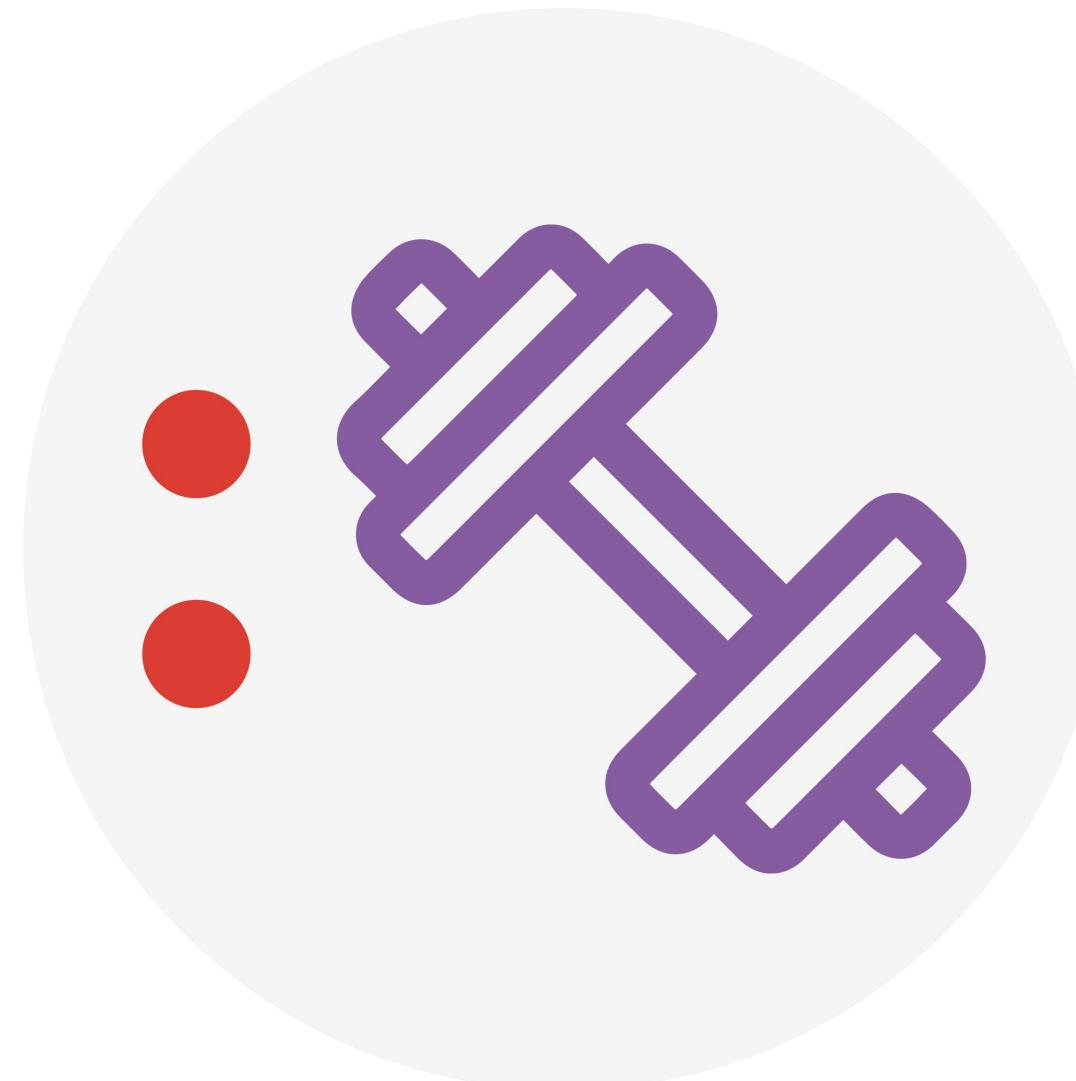
Relationship Modifiers

Use the USERRELATIONSHIP function to make an inactive relationship active during the evaluation of the CALCULATE function

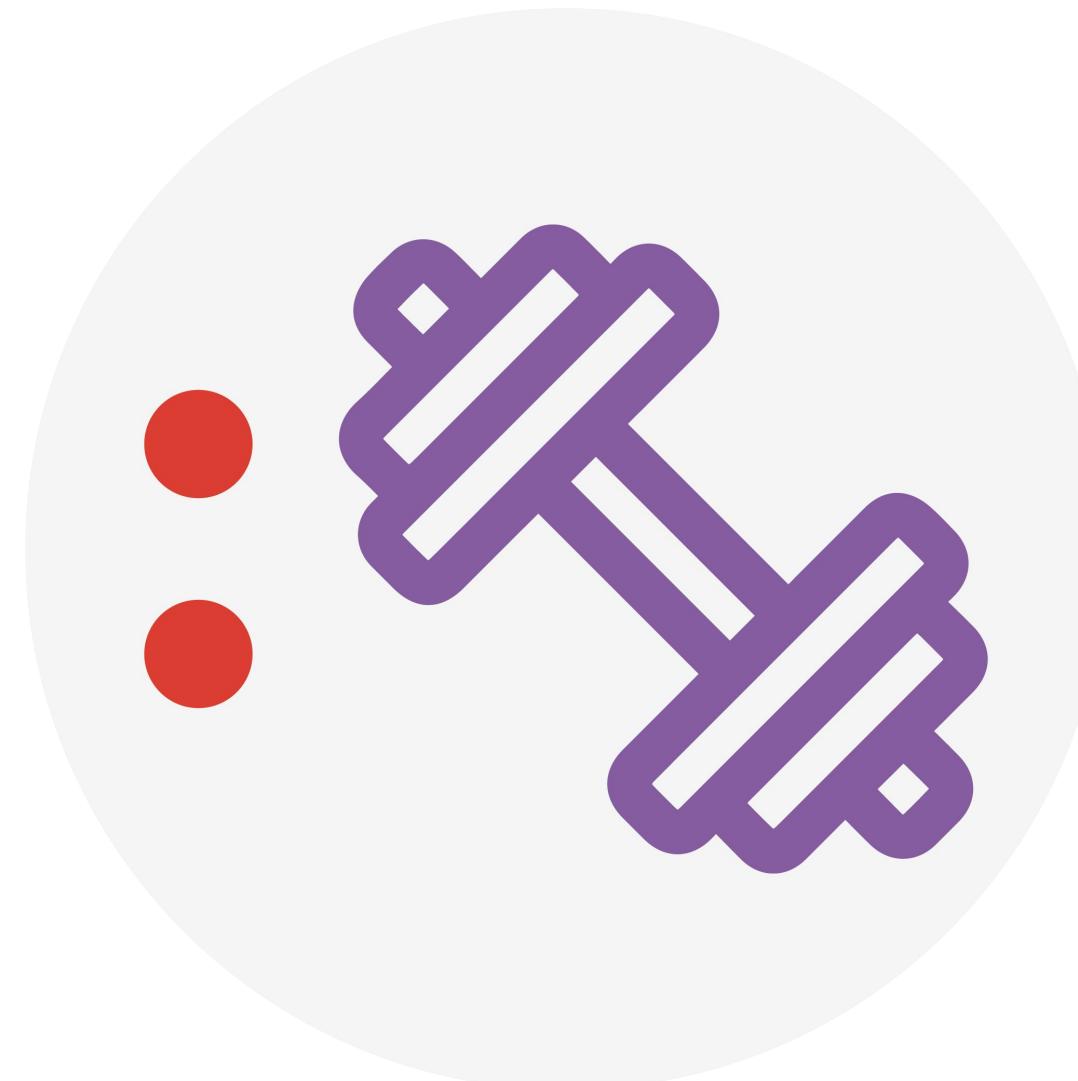


```
Revenue Shipped =  
CALCULATE(  
    [Revenue],  
    USERRELATIONSHIP(  
        'Date'[DateKey],  
        'Sales'[ShipDateKey]  
    )  
)
```

Live Demo



Power Cycle Lab Exercise 4



Knowledge Check 5



Lesson 6

Time Intelligence DAX Patterns

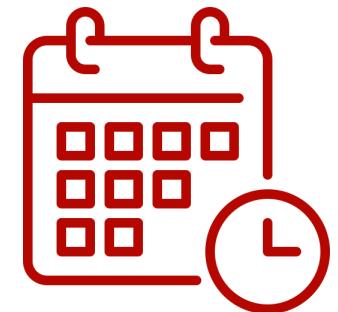
After completing this module, you will be able to:

- Define time intelligence
- Use time intelligence functions
- Understand common time intelligence patterns

Define Time Intelligence

Time Intelligence relates to calculations over dates, months, quarters, or years

- It involves modifying the filter context for date filters
- The model requires a date table
- The date table can be created using the CALENDAR or CALENDARAUTO functions



Define Time Intelligence

Example

Month	Revenue	Revenue YTD
2019 Jan	\$889,902.00	\$889,902.00
2019 Feb	\$837,304.45	\$1,727,206.45
2019 Mar	\$900,089.70	\$2,627,296.15
2019 Apr	\$6,366,809.65	\$8,994,105.80
2019 May	\$5,140,577.65	\$14,134,683.45
2019 Jun	\$3,801,978.60	\$17,936,662.05
2019 Jul	\$3,224,809.45	\$21,161,471.50
2019 Aug	\$2,705,752.60	\$23,867,224.10
2019 Sep	\$2,732,810.85	\$26,600,034.95
2019 Oct	\$6,104,340.55	\$32,704,375.50
2019 Nov	\$5,294,826.15	\$37,999,201.65
2019 Dec	\$4,271,109.60	\$42,270,311.25
Total	\$42,270,311.25	\$42,270,311.25

Use Time Intelligence functions

DAX has many inbuilt Time Intelligence functions

- For example, TOTALYTD and PREVIOUSYEAR
- It is possible to replace them using CALCULATE
- However, that requires more work and could be slower for Power BI to evaluate

Common Time Intelligence Patterns

- Cumulative totals
- Period comparisons
- New customer orders by month
- Stock on-hand value

Summarization over time

- DATESYTD – Returns a single-column table of dates
- TOTALYTD – Evaluates expression for year-to-date (YTD)
- DATESBETWEEN – Returns single-column table with range dates
- DATESINPERIOD – Returns single-column table with range of dates

Shifting time periods

- DATEADD – Returns single-column table of dates shifted by interval
- PARALLELPERIOD – Returns single-column table of parallel dates
- SAMEPERIODLASTYEAR – Returns dates shifted back one year
- Many others – NEXTDAY, NEXTMONTH, NEXTQUARTER, NEXTYEAR, PREVDAY, PREVMONTH, PREVQUARTER ,PREVYEAR

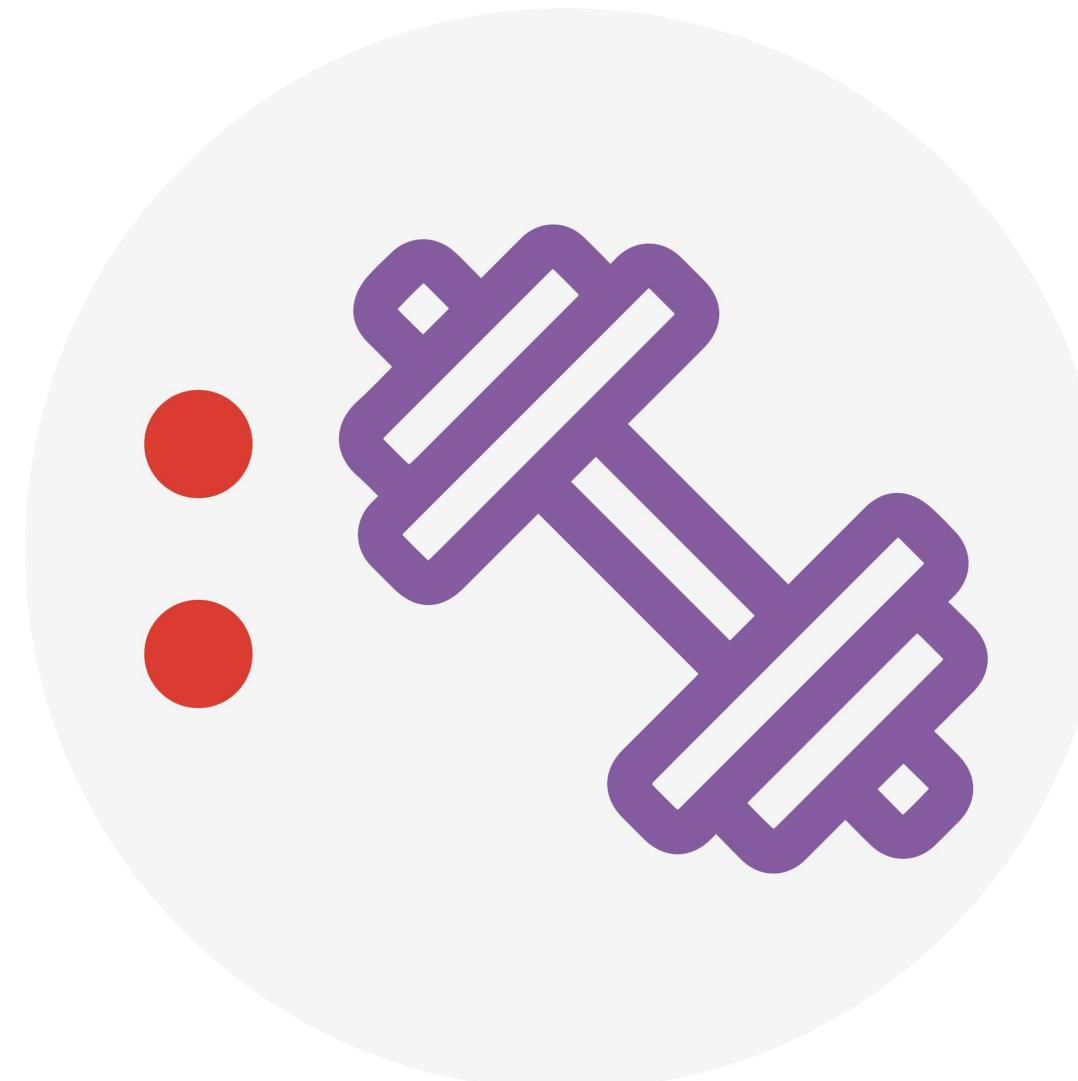
Shifting time periods

- DATEADD – Returns single-column table of dates shifted by interval
- PARALLELPERIOD – Returns single-column table of parallel dates
- SAMEPERIODLASTYEAR – Returns dates shifted back one year
- Many others – NEXTDAY, NEXTMONTH, NEXTQUARTER, NEXTYEAR, PREVDAY, PREVMONTH, PREVQUARTER ,PREVYEAR

