

# Analytics at scale with Power BI and Azure Synapse



# Met dank aan onze partners

Platinum partners



Gold partners



Power BI Gebruikersdag 2022



# Met dank aan onze partners

## Silver partners



## Community partners



Power BI Gebruikersdag 2022





# Dave Ruijter

Solution Architect Data & Analytics  
Blue Rocket Data Consulting & Solutions



-  [dave@blue-rocket.it](mailto:dave@blue-rocket.it)
-  [@DaveRuijter](https://twitter.com/DaveRuijter)
-  [linkedin.com/in/DaveRuijter](https://linkedin.com/in/DaveRuijter)
-  [ModernData.ai](https://ModernData.ai)



# Marc Lelijveld

Data & Analytics consultant  
Macaw Netherlands



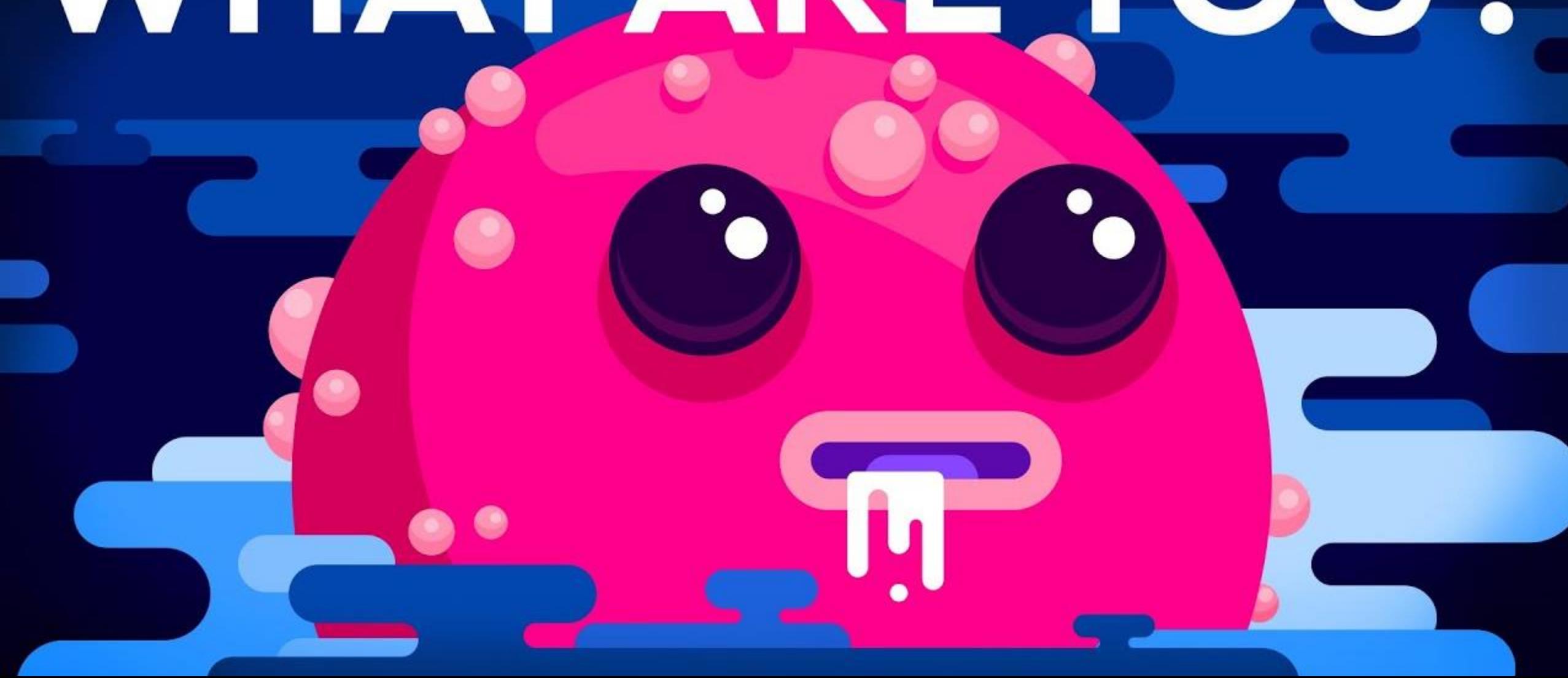
✉ marc.lelijveld@outlook.com

🐦 @MarcLelijveld

linkedin.com/in/MarcLelijveld

🌐 Data-Marc.com

# WHAT ARE YOU?





# Break times

Tea and coffee available all day

- 10:30 to 11:00 Refreshments with snacks
- 12:30 to 13:30 Lunch
- 15:30 to 16:00 Afternoon break with snacks

# Agenda

- Challenges
  - Data Platform
  - Power BI
- Deep Dive
  - Data Platform
  - Power BI
- Lunch
- Hybrid Tables & Incremental Refresh
- Refresh Challenges & Orchestration
- Backup & Restore
- Scaling
- Monitoring
- Wrap-up

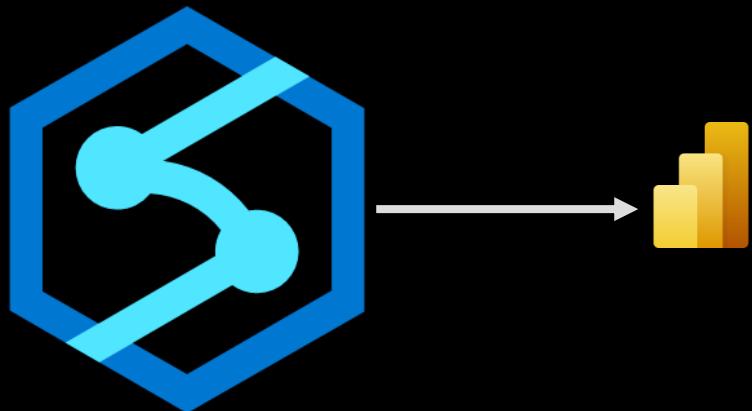
# After this session

Design and implement	Orchestrate	Performance	Cost
Better design and implement complex data models, including hybrid tables, aggregations, and combined storage modes (import, DirectQuery , dual).	Orchestrate the end-to-end data processing, with a pipeline chain from data ingest in the data lake house to the incremental Power BI dataset refresh.	Use different techniques to identify performance bottlenecks in your solutions and how to solve those ("does it fold"?).	Implement a cost-efficient solution, that still meets the scalability demands.

