



# MASTERING POWER BI:

Performance Strategies for Big  
Data and Rapid Analytics





# Housekeeping

- Please silence your phones. If you need to take a call, feel free to step outside and come back in.
- Sessions are being recorded and will be available after.
- Please use this QR code to take the session survey before heading to the next session.
- Survey responses get you more entries into the raffle at the end of the day. (prizes included surface headphones, Smart Ray Bans, RayBan Meta Smart Bluetooth Glasses, and lots more).
- Wifi Info: BusinessTechnologySummit  
Password: journeyteam!



# Presenters



**NATHAN GIULLIAN**  
BI SOLUTION DEVELOPER



**JOSH JAMES**  
BI SOLUTION DEVELOPER



**MELISSA SPILLE**  
FRIEND AND PARTNER











## Have you ever...

Implemented a Business Intelligence solution without the necessary:

- Time,
- Budget, or
- Know-how







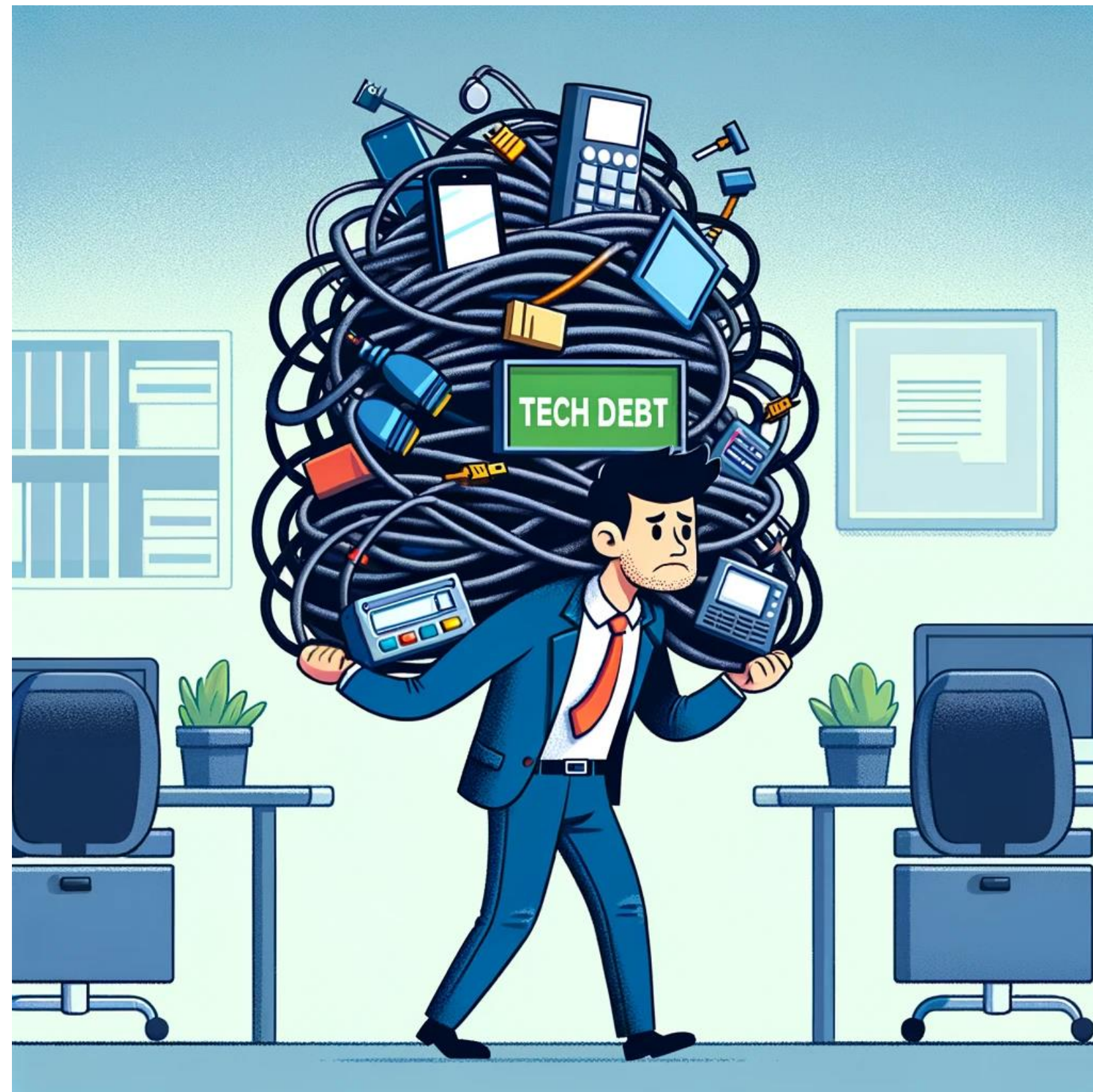








# How much time do you spend on technical debt?



BTS Mastering Power BI - Technical Debt Survey

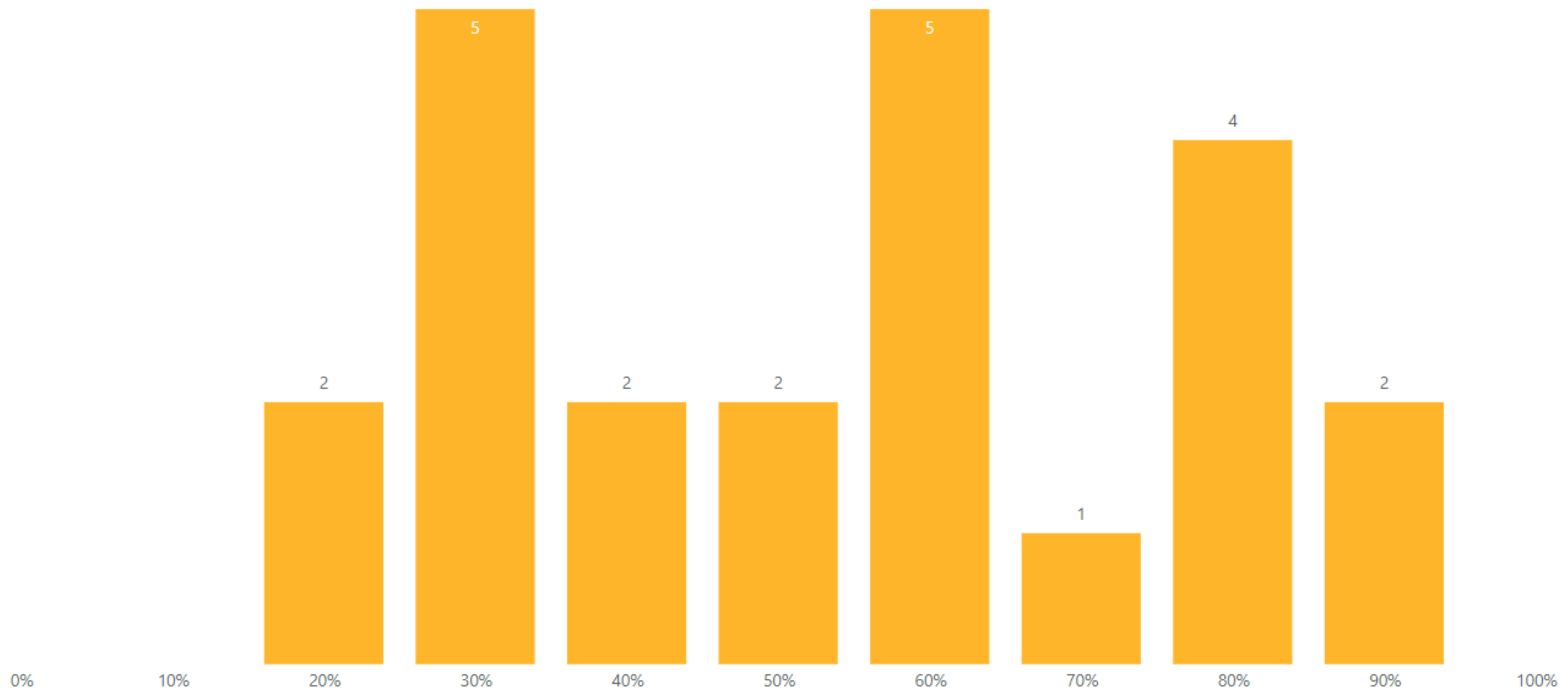


# How much time do you spend on technical debt?

**54%**  
Average Response



Filters





Multiple studies estimate the average organization wastes **23%-42%** of their development time on technical debt.

