Exploring Fabric Semantic Link for Power BI Folks

Data-Marc.com

Big thank you to our great sponsors

data on
Power your data

itm8®

twoday

backstage

CatMan® Solution

INSPARI
a valantic company

Viz.fo

Learning objectives

Fabric	Semantic Link	Query	Document
Understand what Microsoft Fabric is, how this relates to Semantic Link and where it is positioned.	Know exactly what Semantic Link is, how you can use it in your benefit to power your solutions.	Be able to query data and meta data of your semantic model using Semantic Link.	Document your Semantic Models by taking advantage of Semantic Link.

This session is…

✓

- An introduction to Fabric Semantic Link
- Perfectly fitted if you never wrote any line of Python code before

It is not…

✗

- A deep dive on anything Python, Fabric or Notebooks
- Best practices on how you should build data platforms

Expected level:

- Power BI Folks: 300 (advanced but doable 😳)
- Data Engineers: 100 (easy peasy lemon squeezy 🍋)

Marc Lelijveld

Technical Evangelist | Solution Architect
Macaw Netherlands

MVP Microsoft® Most Valuable Professional

sessionize MOST ACTIVE SPEAKER 2022

sessionize MOST ACTIVE SPEAKER 2023

@MarcLelijveld

linkedin.com/in/MarcLelijveld

Data-Marc.com

DutchFabricUsergroup.com

FAVORITE STUFF:

Agenda for today

Introduction to Fabric

What are notebooks?

Understanding Semantic Link

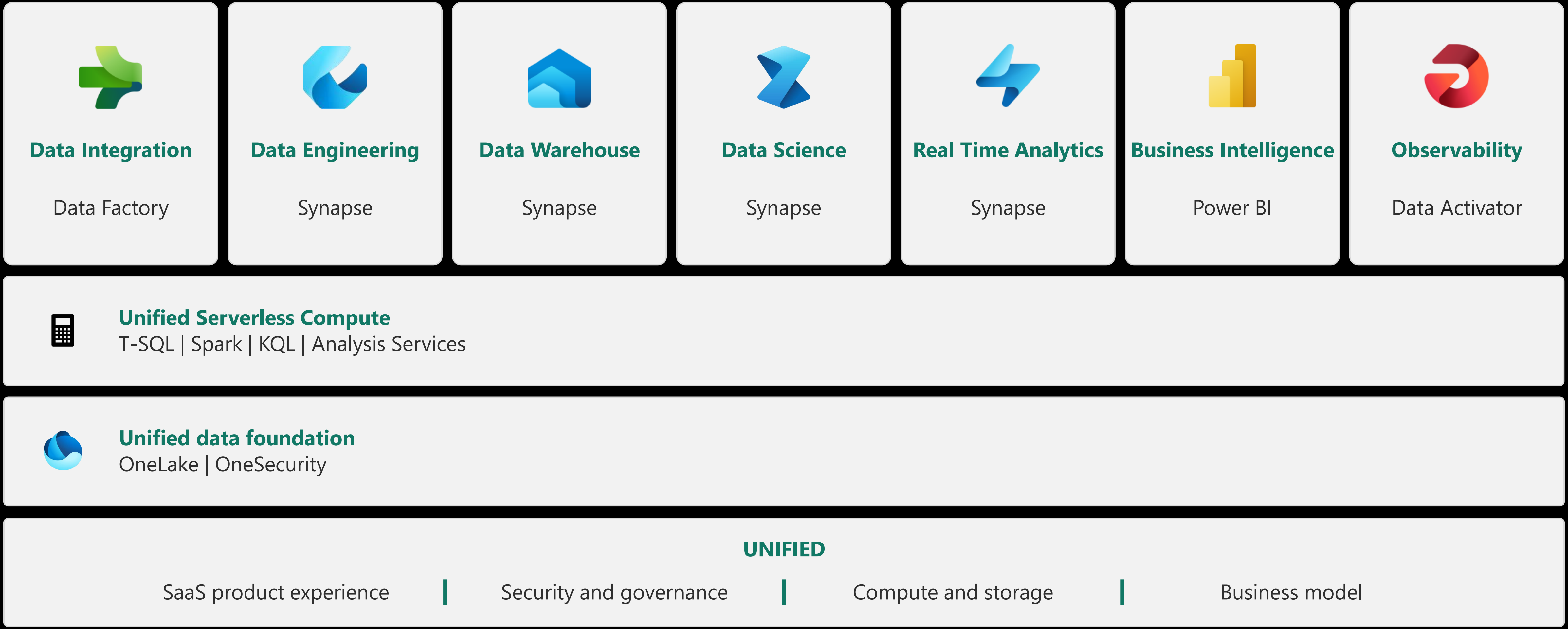
Use cases

Document your Semantic Model

Considerations

Introduction to Fabric

Unified Data Analytics platform.



Data Integration

Data Factory

Data Engineering

Synapse

Data Warehouse

Synapse

Data Science

Synapse

Real Time Analytics

Synapse

Business Intelligence

Power BI

Observability

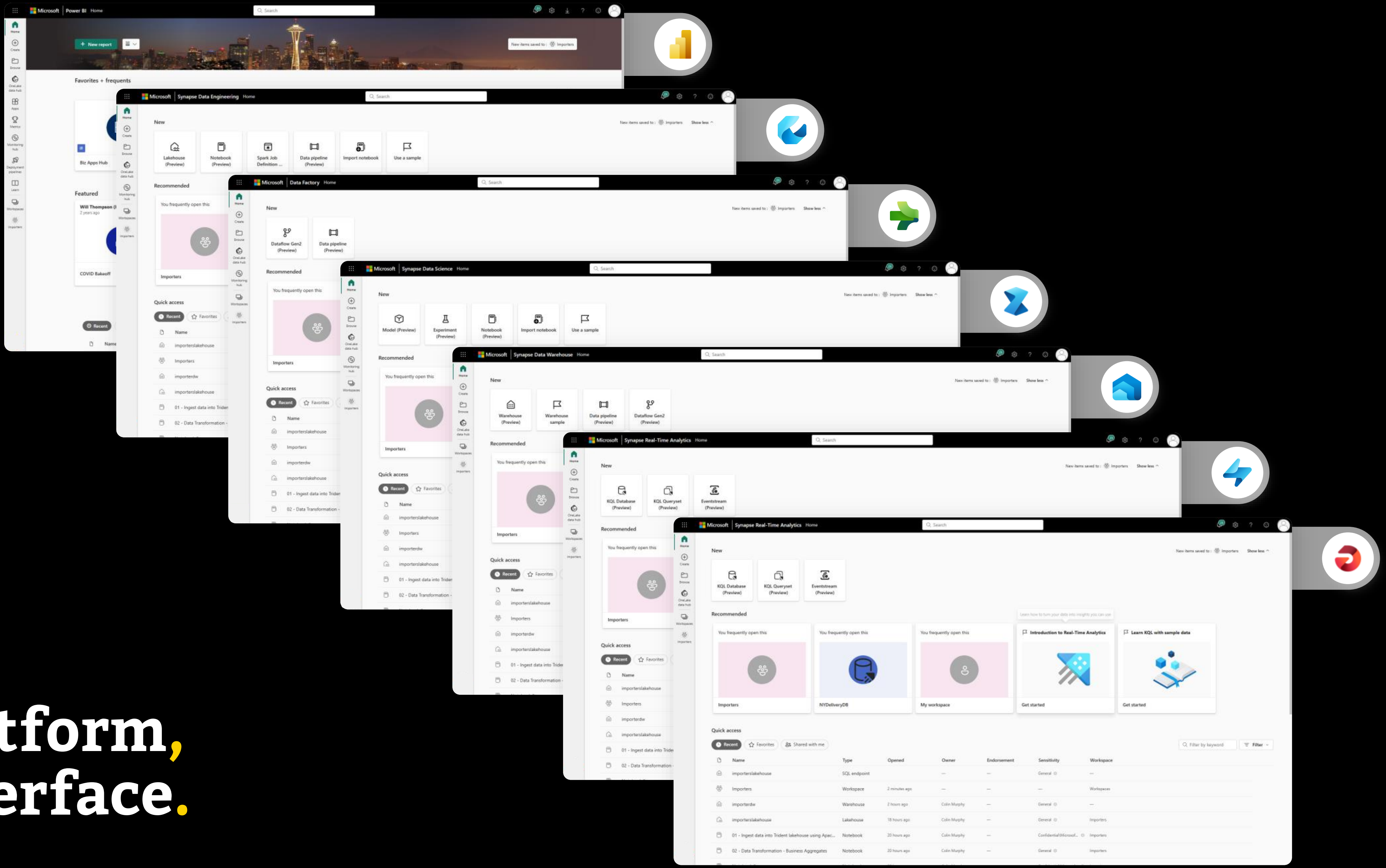
Data Activator

Unified Serverless Compute
T-SQL | Spark | KQL | Analysis Services

Unified data foundation
OneLake | OneSecurity

UNIFIED

SaaS product experience | Security and governance | Compute and storage | Business model


Microsoft | Power BI Home
Microsoft | Synapse Data Engineering Home
Microsoft | Data Factory Home
Microsoft | Synapse Data Science Home
Microsoft | Synapse Data Warehouse Home
Microsoft | Synapse Real-Time Analytics Home

Microsoft | Synapse Real-Time Analytics Home

Search

New

New items saved to : Importers Show less ^

KQL Database (Preview)

KQL Queryset (Preview)

Eventstream (Preview)

Recommended

Learn how to turn your data into insights you can use

You frequently open this Importers

You frequently open this NYDeliveryDB

You frequently open this My workspace

Introduction to Real-Time Analytics Get started

Learn KQL with sample data Get started

Quick access

Recent | Favorites | Shared with me

Filter by keyword | Filter

Name	Type	Opened	Owner	Endorsement	Sensitivity	Workspace
importerslakehouse	SQL endpoint		—	—	General	—
Importers	Workspace	2 minutes ago	—	—	—	Workspaces
importerdw	Warehouse	2 hours ago	Colin Murphy	—	General	—
importerslakehouse	Lakehouse	18 hours ago	Colin Murphy	—	General	Importers
01 - Ingest data into Trident lakehouse using Apac...	Notebook	20 hours ago	Colin Murphy	—	Confidential\Microsof...	Importers
02 - Data Transformation - Business Aggregates	Notebook	20 hours ago	Colin Murphy	—	General	Importers

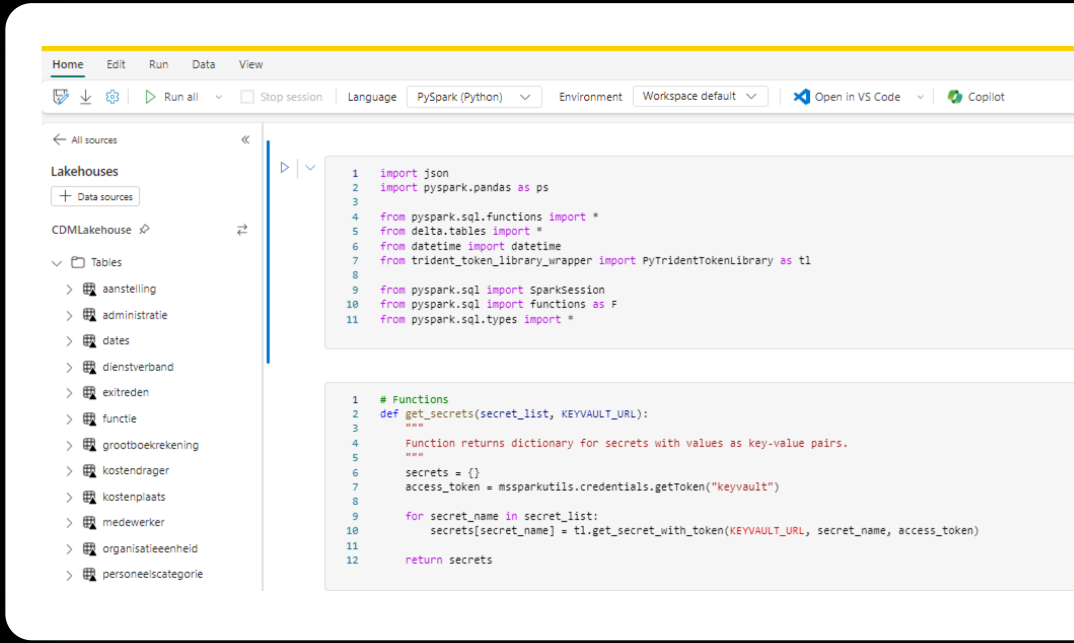
One platform,
One interface.

