

Exploring Fabric Semantic Link for Power BI Folks



Partners

Data Point Prague 2024



**Data
Brothers**



OKškolení
Powered by OKsystem



Tabular Editor



Learning objectives

Fabric

Understand what Microsoft Fabric is, how this relates to Semantic Link and where it is positioned.

Semantic Link

Know exactly what Semantic Link is, how you can use it in your benefit to power your solutions.

Query

Be able to query data and meta data of your semantic model using Semantic Link.

Document

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



@MarcLelijveld



linkedin.com/in/MarcLelijveld



Data-Marc.com



DutchFabricUsergroup.com

FAVORITE STUFF:

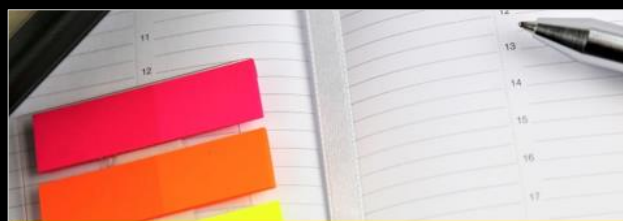


Agenda for today



Introduction to Fabric

 Data-Marc.com



What are notebooks?

 Data-Marc.com



Understanding Semantic Link

 Data-Marc.com



Use cases

 Data-Marc.com



Document your Semantic Model

 Data-Marc.com



Considerations

 Data-Marc.com



Introduction to Fabric

Microsoft Fabric

Complete Analytics Platform

Everything, unified

SaaS-ified

Secured and governed

Lake centric and open

OneLake

One Copy

Open at every tier

Empower Every Business User

Familiar and intuitive

Built into Microsoft 365

Insight to action

AI Powered

Copilot accelerated

ChatGPT on your data

AI driven insights

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

Technology impact

Stages

Ingest	Transform	Model	Visualize
--------	-----------	-------	-----------

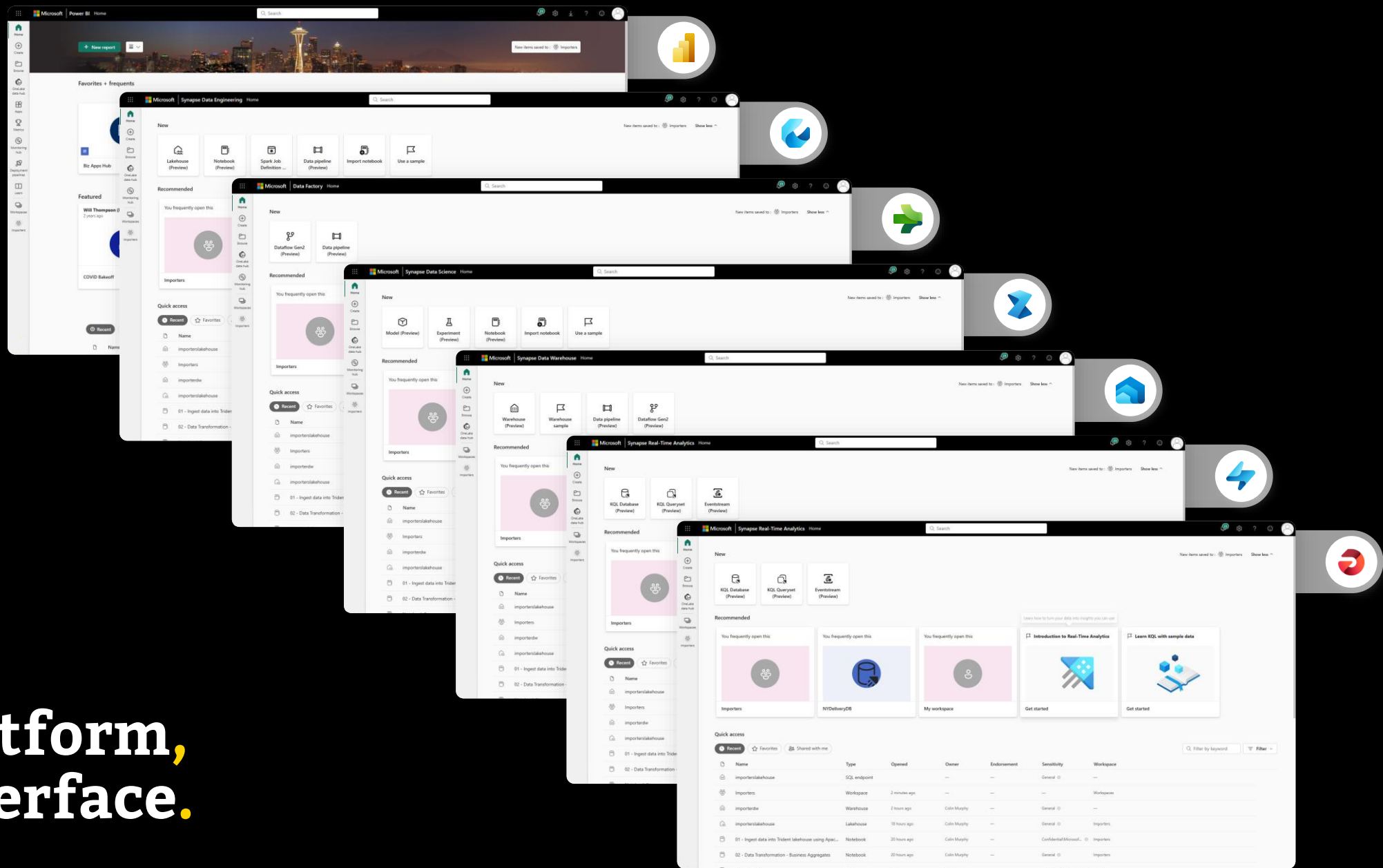
Areas (“Experience”)

<i>Databricks</i>		Warehouse SQL Endpoint	Power BI
Data Factory	Data Engineering		

Artifacts (“Workloads”)

Lakehouse (Bronze / Silver / Gold)	<i>Spark</i>	Semantic Model	Report Paginated Report Metrics
Pipelines	<i>SQL</i>		
Notebooks	<i>Python</i>		
Dataflows (gen2)	<i>Power Query</i>	<i>DAX</i>	

One platform, One interface.





What are notebooks?