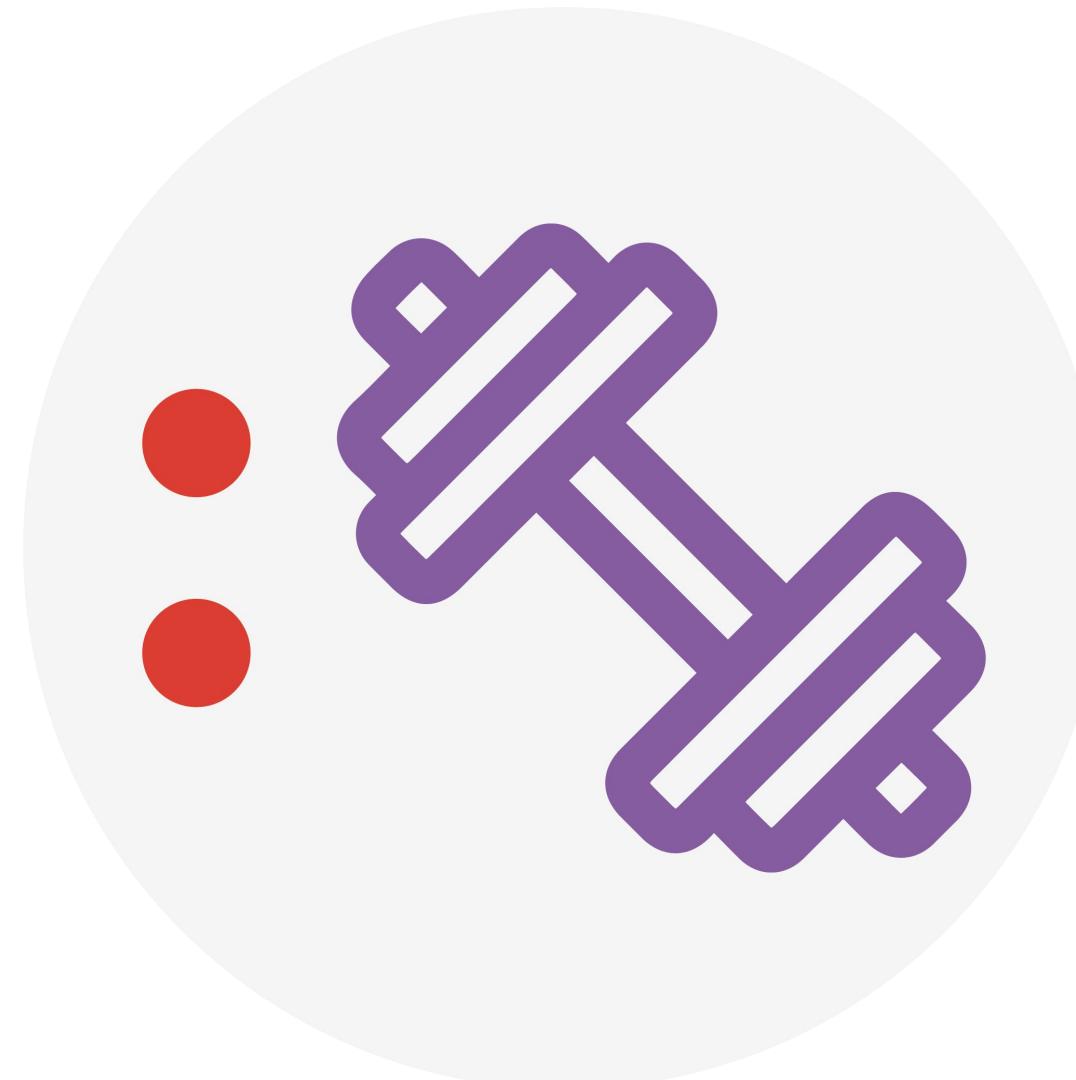
Return a single date

- **FIRSTDATE** – Returns first date in current filter context
- **LASTDATE** – Returns last date in current filter context
- **EOMONTH** - Returns the date in date time format of the last day of the month before or after a specified number of months.

Live Demo



Power Cycle Lab Exercise 5



Knowledge Check 6



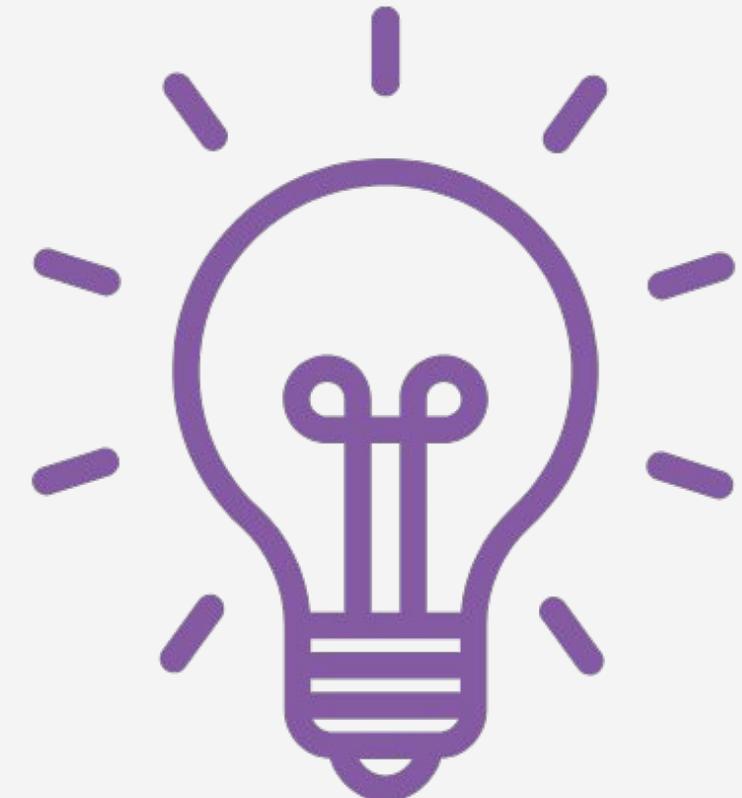
Summary

Today, we learned how to use:

- CALCULATE Function and Modifiers
- Time Intelligence DAX Patterns

In the next session, we will learn about:

- Advanced Report Design Principles
- Row-Level Security
- Power BI App Deployment



DATA S^AOCIETY:

Power BI Intermediate Day 4



Agenda

- Advanced Report Design Principles
- Row-Level Security
- Power BI App Deployment

Warm up

- Check out this [article](#) on tips for creating PowerBi dashboards
- Which tips were the most interesting or surprising? Can you think of other tips that you've used in your own work?
- Share your answers in the chat box



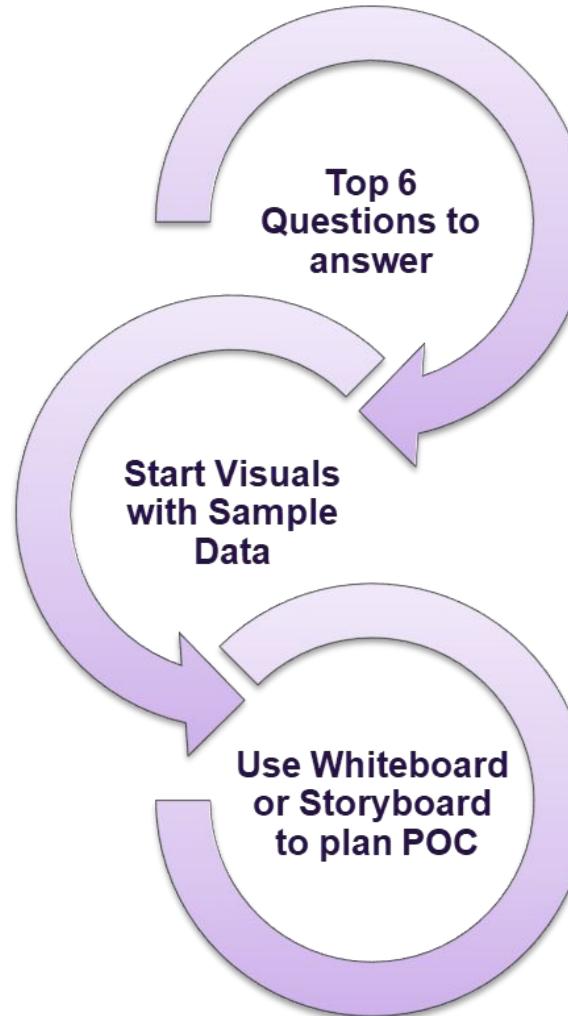
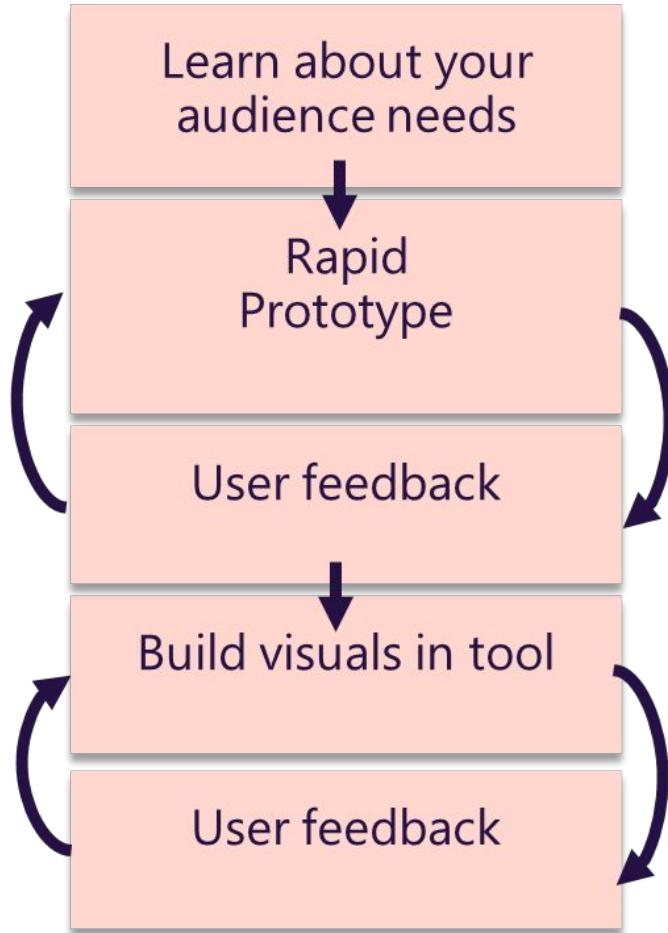
Lesson 7

Advanced Report Design Principles

After completing this module, you will be able to:

- Understand the importance of data storytelling
- Plan report layout, user interface and flow
- Utilize best practices for custom report navigation, report tooltips and drill-through options

Agile Report Creation

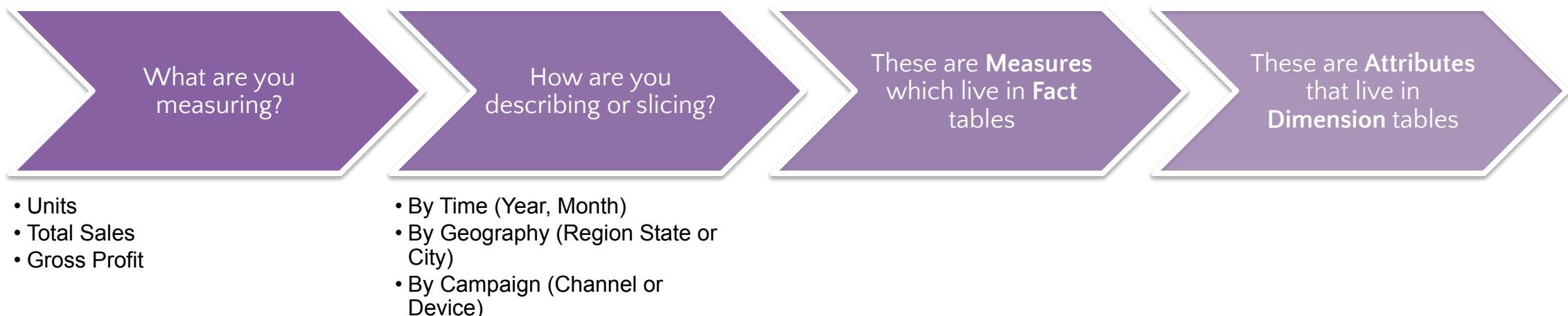


Who? What? When? Where? How Many?
Why?