What are notebooks?

What are notebooks?

- Code first
- Web-based interface
- Cell based code blocks
- Runs on nodes (part of Fabric capacity)
- Often used languages are Python, Spark & Markdown

- Used by data engineers for data ingest, prep and transformations
- Used by data scientist for experiments and models



```python
import json
import pyspark.pandas as ps

from pyspark.sql.functions import *
from delta.tables import *
from datetime import datetime
from trident_token_library_wrapper import PyTridentTokenLibrary as tl

from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql.types import *
```

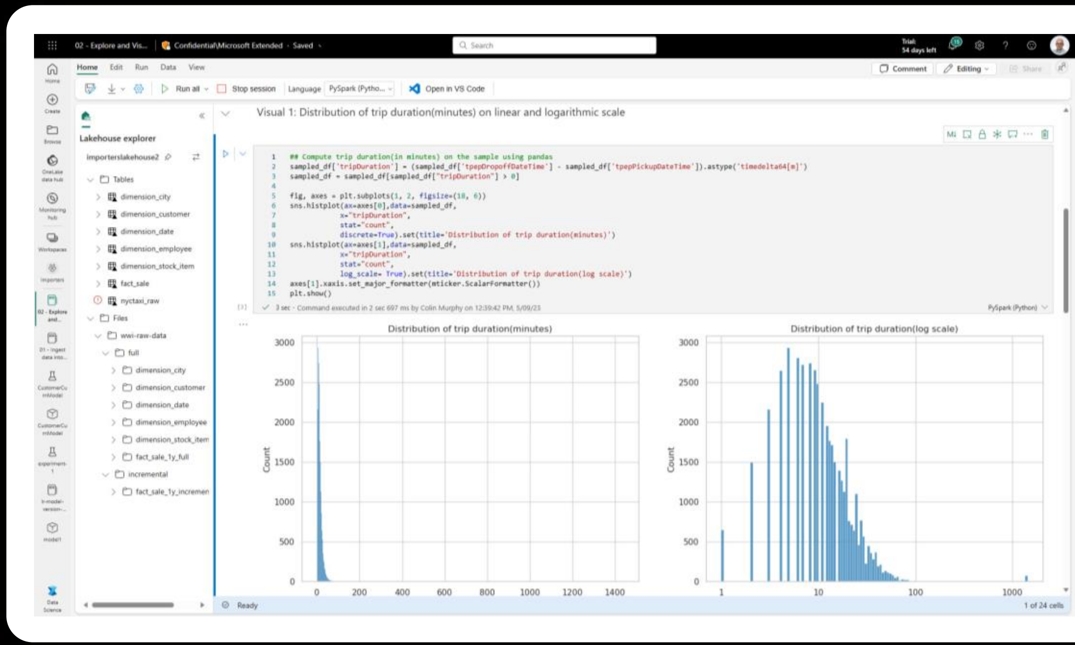
```python
# Functions
def get_secrets(secret_list, KEYVAULT_URL):
    """
    Function returns dictionary for secrets with values as key-value pairs.
    """
    secrets = {}
    access_token = mssparkutils.credentials.getToken("keyvault")

    for secret_name in secret_list:
        secrets[secret_name] = tl.get_secret_with_token(KEYVAULT_URL, secret_name, access_token)

    return secrets
```

Notebook overview

- Manage your Python and R libraries through in-line installs using commands like %pip install
- Advanced notebook development support with ability to reference notebooks in notebooks, and snapshots for easy troubleshooting
- In context monitoring complete with real time advice and error analysis
- Streamline data prep without giving up the power of reproducibility of Python

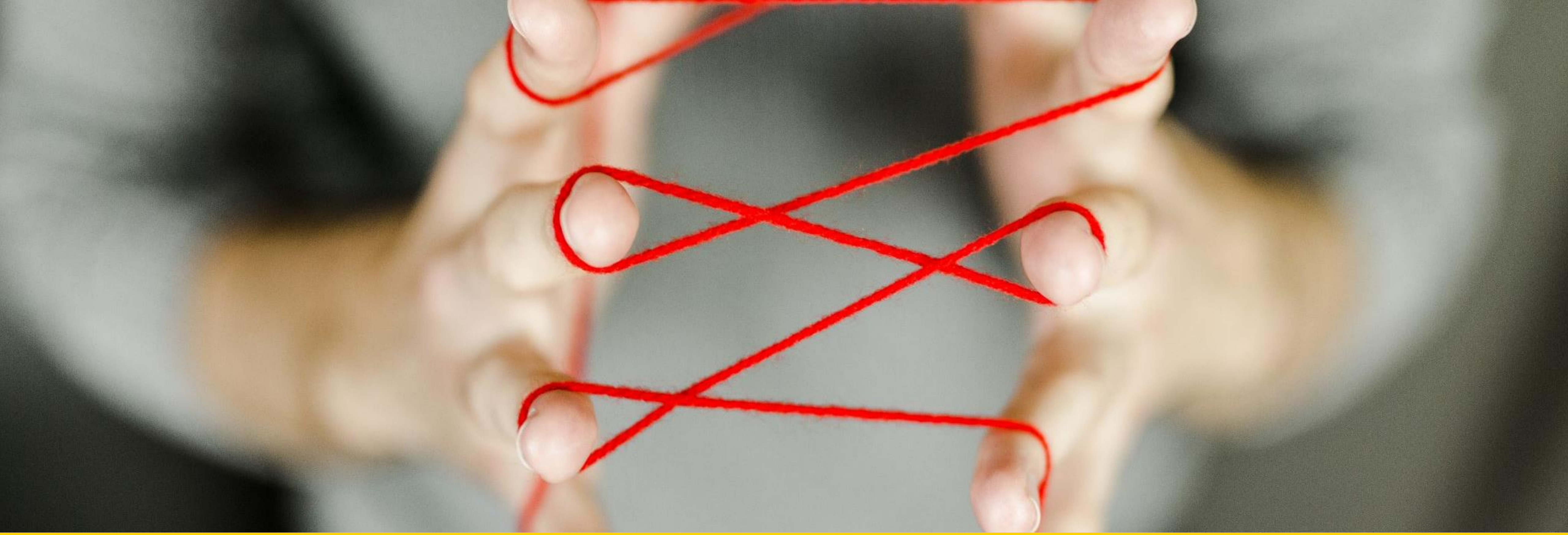


Visual 1: Distribution of trip duration(minutes) on linear and logarithmic scale

```
## Compute trip duration(in minutes) on the sample using pandas
sampled_df['tripDuration'] = (sampled_df['tpepDropoffDateTime'] - sampled_df['tpepPickupDateTime']).astype('timedelta64[m]')
sampled_df = sampled_df[sampled_df["tripDuration"] > 0]

fig, axes = plt.subplots(1, 2, figsize=(18, 6))
sns.histplot(ax=axes[0],data=sampled_df,
             x="tripDuration",
             stat="count",
             discrete=True).set(title='Distribution of trip duration(minutes)')
sns.histplot(ax=axes[1],data=sampled_df,
             x="tripDuration",
             stat="count",
             log_scale= True).set(title='Distribution of trip duration(log scale)')
axes[1].xaxis.set_major_formatter(mticker.ScalarFormatter())
plt.show()
```

[3] ✓ 3 sec - Command executed in 2 sec 697 ms by Colin Murphy on 12:39:42 PM, 5/09/23