- Forbes in 2022

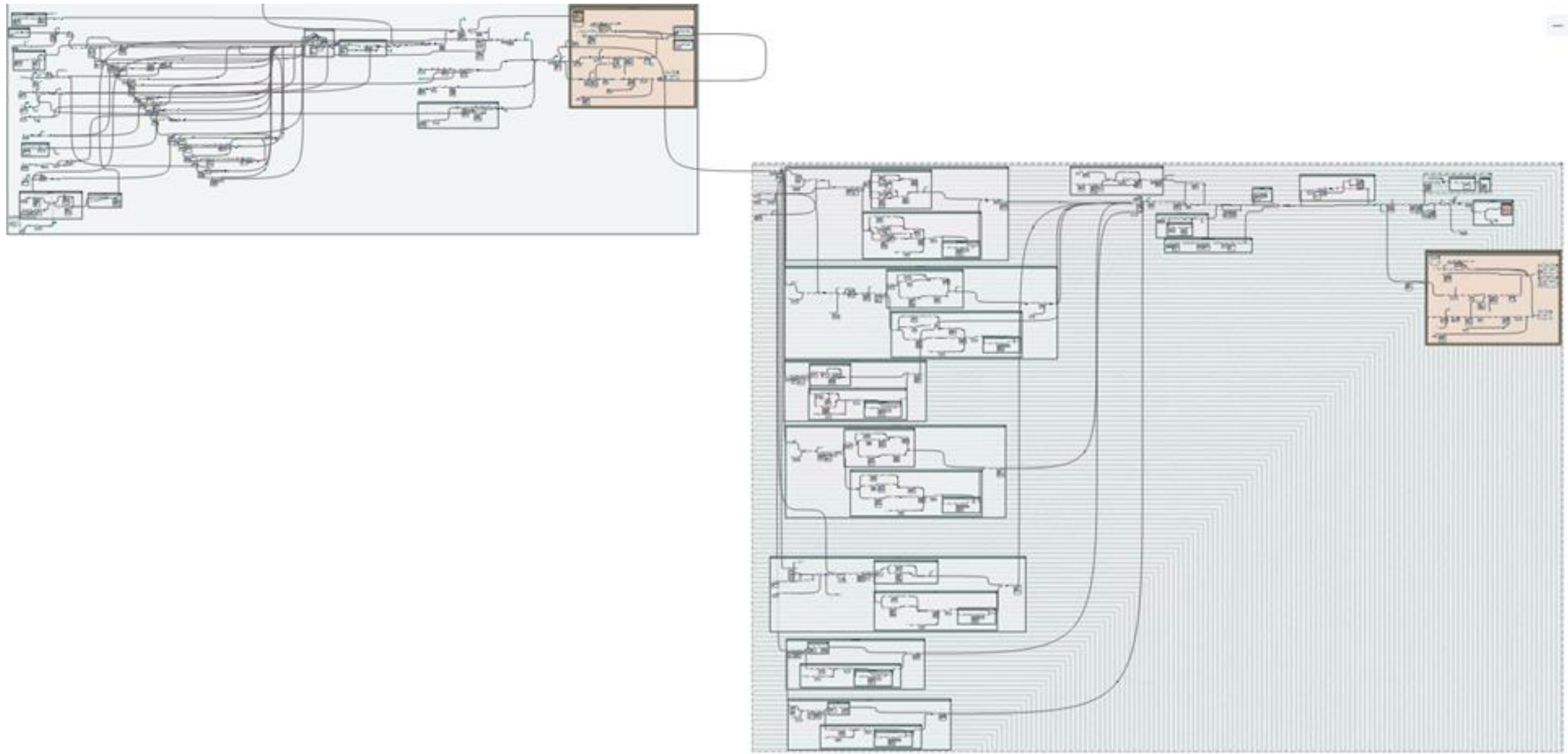
# Paying Off Technical Debt Case Study

Probbio  
MEDICAL

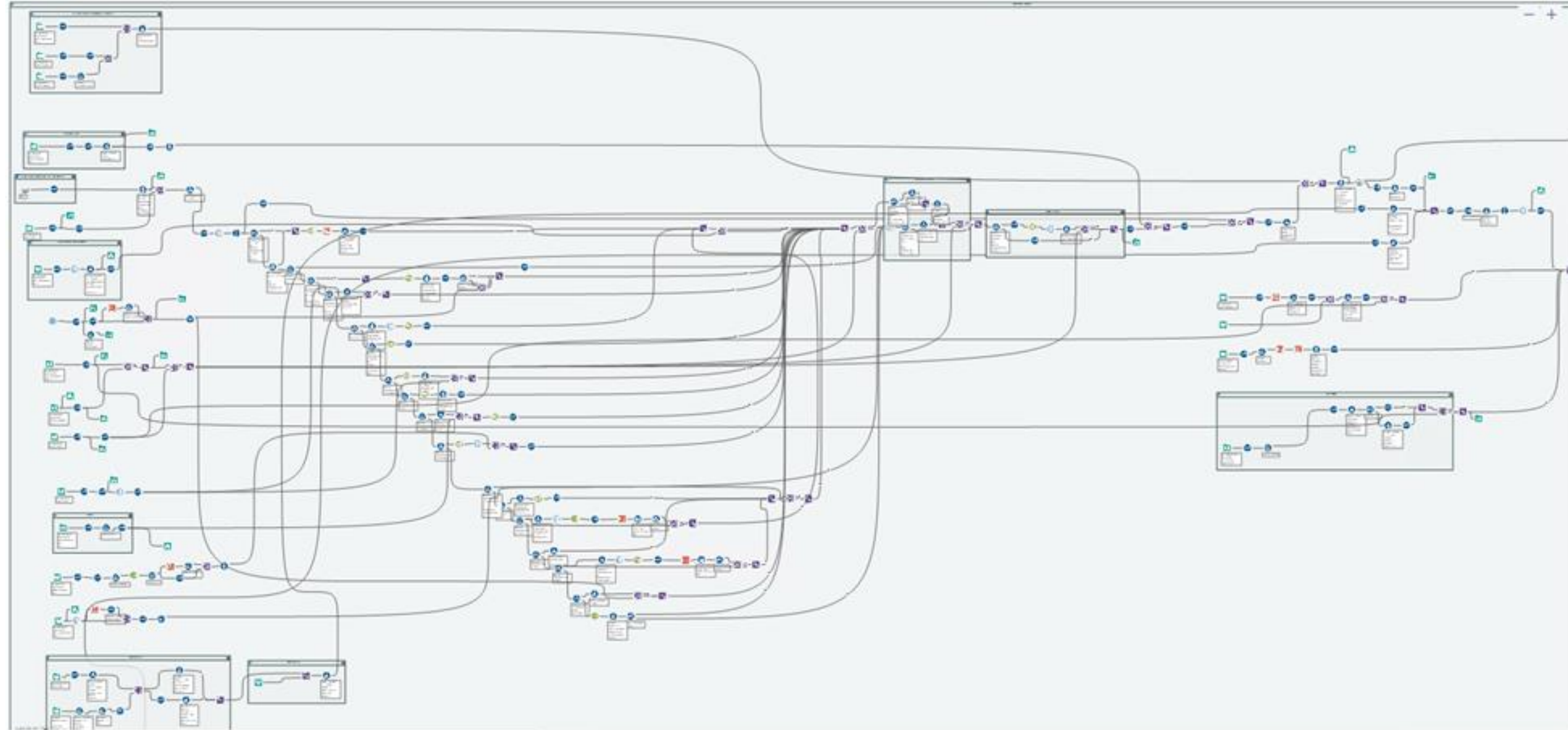


MELISSA SPILLE  
FRIEND AND PARTNER

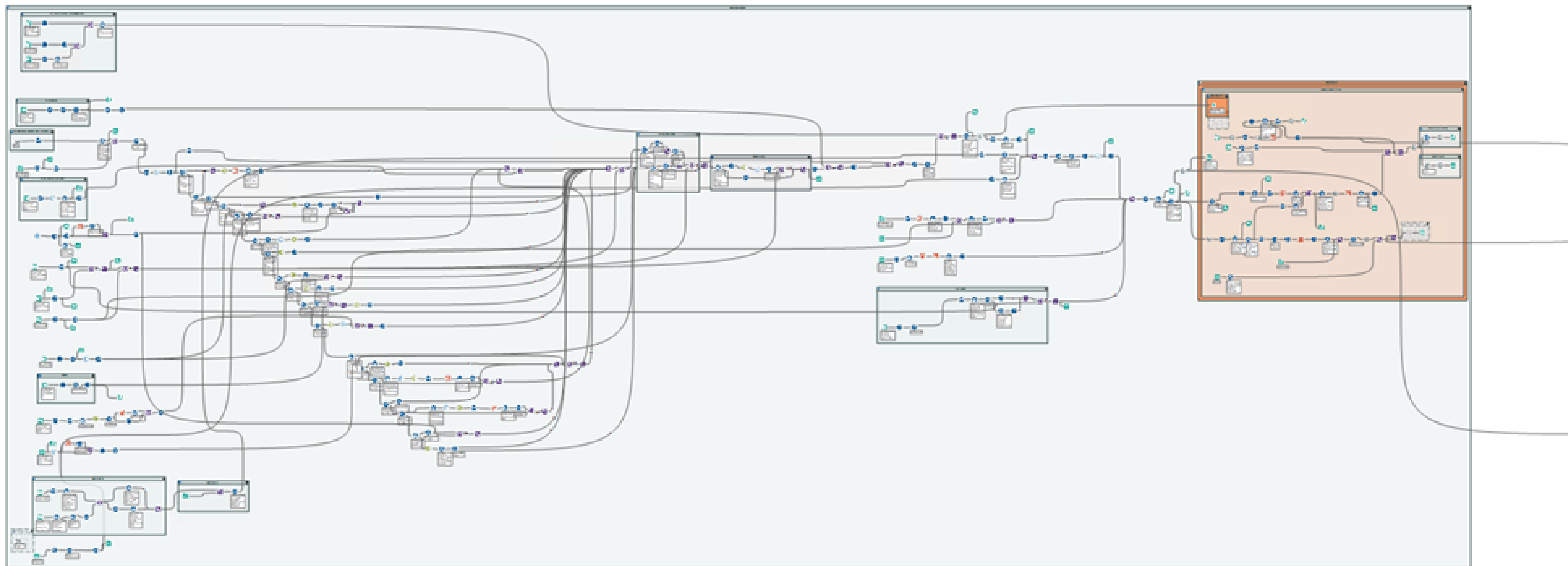
# Legacy Tech Debt Aspect: Alteryx workflow 1



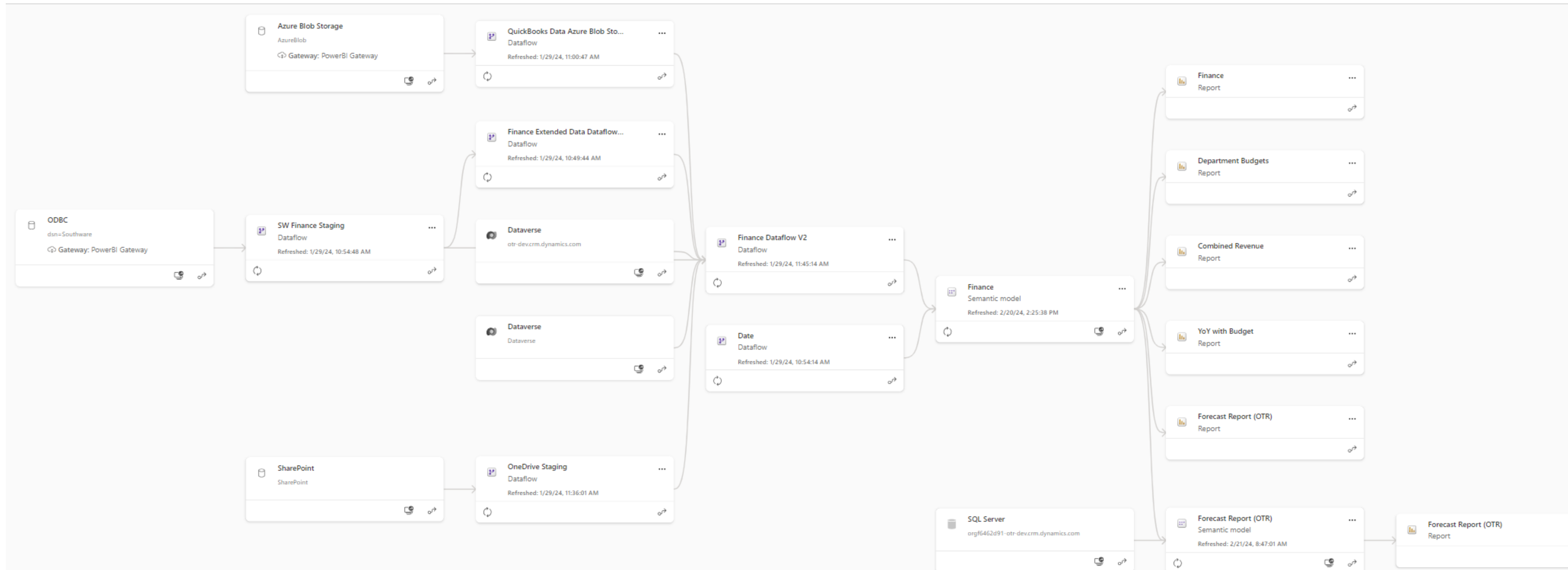
# Legacy Tech Debt Aspect: Alteryx workflow 2



# Legacy Tech Debt Aspect: Alteryx workflow 3



# Current Tech Debt Aspect: LOW



# Legacy Stats

- Only 1 person had knowledge of this setup
- If there were issues with the resulting data, "fixes" were programmed into the flows to get the "right" answer
- Refreshes failed more than they succeeded
- Refreshes took hours
- Data was not reliable, as the infrastructure could be "down" for weeks at a time
- No one trusted the numbers, which led to more discussions about the data and less discussions about business decisions