# Solution challenges

two separate worlds

Data Platform



Power BI







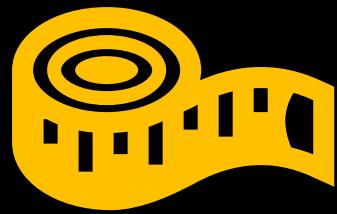
# Data Platform Solution Challenges

- Power BI report can't handle the volume of data
- Showing near real-time data in Power BI report



# Power BI report can't handle the volume

The screenshot shows the Power BI Desktop interface with the title bar "WWI Sales Model - full IMPORT - Power BI Desktop". The Home tab is selected in the ribbon. A central message box says "Unable to save document" and "Power BI Desktop ran out of memory trying to save the data model". The ribbon also displays "1bn" and "Page 1". To the right, there are sections for Filters, Visualizations, Fields, and a list of fields including City, Customer, Date, Employee, Movements, Orders, Payment Method, Purchases, Sales, Stock Holdings, and Stock Item. The status bar at the bottom right indicates "Storage Mode: Mixed".



## Demo

Power BI report can't handle the volume of data



# Should I put everything on DirectQuery instead?



# DirectQuery limitations

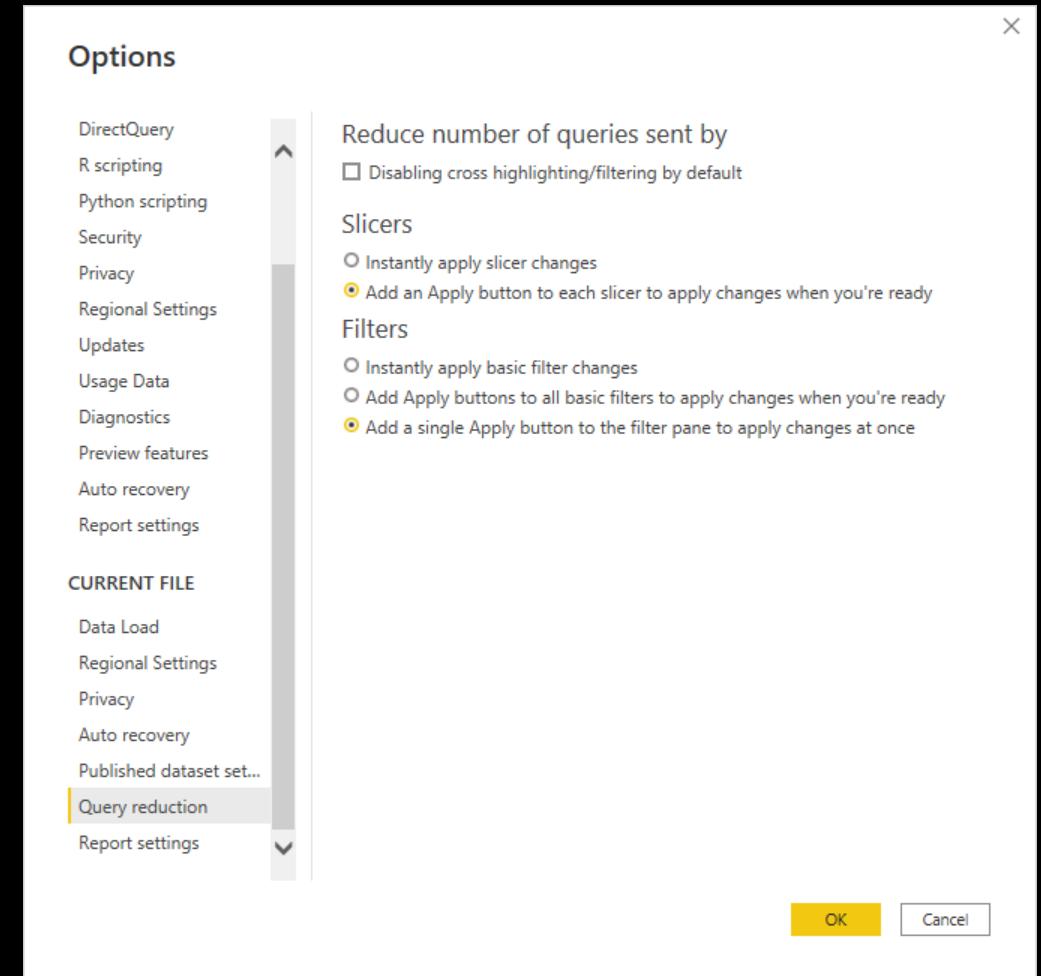
- Limited Power Query capabilities
- DirectQuery != streaming / live! Front-end still caches data
- No built-in date hierarchy (year/quarter/month/day)
- Lowest granularity data is seconds (no milliseconds)
- No parent-child support in DAX with *PATH()*
- Slower end user performance
- 1M row per query
- DAX limitations, only simple calculations possible

# DirectQuery query reduction

Consider requesting to click **Apply** before queries are executed to the source

Applies to

- Slicers
- Filters (filter pane)





Showing near real-time data in  
Power BI



# Showing near real-time data in Power BI

- Refresh takes too long
- Poor end-user performance on DirectQuery
- Streaming datasets only allow one table
- Potentially queries are not foldable, therefore incremental refresh does not work (depending on source)



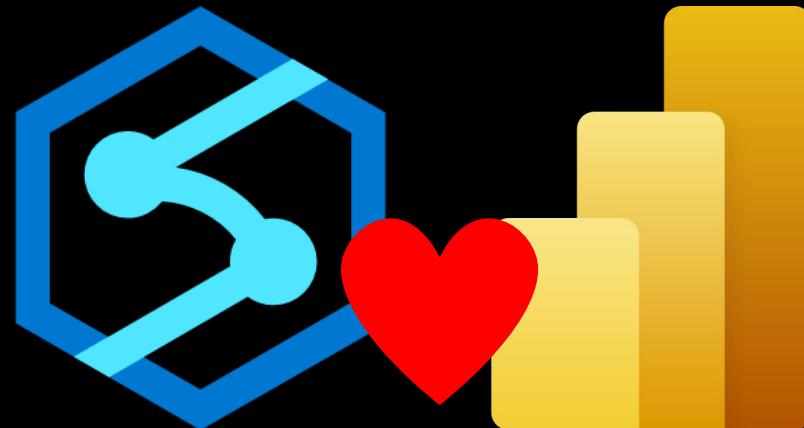
## Demo

Showing near real-time data in Power BI report



# Data Platform Solution Challenges

- Optimize Power BI model
- Use Aggregations
- Use Hybrid Table





# Power BI Solution Challenges

- Loading data from various sources (flat files, databases, APIs)
- Some sources are manually maintained (like mapping tables)
- Data stored on decentralized storages, like SharePoint pages.
- Data from source systems are exported, rather than connected to analytical systems
- Store historical data in Power BI for trend analysis