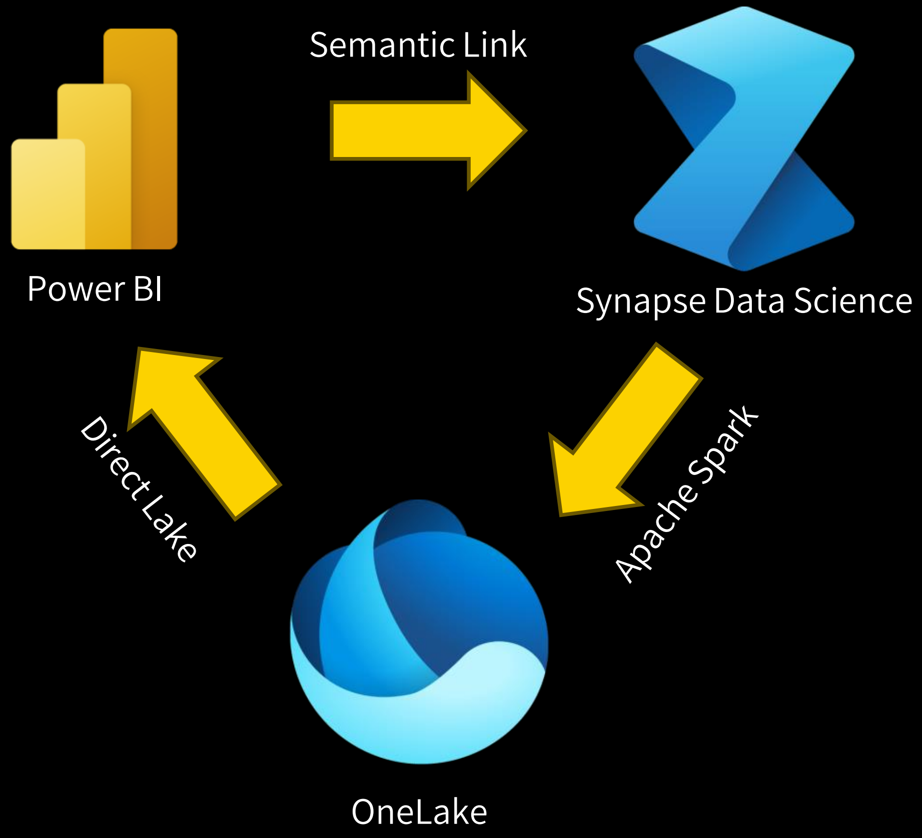


Understanding Semantic Link

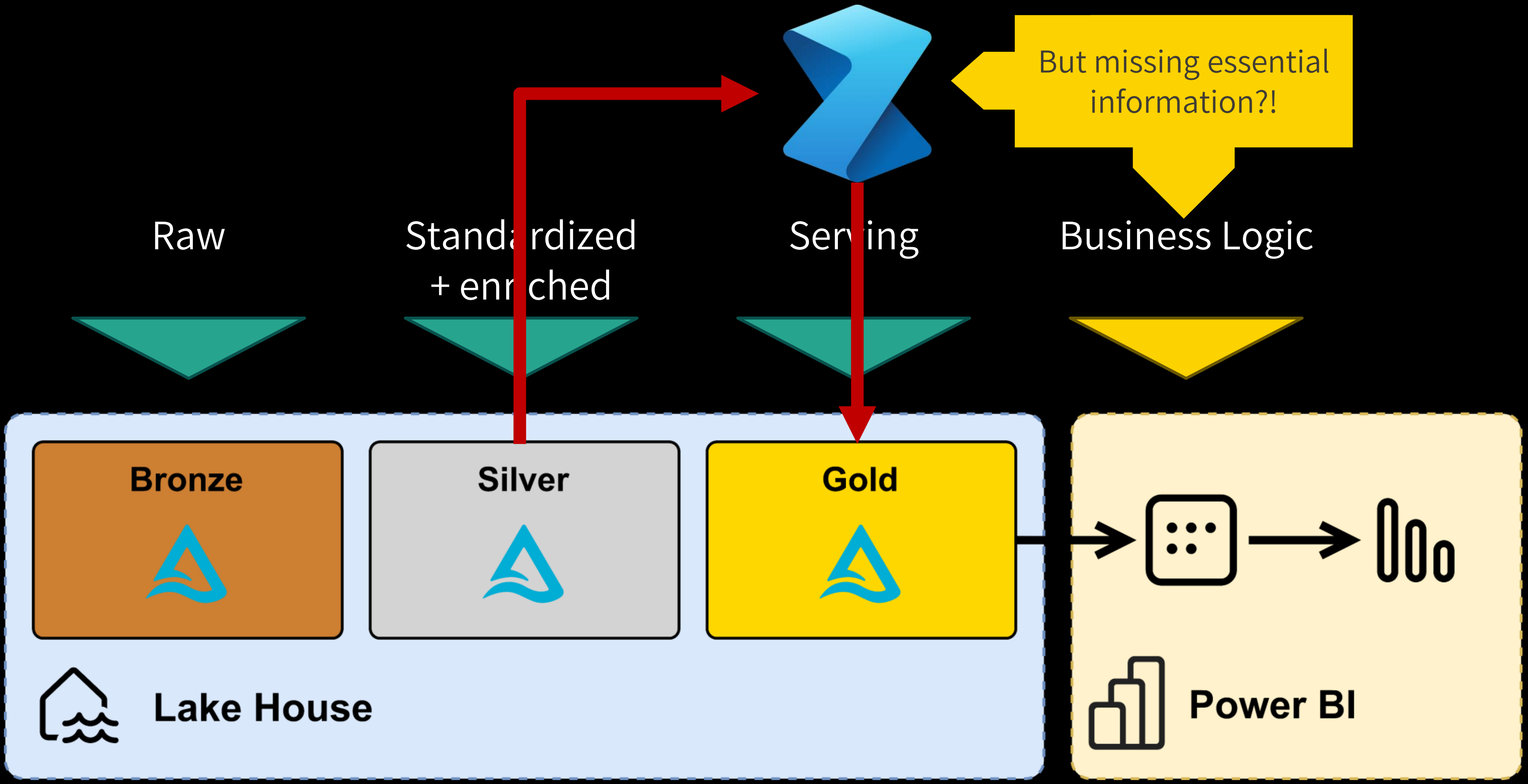
What is Semantic Link exactly?

Semantic Link is a feature in Microsoft Fabric that allows you to connect from ~~Synapse~~ Data Science Notebooks to Power BI Semantic Models.

This feature **only** exists and works in Microsoft Fabric.

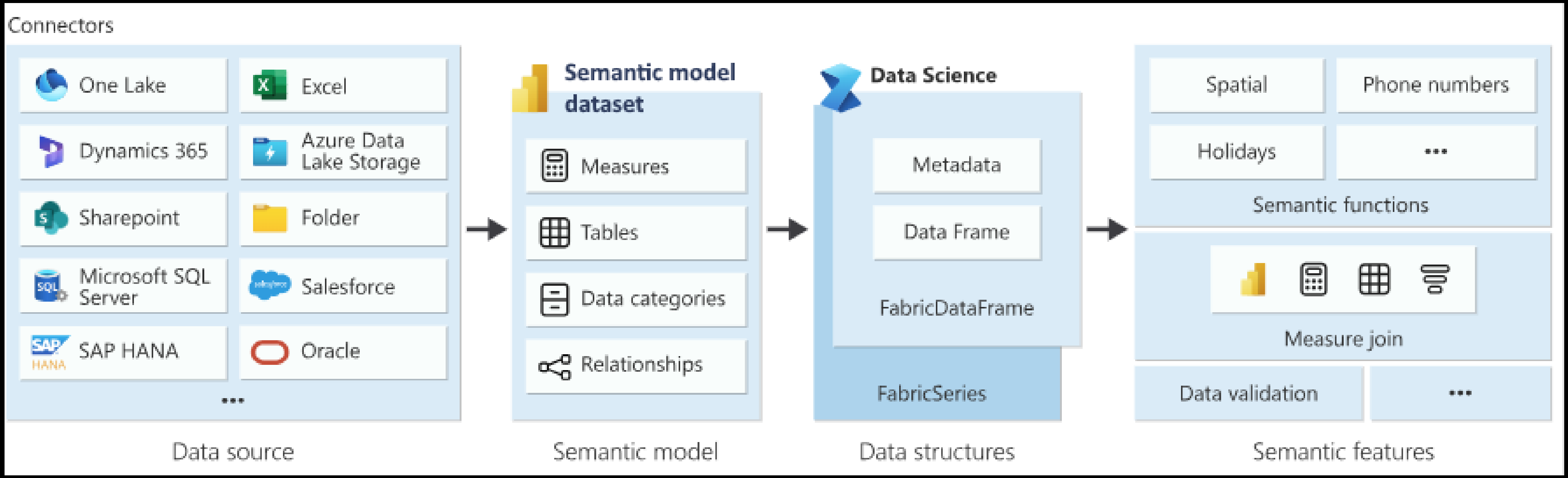


The "classic" Data Science story



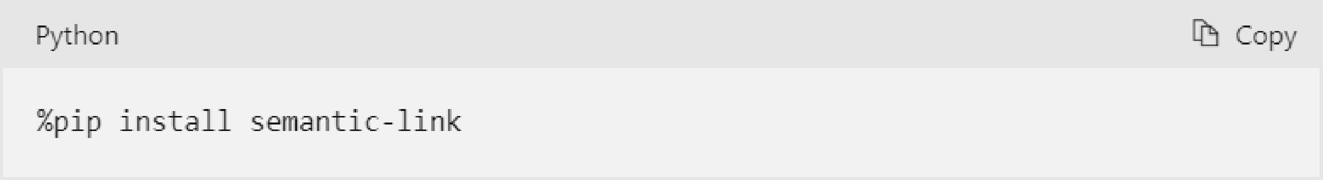
FabricDataFrame data structure

FabricDataFrame is the data structure of Semantic Link. It makes use of pandas DataFrame and adds meta data such as semantic information and lineage.



Getting started

Get the library installed, to begin with



```
%pip install semantic-link
```

This installs the library which allows us to interact with Semantic Models.

~~Planning to build multiple notebooks?~~
~~Consider using a custom environment.~~

Semantic Link or SemPy?

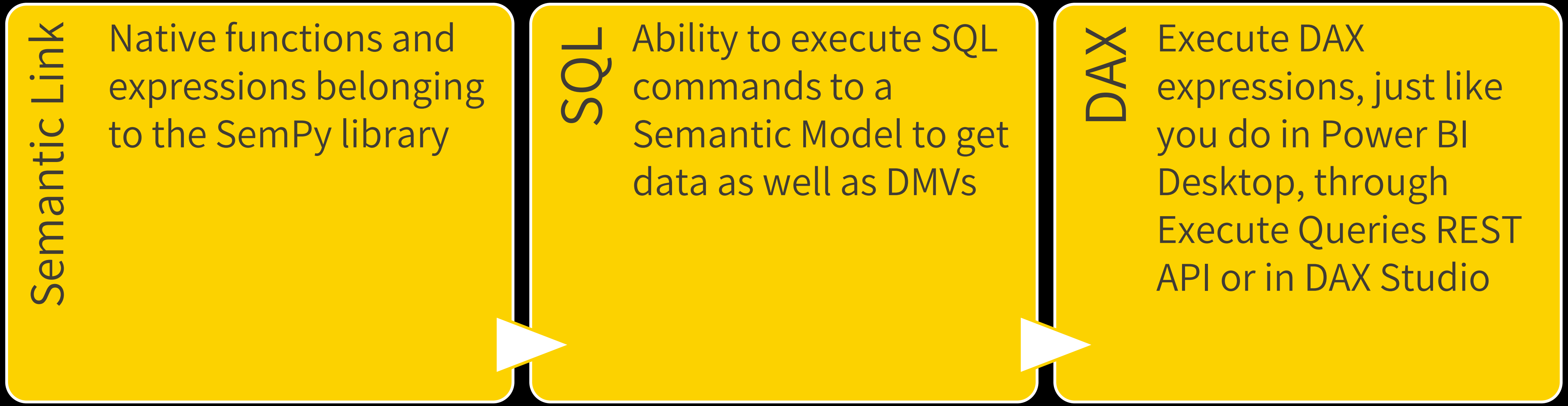
Both names are used, might cause confusion. But there are differences!

Packages	Description
Semantic-link	Meta-package that depends on all individual Semantic Link packages, easy way to install them all at once.
Semantic-link-sempy	The package that only contains the core Semantic Link functionality
Semantic-link-functions-holidays	A package that contains semantic functions for holidays (determine if a day is a holiday etc.)
Semantic-link-geopandas	Semantic Link packages depending on geopandas to work with spatial data, such as GIS.

Multiple languages and Magic Commands

Notebooks support multiple languages and so does Semantic Link.

There are various options to get your started.