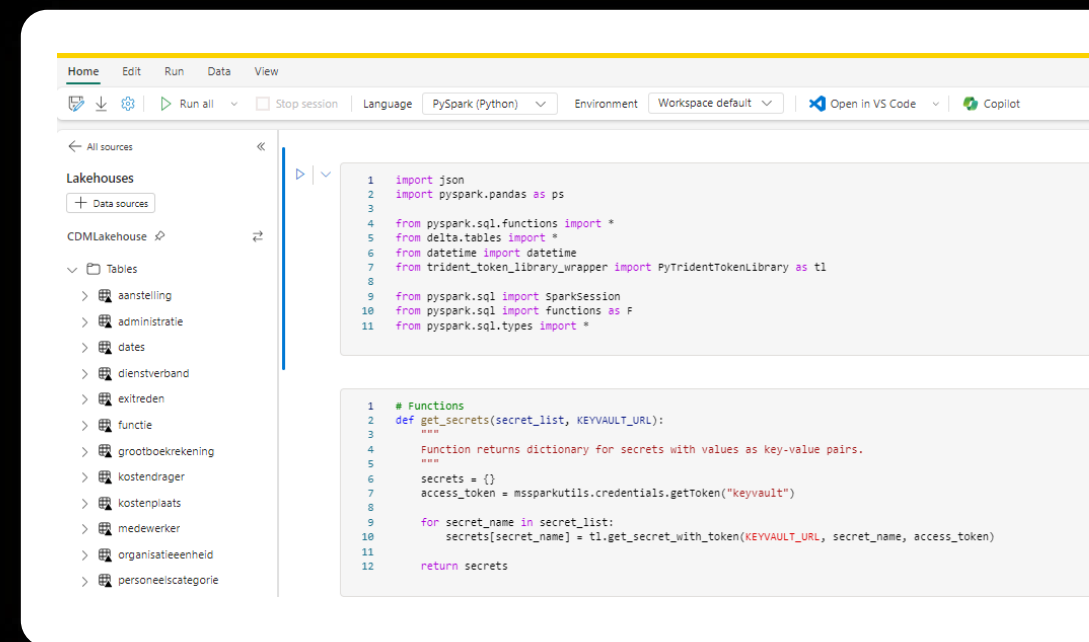
Why is notebooks relevant?

The primary goals of semantic link are to facilitate data connectivity, enable the propagation of semantic information, and seamlessly integrate with established tools used by data scientists, such as notebooks.

Power BI connectivity is at the core of semantic link. Semantic link provides connectivity to semantic models for users of the Python pandas ecosystem and the Apache Spark ecosystem.

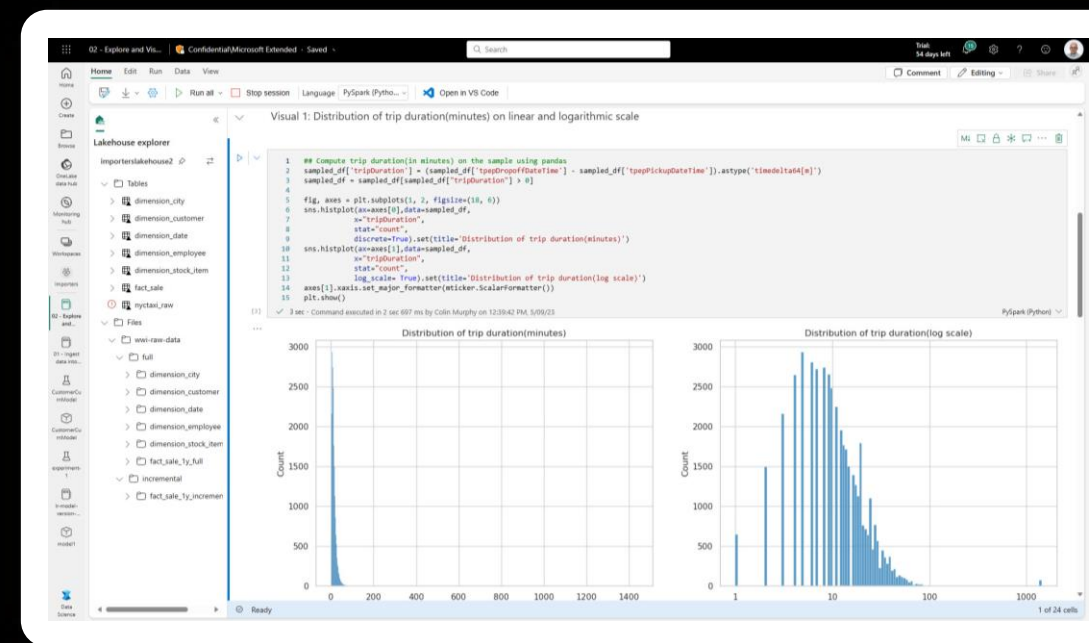
What are notebooks?

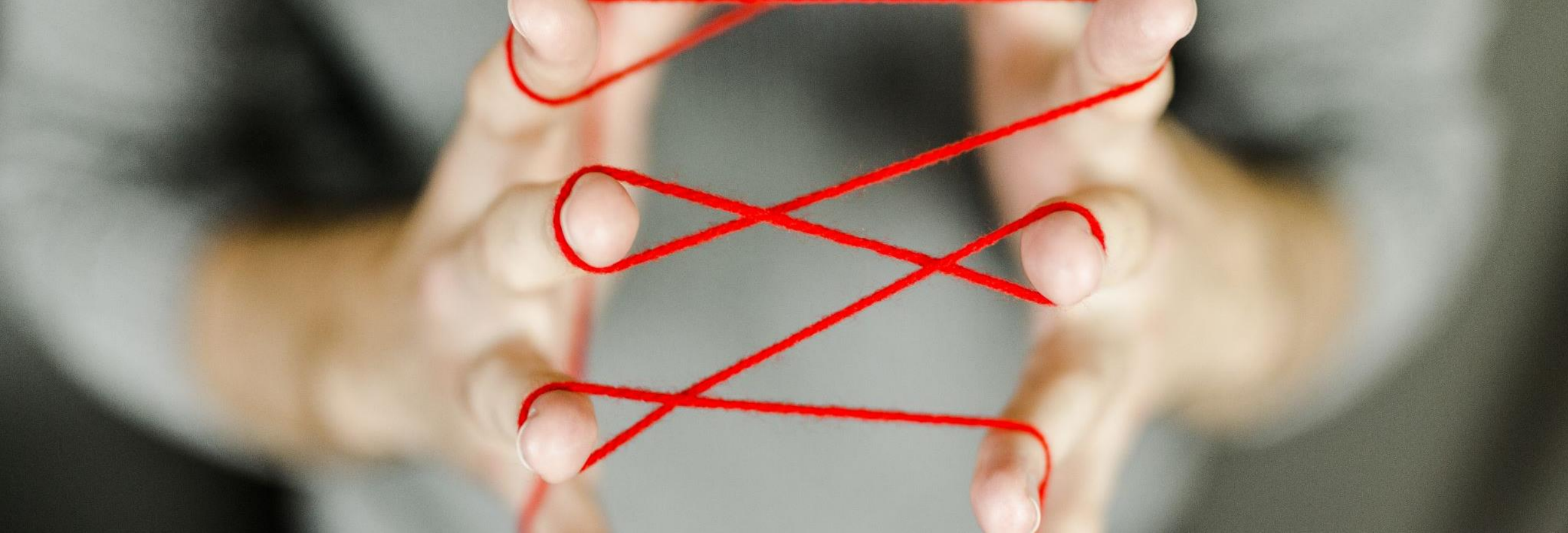
- Code first
 - Web-based interface
 - Cell based code blocks
 - Runs on nodes (part of Fabric capacity)
 - Often used languages are Python, Spark & Markdown
-
- Used by data engineers for data ingest, prep and transformations
 - Used by data scientist for experiments and models