## Demo

Various manually maintained sources



OrderDate  
Last ▾ 1 Select ▾  
No filters applied

Product Category  
All ▾

CountryRegion  
 Canada  
 United Kingdom  
 United States

StateProvince, City  
 Alberta  
 Arizona  
 British Columbia  
 Brunswick

\$708,69K

\$ Sales

2.087

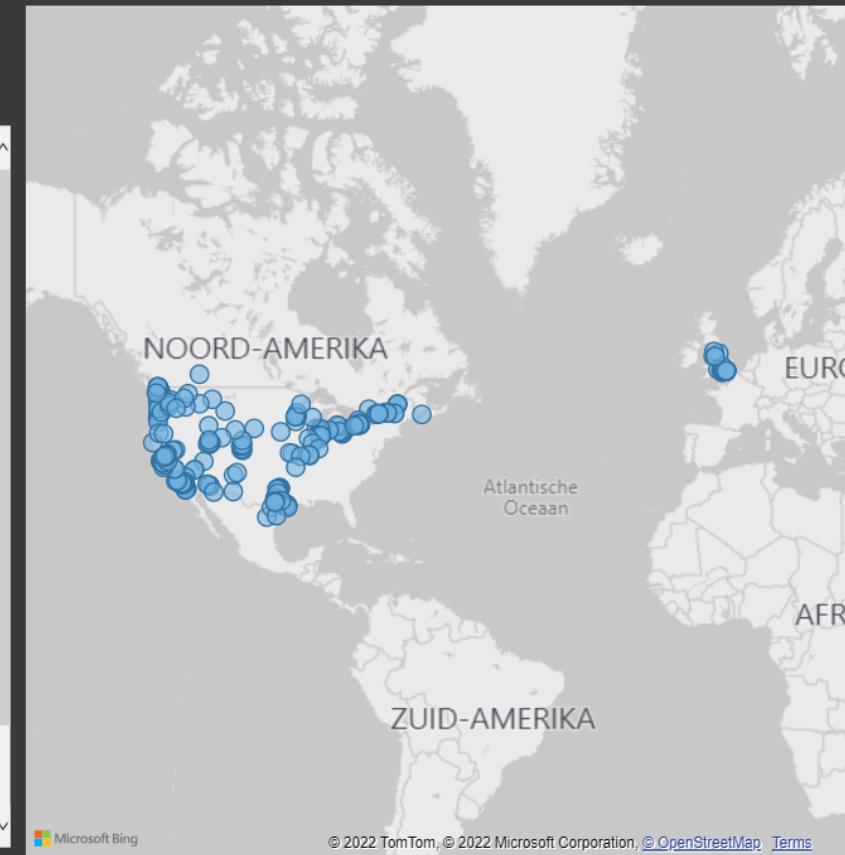
# Order quantity

32

# Orders

Product Category	# Order quantity	\$ Sales	% GT \$ Sales	\$ Price Avg per product
Touring Bikes	252	\$220.655,38	31,14%	\$875,62
Road Bikes	222	\$183.130,30	25,84%	\$824,91
Mountain Bikes	209	\$170.825,89	24,10%	\$817,35
Mountain Frames	128	\$54.949,60	7,75%	\$429,29
Road Frames	60	\$24.346,58	3,44%	\$405,78
Touring Frames	39	\$19.066,26	2,69%	\$488,88
Jerseys	230	\$7.017,88	0,99%	\$30,51
Vests	121	\$4.309,90	0,61%	\$35,62
Cranksets	22	\$3.968,87	0,56%	\$180,40
Shorts	80	\$3.299,80	0,47%	\$41,25
Pedals	84	\$2.996,50	0,42%	\$35,67
Helmets	124	\$2.523,88	0,36%	\$20,35
Bike Racks	32	\$2.304,00	0,33%	\$72,00
Hydration Packs	50	\$1.649,70	0,23%	\$32,99
Bottom Brackets	22	\$1.320,17	0,19%	\$60,01
Deraileurs	21	\$1.296,63	0,18%	\$61,74
Handlebars	27	\$1.192,97	0,17%	\$44,18
Saddles	39	\$1.010,30	0,14%	\$25,91
Gloves	57	\$837,56	0,12%	\$14,69
Brakes	13	\$830,70	0,12%	\$63,90
Socks	66	\$345,21	0,05%	\$5,23
Caps	52	\$277,36	0,04%	\$5,33
Cleaners	55	\$251,88	0,04%	\$4,58
<b>Total</b>	<b>2.087</b>	<b>\$708.690,15</b>	<b>100,00%</b>	<b>\$339,57</b>

\$ Sales by PostalCode

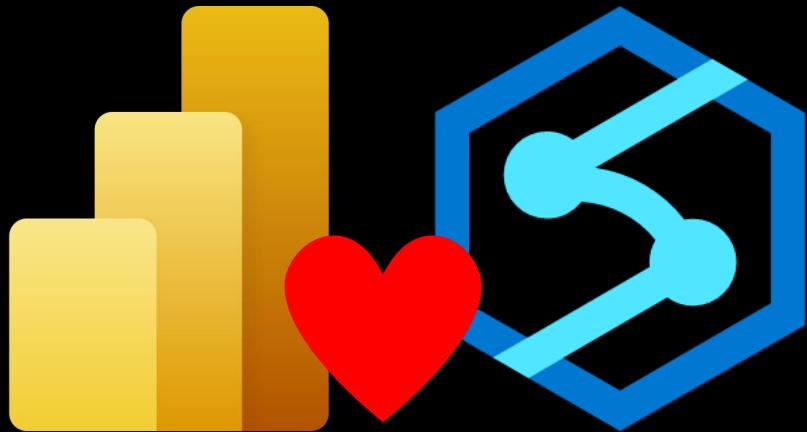


© 2022 TomTom, © 2022 Microsoft Corporation, [OpenStreetMap](#), [Terms](#)



# Power BI Solution Challenges

- Use Synapse Analytics to ingest into a data lakehouse
- Use Layered approach: Bronze, Silver & Gold



# Better together

Data platform deep dive





# Improvement areas

- Ingesting data from APIs using Synapse Analytics
- Store (historical!) data in the delta lakehouse architecture