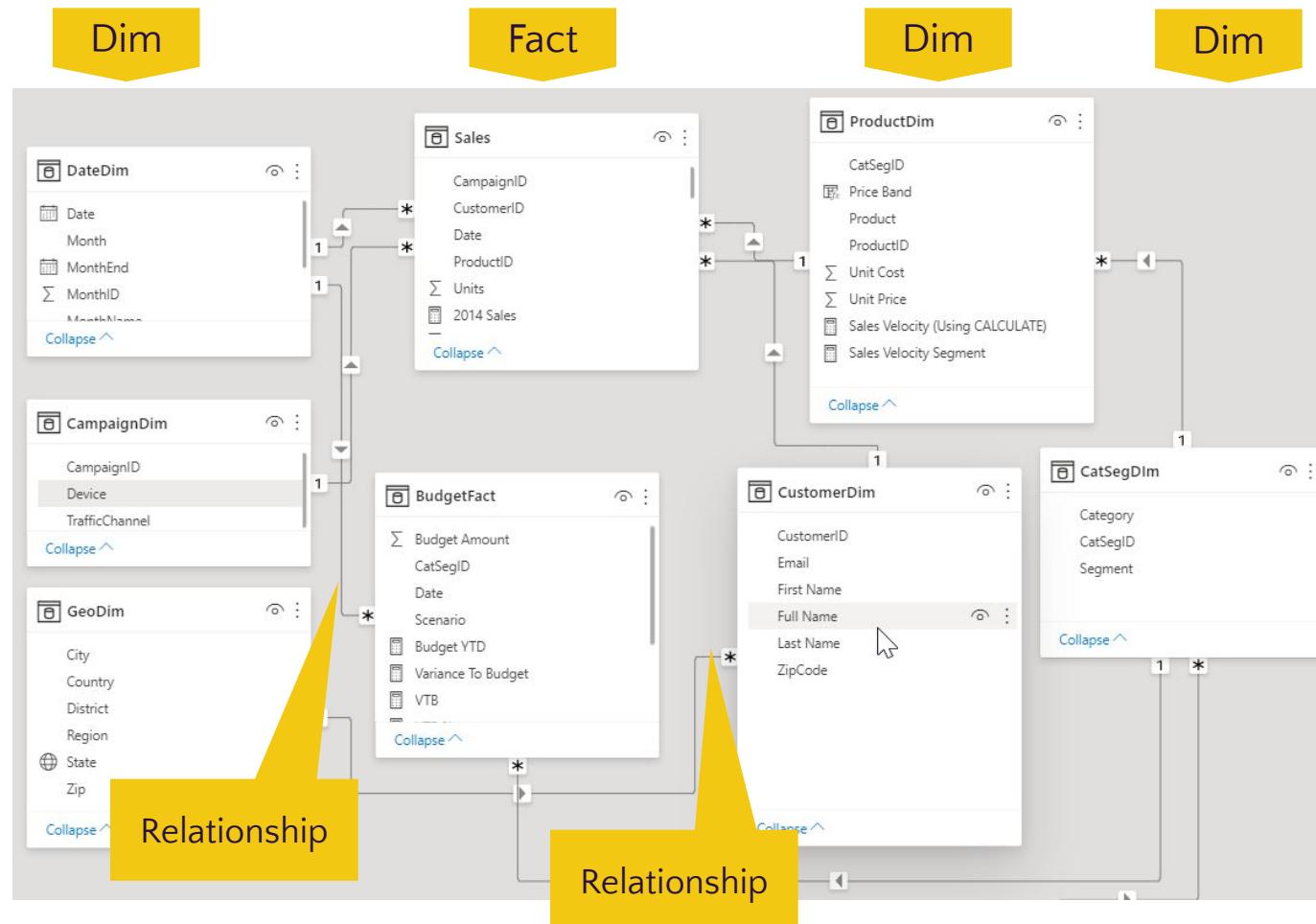
Convert the Story to a Date Model

List your big questions:

- What is my Total Sales for a Selected Year and Region?
- How is my Total Sales doing Year Over Year?
- How are my Units trending for various States in my region?
- How is my Sales doing by Channel, Device, Category for selected Year?
- Which categories are performing best to worst by Total Sales ?

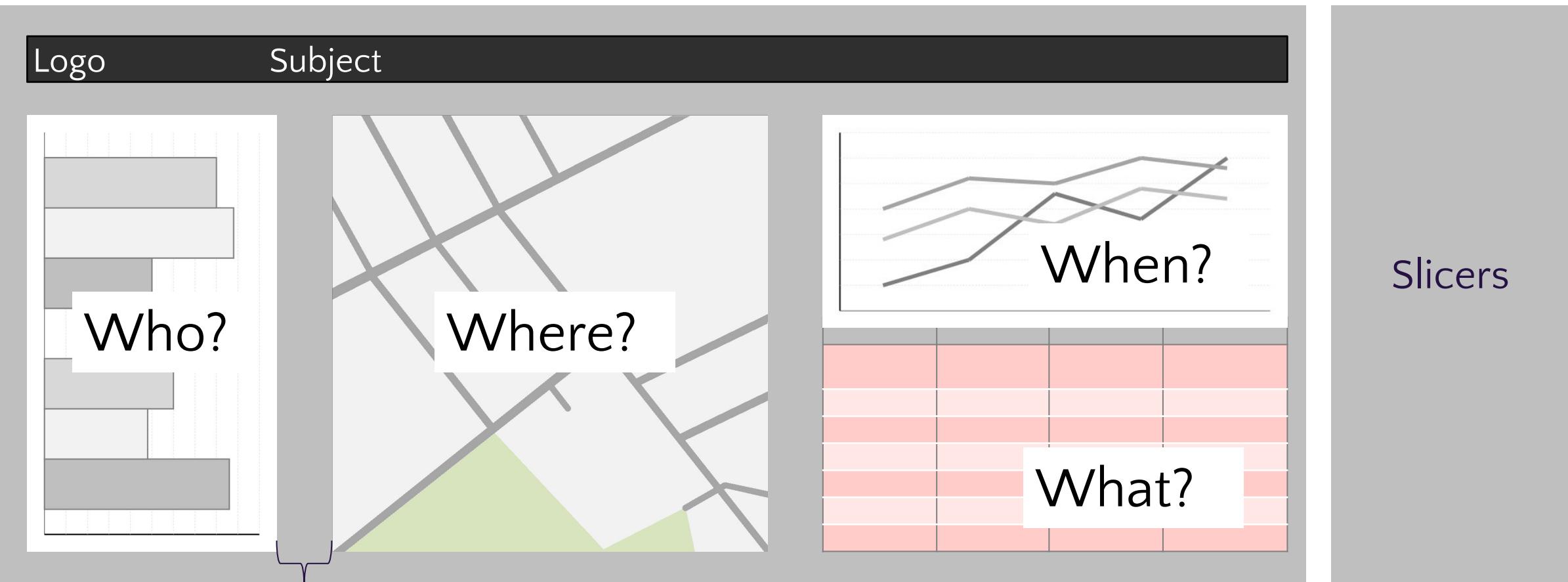


Convert Story to a Data Model



- Measures (e.g., Units or Sales) live in fact tables
- Descriptive attributes (e.g., Campaign, Customer name) live in dimension tables
- Relationships tie the data together so you can slice your measures by your attributes

Use Storyboarding to Design Report Layout

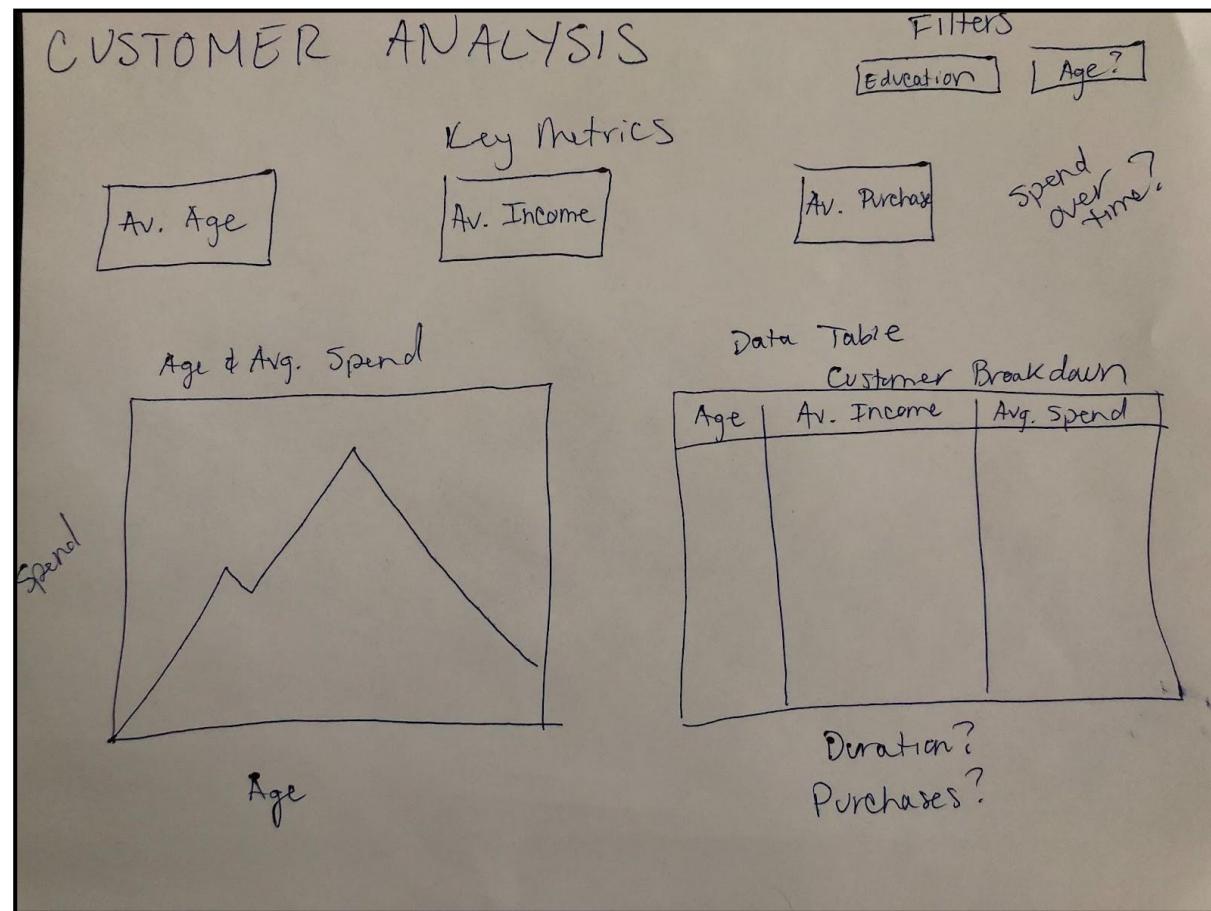


Control this space. If all objects tell the same story, use the same gap space. If some objects tell a different story, make the gap bigger

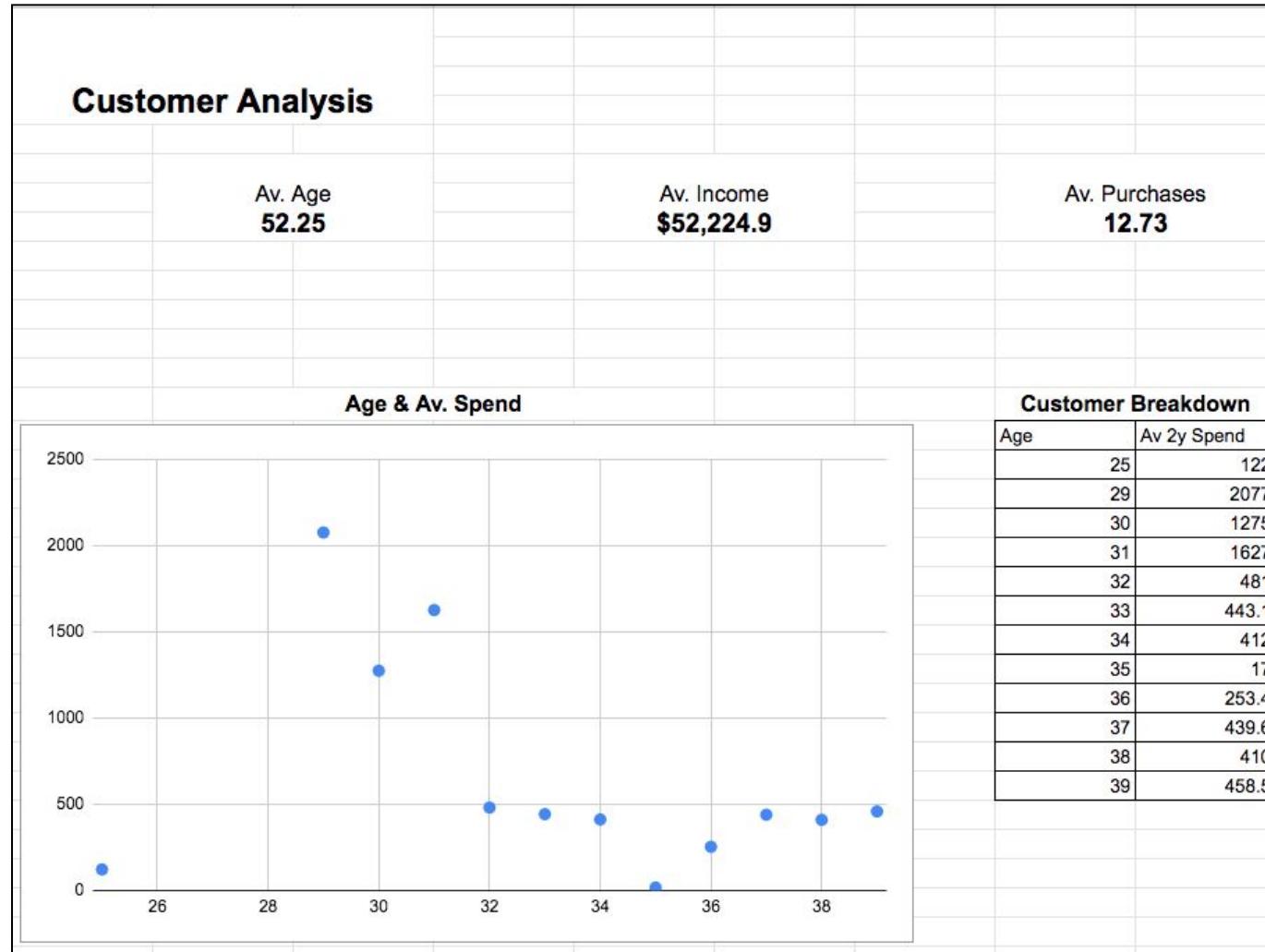
Visual Selection

Comparison	Data Over Time	Correlation	Distribution	Part-to-whole	Ranking
 Bar Chart	 Bar Chart	 Bubble Chart	 Bubble Chart	 Donut	 Ord. Column
 Grouped Bar	 Line Chart	 Column Line	 Grouped Bar	 Stacked Bar	 Ordered Bar
 Line Chart	 Stacked Bar	 Scatterplot		 Treemap	 Ribbon
 Bubble Chart	 Area Chart				 Decomp. Tree
 Area Chart	 Stacked Area				 Funnel
 Stacked Bar	 Bubble Chart				
 Ribbon	 Waterfall				
 Shape Map					