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



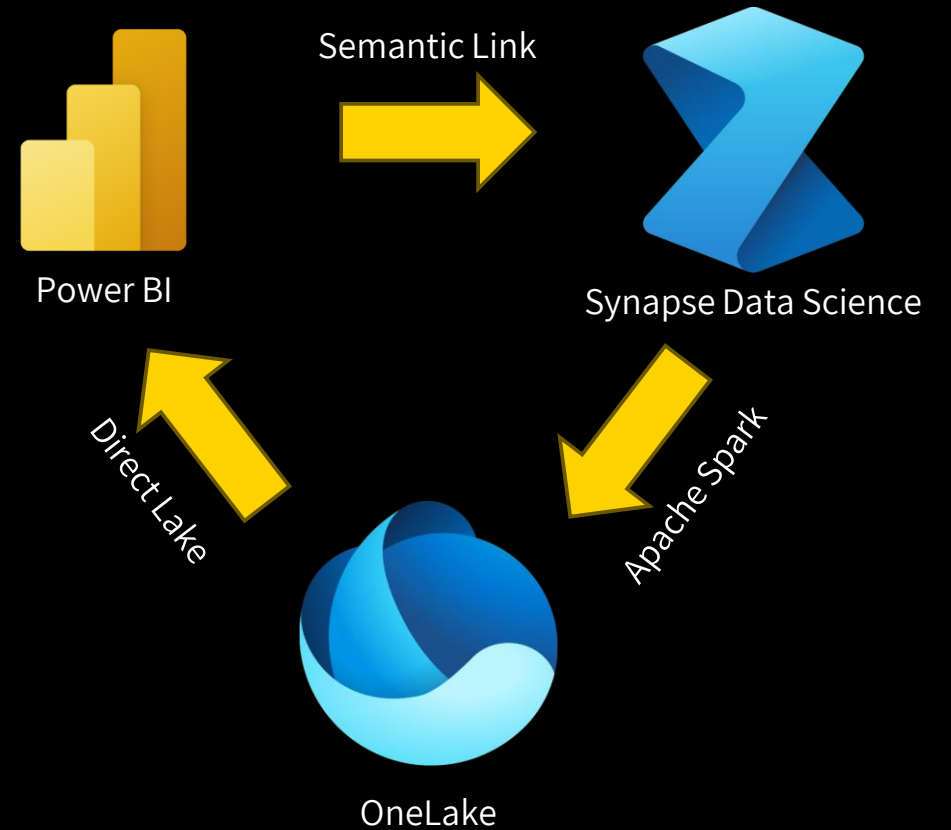


Understanding Semantic Link

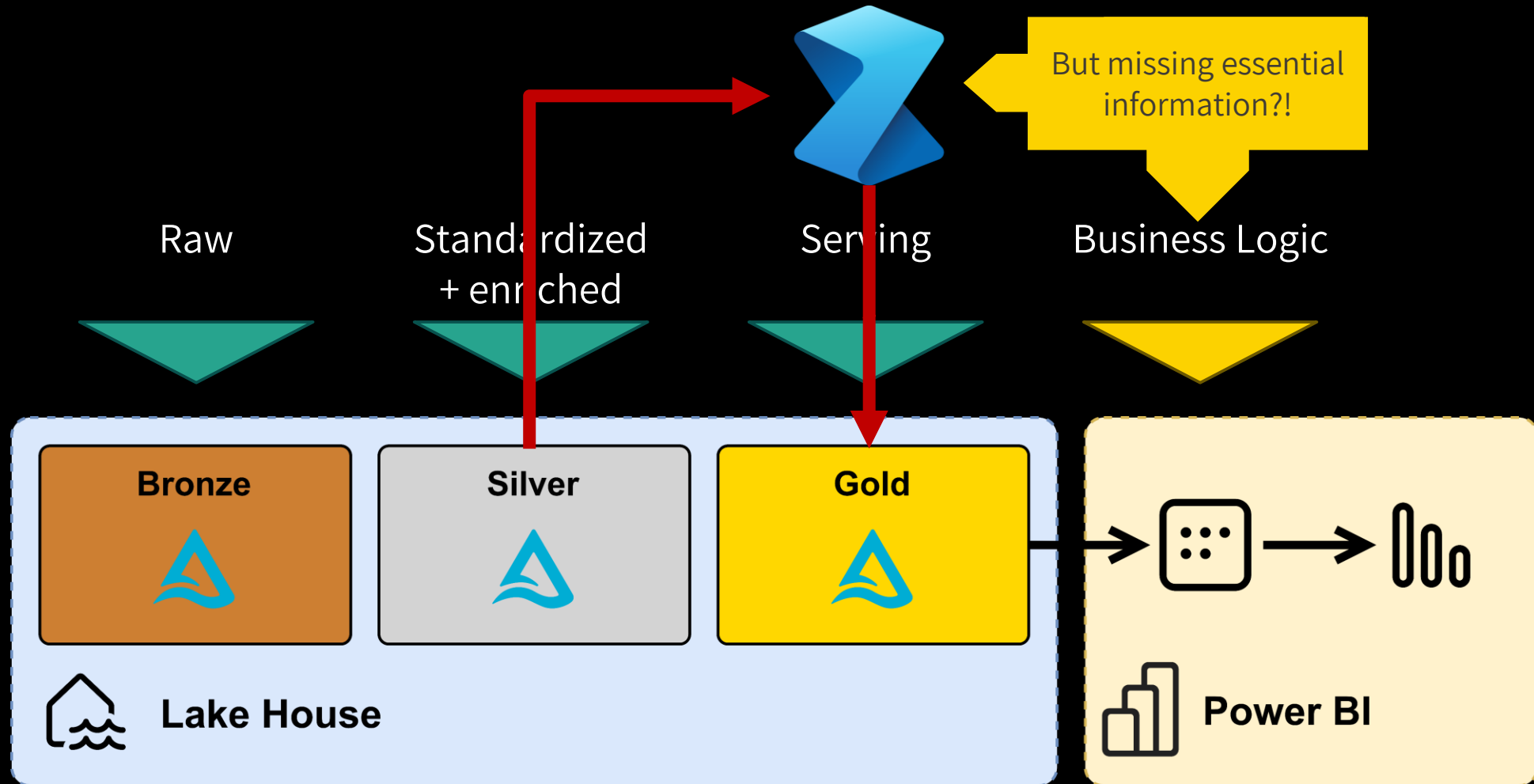
What is Semantic Link exactly?