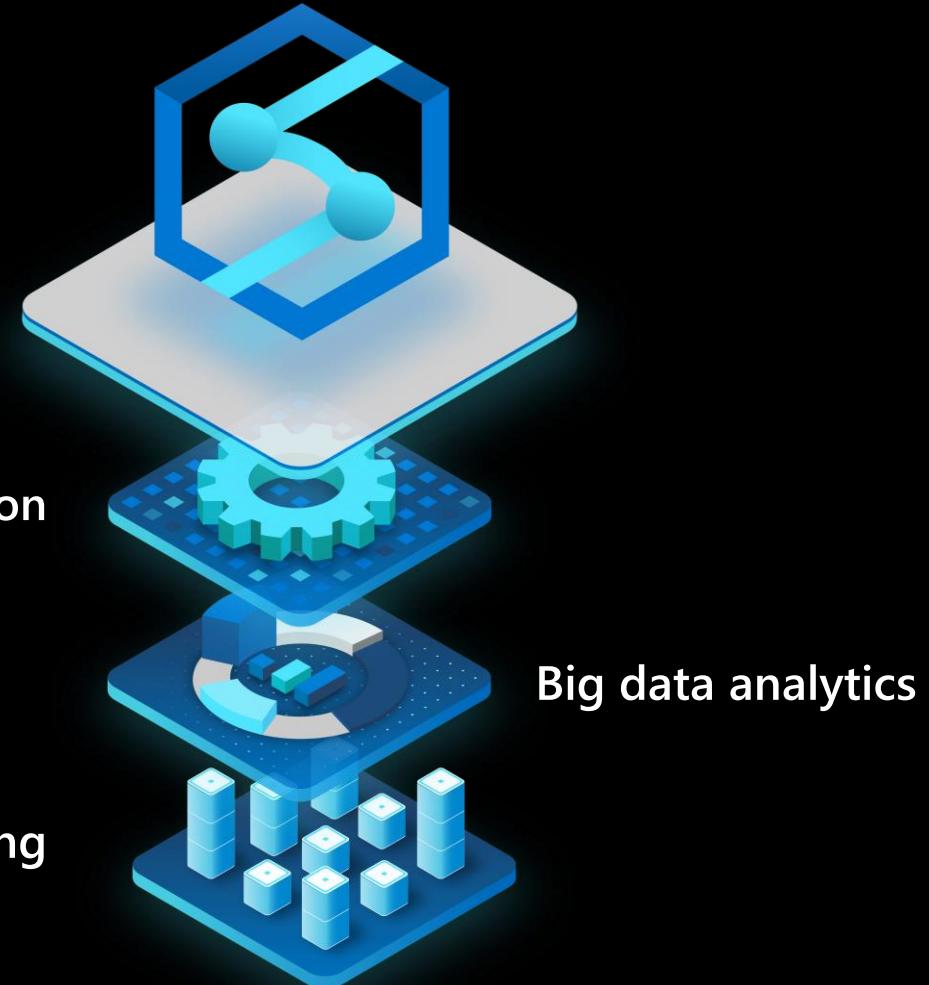
# Azure Synapse Analytics

**The first unified, cloud native platform for converged analytics**

Azure Synapse is the only unified platform for analytics, blending big data, data warehousing, and data integration into a single cloud native service for end-to-end analytics at cloud scale.

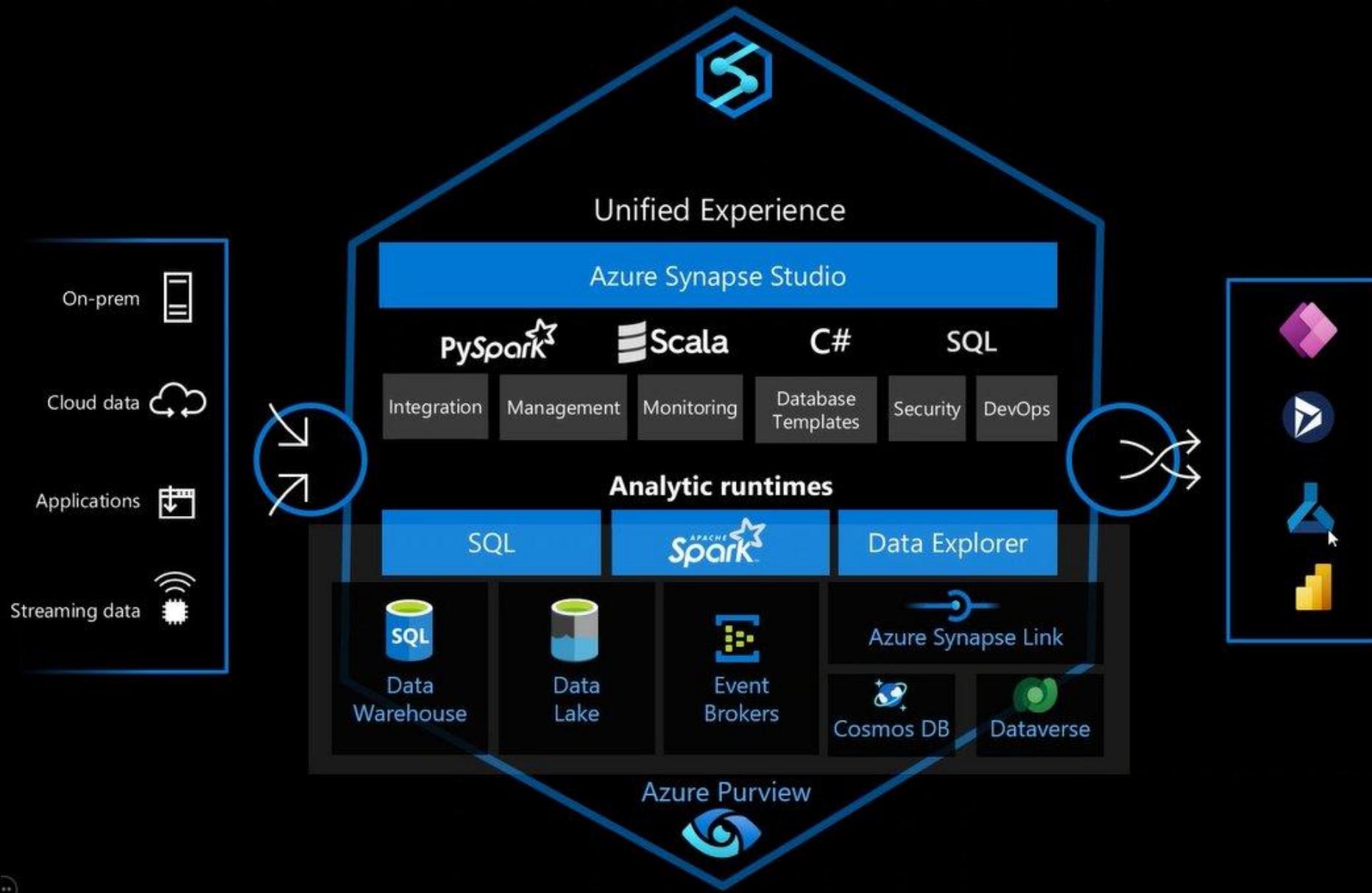
**Data integration**

**Data warehousing**



# Azure Synapse Analytics

The first unified, cloud native platform for converged analytics





# Ingesting data from APIs using Synapse Analytics

- Pipeline (same as ADF)
- Data Flow (same as ADF) *not to be confused with dataflows in PBI*
- Spark Notebook (4 languages available)
- Wrangling Dataflows (Same as ADF)



# Hybrid data integration

## Cloud native ETL/ELT

95+ connectors available

Secure connectivity to on-premise data sources, other clouds, and SaaS applications