Semantic Link

Native functions and expressions belonging to the SemPy library

SQL

Ability to execute SQL commands to a Semantic Model to get data as well as DMVs

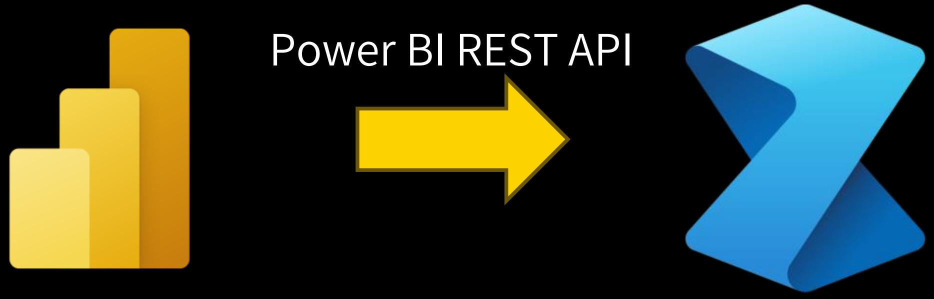
DAX

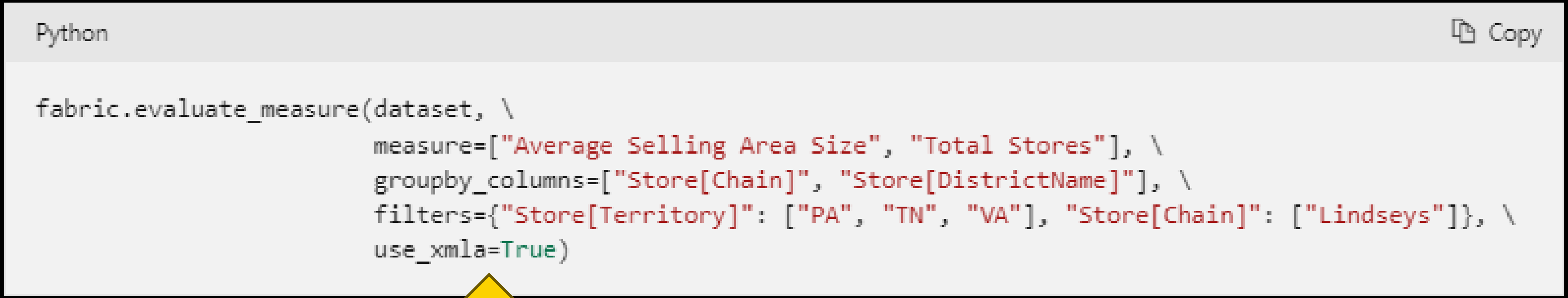
Execute DAX expressions, just like you do in Power BI Desktop, through Execute Queries REST API or in DAX Studio

Demo – First exploration of Semantic Link

Connectivity

Default uses the Power BI REST API. For certain operations, the XMLA endpoint might be more useful. With *use_xmla=True* you can direct the connection of XMLA.





```python
fabric.evaluate_measure(dataset, \
                        measure=["Average Selling Area Size", "Total Stores"], \
                        groupby_columns=["Store[Chain]", "Store[DistrictName]"], \
                        filters={"Store[Territory]": ["PA", "TN", "VA"], "Store[Chain]": ["Lindseys"]}, \
                        use_xmla=True)
```

One level further: Semantic Link Labs

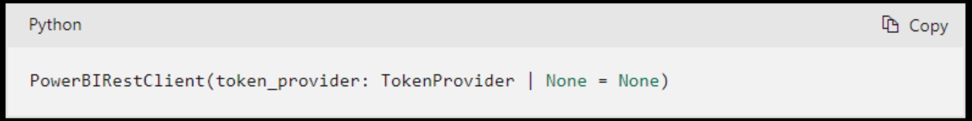
- Migrate import semantic models to Fabric solutions
- Translate model meta data to various languages
- Refresh specified tables/partitions/…
- Run DAX INFO functions to gain information about your model
- Execute Best Practice Analyzer from Fabric notebooks
- Run Vertipaq analyzer to gain statistics from your model
- Identify potential Direct Lake fallbacks

Use cases

Orchestration

Refresh your semantic models via a notebook and trigger dependent actions.

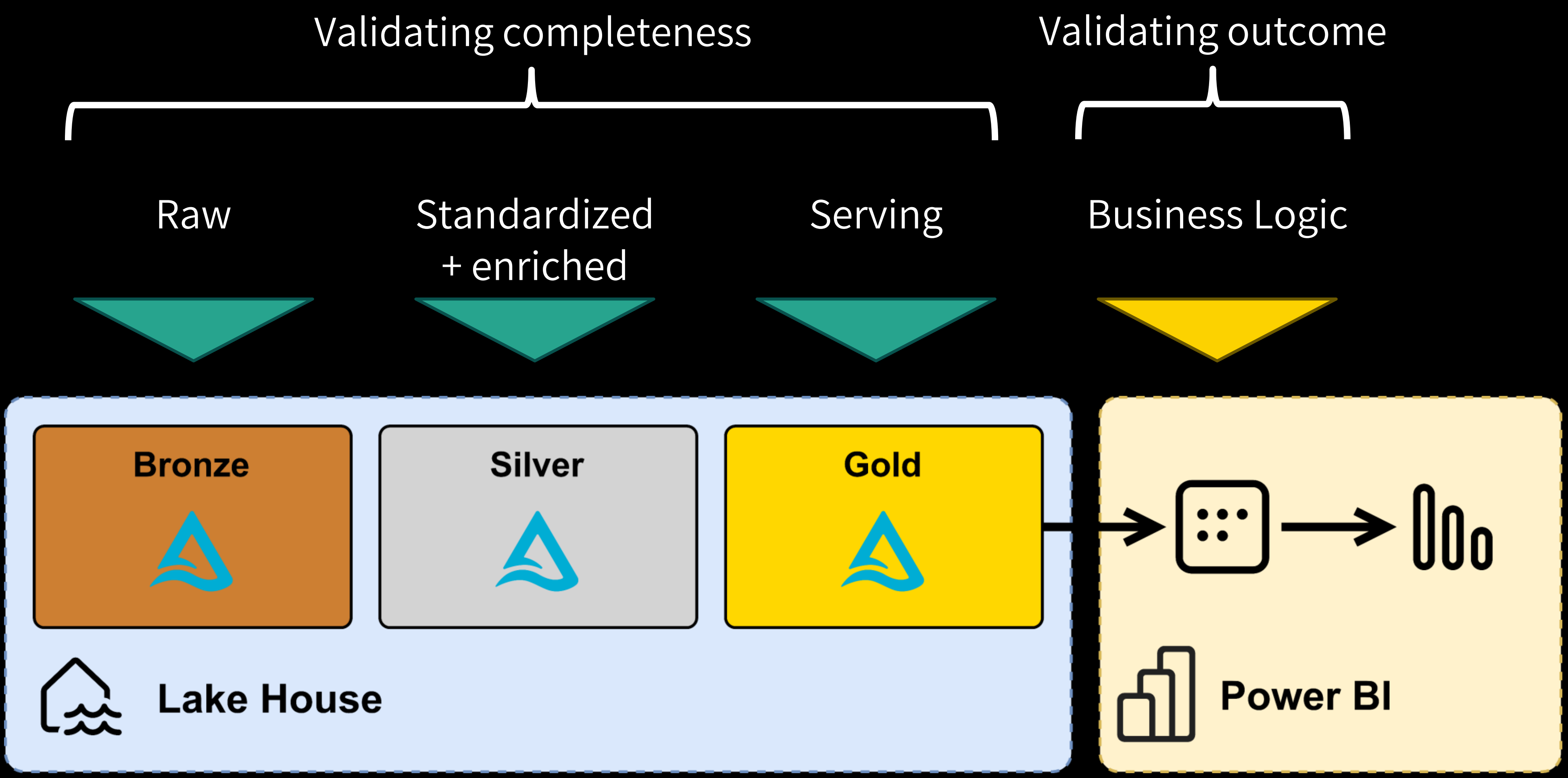
- Given Semantic Link uses the REST APIs, you can orchestrate not only your semantic model refresh, but also trigger upstream dataflows for example
- Refresh individual tables, partitions or reprocess partitions through enhanced refresh API
- Anything else what is possible with the REST API



```python
PowerBIRestClient(token_provider: TokenProvider | None = None)
```

Demo – REST API usage for refreshes

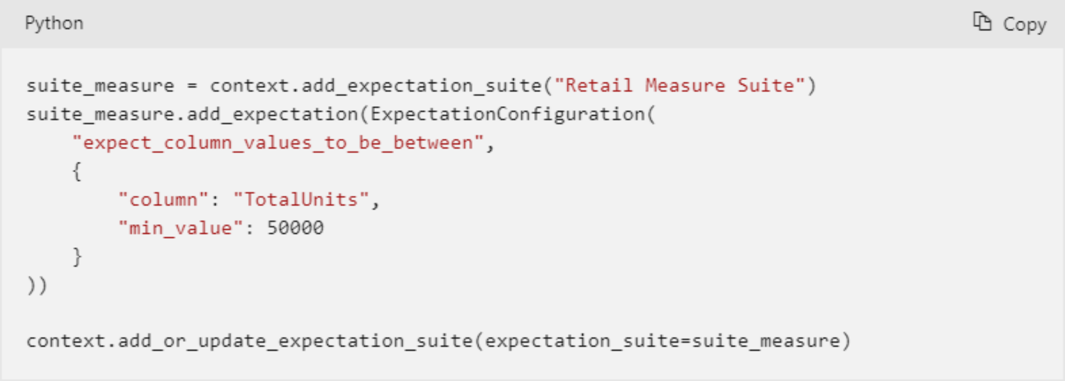
Data validation



Data validation

Data validation based on enrichments (measures) in Semantic Model, therefore different than validation on Gold layer in lakehouse.

- Makes use of public library Great Expectations
- Can be run against Tables, Measures, DMVs
- Sets rules on data types and validates them
  - E.g. Postal codes needs to have 4 numbers and 2 letters (in Dutch system)
  - E.g. Value in column X must be in range between A and B
  - E.g. Units Sold should always be a full number, no decimals



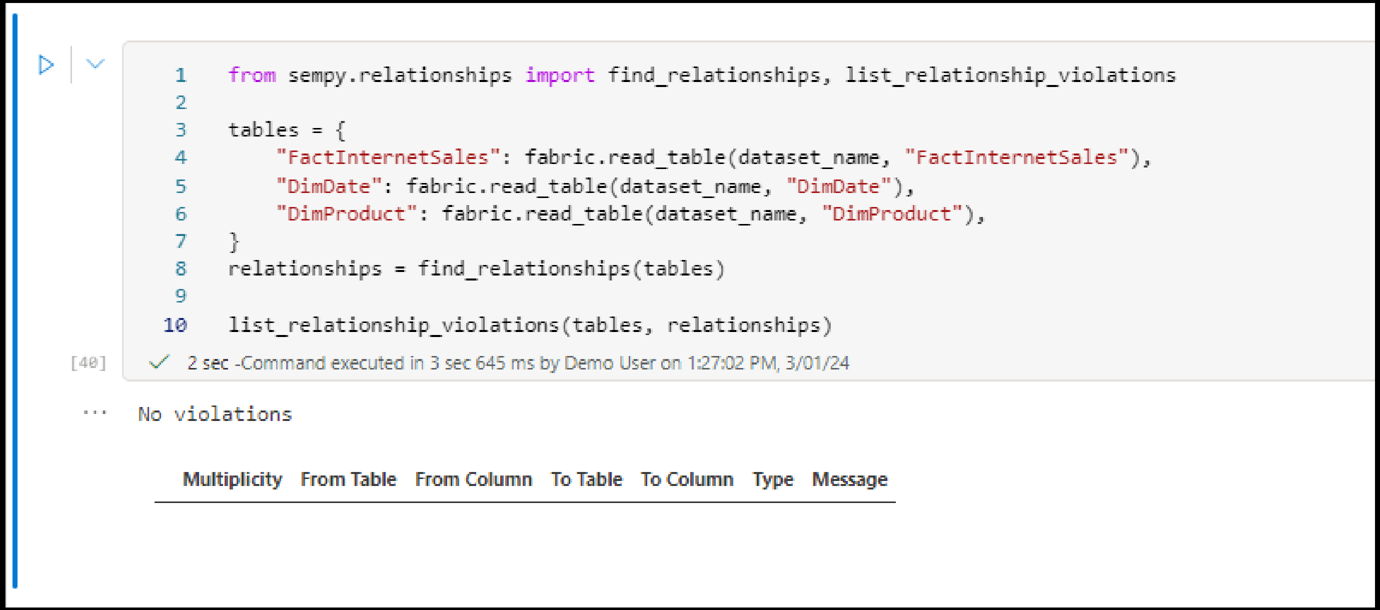
```python
suite_measure = context.add_expectation_suite("Retail Measure Suite")
suite_measure.add_expectation(ExpectationConfiguration(
    "expect_column_values_to_be_between",
    {
        "column": "TotalUnits",
        "min_value": 50000
    }
))

context.add_or_update_expectation_suite(expectation_suite=suite_measure)
```

Semantic Model Quality

Mainly depending on DMVs to query Semantic Model meta data and ability to trace dependencies in queries or relationship integrity for example.