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.

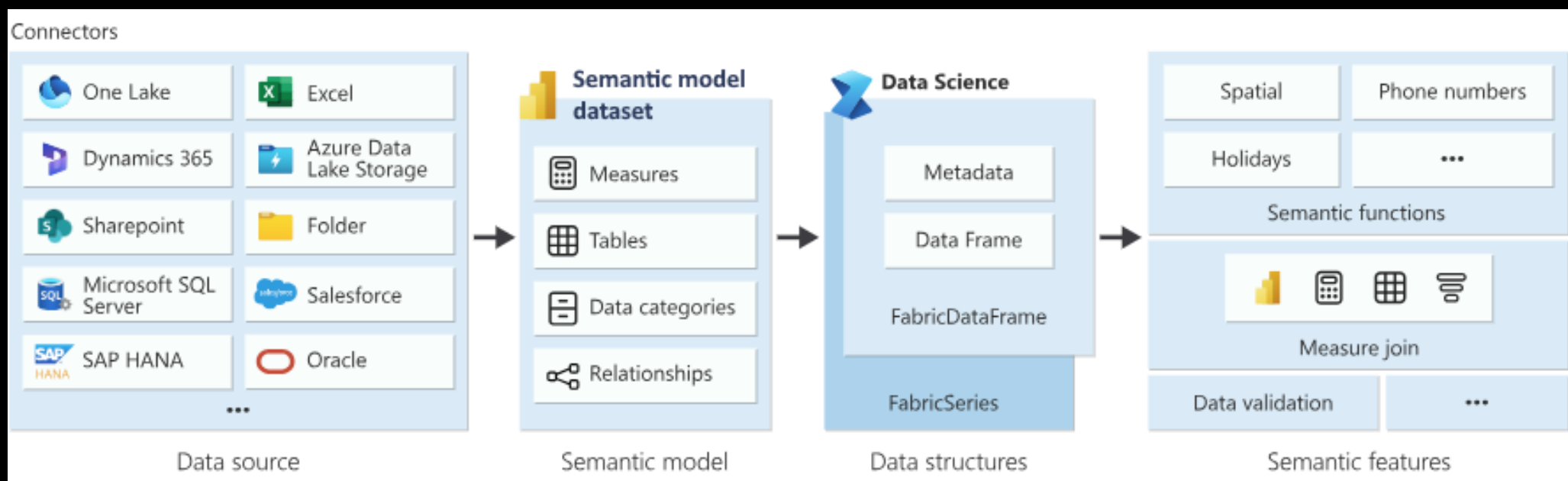


The “classic” Data Science story



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

Python

 Copy

```
%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

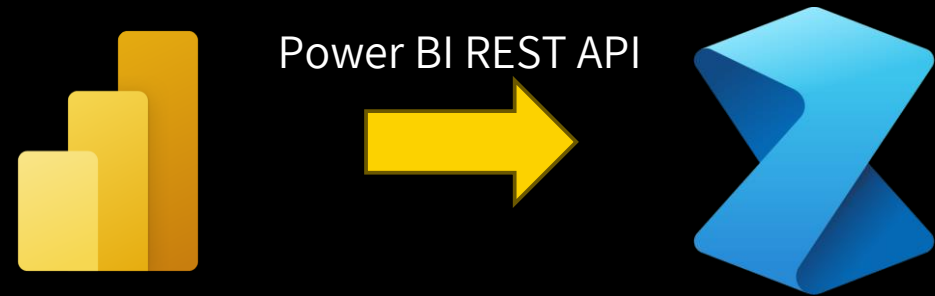
Execute DAX expressions, just like you do in Power BI Desktop, through Execute Queries REST API or in DAX Studio




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

 Copy