Code-first and low/no code design interfaces

Schedule and Event based triggering





# Code-free data wrangling

## No/low-code data transformation

Excel-like interface is familiar to users

Transform data to desired shape completely visually

Operationalize into pipelines

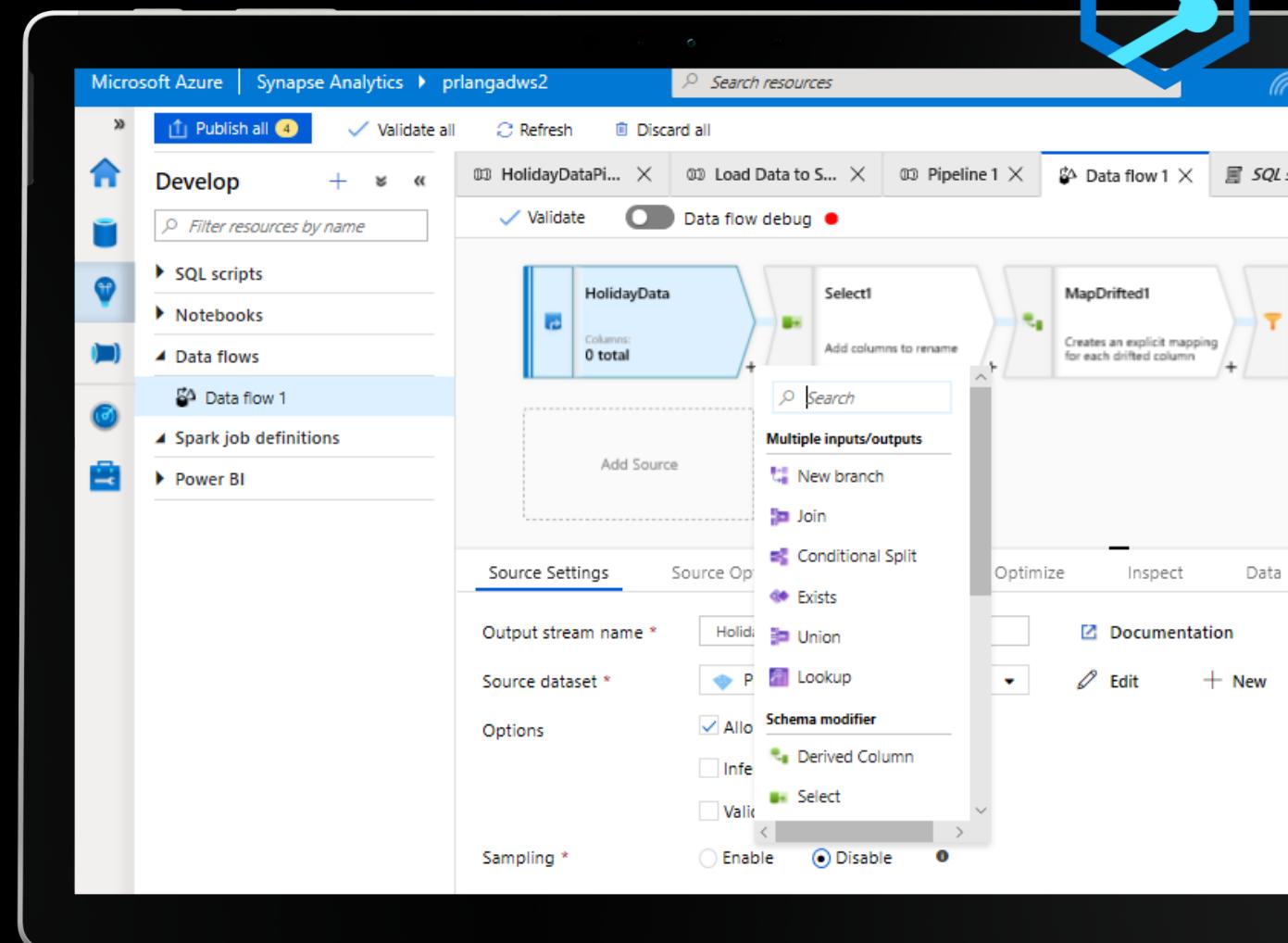
The screenshot shows the Microsoft Azure Synapse Analytics Power Query Editor interface. The top navigation bar includes 'Microsoft Azure', 'Synapse Analytics', 'wsazuresynapseanalytics', and a search bar. Below the navigation is a toolbar with various icons for validation, publishing, and discarding. The main workspace is titled 'PQSalesPrep' and shows a 'Settings' section with a note about Power Query M functions. The 'Home' tab is selected, displaying a ribbon of tools: Enter data, Options, Manage parameters, Refresh, Advanced editor, Properties, Choose columns, Remove columns, Manage columns, Keep rows, Remove rows, Reduce rows, Sort, Split column, Group by, Replace values, and Transform. To the right of the ribbon is a data preview table with columns: ab storeId, ab productCode, 12 quantity, 1.2 logQuantity, ab advertising, ab price, ab weekStarting, and ab id. The table contains 17 rows of data related to surface.go products. At the bottom of the table, it says 'Columns: 8 Rows: 99+'. The overall interface is clean and modern, designed for data wrangling and transformation.

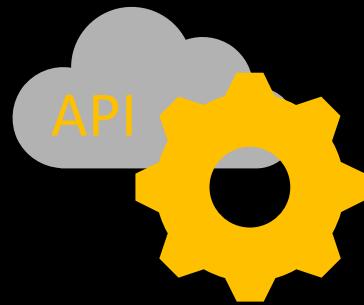


# Hybrid data integration

## Data Flows

*not to be confused with dataflows in Power BI*





## — Demo API ingest



# Store (historical!) data in the lake

- The Layered approach: Bronze, Silver & Gold
- Keep original raw data, build up history -> **bronze**
- Cleanse and refine data, standard file format -> **silver**
- Aggregate, prepare, transform, merge, make start schema -> **gold**



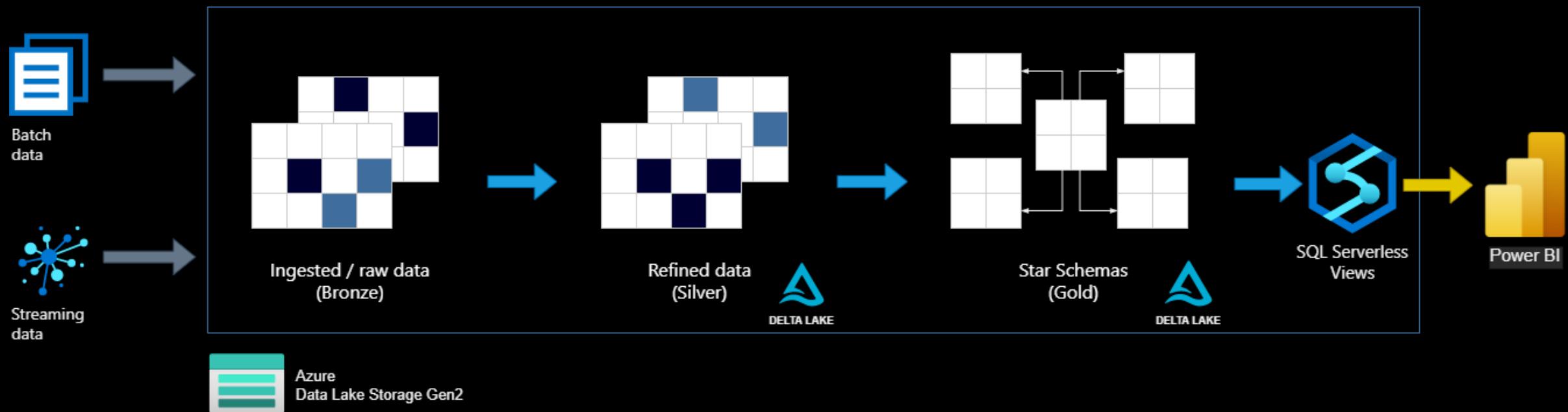
# Store (historical!) data in the lake

- The Layered approach:  
Bronze, Silver & Gold
- Keep original raw data, build up  
history -> **bronze**
- Cleanse and refine data, standard file  
format -> **silver**
- Aggregate, prepare, transform, merge,  
make start schema -> **gold**