E.g. Is the one-side of your relationship, really unique?



```python
from sempy.relationships import find_relationships, list_relationship_violations

tables = {
    "FactInternetSales": fabric.read_table(dataset_name, "FactInternetSales"),
    "DimDate": fabric.read_table(dataset_name, "DimDate"),
    "DimProduct": fabric.read_table(dataset_name, "DimProduct"),
}
relationships = find_relationships(tables)

list_relationship_violations(tables, relationships)
```

[40] ✓ 2 sec -Command executed in 3 sec 645 ms by Demo User on 1:27:02 PM, 3/01/24

```
No violations
```

Multiplicity	From Table	From Column	To Table	To Column	Type	Message

Query data to use elsewhere

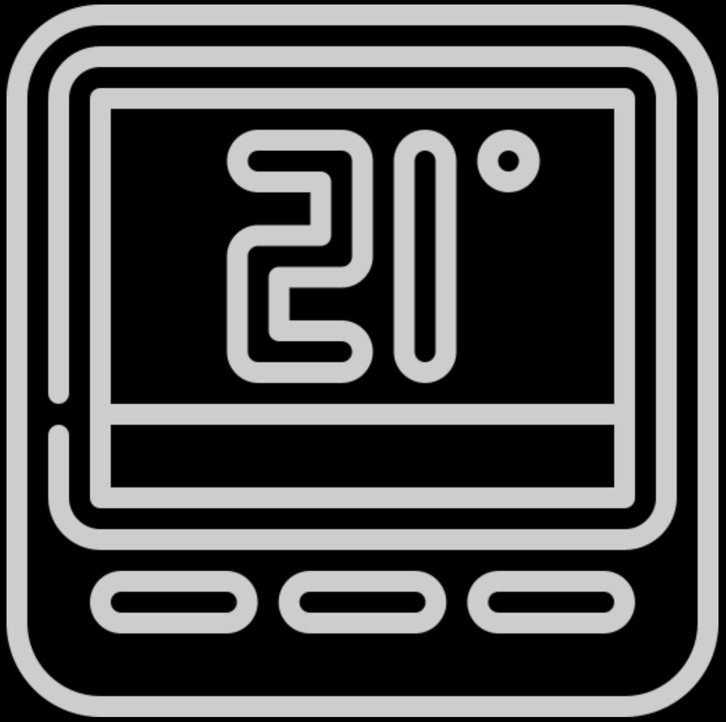
Imagine you invested a lot of time to bring together various data sources. Your Semantic Model turns into a small data warehouse solution. The ability to query data using Semantic Link opens all sorts of new options also to get your data out of Power BI again.

Should you? **NO!**

Power BI is not an ETL tool. Do your data transformations as far upstream as possible – Roche’s Maxim

Warm-up Direct Lake Semantic Models

- When using a Semantic Model with **Direct Lake** storage mode (Fabric only), your data is loaded on-demand to memory.
- This means, only columns that are queried are loaded into the capacities memory. Once loaded, the column will get a **temperature**.
- Every time a new column is loaded, there is a slight **performance** impact since data must be loaded from storage to memory.
- Over time, temperature will drop to zero, and eventually data will **evicted** from memory.



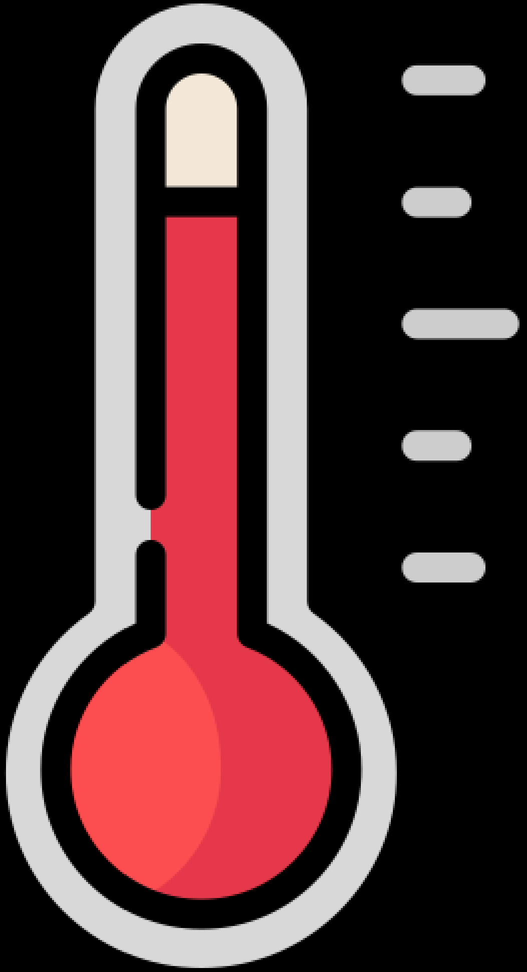
Warm-up Direct Lake Semantic Models

What will be evicted?

Basically, your data will be evicted from active memory, that you want to always have available!

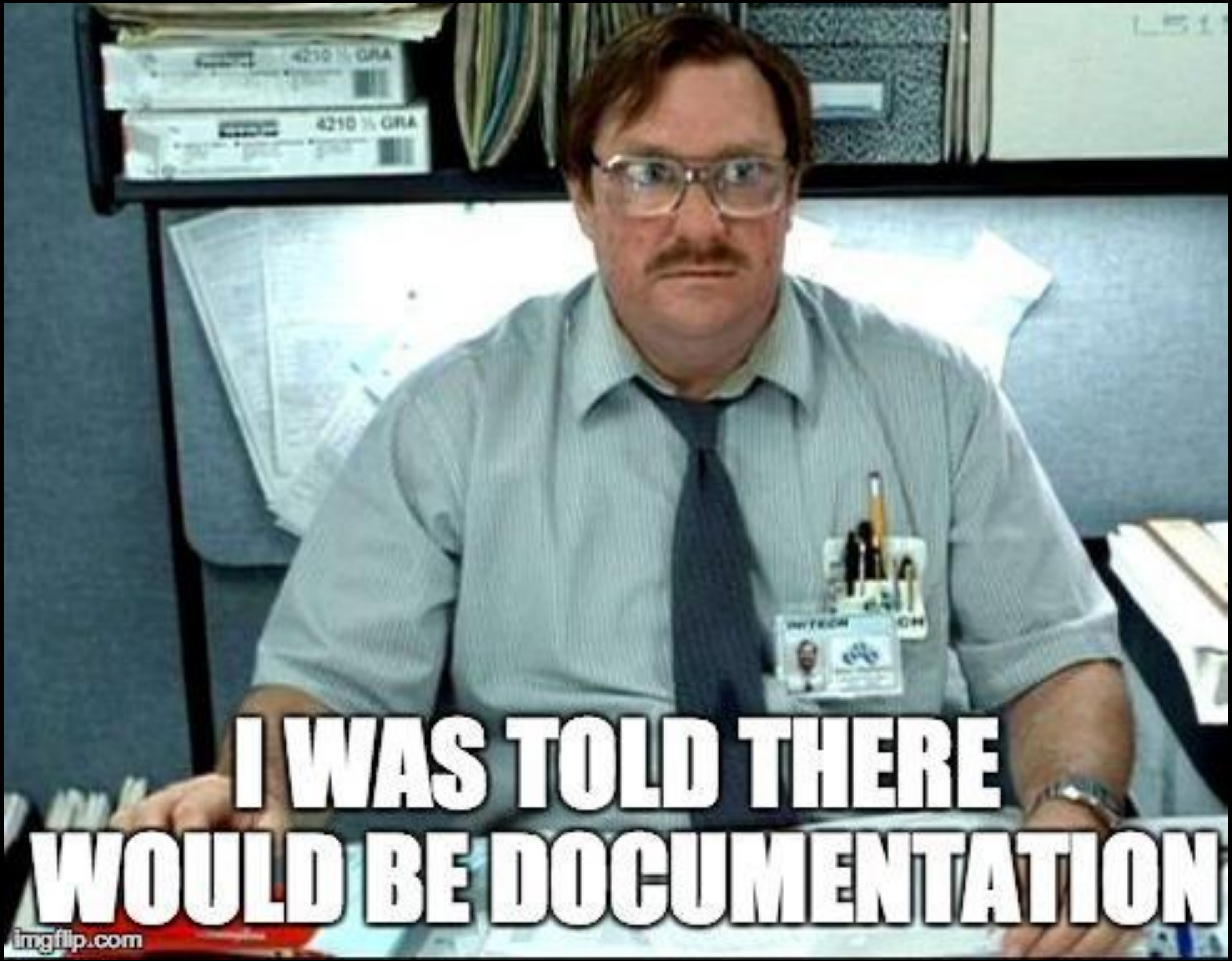
How can you influence that?

Consider setting up a process (notebook, other automated setup) to pro-actively execute queries to keep certain data **WARM**!