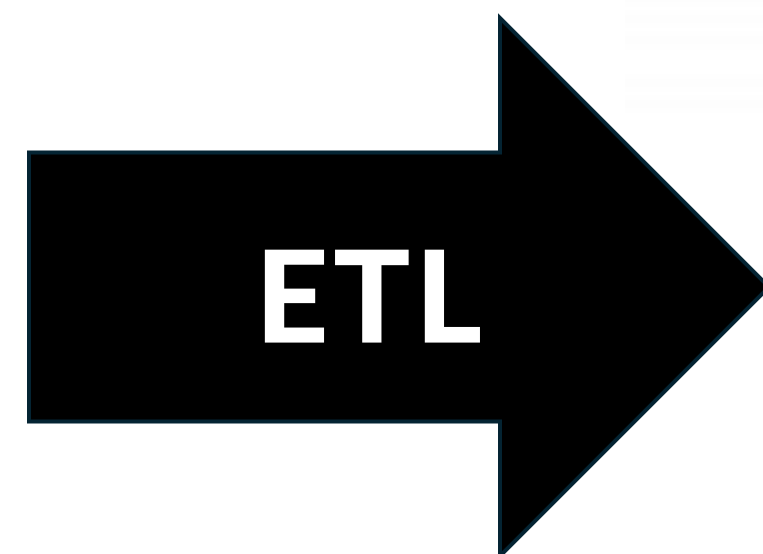
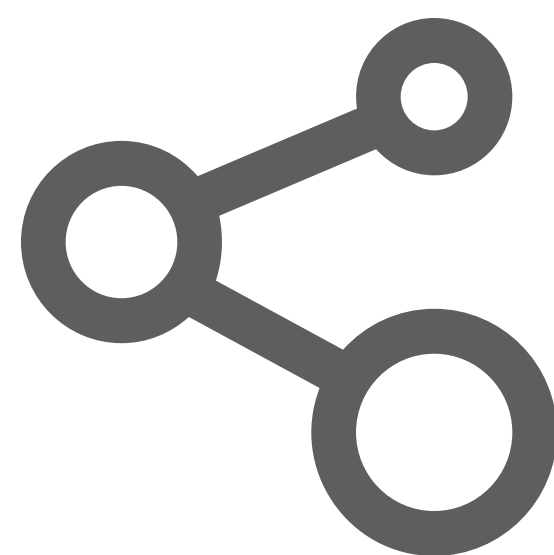
# Current Stats

- Building an in-house team, so more than 1 more person is involved in maintenance, report creation, pipeline creation
  - For our existing 5 pipelines, 1 person manages it all
  - If we didn't address our tech debt, it would have required a team of at least 4
- If data is inaccurate, the ERP is updated some reports available direct from the ERP match PBI
- Refreshes are reliable
  - The only failures we have is when a dialogue box is open on a small dataset that we are pulling in
  - Refreshes take about 15 minutes
- Data is 100% reliable and is validated quarterly
- PBI is considered truth
  - We have business-wide buy in
- This system is fully scalable
  - The infrastructure was built in such a way that we can easily build different reports without having to adjust the dataflows



# Places to Watch for Power BI Tech Debt

DATA SOURCE(S)



REPORT CONSUMER(S)



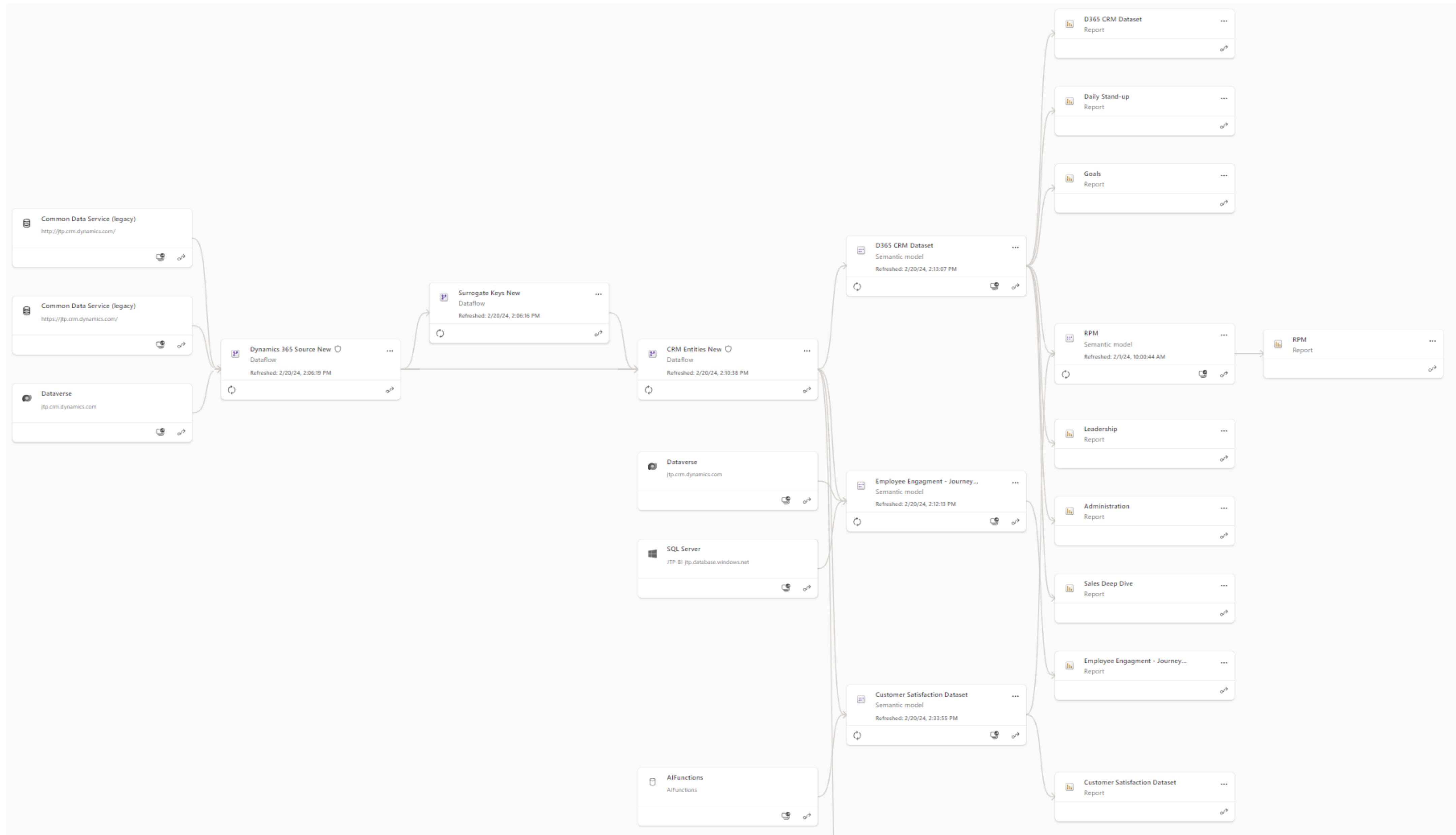
1. REPORTING
2. DEVELOPMENT
3. GOVERNANCE

# ETL Performance

1. Minimize data to process
  - Remove unnecessary data
  - Aggregate data, when appropriate
  - Use Incremental Refreshes when possible
2. Push transformations as far upstream as possible
3. Use ELT methodology rather than ETL (i.e., stage your data)



# **Staging Dataflows Demo**



# Stage with query folding – look for:

The screenshot displays a software interface with several key components:

- Data source query:** A text area containing a SQL query with various date and numeric conversions and field selections.
- Applied steps:** A vertical list of steps including 'Source', 'Navigation', and 'Choose col...'. The 'Source' step is currently selected.
- Query plan:** A diagram showing two nodes: 'Value.NativeQuery' (Remote) and 'CommonDataService...' (Remote server). An arrow points from the CommonDataService node to the Value.NativeQuery node. The Value.NativeQuery node includes a 'View details' link.
- Value.NativeQuery window:** A detailed view of the query, showing the full SQL text.