Bronze	Silver	Gold
Raw data	Apply metadata	Implement business rules
All history, system replayable	Protect data (GDPR)	Fit for purpose
	Current & historical view	



# Using Delta Lakehouse with layered approach





# Warm-up time of Serverless SQL pools

## SQL requests

Refresh Edit columns

Filter by keyword

Local time : **Last 24 hours**

Status : **All**

Pool : **Built-in**

Add filter

Showing 1 - 4 of 4 items

Request ID ↑	Request content ↑	Submit time ↑	Duration	Data processed
44196195	SELECT TOP (1000... <a href="#">More</a>	3/6/22, 10:45:14 AM	14 sec	5.12 GB
44190548	SELECT TOP (1000... <a href="#">More</a>	3/6/22, 10:43:13 AM	16 sec	4.34 GB
44187683	SELECT TOP (1000... <a href="#">More</a>	3/6/22, 10:42:32 AM	14 sec	6.12 GB
44178591	SELECT TOP (1000... <a href="#">More</a>	3/6/22, 10:39:55 AM	20 sec	6.11 GB

Sales Territory	\$ Sales
External	\$12,540,565,587.63
Far West	\$130,724,597,536.11
Great Lakes	\$137,062,180,086.18
Mideast	\$170,472,095,945.05
New England	\$58,853,530,841.09
Plains	\$154,405,133,753.92
Rocky Mountain	\$72,387,209,160.70
Southeast	\$258,110,199,285.25
Southwest	\$141,628,285,606.20
Total	\$1,136,183,797,802.13



# Best practices for serverless SQL pools

- Azure AD Pass-through Authentication performance <= shared access signature credentials
- Colocate
- Same region
- Convert large CSV and JSON files to Parquet
- Try to optimize storage layout by using partitioning and keeping your files in the range between 100 MB and 10 GB
- Use appropriate data types (smallest, integer-based, varchar)
- Use filename and filepath functions to target specific partitions



# Demo Data Platform Lakehouse architecture

# Better together

Power BI deep dive





# Improvement areas

- Data model
- Query Folding
- Aggregations
- Storage modes
- Hybrid tables



# But before we start changing our solution, let's measure...

- Refresh durations
- Model Size
- Vertipaq Analyzer
- Performance Analyzer
- Query folding applied?

Perfect E2E.pbix			
Total Size <b>74,89 MB</b>	Last Data Refresh <span>1-3-2022 20:05:54 +01:00</span>	Analysis Date <span>1-3-2022 20:05:55 +01:00</span>	
Compatibility 1550	Tables 7	Columns 124	Server localhost:60032



# Performance analyzer in Power BI Desktop

The screenshot shows the 'Performance analyzer' window in Power BI Desktop. It lists various tasks and their execution times in milliseconds (ms). The tasks include '0.0%', 'Simple Image', 'Net Sales vs "What If" Analysis', 'OneNote', 'What If...', 'Return Rate', 'Net Sales (Forecast)', 'Extra Profit', 'Card', 'Returns', 'OneNote', 'Button', 'Last Refresh: Jun 30th, 2019 / ...', '\$30,772', '+17.1%', '1715', and 'What If" Analysis Forecast'. The 'Duration (ms)' column is sorted in descending order, with the first few items being the longest. A red box highlights the 'Duration (ms)' column header.

Name	Duration (ms)
0.0%	2279
Simple Image	1440
Net Sales vs "What If" Analysis	4331
OneNote	2391
"What If" Analysis Forecast	2543
Changed a slicer	50
What If...	-
What If...	128
Return Rate	1028
Net Sales (Forecast)	1618
Extra Profit	2046
Card	1425
"What If" Analysis Forecast	1890
Returns	1723
OneNote	2249
	1155
	1723
	1722
	1723
Button	398
Last Refresh: Jun 30th, 2019 / ...	397
\$30,772	2107
+17.1%	1528
	1715
"What If" Analysis Forecast	388
Simple Image	2721

Learn more about optimizing your report's performance on our [support site](#). Find help tuning your report from specialist Power BI partners on [AppSource](#).

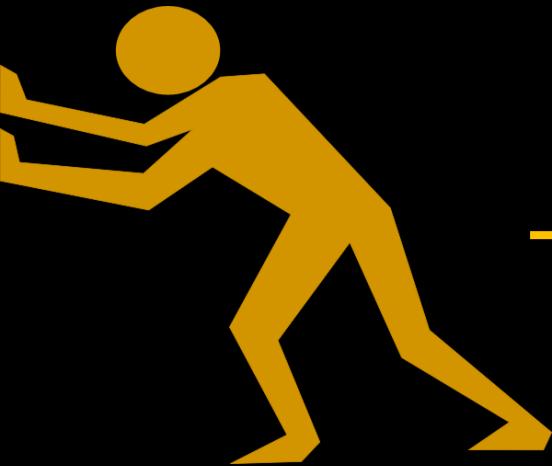
- DAX Query
- Visual Display
- Other
  - Preparing queries
  - Waiting for other visuals to complete
  - Other background processes