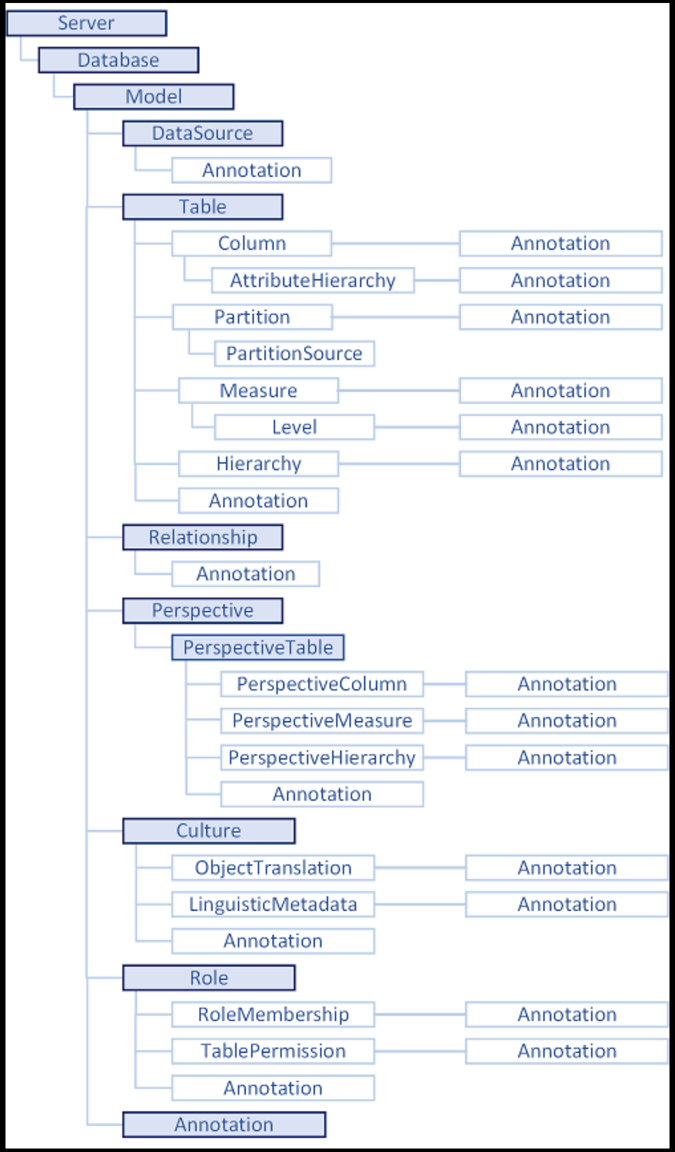
Demo – Direct Lake data warm-up

Document your Semantic Model


I WAS TOLD THERE
WOULD BE DOCUMENTATION
imgflip.com

Data model metadata

- Matches Analysis Services metadata
- Model.bim
- Tabular Object Model (TOM)
- Open format (json)
- Now, also TMDL (February 2024 update)

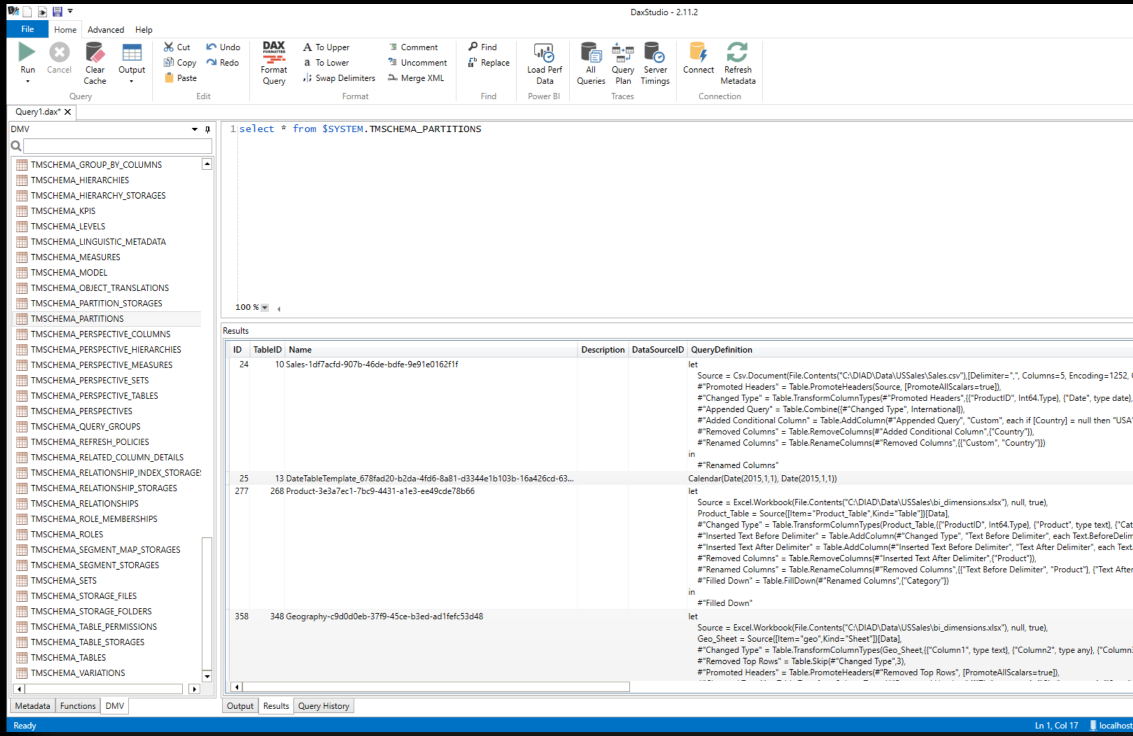


- Server
  - Database
    - Model
      - DataSource
        - Annotation
      - Table
        - Column
          - Annotation
          - AttributeHierarchy
            - Annotation
        - Partition
          - Annotation
          - PartitionSource
        - Measure
          - Annotation
          - Level
            - Annotation
        - Hierarchy
          - Annotation
        - Annotation
      - Relationship
        - Annotation
      - Perspective
        - PerspectiveTable
          - PerspectiveColumn
            - Annotation
          - PerspectiveMeasure
            - Annotation
          - PerspectiveHierarchy
            - Annotation
          - Annotation
      - Culture
        - ObjectTranslation
          - Annotation
        - LinguisticMetadata
          - Annotation
        - Annotation
      - Role
        - RoleMembership
          - Annotation
        - TablePermission
          - Annotation
        - Annotation
      - Annotation

Dynamic Management Views

Analysis Services Dynamic Management Views (DMVs) are queries that return information about model objects, server operations, and server health.

- DB Schema	=	Database model
- DISCOVER	=	Operations & Sessions
- TM Schema	=	Tabular = Power BI / AAS
- MD Schema	=	MDX = Multidimensional



- TMSCHEMA_GROUP_BY_COLUMNS
- TMSCHEMA_HIERARCHIES
- TMSCHEMA_HIERARCHY_STORAGES
- TMSCHEMA_KPIS
- TMSCHEMA_LEVELS
- TMSCHEMA_LINGUISTIC_METADATA
- TMSCHEMA_MEASURES
- TMSCHEMA_MODEL
- TMSCHEMA_OBJECT_TRANSLATIONS
- TMSCHEMA_PARTITION_STORAGES
- TMSCHEMA_PARTITIONS
- TMSCHEMA_PERSPECTIVE_COLUMNS
- TMSCHEMA_PERSPECTIVE_HIERARCHIES
- TMSCHEMA_PERSPECTIVE_MEASURES
- TMSCHEMA_PERSPECTIVE_SETS
- TMSCHEMA_PERSPECTIVE_TABLES
- TMSCHEMA_PERSPECTIVES
- TMSCHEMA_QUERY_GROUPS
- TMSCHEMA_REFRESH_POLICIES
- TMSCHEMA_RELATED_COLUMN_DETAILS
- TMSCHEMA_RELATIONSHIP_INDEX_STORAGE:
- TMSCHEMA_RELATIONSHIP_STORAGES
- TMSCHEMA_RELATIONSHIPS
- TMSCHEMA_ROLE_MEMBERSHIPS
- TMSCHEMA_ROLES
- TMSCHEMA_SEGMENT_MAP_STORAGES
- TMSCHEMA_SEGMENT_STORAGES
- TMSCHEMA_SETS
- TMSCHEMA_STORAGE_FILES
- TMSCHEMA_STORAGE_FOLDERS
- TMSCHEMA_TABLE_PERMISSIONS
- TMSCHEMA_TABLE_STORAGES
- TMSCHEMA_TABLES
- TMSCHEMA_VARIATIONS

```
select * from $SYSTEM.TMSCHEMA_PARTITIONS
```