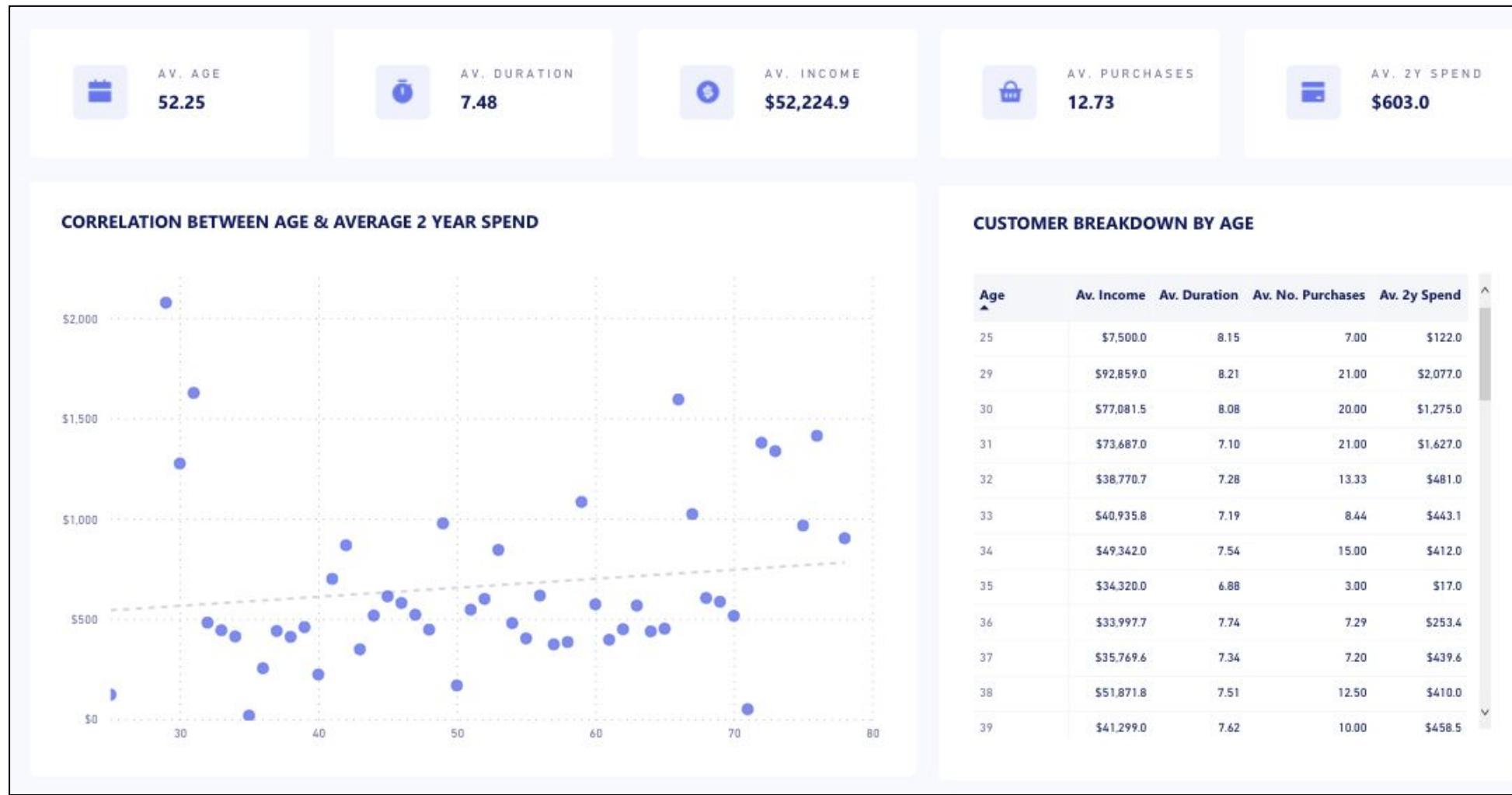
Whiteboard Mock-up



Wireframe



Final Product



Page Navigation

Home

Global Sales

EU Sales

NA Sales

JP Sales

Details

Company Analysis

Genre Analysis

Platform Analysis

Global Sales by Year and Company

Company Microsoft ● Nintendo ● PC ● Sony

100%

50%

0%

1990 1995 2000 2005 2010 2015 2020

Year

Global Sales

Microsoft

Sony

PC

Nintendo

Game

Company

Critic Score

Rating

Year

2002

Spider-Man: The Movie

Nintendo

78 E

2002

SpongeBob SquarePants: Revenge of the Flying Dutchman

Nintendo

71 E

2002

Spyro 2: Season of Flame

Nintendo

76 E

2002

Star Fox Adventures

Nintendo

82 T

2002

Super Mario Sunshine

Nintendo

92 E

2002

Super Monkey Ball 2

Nintendo

87 E

2002

The Legend of Zelda: A Link to the Past

Nintendo

95 E

2002

The Legend of Zelda: The Wind Waker

Nintendo

96 E

2002

Age of Mythology

PC

89 T

Year	Game	Company	Critic Score	Rating
2002	Spider-Man: The Movie	Nintendo	78	E
2002	SpongeBob SquarePants: Revenge of the Flying Dutchman	Nintendo	71	E
2002	Spyro 2: Season of Flame	Nintendo	76	E
2002	Star Fox Adventures	Nintendo	82	T
2002	Super Mario Sunshine	Nintendo	92	E
2002	Super Monkey Ball 2	Nintendo	87	E
2002	The Legend of Zelda: A Link to the Past	Nintendo	95	E
2002	The Legend of Zelda: The Wind Waker	Nintendo	96	E
2002	Age of Mythology	PC	89	T

Company Analysis

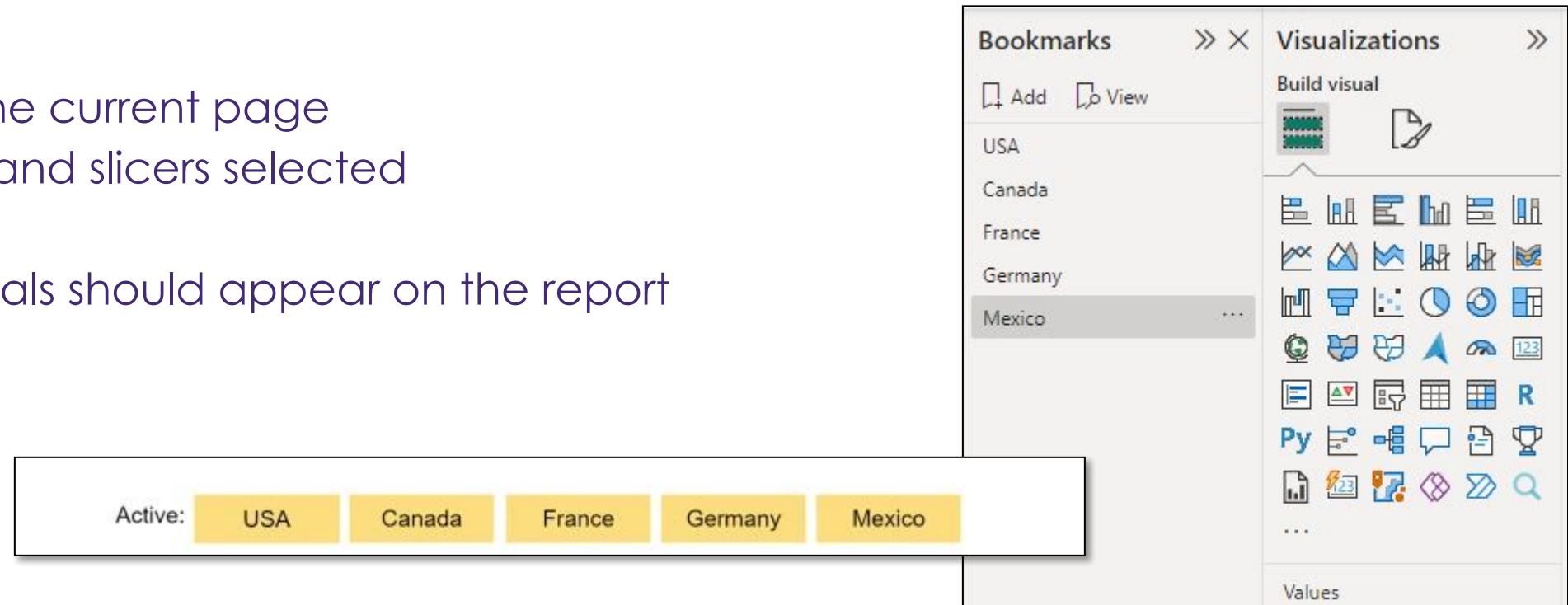
Genre Analysis

Platform Analysis

Bookmark

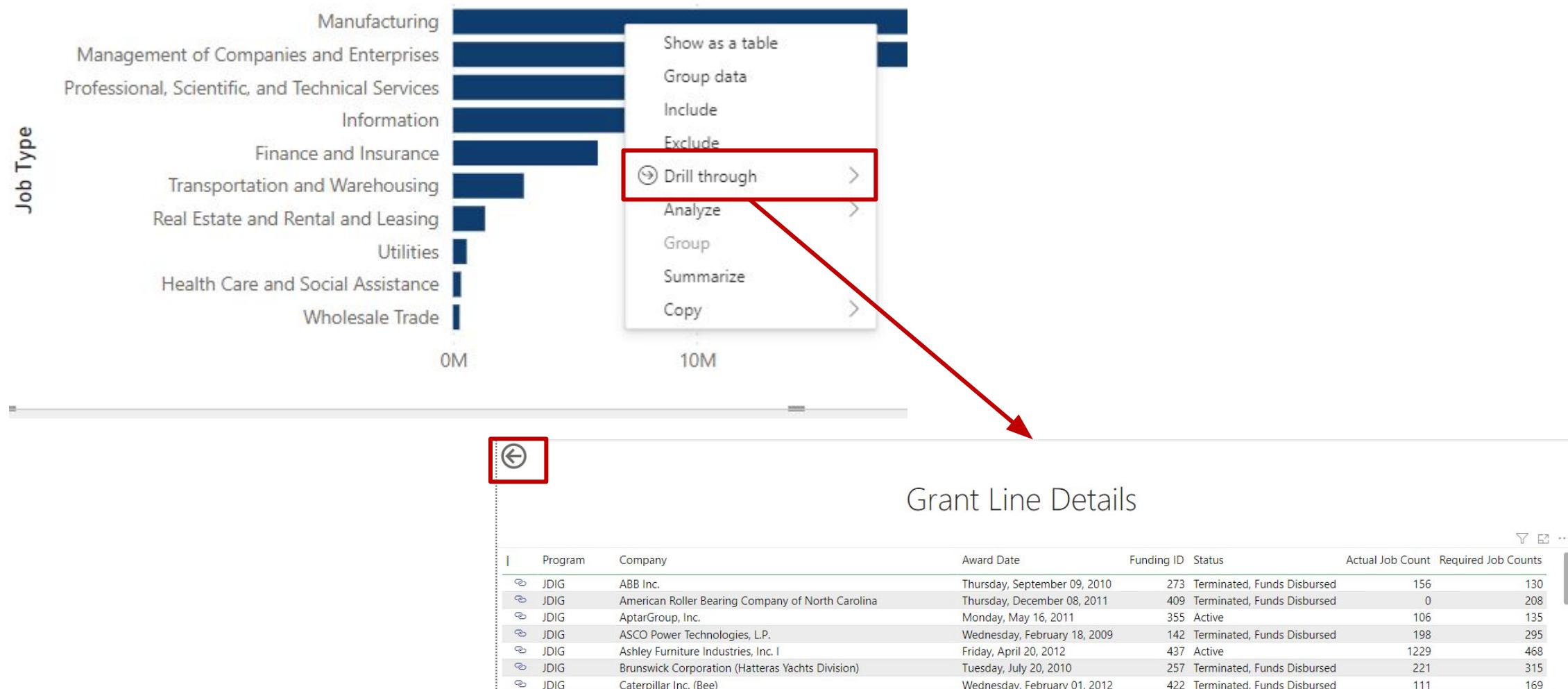
Bookmarks let you save interesting states as part of your report. Once you have a list of bookmarks, you can use these in several ways including organizing and transitioning visuals:

- Includes the current page
- Any filters and slicers selected
- Sort Order
- Which visuals should appear on the report