# Vertipaq analyzer

See where your data volume is

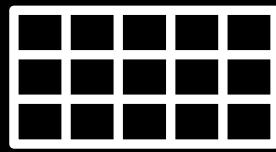
VertiPaq Analyzer Metrics									
Tables		Cardinality	Table Size	Col Size	Data	Dictionary	Hier Size	Encoding	Data Type
↳ Sales Agg		1.933.444	55.428.208	55.422.984	29.854.056	21.688.736	3.880.192	Many	-
Total Including Tax		119.940	55.428.208	11.252.400	4.973.680	5.319.152	959.568	HASH	Double
Tax Amount		119.137	55.428.208	11.241.312	4.972.888	5.315.288	953.136	HASH	Double
Total Excluding Tax		118.952	55.428.208	11.241.120	4.973.680	5.315.776	951.664	HASH	Double
Profit		113.830	55.428.208	11.139.328	4.975.824	5.252.816	910.688	HASH	Double
Invoice Date Key		1.444	55.428.208	3.109.056	3.013.344	84.112	11.600	HASH	DateTime
Delivery Date Key		1.443	55.428.208	3.109.016	3.013.344	84.072	11.600	HASH	DateTime
Count Rows		9.722	55.428.208	2.356.128	1.974.104	304.200	77.824	HASH	Int64
Customer Key		403	55.428.208	1.929.700	1.916.016	10.420	3.264	HASH	Int64
Salesperson Key		101	55.428.208	44.804	41.176	2.780	848	HASH	Int64
RowNumber-2662979B-1795-4F74-8F37-6A1BA8059B61		0	55.428.208	120	0	120	0	VALUE	Int64
↳ Date		3.287	23.045.462	22.981.430	73.096	22.805.342	102.992	Many	-
↳ Customer		403	6.436.260	6.436.260	2.448	6.419.380	14.432	Many	-
↳ Employee		213	2.157.356	2.157.356	1.120	2.151.460	4.776	Many	-
↳ StockItem		0	8.576	8.576	160	8.416	0	HASH	-
↳ Sales		0	8.392	8.352	176	8.176	0	HASH	-
↳ sales_model Employee		0	8.352	8.352	176	8.176	0	HASH	-
↳ City		0	6.496	6.496	112	6.384	0	HASH	-



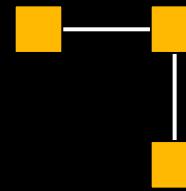
# — Performance & Vertipaq Analyzer



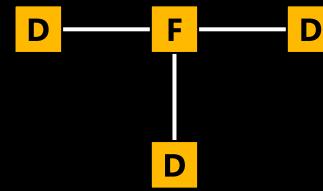
# Care about the data model!



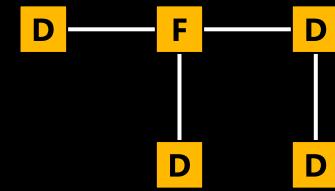
Flat



Normalized  
(n<sup>th</sup> normal form)



Star



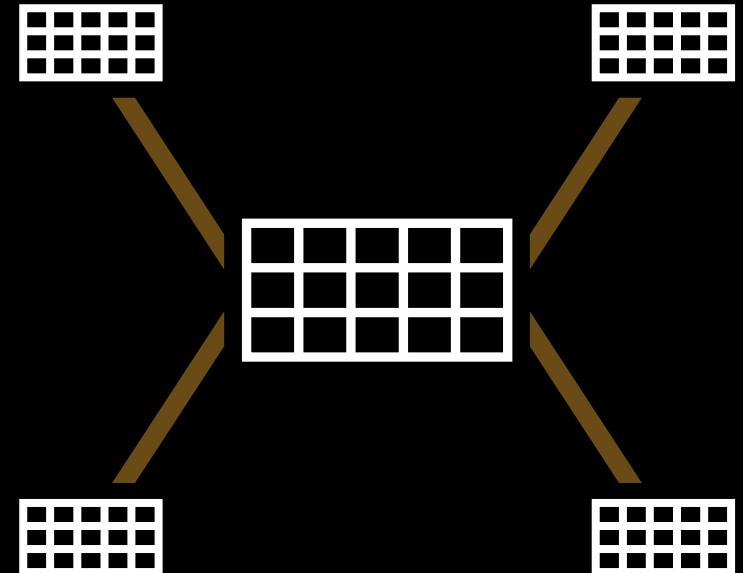
Snowflake



# Star schema all the things!

## Facts

- Contains **numerical information** about a business process or items to be aggregated
- Aggregations provide totals, averages, etc.  
Power BI implements these using **Measures**
- Usefulness limited without context  
Context is provided by **dimensions** that slice the data



## Dimensions

- Contains **descriptive information** that define how a fact should roll up.
- Examples: Date, Month, Customer, Geography, Product, Payment type.
- Without dimensions there is no context.
- Also called: Lookup table on steroids.



# Care about the data model

	Flat table	Normalized	Star schema	Snowflake
Performance for analytics	Low	Medium	High	High
Development effort	Low	High	High	High
Query volume and complexity	Low volume Low complexity	High volume Low complexity	Low volume High complexity	Low volume High complexity
Intended for	No database	CRM / ERP / Applications	Analytical systems / data warehouses	Analytical systems / data warehouses
Compression	Row	Row	Column	Column



# Aggregations

## Benefits

- Report visualizations are faster
- Balanced architecture by combining DirectQuery and Import storage modes

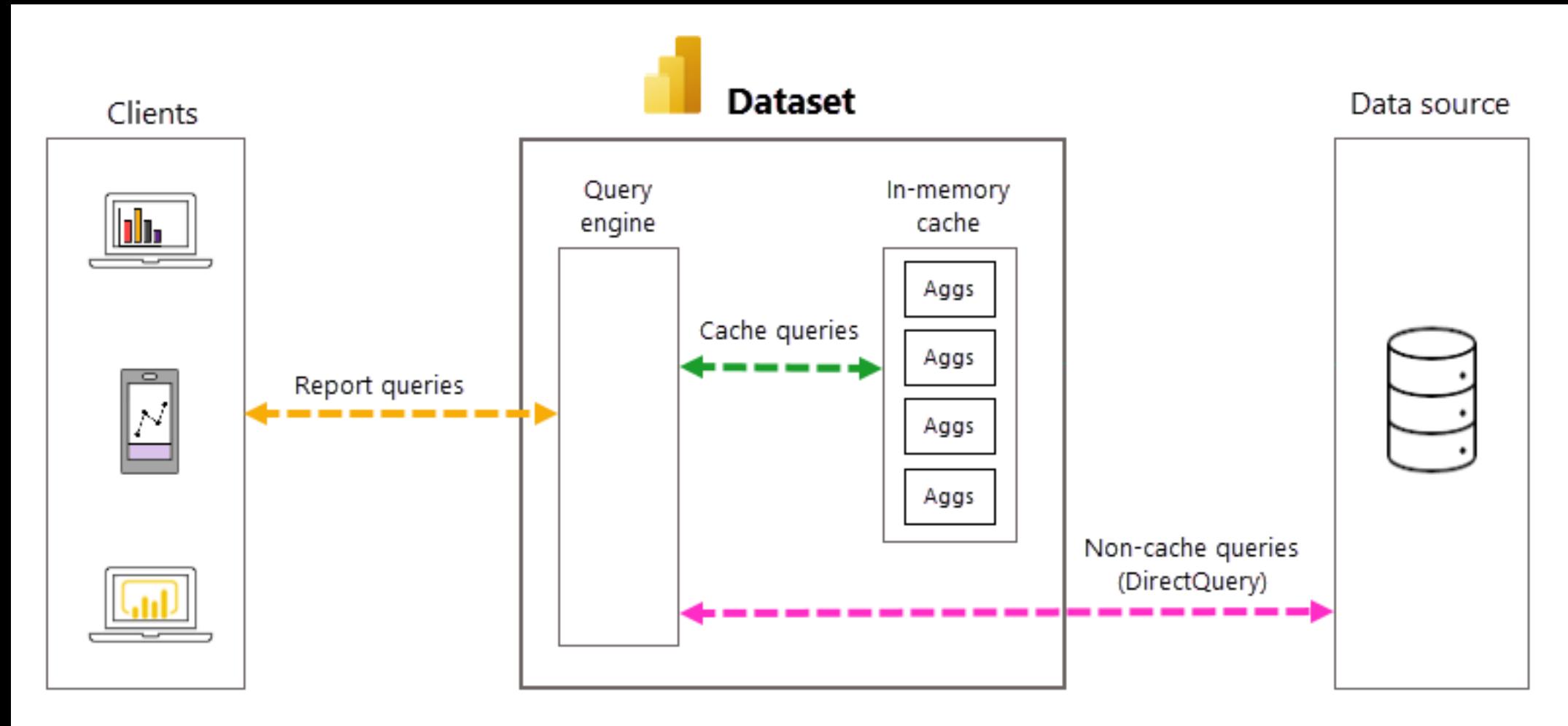
**Store data at a higher level of granularity than the original table**