**CHECK THE DATA SOURCE QU**

**AND THE QUERY PLAN**

# Staging Data in Dataflows

1. Reduce load on the source system
2. Speed up transformations by performing them in a downstream dataflow
3. Check to make sure the staging dataflow is query folding
  - Only include steps to filter out rows and remove columns that aren't needed
  - Don't merge queries, create surrogate keys, or other more complex transformations that will break query folding

# Power BI Report Performance

## The Three Main Influences

1. Data Model Size
2. Data Model Infrastructure
3. DAX Query

# Data Model Size

**Remember: ALL data in your model take up space in your model**

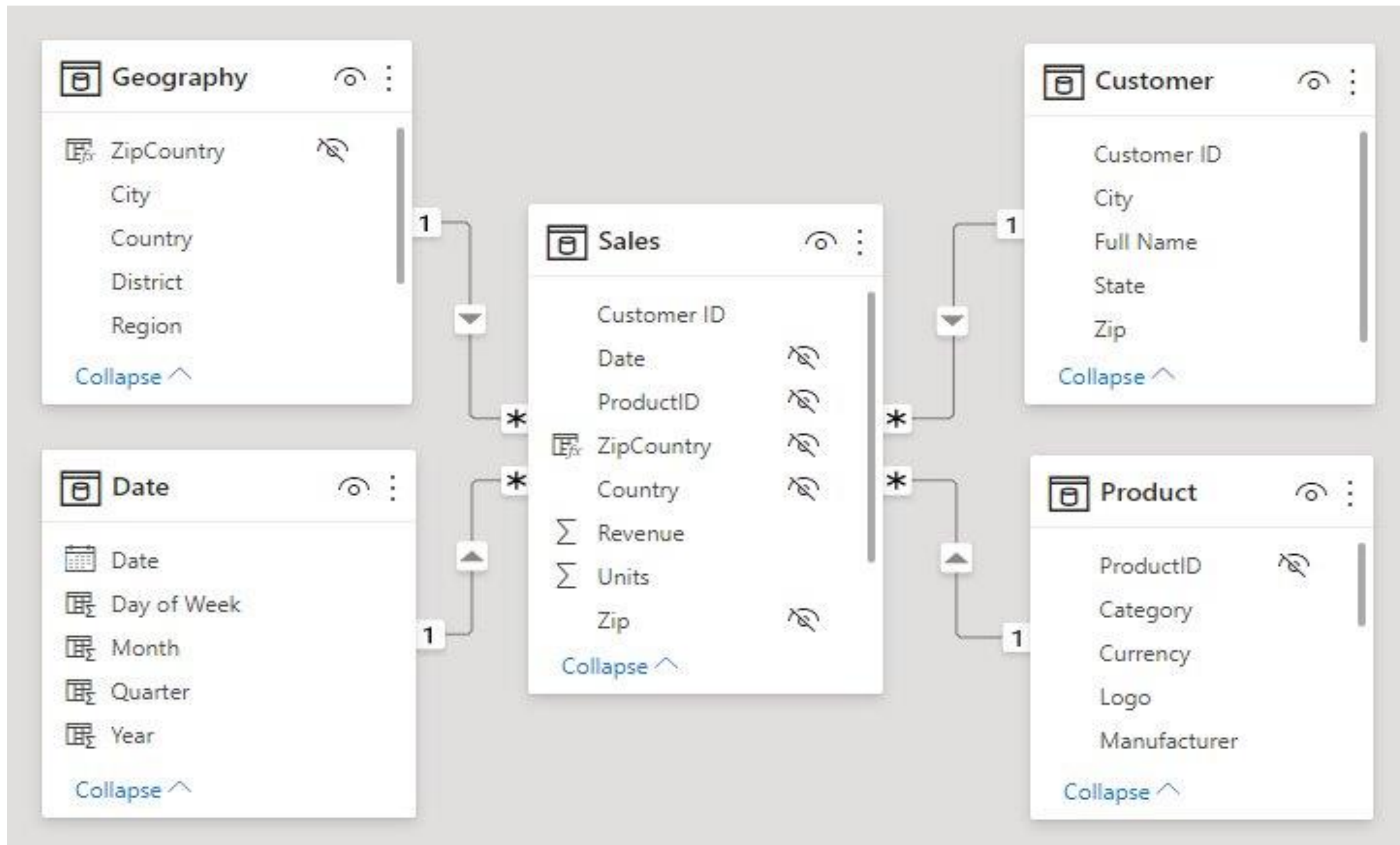
1. Remove unnecessary data (i.e., Start with the end in mind and only keep data that support that end)
2. Aggregate data to lowest granularity required (or use Aggregate Tables when different granularities are required)
3. Use the Star Schema Methodology (i.e., normalize flat tables)
4. Use Surrogate Keys
5. Use Measures instead of columns/calculated columns



# Data Model Infrastructure

**Let the Star Schema Methodology be your guide**

1. Clearly identify and parse Fact and Dimension tables
2. Use surrogate keys to connect tables
3. Active relationships are:
  - a. Between Fact and Dimension tables (never Fact-to-Fact, rarely Dimension-to-Dimension)
  - b. Rarely Many-to-Many (only used when grain requires AND only one-directional)
  - c. Rarely Bi-Directional (use measures to trigger bi-directional behavior, when needed)








# Star Schema Demo


# Create a Star Schema

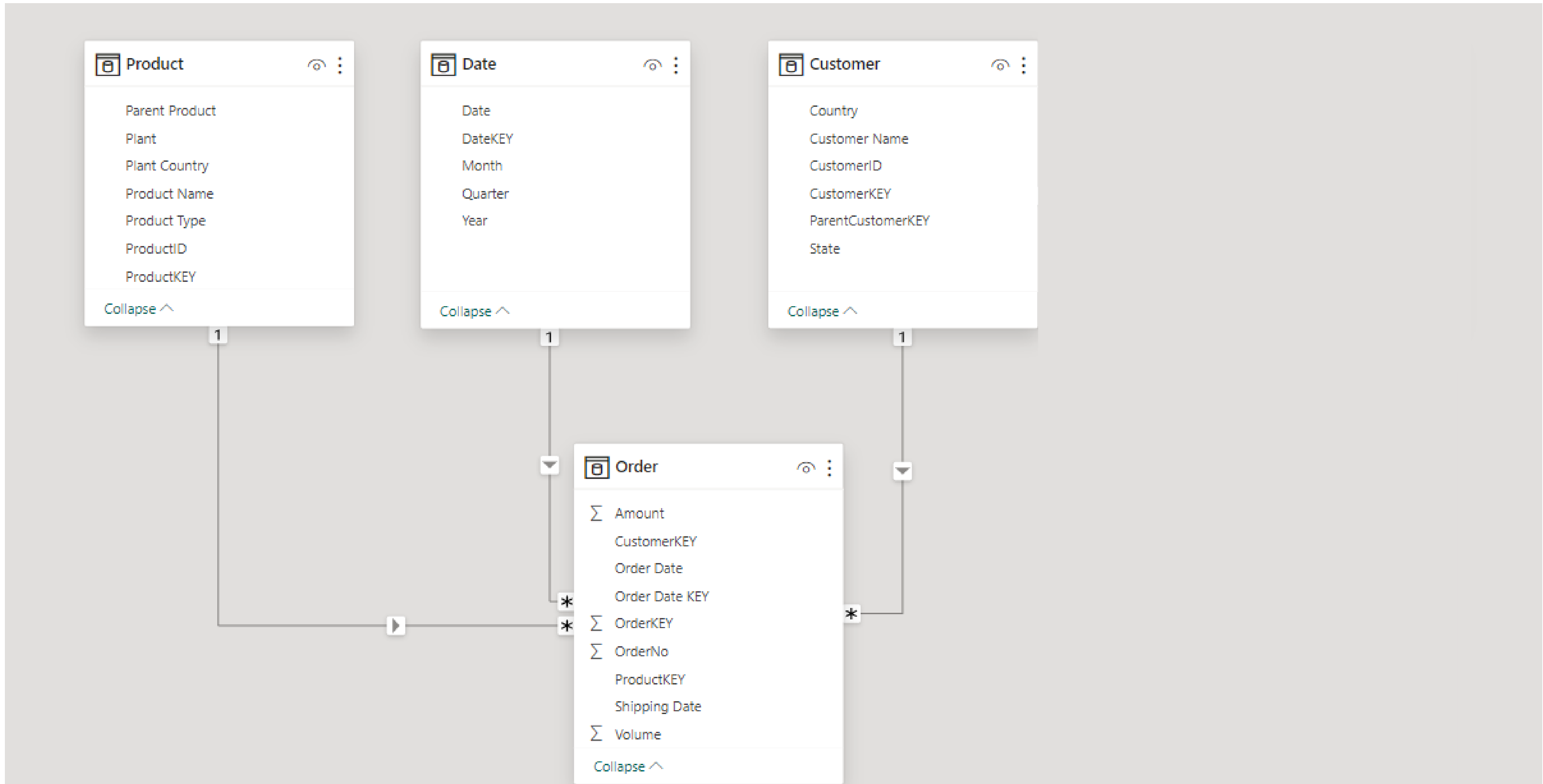
## Common Scenarios

1. Flat Tables (one table with Fact and Dimensional fields)
  - Divide Flat Table into Fact and Dimension tables (Star Schema)