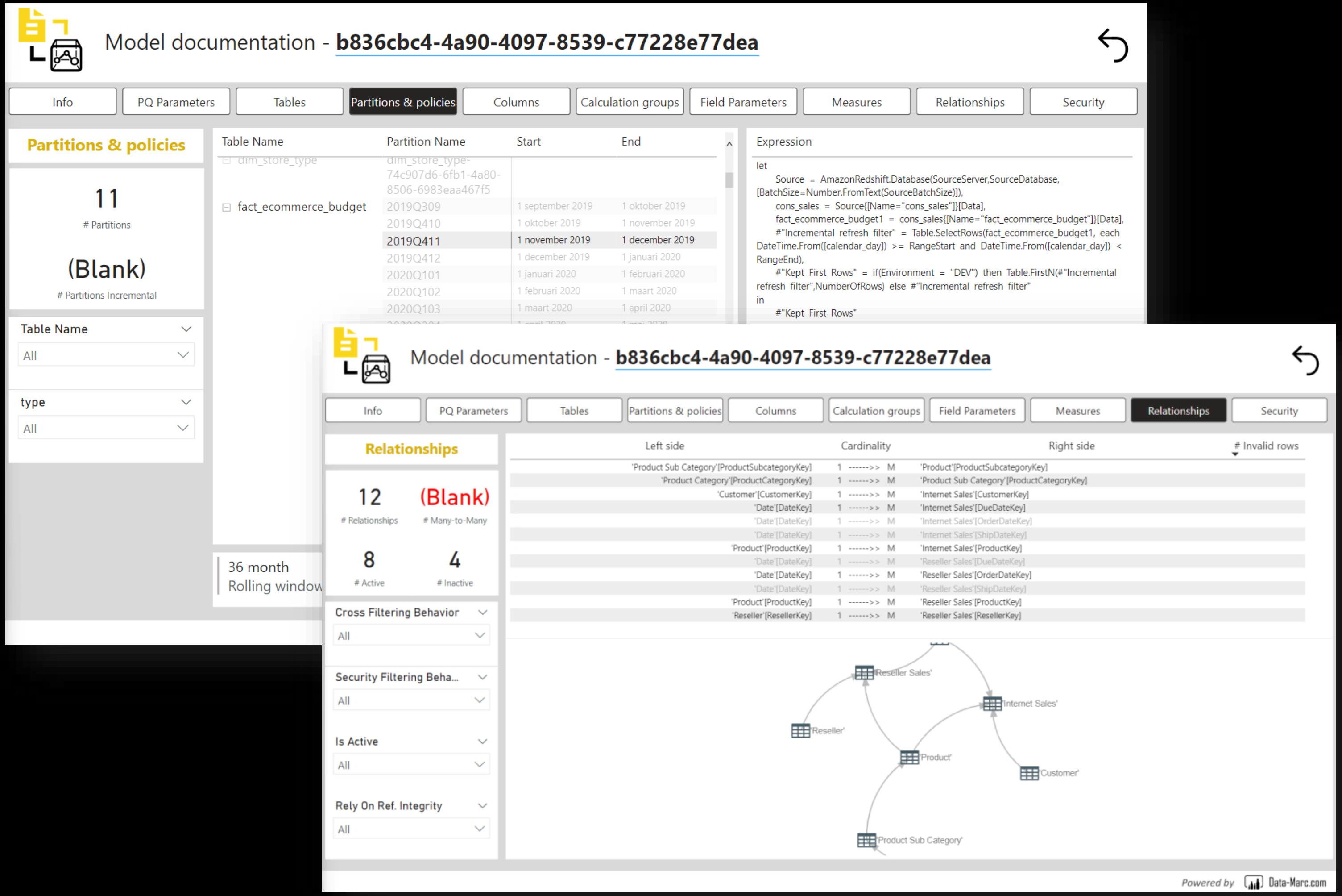
Results

ID	TableID	Name	Description	DataSourceID	QueryDefinition
24	10	Sales-1df7acfd-907b-46de-bdfe-9e91e0162f1f			let Source = Csv.Document(File.Contents("C:\DIAD\Data\USSales\Sales.csv"),[Delimiter=",", Columns=5, Encoding=1252, #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]), #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"ProductID", Int64.Type}, {"Date", type date}, #"Appended Query" = Table.Combine({#"Changed Type", International}), #"Added Conditional Column" = Table.AddColumn(#"Appended Query", "Custom", each if [Country] = null then "USA #"Removed Columns" = Table.RemoveColumns(#"Added Conditional Column",{"Country"}), #"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"Custom", "Country"}}) in #"Renamed Columns"
25	13	DateTableTemplate_678fad20-b2da-4fd6-8a81-d3344e1b103b-16a426cd-63...			Calendar(Date(2015,1,1), Date(2015,1,1))
277	268	Product-3e3a7ec1-7bc9-4431-a1e3-ee49cde78b66			let Source = Excel.Workbook(File.Contents("C:\DIAD\Data\USSales\bi_dimensions.xlsx"), null, true), Product_Table = Source{[Item="Product_Table",Kind="Table"]}[Data], #"Changed Type" = Table.TransformColumnTypes(Product_Table,{{"ProductID", Int64.Type}, {"Product", type text}, {"Cat #"Inserted Text Before Delimiter" = Table.AddColumn(#"Changed Type", "Text Before Delimiter", each Text.BeforeDelim #"Inserted Text After Delimiter" = Table.AddColumn(#"Inserted Text Before Delimiter", "Text After Delimiter", each Text. #"Removed Columns" = Table.RemoveColumns(#"Inserted Text After Delimiter",{"Product"}), #"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"Text Before Delimiter", "Product"}, {"Text After #"Filled Down" = Table.FillDown(#"Renamed Columns",{"Category"}) in #"Filled Down"
358	348	Geography-c9d0d0eb-37f9-45ce-b3ed-ad1fefc53d48			let Source = Excel.Workbook(File.Contents("C:\DIAD\Data\USSales\bi_dimensions.xlsx"), null, true), Geo_Sheet = Source{[Item="geo",Kind="Sheet"]}[Data], #"Changed Type" = Table.TransformColumnTypes(Geo_Sheet,{{"Column1", type text}, {"Column2", type any}, {"Column #"Removed Top Rows" = Table.Skip(#"Changed Type",3), #"Promoted Headers" = Table.PromoteHeaders(#"Removed Top Rows", [PromoteAllScalars=true]),

Data-Marc.com

Power BI Model Documenter

Power BI External Tool that let's you document your Power BI Solution by generating a VPAX file and visualizing in a Power BI report.



Model documentation - b836cbc4-4a90-4097-8539-c77228e77dea

Info | PQ Parameters | Tables | Partitions & policies | Columns | Calculation groups | Field Parameters | Measures | Relationships | Security

Partitions & policies

11
# Partitions

(Blank)
# Partitions Incremental

Table Name: All

type: All

Table Name	Partition Name	Start	End
dim_store_type	dim_store_type-74c907d6-6fb1-4a80-8506-6983eaa467f5		
fact_ecommerce_budget	2019Q309	1 september 2019	1 oktober 2019
	2019Q410	1 oktober 2019	1 november 2019
	2019Q411	1 november 2019	1 december 2019
	2019Q412	1 december 2019	1 januari 2020
	2020Q101	1 januari 2020	1 februari 2020
	2020Q102	1 februari 2020	1 maart 2020
	2020Q103	1 maart 2020	1 april 2020

Expression

```
let
    Source = AmazonRedshift.Database(SourceServer,SourceDatabase,
[BatchSize=Number.FromText(SourceBatchSize)]),
    cons_sales = Source{[Name="cons_sales"]}[Data],
    fact_ecommerce_budget1 = cons_sales{[Name="fact_ecommerce_budget"]}[Data],
    #"Incremental refresh filter" = Table.SelectRows(fact_ecommerce_budget1, each
DateTime.From([calendar_day]) >= RangeStart and DateTime.From([calendar_day]) <
RangeEnd),
    #"Kept First Rows" = if(Environment = "DEV") then Table.FirstN(#"Incremental
refresh filter",NumberOfRows) else #"Incremental refresh filter"
in
    #"Kept First Rows"
```

36 month
Rolling window

Model documentation - b836cbc4-4a90-4097-8539-c77228e77dea

Info | PQ Parameters | Tables | Partitions & policies | Columns | Calculation groups | Field Parameters | Measures | Relationships | Security

Relationships

12
# Relationships

(Blank)
# Many-to-Many

8
# Active

4
# Inactive

Cross Filtering Behavior: All

Security Filtering Beha...: All

Is Active: All

Rely On Ref. Integrity: All

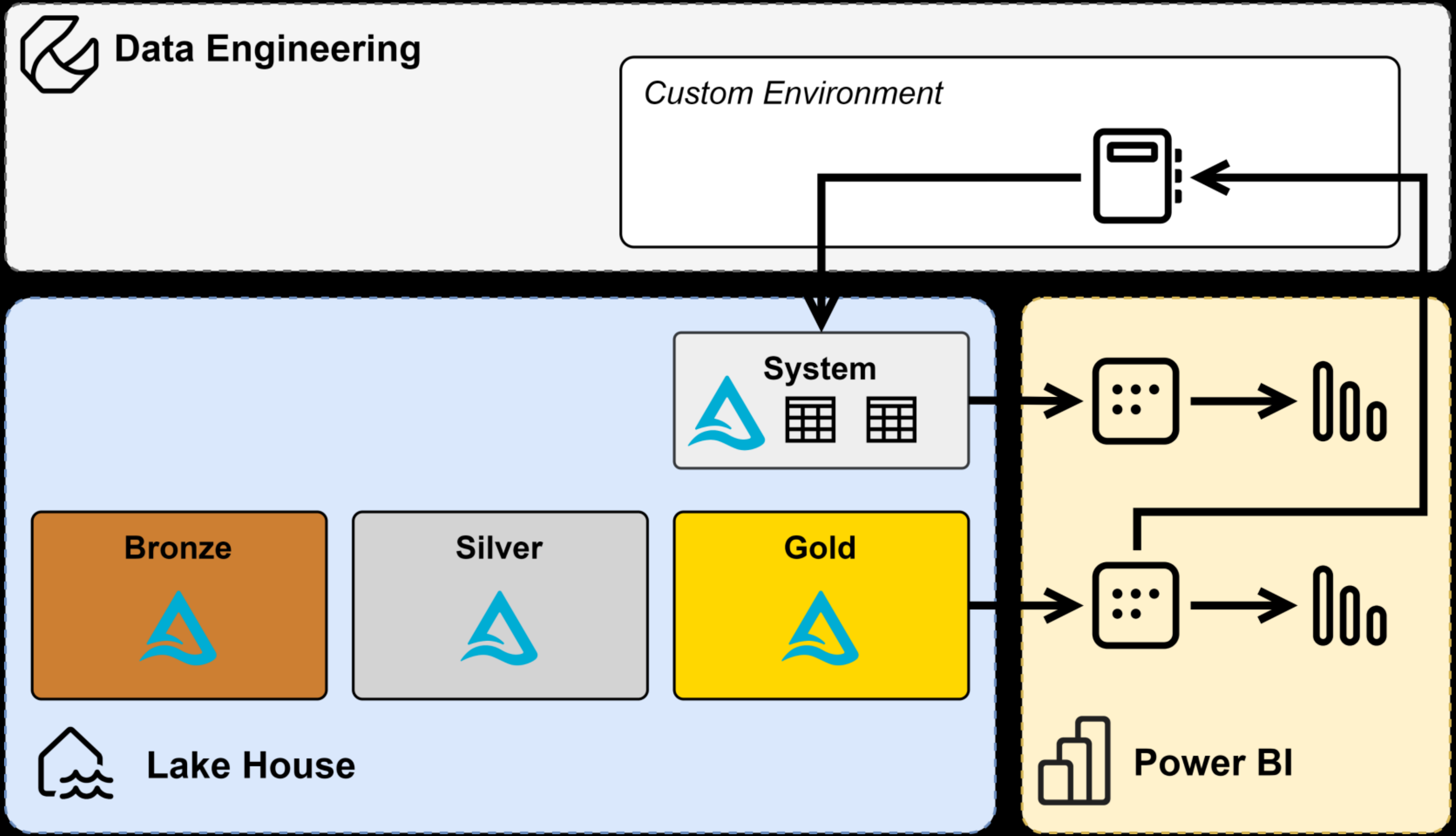
Left side	Cardinality	Right side	# Invalid rows
'Product Sub Category'[ProductSubcategoryKey]	1 ------>> M	'Product'[ProductSubcategoryKey]	
'Product Category'[ProductCategoryKey]	1 ------>> M	'Product Sub Category'[ProductCategoryKey]	
'Customer'[CustomerKey]	1 ------>> M	'Internet Sales'[CustomerKey]	
'Date'[DateKey]	1 ------>> M	'Internet Sales'[DueDateKey]	
'Date'[DateKey]	1 ------>> M	'Internet Sales'[OrderDateKey]	
'Date'[DateKey]	1 ------>> M	'Internet Sales'[ShipDateKey]	
'Product'[ProductKey]	1 ------>> M	'Internet Sales'[ProductKey]	
'Date'[DateKey]	1 ------>> M	'Reseller Sales'[DueDateKey]	
'Date'[DateKey]	1 ------>> M	'Reseller Sales'[OrderDateKey]	
'Date'[DateKey]	1 ------>> M	'Reseller Sales'[ShipDateKey]	
'Product'[ProductKey]	1 ------>> M	'Reseller Sales'[ProductKey]	
'Reseller'[ResellerKey]	1 ------>> M	'Reseller Sales'[ResellerKey]	

Powered by Data-Marc.com

Data-Marc.com

Conceptual overview

Using Semantic-Link to read Meta Data from an existing Sematic Model, which is saved in a Lake House, with a Semantic Model and Report on top to visualize the output.