```
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)
```




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

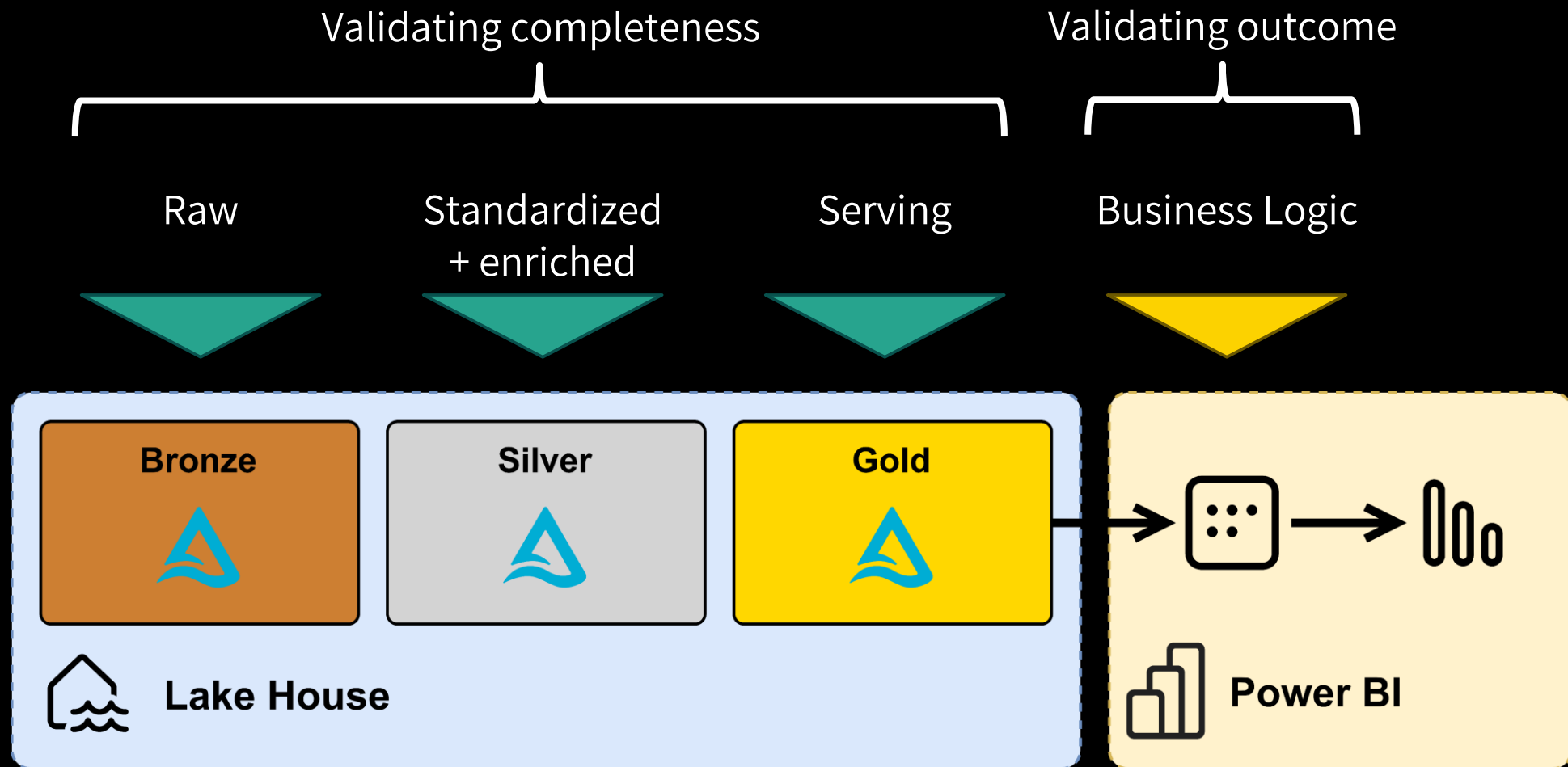
Copy