 Flat Order  

- Amount
- Customer Address
- Customer Name
- Order Date
- OrderNo
- Parent Customer
- Product Name
- Product Type
- Shipping Date

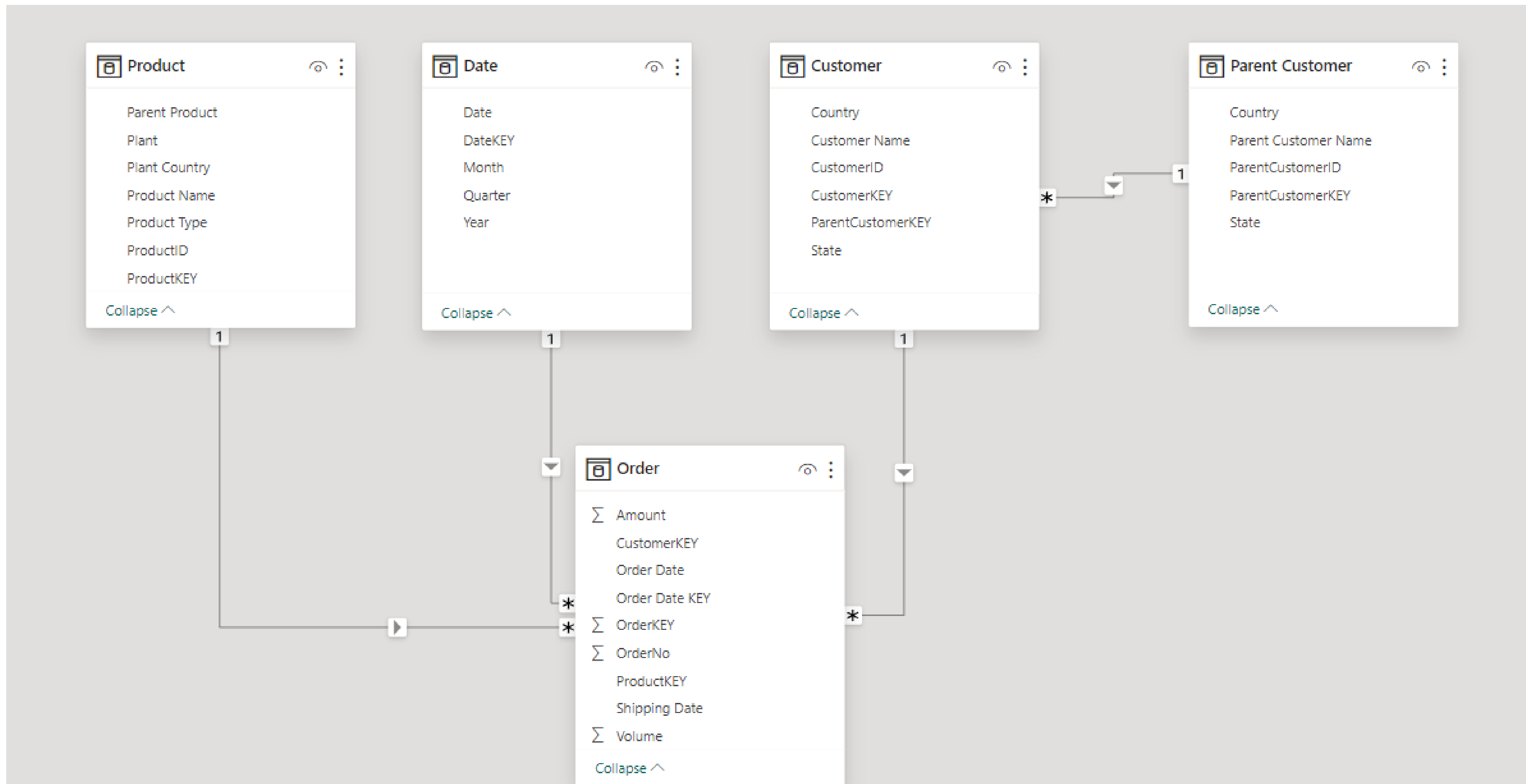
[Collapse](#) 