Report Drill-through

Total Actual Award by Job Type



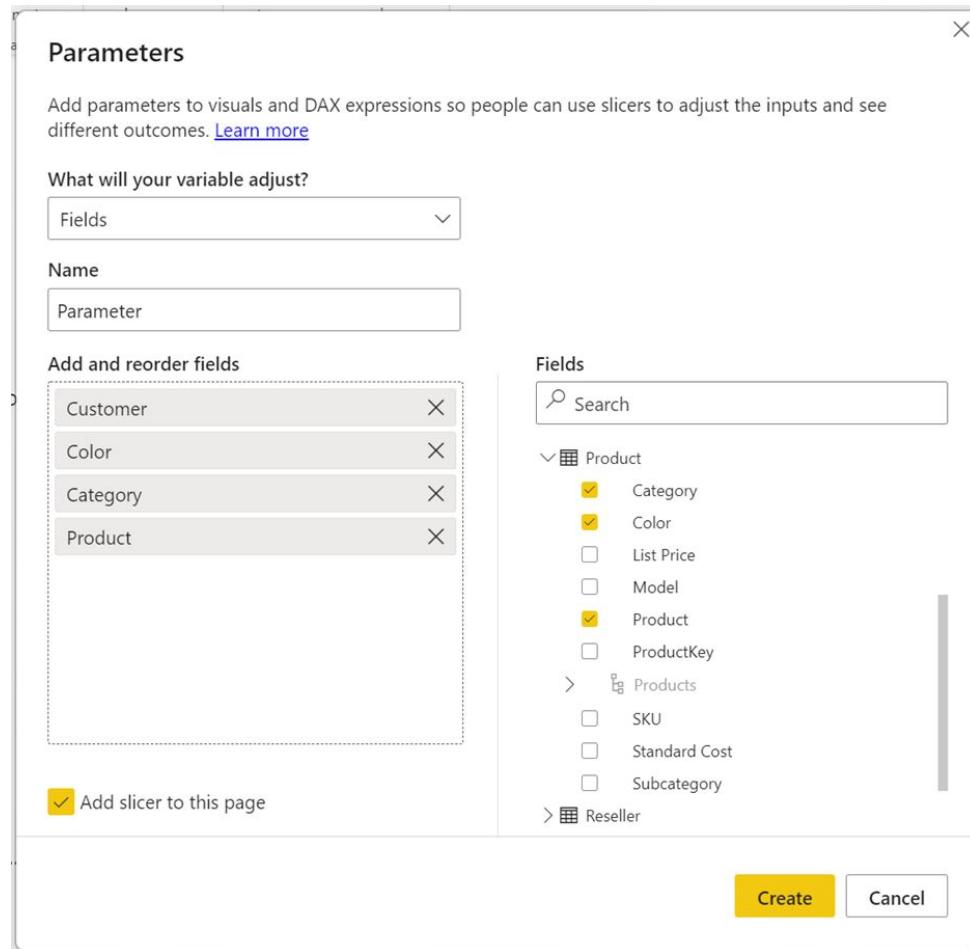
Custom Tooltip



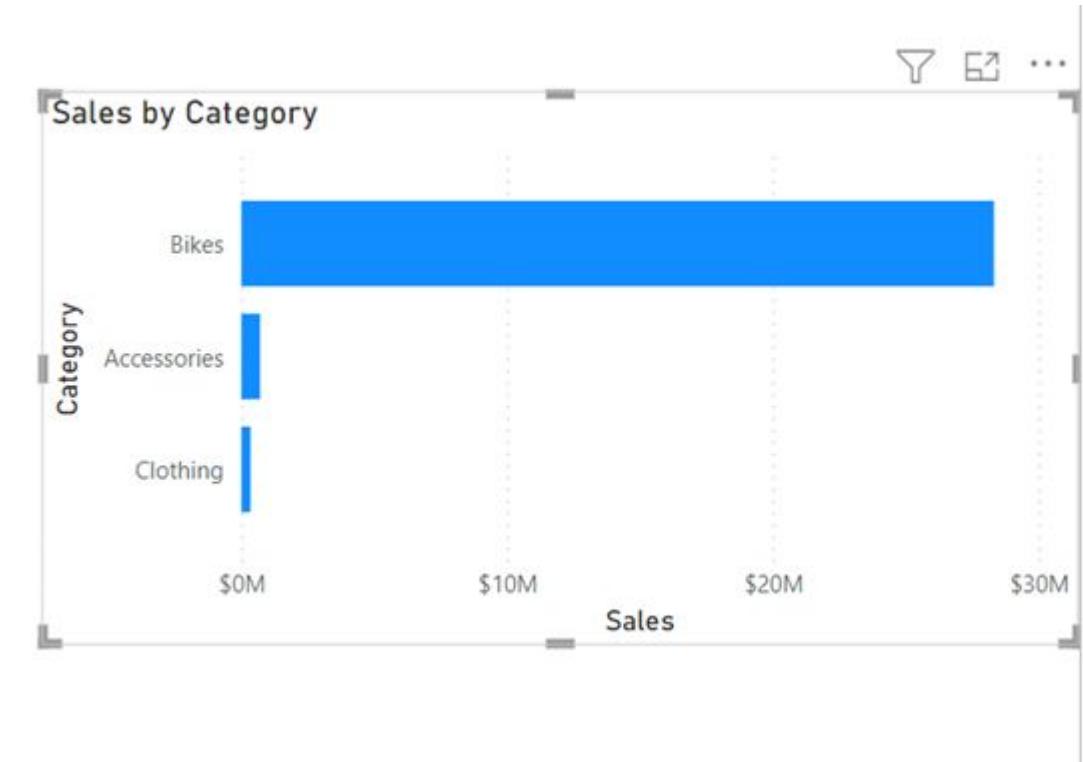
Custom tooltips in Power BI let you create personalized and interactive data point descriptions that appear when users hover over visuals, enhancing the user experience with tailored information and insights.



Field Parameters



Parameter
 Customer
 Category
 Color
 Product



Performance Analyzer

Built-in performance analysis

- Each visual has three components
 - How long did the DAX query take?
 - How long did my visual take to render?
 - How long was everything else?

The screenshot shows the Power BI Performance Analyzer interface. At the top, there are four buttons: 'Pause visuals' (bar chart), 'Refresh visuals' (refresh icon), 'Optimization presets' (chart with gear), and 'Performance analyzer' (clock icon). The 'Performance analyzer' button is highlighted with a red box. To its right is a 'Review' button. Below these are two tabs: 'Report' and 'Performance analyzer'. The 'Performance analyzer' tab is selected and has a red border around it. It contains a table titled 'Performance analyzer' with columns 'Name' and 'Duration (ms)'. The table shows several recorded queries with their durations: Recording started 9/27/2023 5:21:00 PM, 237 ms, 638 ms, 79 ms, 18 ms, 541 ms, 637 ms, 580 ms, 670 ms, and 715 ms. Below the table is a section titled 'Options' with two sections: 'GLOBAL' and 'CURRENT FILE'. Under 'GLOBAL', there are checkboxes for 'Reduce number of queries sent by' (Disabling cross highlighting/filtering by default) and 'Show an Apply button and only send queries once for' (Slicer selections, Filter selections). Under 'CURRENT FILE', there are checkboxes for 'Data Load', 'Regional Settings', 'Privacy', 'Auto recovery', 'DirectQuery', and 'Query reduction'. At the bottom right are 'OK' and 'Cancel' buttons.

Name	Duration (ms)
Recording started 9/27/2023 5:21:00 PM	-
	-
	237
	638
	79
	18
	541
	637
	580
	670
	715

Options

GLOBAL

Reduce number of queries sent by
 Disabling cross highlighting/filtering by default

Show an Apply button and only send queries once for
 Slicer selections
 Filter selections

CURRENT FILE

Data Load
Regional Settings
Privacy
Auto recovery
DirectQuery
 Query reduction
Report settings

Use Query Reduction Features

Scenario

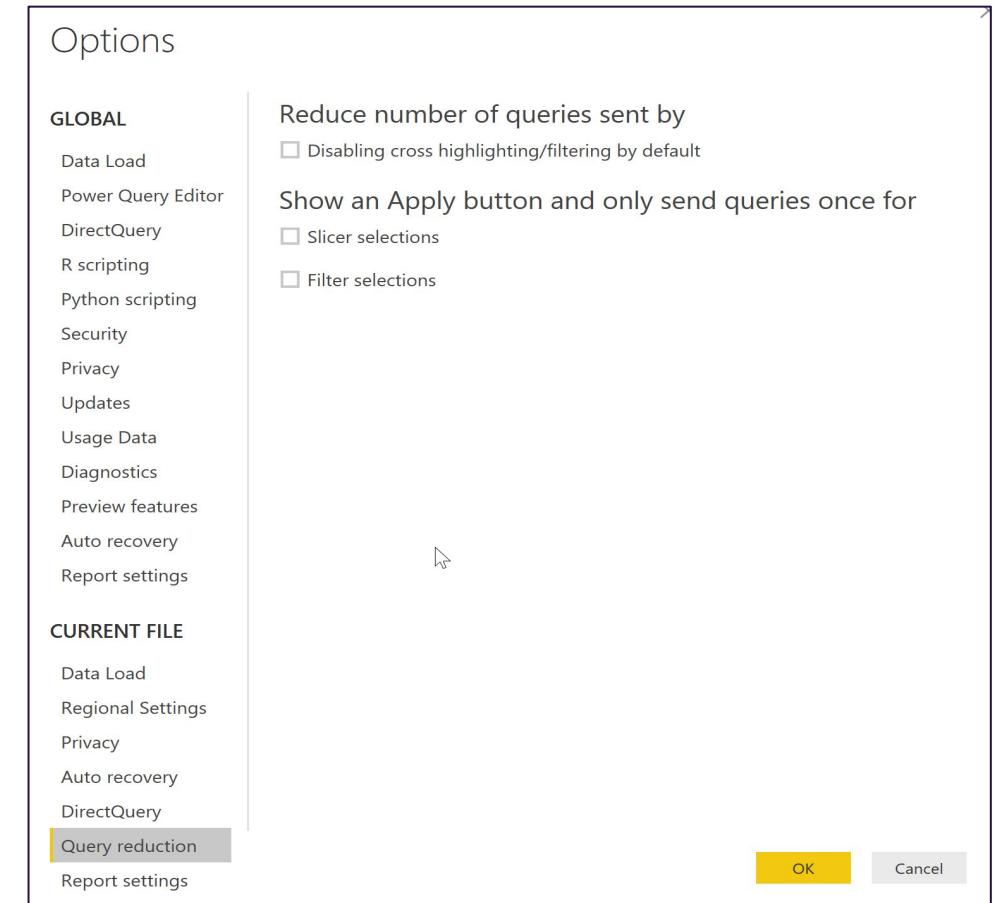
- Previously discussed high number of visuals/slicers

Why is it desired?

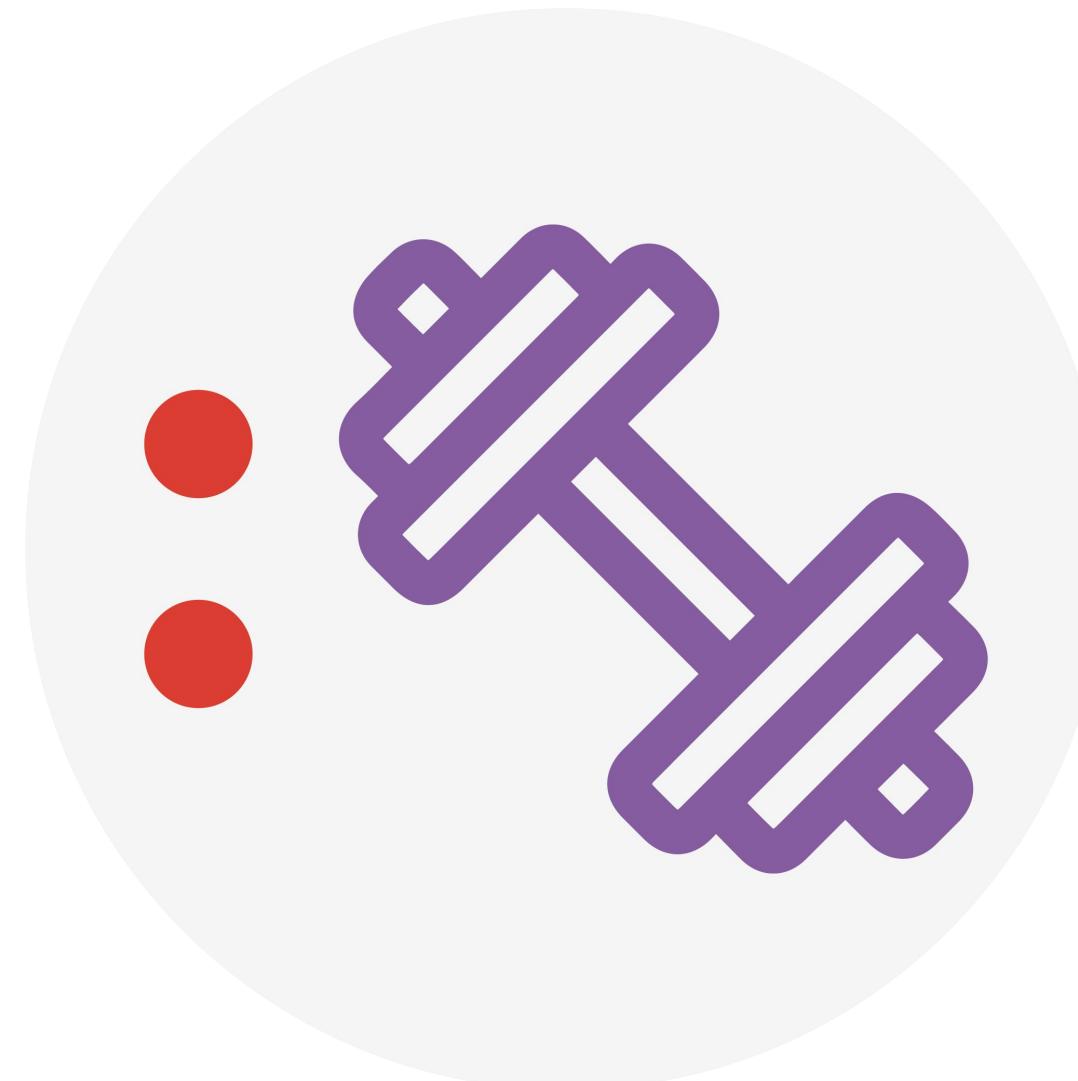
- Lower memory load, less data to fetch and process