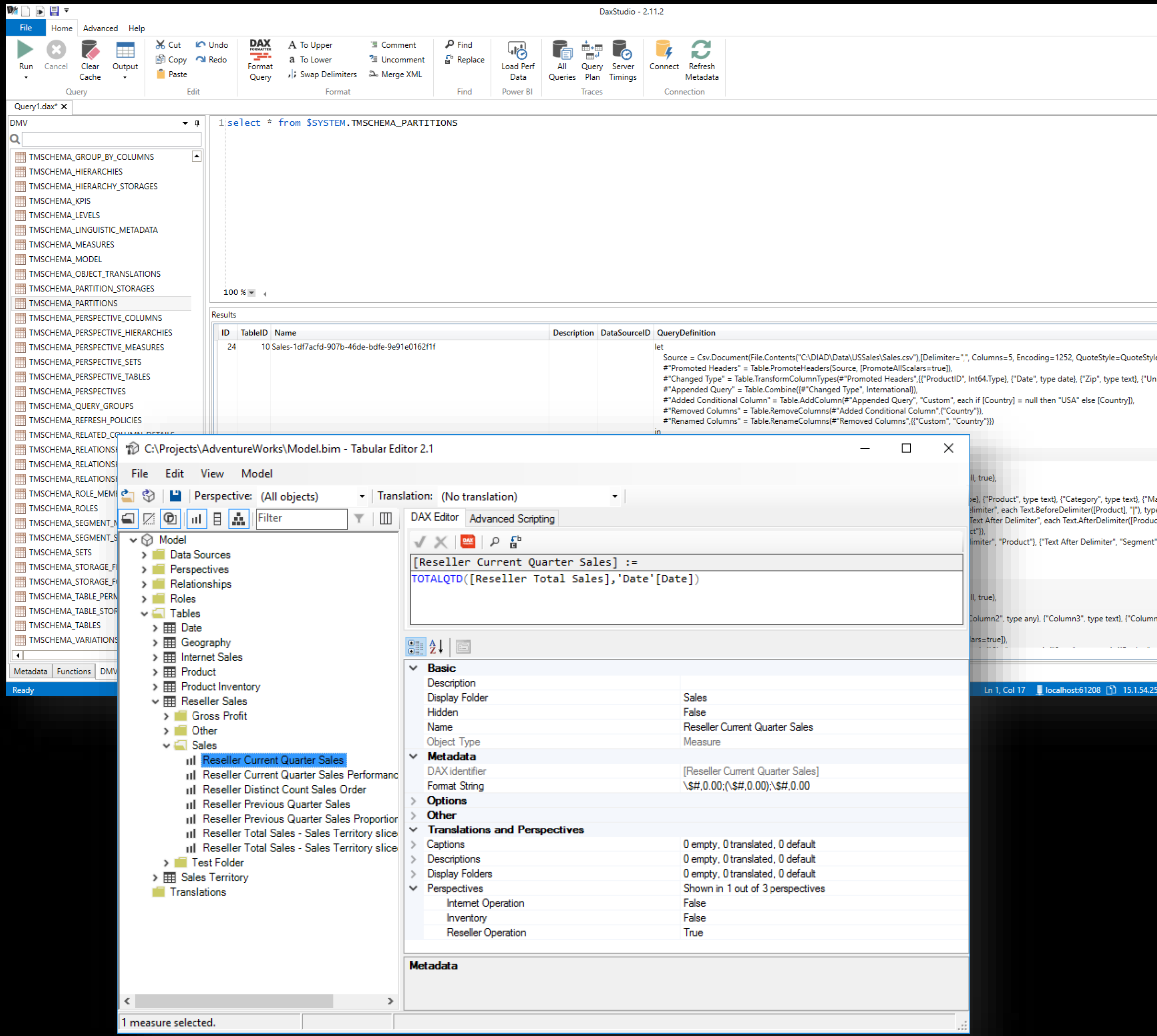
Demo – Document your solution using Semantic Link

Considerations

Sounds familiar?

We could already do a lot using Tabular Editor / DAX Studio and **XMLA endpoints** for Semantic Model quality checks, but requires user input.

Executing DMVs and model documentation can be done using **External Tools** in PBI Desktop, but cannot be refreshed/updated easily.



```
select * from $SYSTEM.TMSCHEMA_PARTITIONS
```

ID	TableID	Name	Description	DataSourceID	QueryDefinition
24	10	Sales-1df7acfd-907b-46de-bdfe-9e91e0162f1f			let

```
let
    Source = Csv.Document(File.Contents("C:\DIAD\Data\USSales\Sales.csv"),[Delimiter=",", Columns=5, Encoding=1252, QuoteStyle=QuoteStyle
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"ProductID", Int64.Type}, {"Date", type date}, {"Zip", type text}, {"Un
    #"Appended Query" = Table.Combine({#"Changed Type", International}),
    #"Added Conditional Column" = Table.AddColumn(#"Appended Query", "Custom", each if [Country] = null then "USA" else [Country]),
    #"Removed Columns" = Table.RemoveColumns(#"Added Conditional Column",{"Country"}),
    #"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"Custom", "Country"}})
in
```

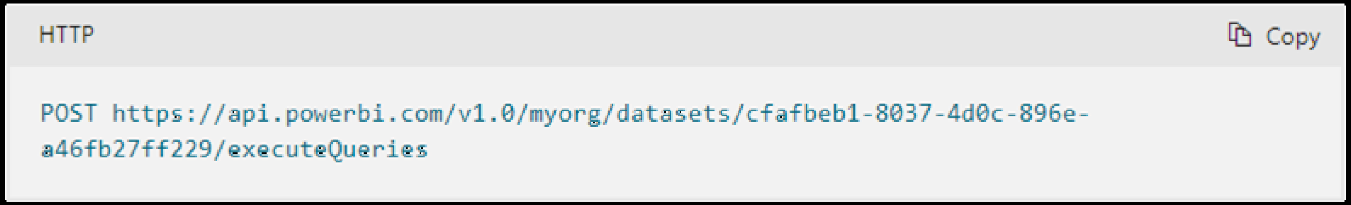
```
[Reseller Current Quarter Sales] :=
TOTALQTD([Reseller Total Sales],'Date'[Date])
```

Basic	
Description	
Display Folder	Sales
Hidden	False
Name	Reseller Current Quarter Sales
Object Type	Measure
Metadata	
DAX identifier	[Reseller Current Quarter Sales]
Format String	\$#,0.00;(\$#,0.00);\$#,0.00
Options	
Other	
Translations and Perspectives	
Captions	0 empty, 0 translated, 0 default
Descriptions	0 empty, 0 translated, 0 default
Display Folders	0 empty, 0 translated, 0 default
Perspectives	Shown in 1 out of 3 perspectives
Internet Operation	False
Inventory	False
Reseller Operation	True

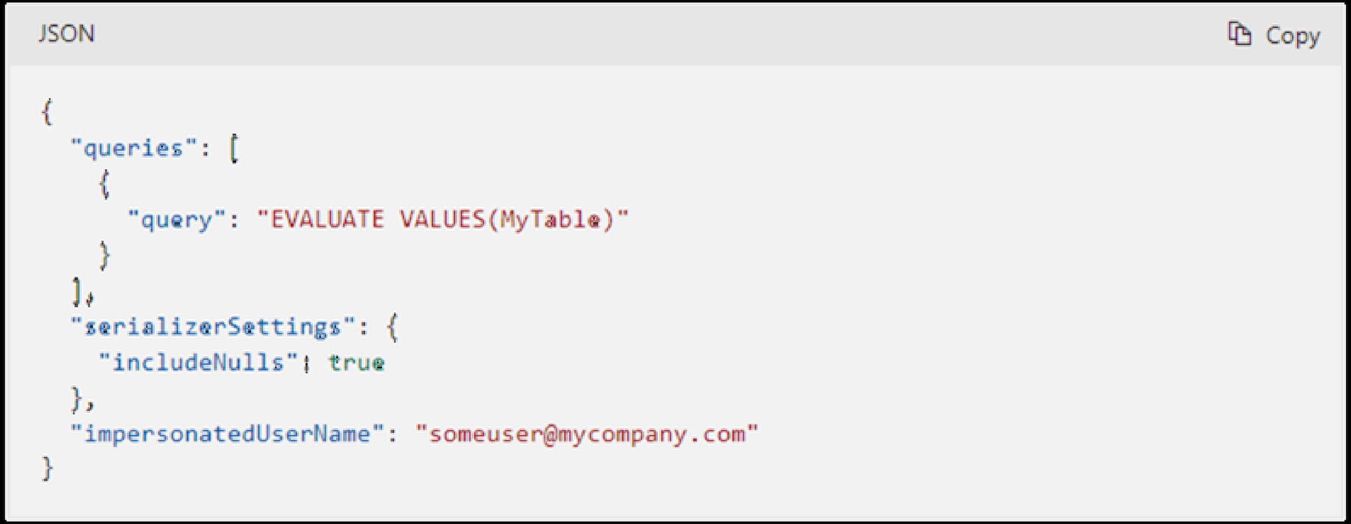
Data-Marc.com

Sounds familiar?

Data validation use cases, we could already do this using the **Execute Queries REST API**, but is more complex to setup yourself.



```
POST https://api.powerbi.com/v1.0/myorg/datasets/cfafbeb1-8037-4d0c-896e-
a46fb27ff229/executeQueries
```



```
{
  "queries": [
    {
      "query": "EVALUATE VALUES(MyTable)"
    }
  ],
  "serializerSettings": {
    "includeNulls": true
  },
  "impersonatedUserName": "someuser@mycompany.com"
}
```

Wrap-up and resources

Keep in mind that...

- Semantic Link as a whole, only works in **Fabric Notebooks**
- Semantic Link is currently in ~~**Preview**~~ **GENERAL AVAILABLE**
- Some use cases only apply to **Fabric specific** solutions (warm-up story)
- There is **limited** content available online

Wrap up

LET'S RECAP...

- Semantic Link allows you to **connect** to your Semantic Model via a Notebook.
- It only works in **Fabric Notebooks**, no limitations on SKUs.
- You can query **data**, **Dynamic Management Views** and any kind of **meta data**.
- Semantic Link allows you to validate both **semantic model quality** and **data quality**.
- Can be used to **extract data** from Power BI to other tools, but **you shouldn't** IMO.
- It perfectly works to **generate documentation** that updates as part of you end-to-end pipeline after refreshes.

Resources

Semantic-Link overview documentation
https://learn.microsoft.com/en-us/fabric/data-science/semantic-link-overview

Semantic-Link Python reference
https://learn.microsoft.com/en-us/python/api/semantic-link-sempy

Fabric Semantic Link and Use Cases by Sandeep Pawar
https://fabric.guru/fabric-semantic-link-and-use-cases

Refreshing (historical) partitions in Power BI Incremental Refresh Semantic Models using Fabric Semantic Link
https://data-marc.com/2024/05/28/dynamically-refreshing-historical-partitions-in-power-bi-incremental-refresh-semantic-models-using-fabric-semantic-link/

These slides
https://github.com/marclelijveld/Slide-decks

Thanks for attending!

Marc Lelijveld
Technical Evangelist | Architect
Macaw Netherlands

Microsoft® Most Valuable Professional MVP

sessionize MOST ACTIVE SPEAKER 2022

sessionize MOST ACTIVE SPEAKER 2023

@MarcLelijveld

linkedin.com/in/MarcLelijveld

Data-Marc.com

Session Evals