```
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 Copy

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?

```
1 from sempy.relationships import find_relationships, list_relationship_violations
2
3 tables = {
4     "FactInternetSales": fabric.read_table(dataset_name, "FactInternetSales"),
5     "DimDate": fabric.read_table(dataset_name, "DimDate"),
6     "DimProduct": fabric.read_table(dataset_name, "DimProduct"),
7 }
8 relationships = find_relationships(tables)
9
10 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

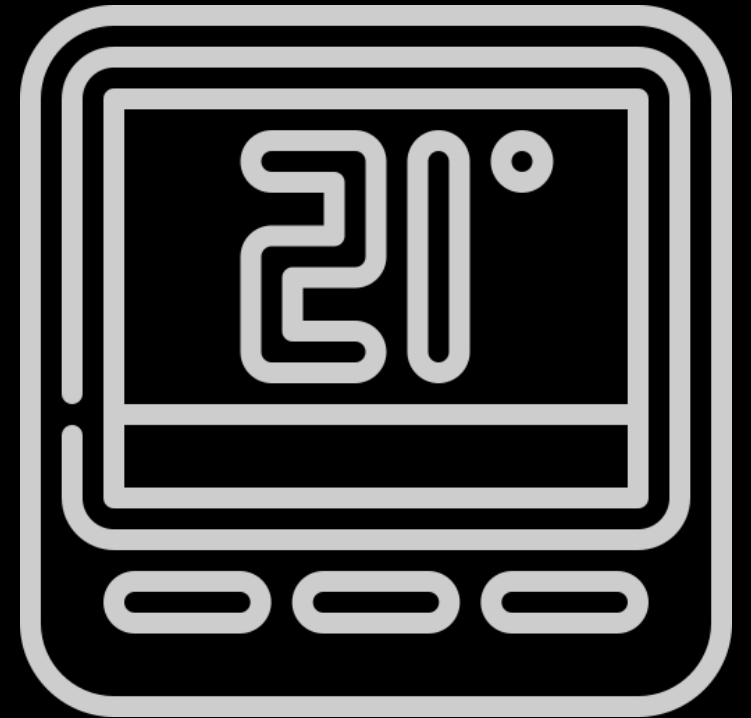
Imagine you invested a lot of time to bring together various data sources. Your Semantic Model turns into a small data warehouse solution. The ability to query data using Semantic Link opens all sorts of new options also to get your data out of Power BI again.