Proposed solutions

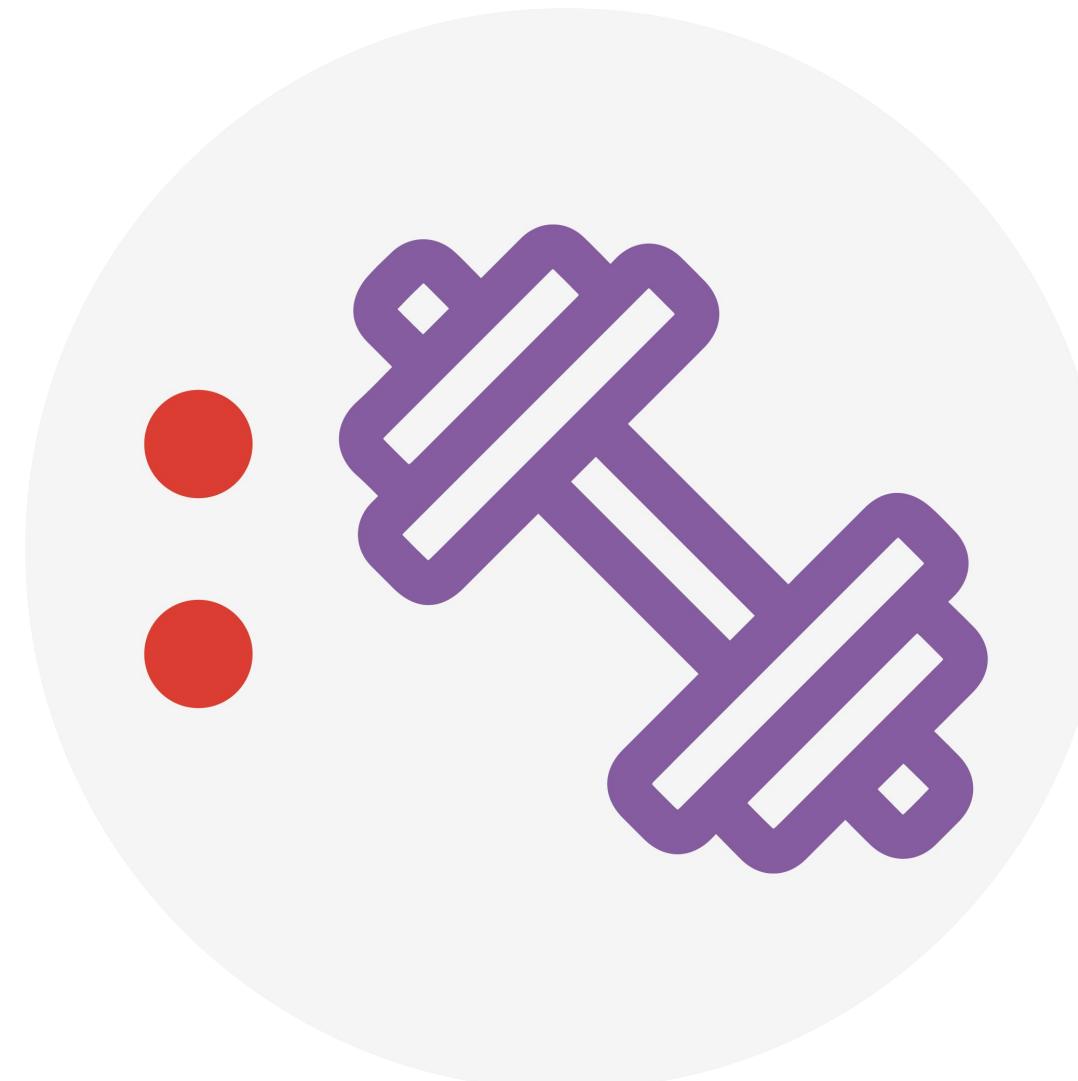
- consider no cross-highlighting, and
- adding Apply buttons to slicers and filters



Live Demo



Power Cycle Lab Exercise 6



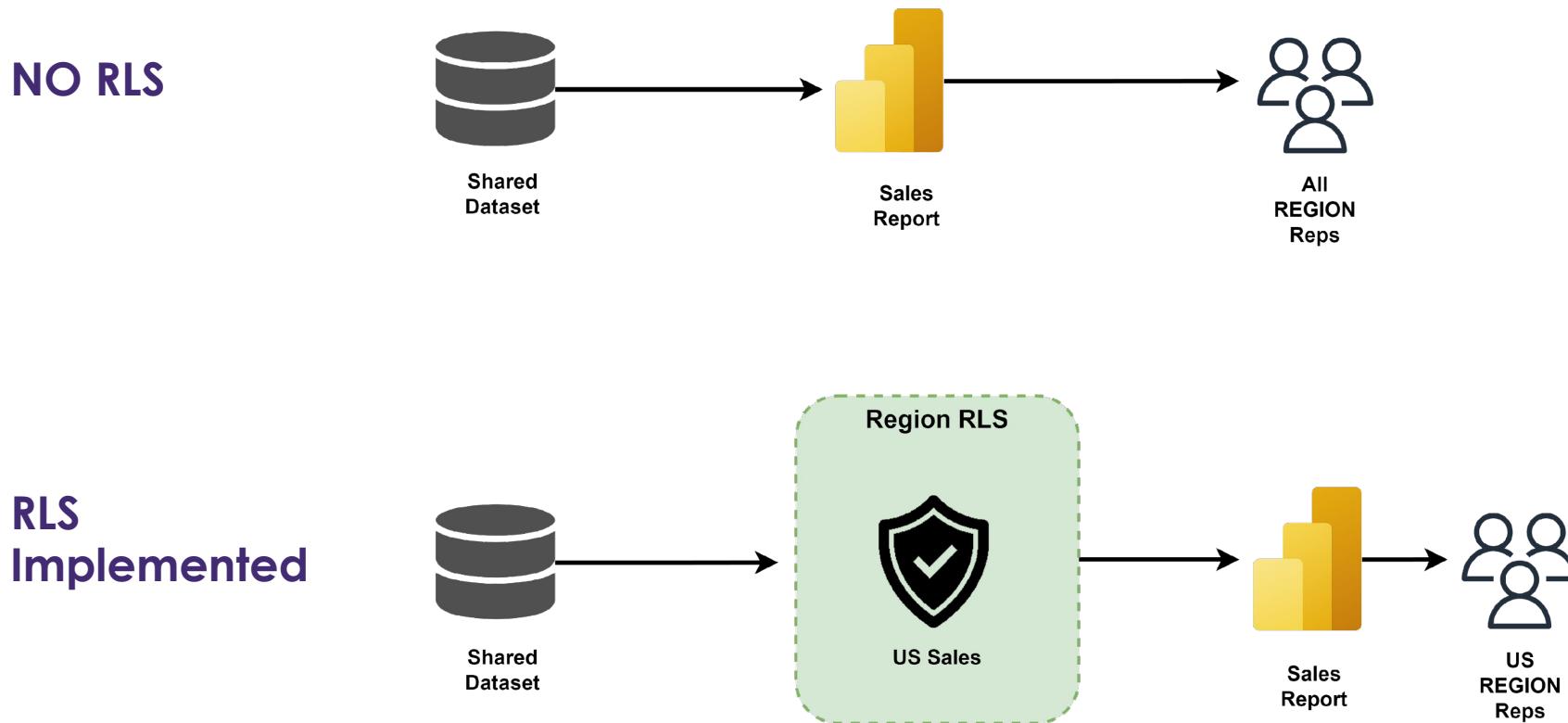
Lesson 8

Row-Level Security

After completing this module, students will be able to:

- Define RLS
- Implement Static RLS
- Implement Dynamic RLS

What is Row-Level Security



Row-Level Security (RLS) in Power BI is a security feature that allows you to control access to data at a granular level. With RLS, you can define rules that restrict which rows of data users can see in a dataset or report based on their roles.

Static Row-Level Security

Manage roles

Roles
Automotive
Clothing
Game
Sports

Tables

Employees

Product

Table filter DAX expression

`[department] = "Game"`

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

Save

Cancel



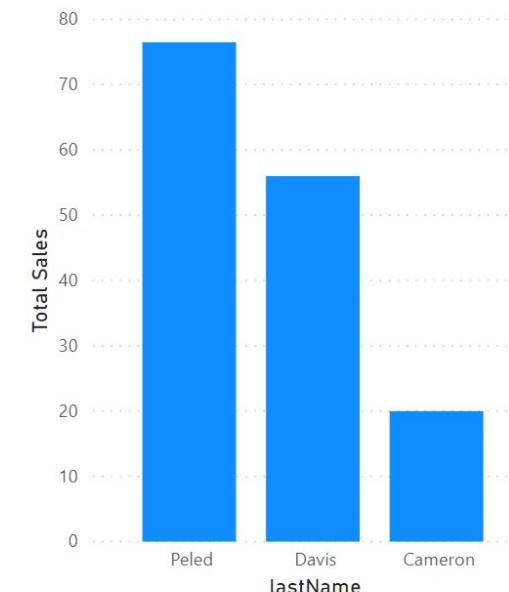
152.47

Total Sales

Top Products Sales

product	department	Total Sales
Spider, spider	Game	44.50
Invest in it All	Game	31.99
Santo Domingo	Game	31.00
Settlers of Air	Game	24.99
Lords of Avalon	Game	19.99
Total		152.47

Top Employees



Setting Up RLS in Power BI Service

The screenshot shows the Power BI service workspace with a list of reports and datasets. The dataset 'Lab 7 -RLS' is selected and highlighted with a red box. A context menu is open over the dataset, with another red box highlighting the 'Security' option at the bottom of the menu.

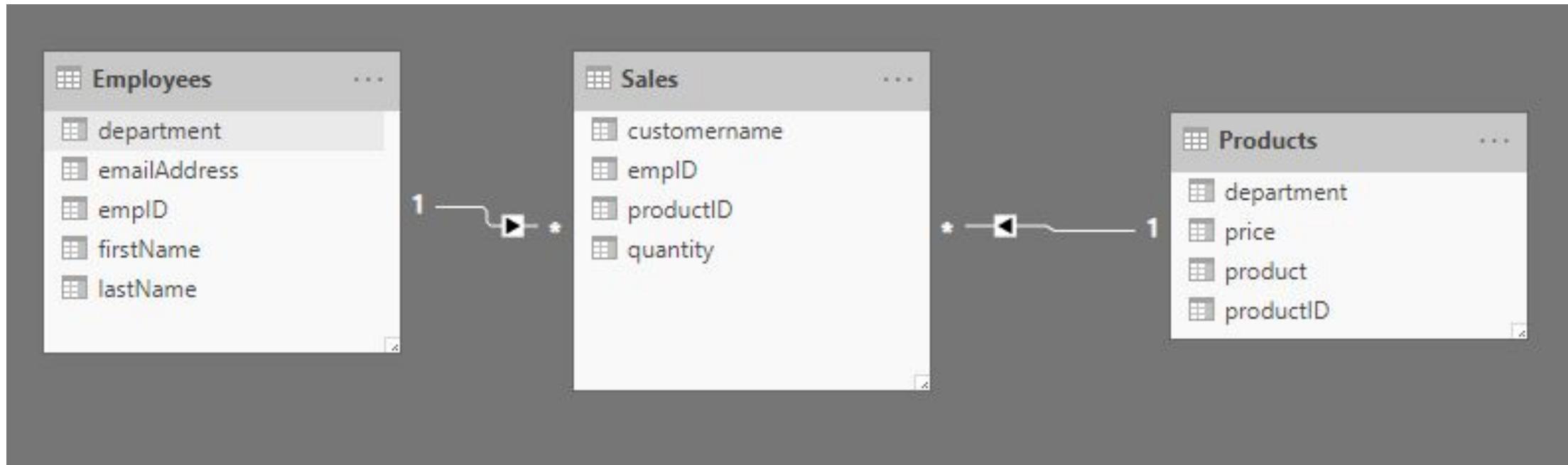
From your Power BI service workspace you will need to select the dataset security settings

The screenshot shows the 'Row-Level Security' configuration page. It lists two regions: 'Canada Region (0)' and 'US Region (0)'. On the right, there's a 'Members (0)' section titled 'People or groups who belong to this role'. It contains a list of users: 'Shannon Lynch' and 'Tripat Gill', each with a delete icon. Below the list is a placeholder 'Enter email addresses' and a green 'Add' button. A red arrow points from the 'Canada Region (0)' section towards the user list.

Add the users that should be assigned to the applicable RLS role

RLS Security

Secure reports and workspace by sharing them to Active Directory users and groups.



Dynamic RLS

Manage roles

Roles

EmployeeEmailAddress

Create

Delete

Tables

Employees



...

Products

...

Sales

...