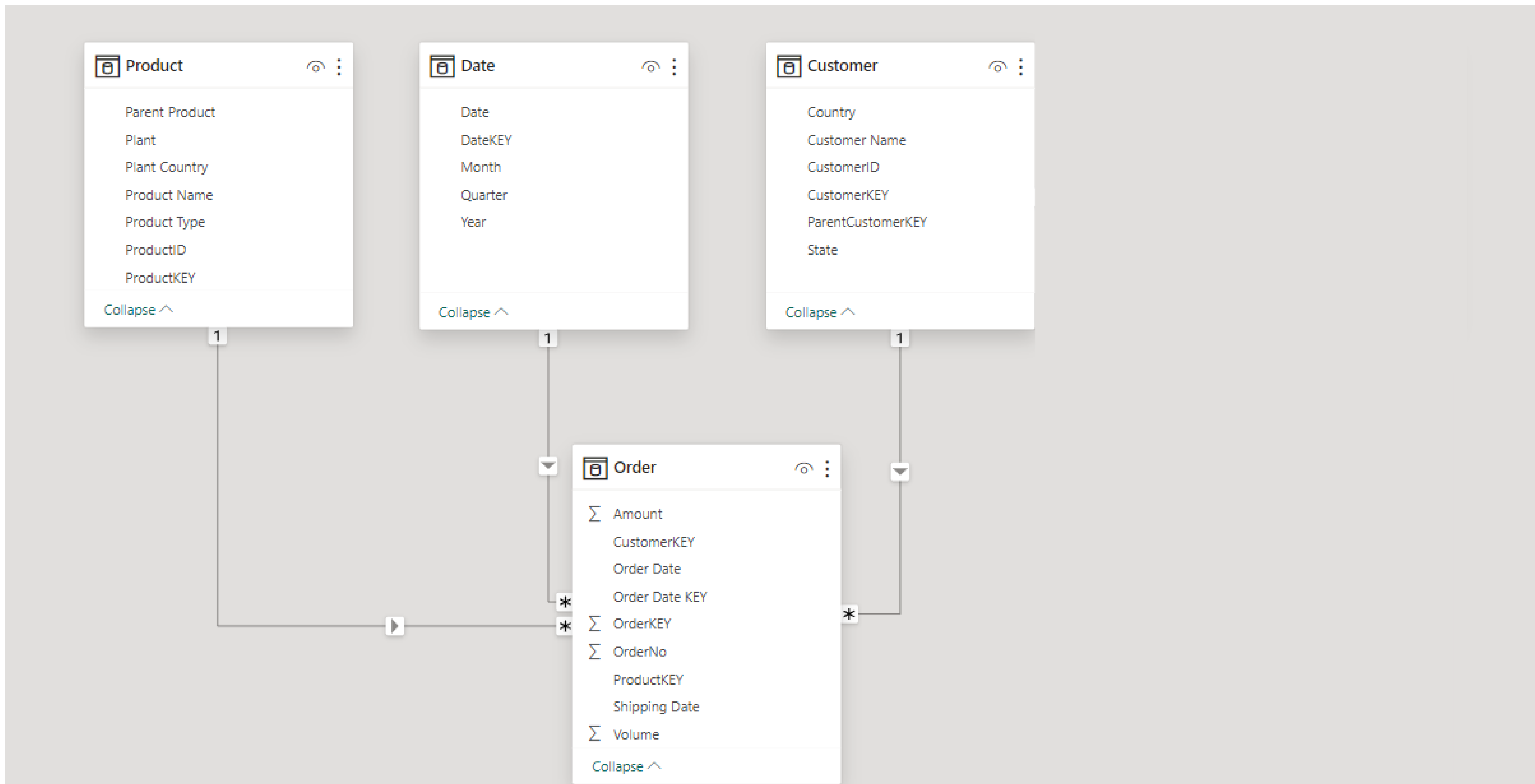
# Create a Star Schema

## Common Scenarios

1. Flat Tables (one table with Fact and Dimensional fields)
  - Divide Flat Table into Fact and Dimension tables (Star Schema)
2. Flake Tables (Dimension-to-Dimension relationships)
  - JOIN or merge into lowest grain
  - Push keys to Fact table