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.



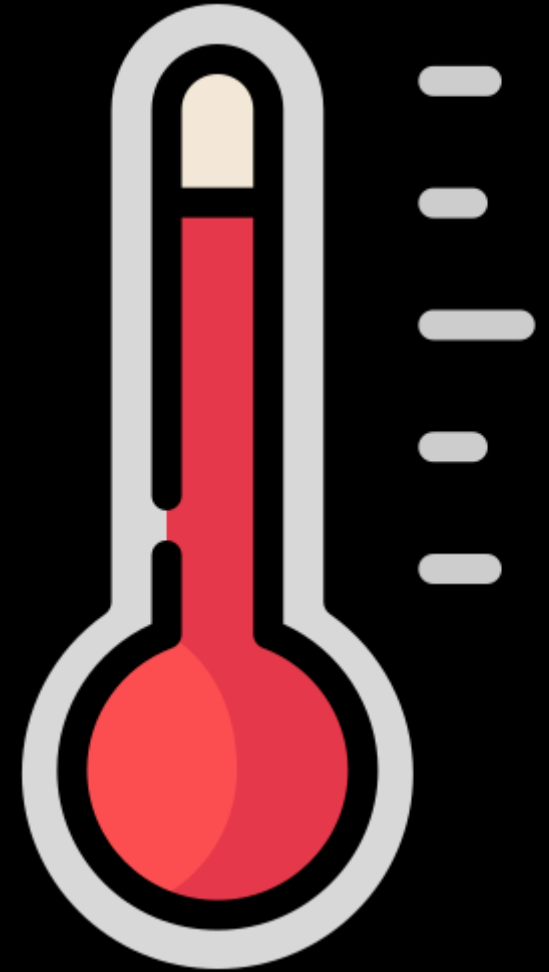
Warm-up Direct Lake Semantic Models

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**!

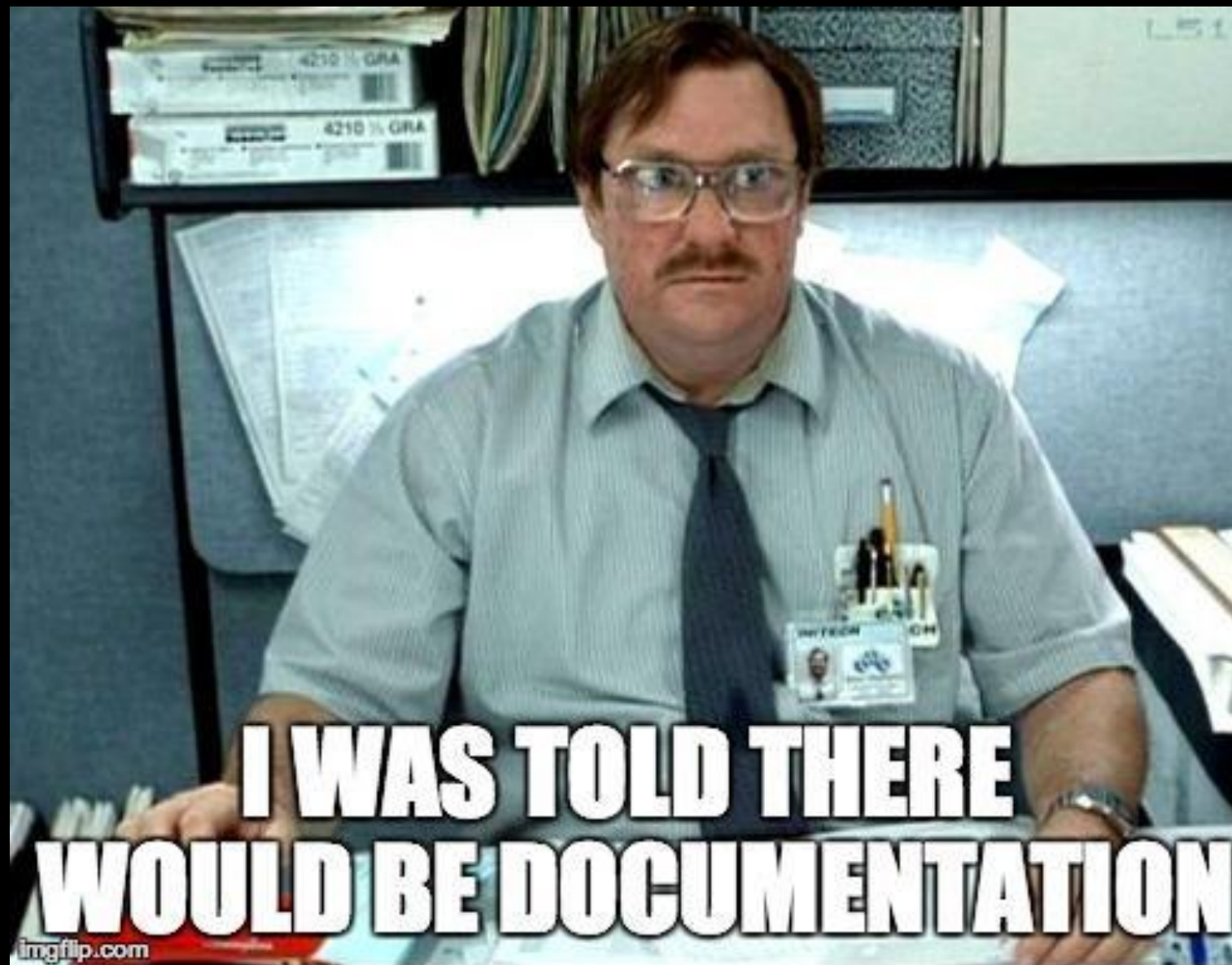




Demo – Direct Lake data
warm-up



Document your Semantic Model

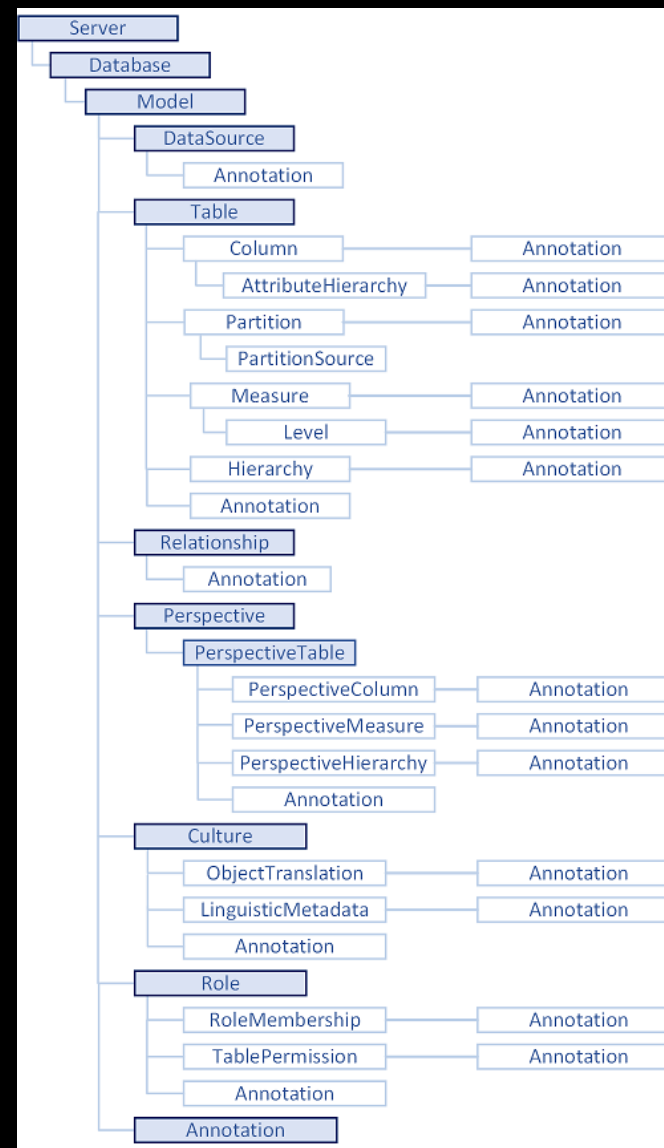


**I WAS TOLD THERE
WOULD BE DOCUMENTATION**

imgflip.com

Data model metadata

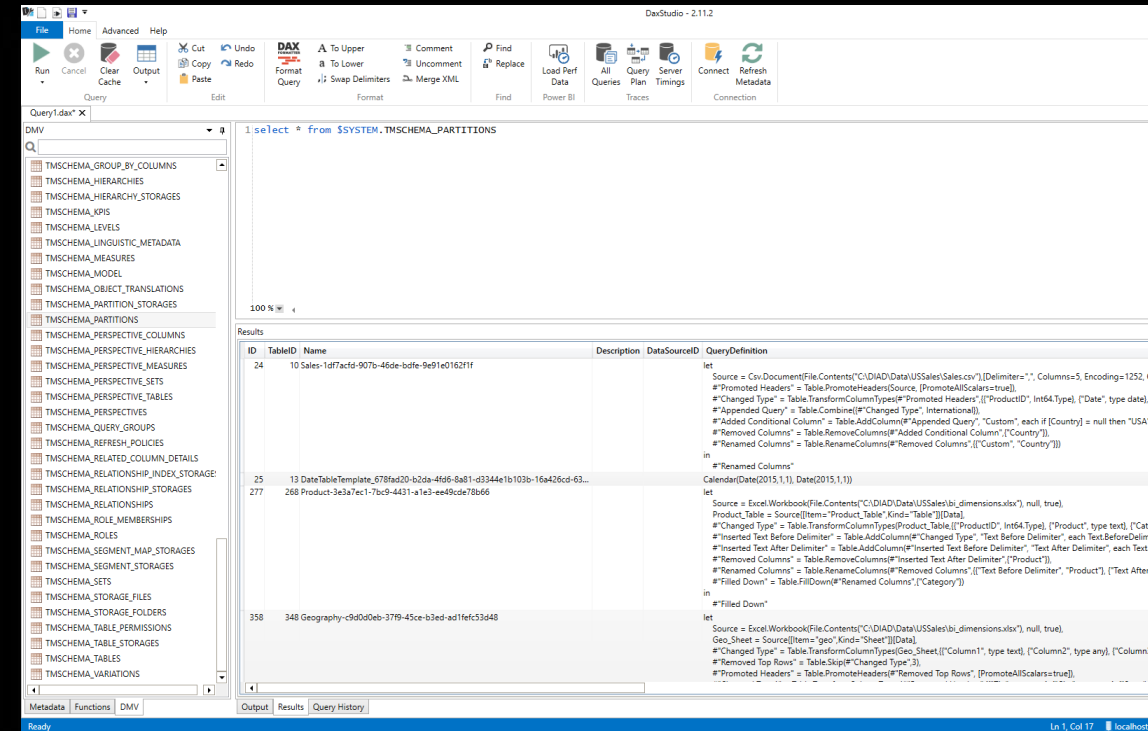
- Matches Analysis Services metadata
- Model.bim
- Tabular Object Model (TOM)
- Open format (json)
- Now, also TMDL (February 2024 update)



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



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	

36 month Rolling window

Expression