Table filter DAX expression

[emailAddress] = userprincipalname()

Row-Level Security

Eastern US (0)

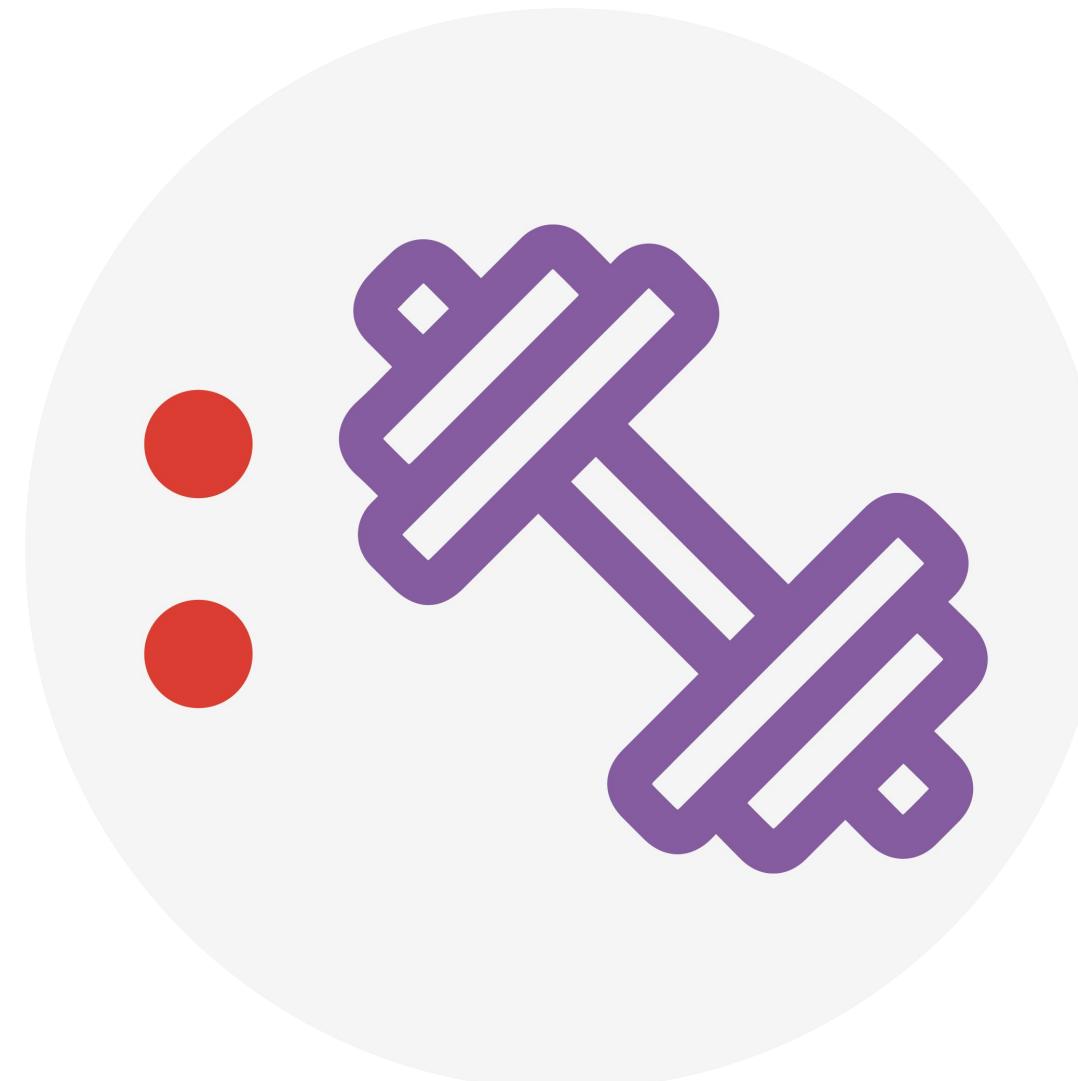
Members (0)

People or groups who belong to this role

Enter email addresses

Add

Live Demo



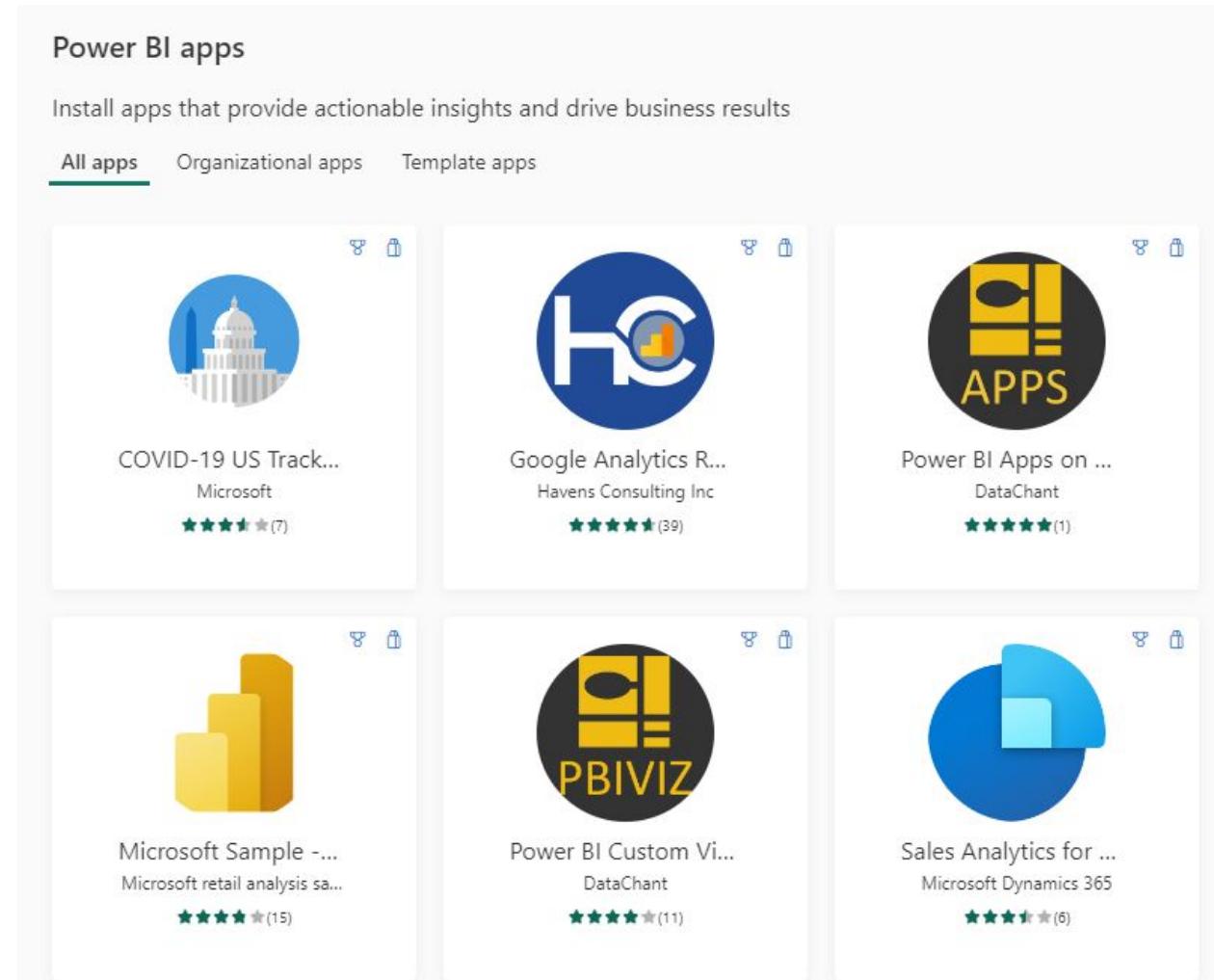
Knowledge Check 7



Power BI App Deployment

Publishing your report App

- Within Power BI, you have the capability to craft formalized content packages and subsequently share them widely in the form of apps
- These apps are developed within workspaces, fostering collaborative work on Power BI content alongside your peers
- After completion, you can release the finalized app to a broad audience within your organization

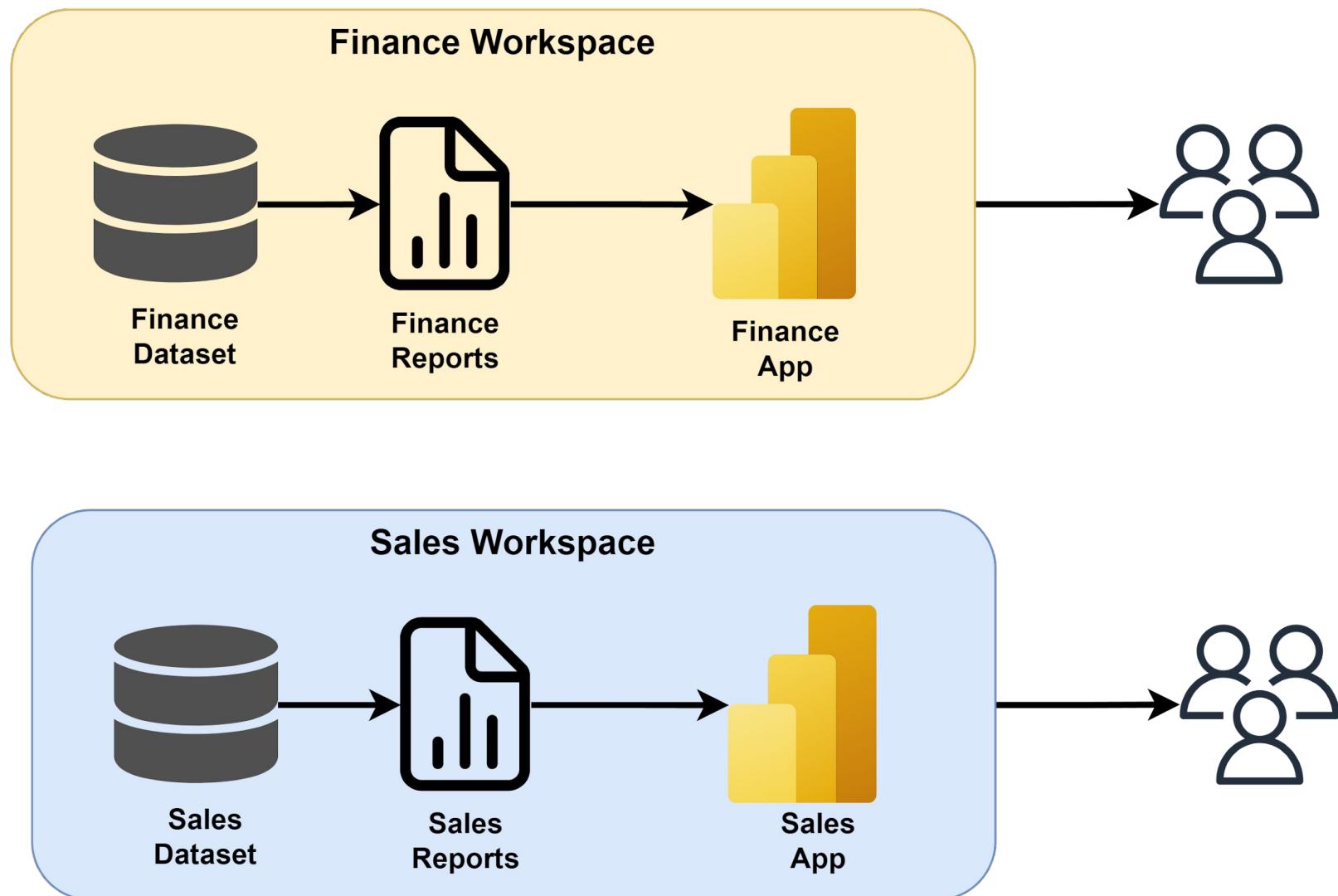


App Benefits of Report Distribution

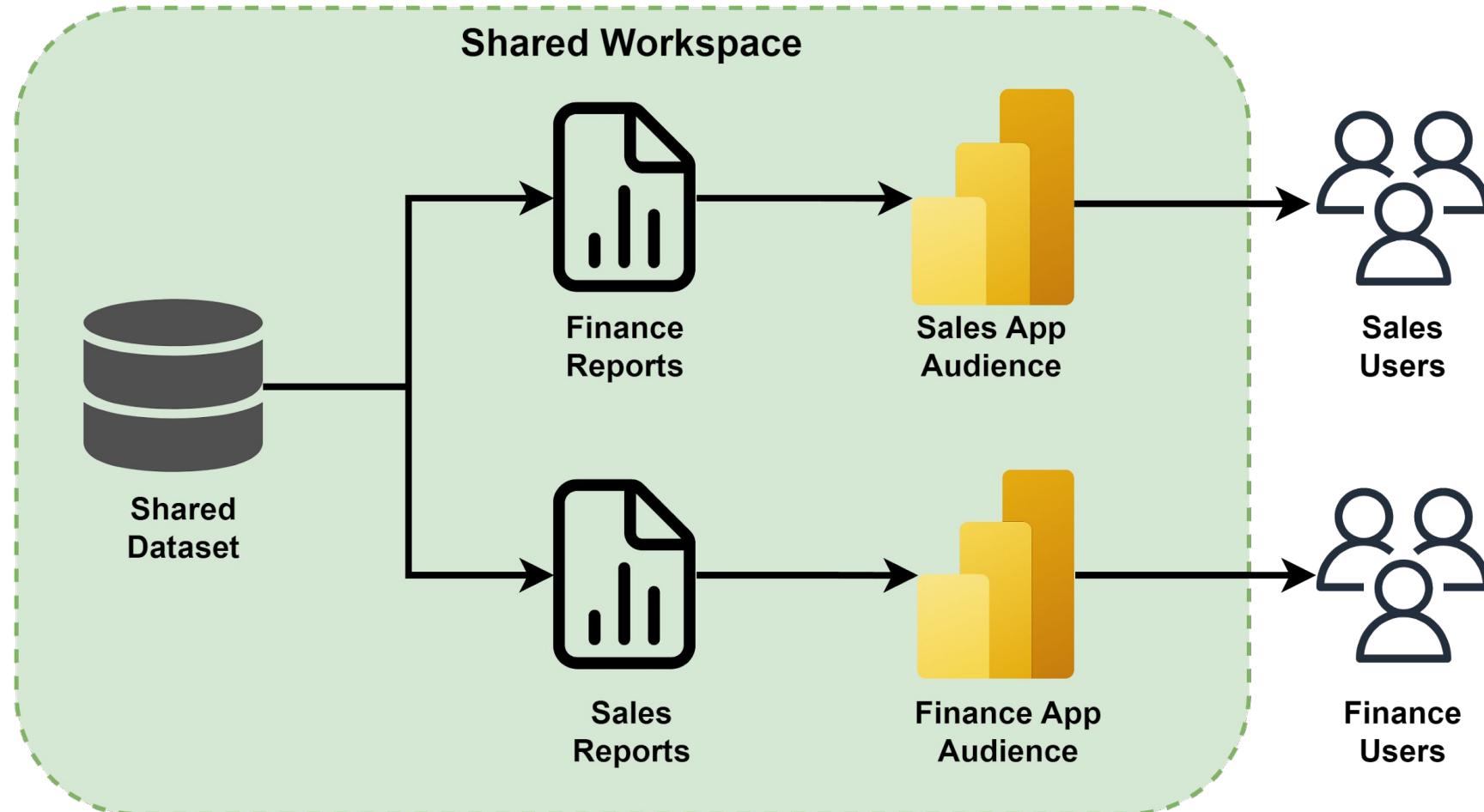
- ✓ Streamlined Access
- ✓ Enhanced Security
- ✓ Consistency
- ✓ Simplified Sharing
- ✓ Usage Metrics
- ✓ Mobile Access
- ✓ Automatic Updates
- ✓ User-Friendly Experience
- ✓ Scalability
- ✓ Version Control



Strategy #1: Separate Workspaces for Each User Group



Strategy #2: Shared Workspace for Multiple User Groups



Create and publish your App

Power BI Power BI Intermediate Training 1 ? ? ? ?