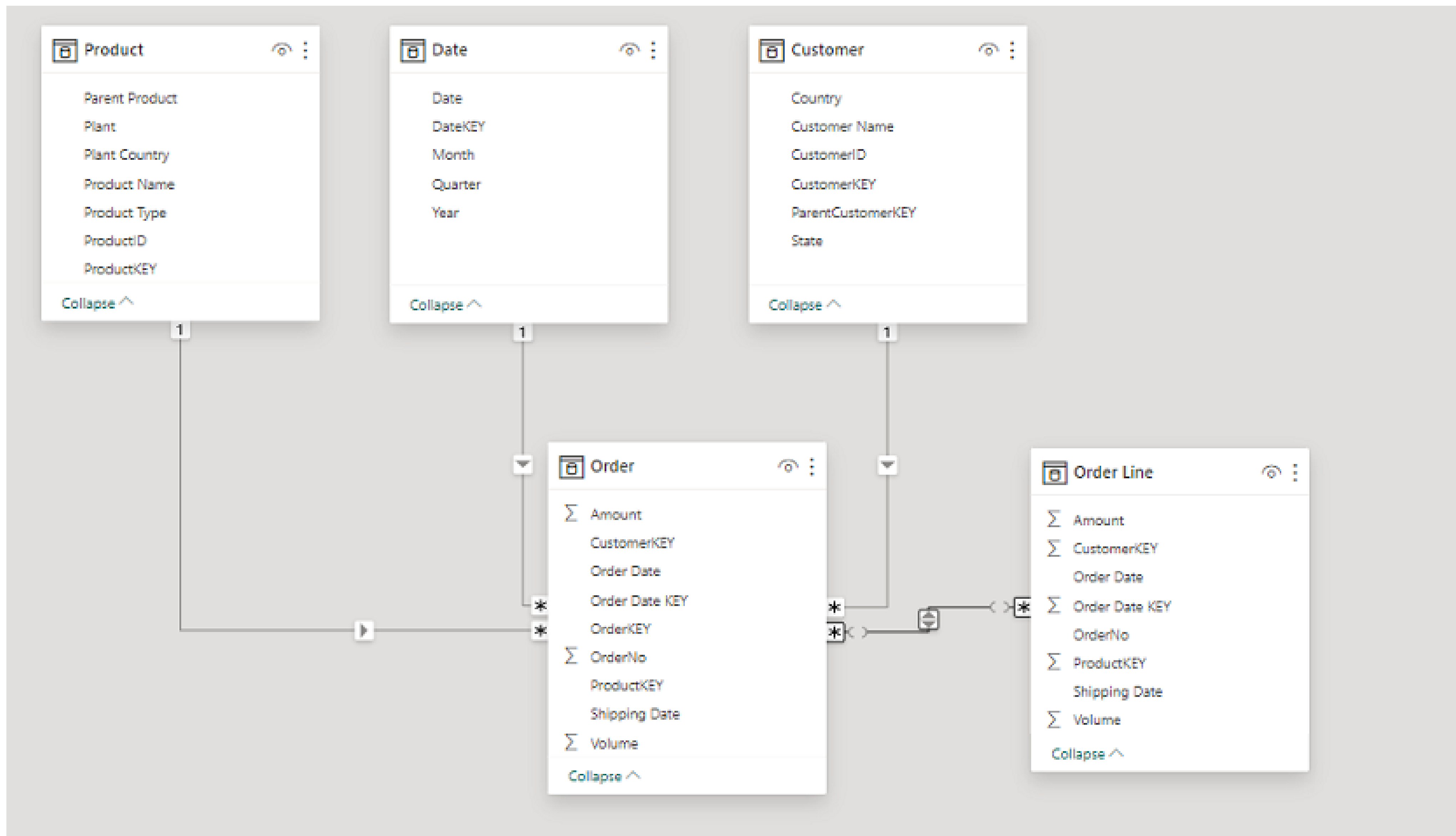


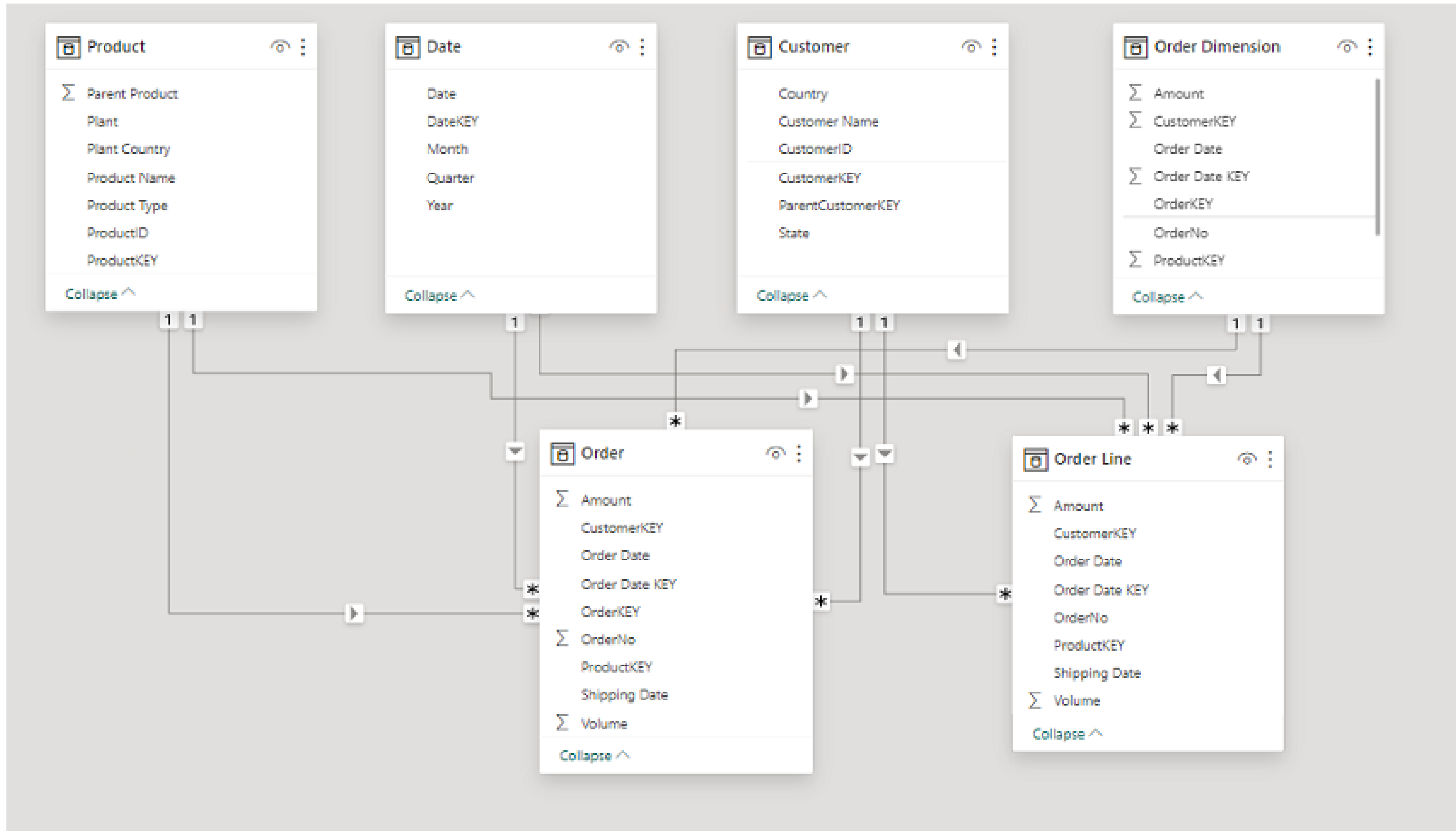


# Create a Star Schema

## Common Scenarios

1. Flat Tables (one table with Fact and Dimensional fields)
  - Divide Flat Table into Fact and Dimension tables (Star Schema)
2. Flake Tables (Dimension-to-Dimension relationships)
  - JOIN or merge into lowest grain
  - Push keys to Fact table
3. Header/Detail Fact Tables (Fact-to-Fact relationships)
  - Divide Header Fact table into Dimension and Fact tables





# DAX Query

## DAX Optimization

1. Keep DAX queries as simple as possible
  - Only use iterators when necessary (e.g., SUMX, AVERAGEX)
  - Avoid overusing variables (VAR)
  - Avoid over-nesting measures
2. When a visual takes a long time to load:
  - Use Performance Analyzer to find potential issues
  - Use DAX Studio to dive deeper into the issues



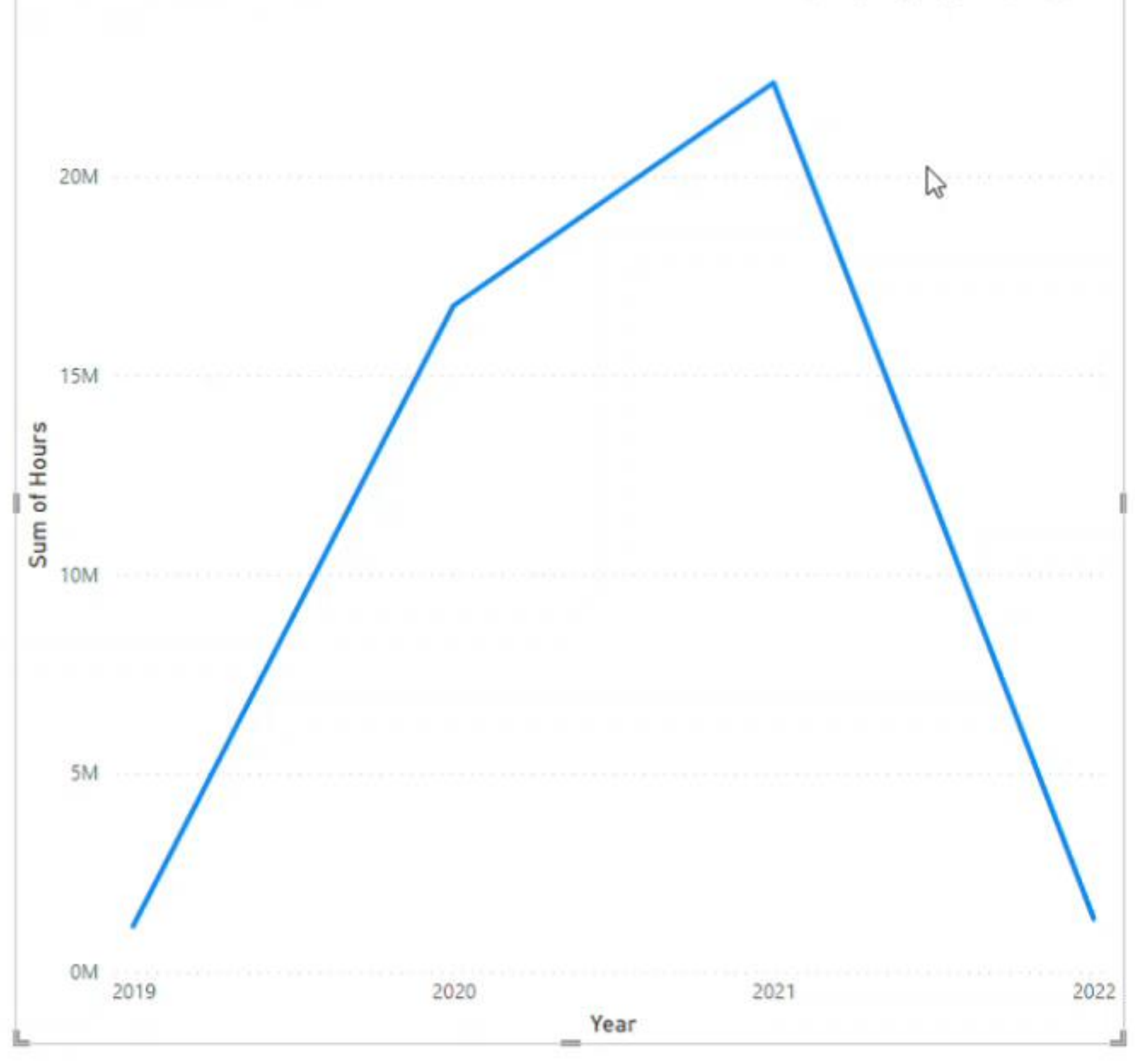
# DAX Demo

Pause Refresh Optimization Performance Apply all slicers  
visuals visuals presets analyzer button  
Queries Report Review Apply

MTD Hours by Year, MonthNumberOfYear and Date



Sum of Hours by Year



Filters

Search

Filters on this visual

- Date is (All)
- MonthNumberOfYear is (All)
- Sum of Hours is (All)
- Year is (All)

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Visualizations

Build visual

X-axis

- Year
- MonthNumberOfYear
- Date

Y-axis

- Sum of Hours

Secondary y-axis