```
let
    Source = AmazonRedshiftDatabase(SourceServer,SourceDatabase,
    [BatchSize=Number.From(Text.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"
```

Relationships

12 (Blank)
Relationships # Many-to-Many

8 4
Active # Inactive

Cross Filtering Behavior

All

Security Filtering Beha...

All

Is Active

All

Rely On Ref. Integrity

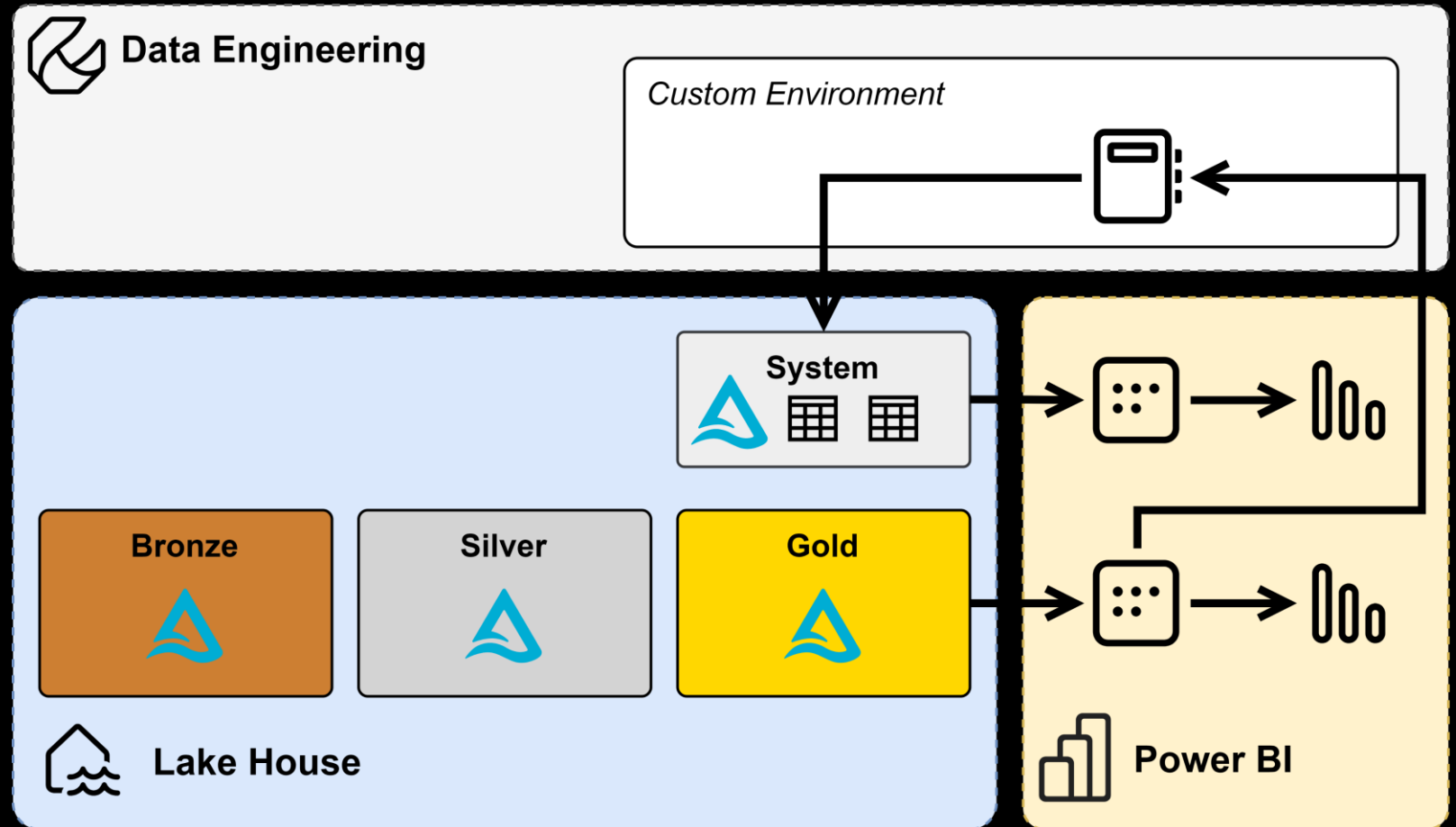
All

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

Conceptual overview

Using Semantic-Link to read Meta Data from an existing Semantic 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

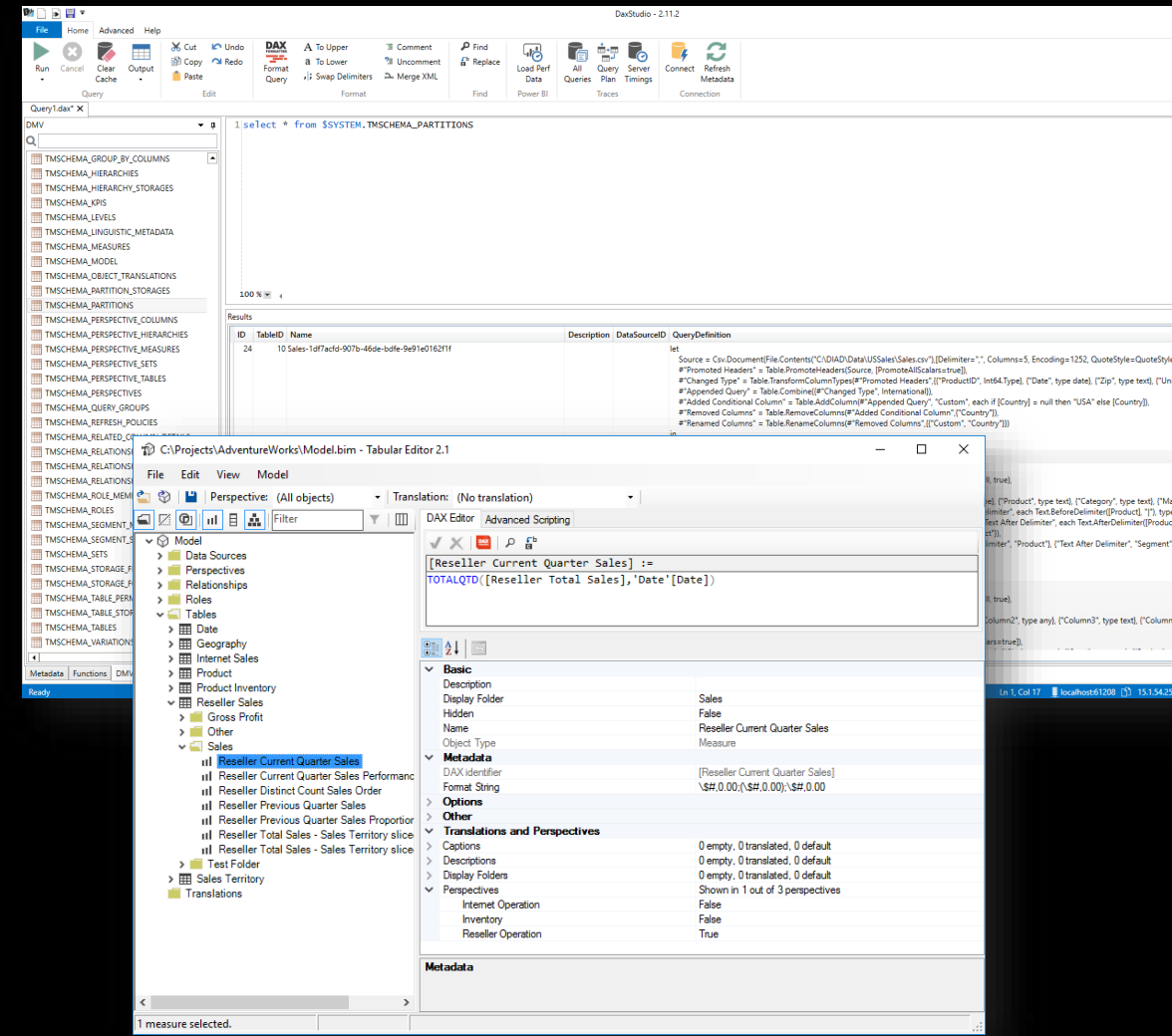


Considerations

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.



Sounds familiar?

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

HTTP

Copy

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

JSON

Copy

```
{
  "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 your 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

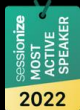
<https://github.com/marclelijveld/Slide-decks>

Thanks for attending!



Marc Lelijveld

Technical Evangelist | Architect
Macaw Netherlands



@MarcLelijveld



[linkedin.com/in/MarcLelijveld](https://www.linkedin.com/in/MarcLelijveld)



Data-Marc.com