Power BI Intermediate Training

+ New + Upload + Create app Manage access Workspace settings Filter by keyword Filter Sort

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Included in app
Lab 6 -Advanced Visualization Completed	Report	Power BI Intermediate...	9/17/23, 10:56:37 PM	—			
Lab 6 -Advanced Visualization Completed	Dataset	Power BI Intermediate...	9/17/23, 10:56:37 PM	N/A			

① Setup **② Content*** **③ Audience***

Build your app

App name *
Power Cycle Report

Description *
Power Cycle operational overview
168 characters left

App logo
 Upload Delete

App theme color


Contact information
 Show app publisher
 Show items contacts from the workspace
 Show specific individuals or groups

Report Content

Power BI Apps

① Setup ② Content* ③ Audience*

Power Cycle Report

+ Add content

Add content

Insert reports, dashboards, and workbooks directly from your workspace. You can also add website links.

Add from workspace Add a link

	Name	Type	Owner	Refreshed
●	Lab 6 -Advanced Visualiza...	Report	Power BI Intermediate...	9/17/23, 10:56:37 PM
●	Lab 7 -RLS	Report	Power BI Intermediate...	8/17/23, 3:04:07 PM

① Setup ② Content ③ Audience

Power Cycle Report

+ Add content Add a link

Add a new section

POWER CYCLE

\$32.51M Sales

\$34M Sale Target

-1.49M Variance

-0.04 Variance %

Sales vs. Target

Use individual visual in PowerPoint

Embed any visual from organizational app or workspace into your PowerPoint slides.

Learn more

Running Total Sales by Date

Intermediate Power BI

DATA SOCIETY © 2020 107

Managing App Audiences

① Setup ② Content ③ Audience

Audience

Manage your audiences and their permissions. Select what content each audience can see by toggling the eye icon.

Power Cycle Rep...

New Audience

+ New Audience



Add up to 10 audience views



Power Cycle Report

Lab 6 -Advanced Visu...

Main Report

Report Details

Lab 7 -RLS



Hide reports that should not be accessible to this audience

POWER CYCLE

\$32.51M
Sales

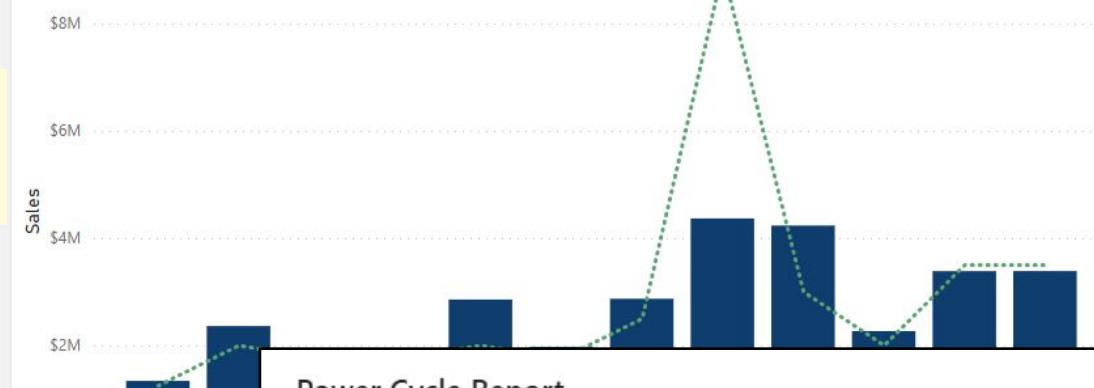
\$34M
Sale Target

-1.49M
Varience

-0.04
Varience %

Sales vs. Target

● Sales ● Sale Target



Power Cycle Report

When you publish an app that has large distribution, it might take a little while to process. Typically, the content will be available within 5-10 minutes, but it can take up to one day.

Publish

Cancel

Last 3 Months

Last 6 Months

Last 12 Months

Main Report

Report Detail

Edit Audience

Power Cycle Report

Grant access to

Entire organization [Learn more](#)

Specific users or groups

SL Shannon Lynch X

Enter a name or email address

> Advanced

Workspace users ?



Determine who needs to have access to this view

Sales by ResellerCountry	
United States	\$20M
Canada	\$6M
France	\$2M
United Kingdom	\$2M
Germany	\$1M
Australia	\$1M
	\$0M \$10M \$20M
Sales	

Running Total Sales by Date



Publish app

Cancel

Link to App

✓ Successfully published

Power Cycle Report

Give people the link below, or direct them to Apps > Get apps in the Power BI service.

<https://app.powerbi.com/Redirect?action=OpenApp&appId=ab92294c-fee3-4bfe-98f9-799429350f55&ctid=451>

Copy

Go to app Close

Power Cycle Report

Lab 6 -Advanced Visualizatio... ^

Main Report

Report Details

POWER CYCLE

Last 3 Months Last 6 Months Last 12 Months Main Report Report Details

\$32.51M Sales \$34M Sale Target -1.49M Varience -0.04 Varience %

Sales vs. Target

Sales Sale Target

Sales

\$8M

\$6M

\$4M

\$2M

\$0M

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Month Short Name

United States \$20M

Canada \$6M

France \$2M

United Kingdom \$2M

Germany \$1M

Australia \$1M

Sales

\$0M \$10M \$20M

Sales by ResellerCountry

Running Total Sales by Date

\$60M

\$40M

Jan 2019 Apr 2019 Jul 2019 Oct 2019 Date

Go back

Intermediate Power BI

Managing App users

The screenshot shows the Microsoft Data Hub interface. On the left, there's a sidebar with icons for 'data hub', 'Apps' (which is highlighted with a red box), 'Metrics', and 'Monitoring'. The main area displays an app card for 'Power Cycle Report'. The card includes a star icon, a three-dot menu, the name 'Shannon Lynch', the date '9/17/23, 11:18:52 PM', and the status 'Org app'. A dropdown menu from the three-dot menu contains 'Edit', 'Delete', and 'Manage permissions', with 'Manage permissions' also highlighted with a red box.

The screenshot shows the 'Power Cycle Report' app settings page. At the top, there's a header with a blue circular icon containing a white 'P', the app name, and a 'Manage audiences' button. Below the header, there are three buttons: '+ Add user', 'Manage audiences' (which is highlighted with a red box), 'Direct access', 'Pending requests', and 'Pending invitations'. The main section has four tabs: 'People and groups with access', 'Email Address', 'Audiences', and 'Options'. Under 'People and groups with access', there's a table with one row showing 'Workspace users' (with a blue circular icon) and 'All' under 'Audiences'.

App Endorsement

The screenshot shows the 'Power BI Intermediate Training' app settings page. At the top, there's a navigation bar with 'New', 'Upload', 'Update app', 'Manage access', 'Workspace settings', and a three-dot menu. The three-dot menu has two options: 'Endorse this app' (which is highlighted with a red box) and 'Feature this app on Home'. Below the navigation bar is a table with columns 'Name', 'Type', and 'Owner'. The 'App' tab is selected. In the main content area, there's a section titled 'Settings for Power Cycle Report' with a 'Promotion' tab selected. Under 'Endorsement', there are three options: 'None' (grayed out), 'Promoted' (selected, highlighted with a red box), and 'Certified' (grayed out). A checked checkbox labeled 'Feature on Home' is also present. At the bottom, there's a table titled 'Apps' with columns 'Name', 'Publisher', 'Published', 'App type', 'Version', and 'Endorsement'. The 'Power Cycle Report' app is listed with 'Shannon Lynch' as the publisher, '9/17/23, 11:18:52 PM' as the published date, 'Org app' as the type, and 'Promoted' as the endorsement status (highlighted with a red box).

Name	Type	Owner
General		
Dashboards		
Datasets		
Workbooks		
Dataflows		
App		

Settings for Power Cycle Report

Endorsement

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

None
The app will appear in search results but isn't endorsed.

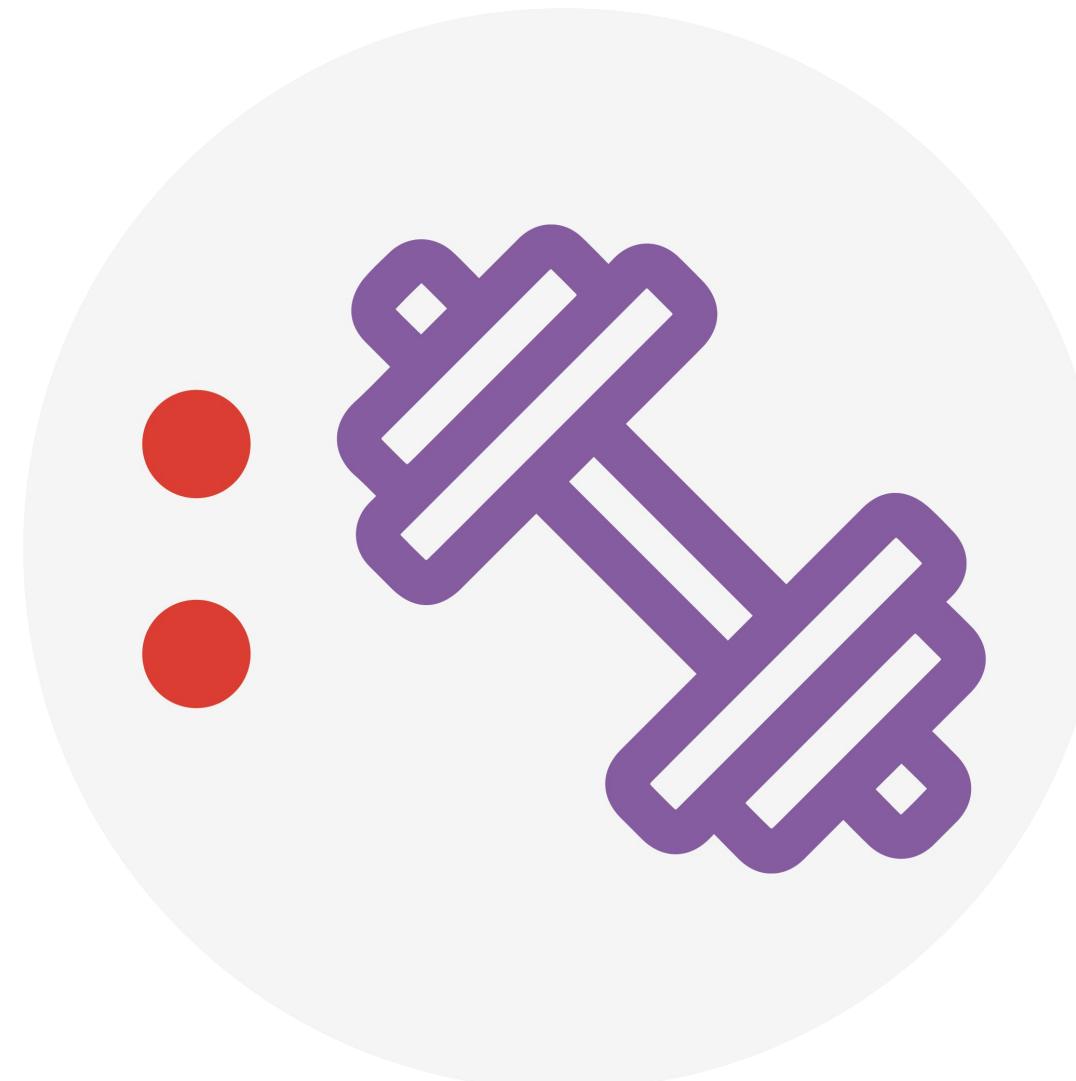
Promoted
When you're ready to distribute the app to your coworkers, promote it to let them know.

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

Feature on Home

Name	Publisher	Published	App type	Version	Endorsement ↑
Power Cycle Report	Shannon Lynch	9/17/23, 11:18:52 PM	Org app	—	Promoted

Live Demo



Knowledge Check 8



Summary

Today, we learned about:

- Advanced Report Design Principles
- Row-Level Security
- Power BI App Deployment



Power BI Support Resources

[Community.PowerBI.com](#) – Community Forum

[Report Theme Gallery](#) – A showcase for stunning report themes

[Data Stories Gallery](#) – Get inspired with Data Stories by other Power BI users

[R-Visuals Gallery](#) – Get inspired by others use of R for analyzing their data

[Store.office.com](#) – Custom PBI visuals and R visuals you can download and use in your story

[Power BI Blog](#) – weekly updates

[User Voice for Power BI](#) – Vote on (or submit) your favorite new ideas for Power BI

[Issues.PowerBI.Com](#) – log issues with the community

[Whitepaper](#) – Creating an Enterprise Class Dashboard *Solution with Power BI*

[Guided Learning Self Service Power BI training](#)

[DAX Formula Language](#) – syntax for DAX

[DAX Patterns](#) – Great website to learn new patterns for the DAX Language

[Power Query Formula Language](#) – syntax for the “Query” language

[Paletton.com](#) – a color scheme generator

<https://unicode-table.com/en/> – Unicode Character Table

[Theme Generator](#)

[Contrast Analyzer](#): a tool that creates a “lens” to show how people with different visual disabilities might see your reports

[Charticulator](#): a tool that helps to build custom visuals



Thank you!

hello@datasociety.com

1100 15th St. NW, Floor 4
Washington, D.C. 20005

(202)600-9635

datasociety.com