Add data fields here

Legend

Add data fields here

Small multiples

Add data fields here

Tooltips

Add data fields here

Drill through

Cross-report

Keep all filters

Data

Search

- Measures Table
- Calendar SQL DQ
- Resources SQL DQ
- Time Entries SQL DQ

# Power BI Development Speed

1. Use Surrogate Keys for relationships
2. Avoid Calculated Columns/Tables
3. Organize and name your data in helpful ways
  - Use business nomenclature for tables and columns
  - Add notes to the measure code, and field or table descriptions whenever you think confusion could arise
  - Don't bring in data that won't be used
  - Hide keys or other fields that are required but could confuse users



# Power BI Governance

1. Use Azure (Entra ID) Security Groups
2. Build out a sensible tenant and workspace architecture
3. Minimize number of Semantic Models
4. Use Power BI Sentinel

# Power BI Sentinel Demo



# Products and Offerings



- Contact us today to get a Power BI Health Check by requesting it on our survey
- Sign up for a Power BI Sherpa so you can build Power BI the right way with expert guidance.



# Q&A



**We love feedback!**  
Please complete the session  
survey for **an extra giveaway  
raffle ticket!**





# Thank You!

**Let's Connect:**

**Nathan Giullian**  
**Josh James**

[info@journeyteam.com](mailto:info@journeyteam.com)

[www.journeyteam.com](http://www.journeyteam.com)