The following aggregations are available:  
count, groupby, max, min, sum, and count table rows

- Aggregated data is stored in-memory (imported), details are accessed through DirectQuery
- You can create the aggregated table in the Data Transformations (Power Query) or in your source (preferred)



# Aggregations





# Aggregations

Aggregation, that hit based on relationships, require *regular* relationships.

Regular relationships include the following storage mode combinations, where both tables are from a single source group:

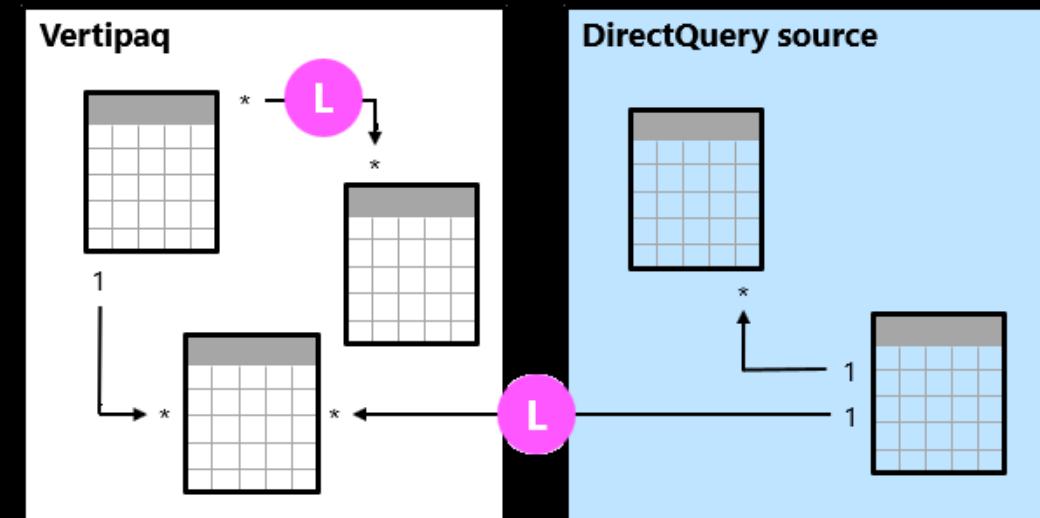
Table on the many side	Table on the 1 side
Dual	Dual
Import	Import or Dual
DirectQuery	DirectQuery or Dual



# Relationships and storage modes

A model relationship is **limited** when there's no guaranteed "one" side. It can be the case for three reasons:

- The relationship uses a Many-to-many cardinality type (even if one or both columns contain unique values)
- The storage mode combination is Import and DirectQuery
- The relationship is cross source group





# Impact of limited relationships

Cross source group relationships have performance implications.

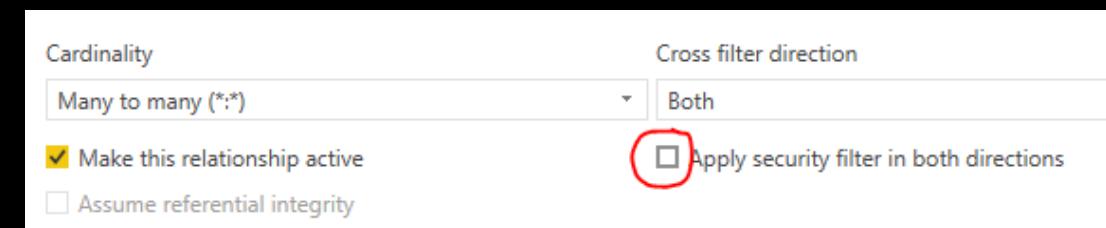
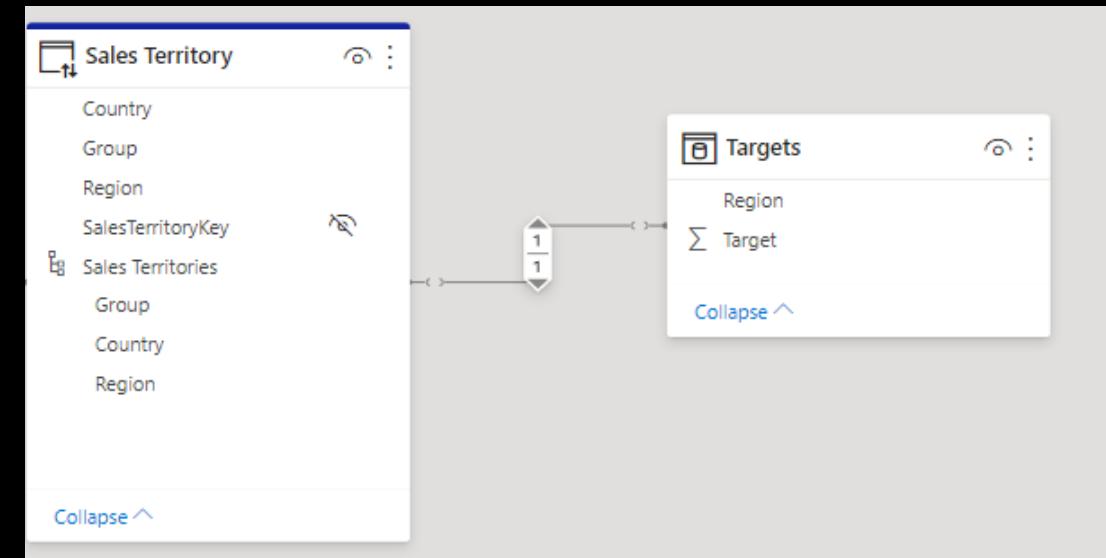
Limited optimization: joins are done on-demand for DirectQuery.

No blank rows: table joins are achieved by using INNER JOIN.

- Blank rows are not added for referential integrity violations

## Additional restrictions:

- RELATED DAX function cannot be used to retrieve the 'one' side of the relationship
- Enforcing RLS requires you to check the following checkbox





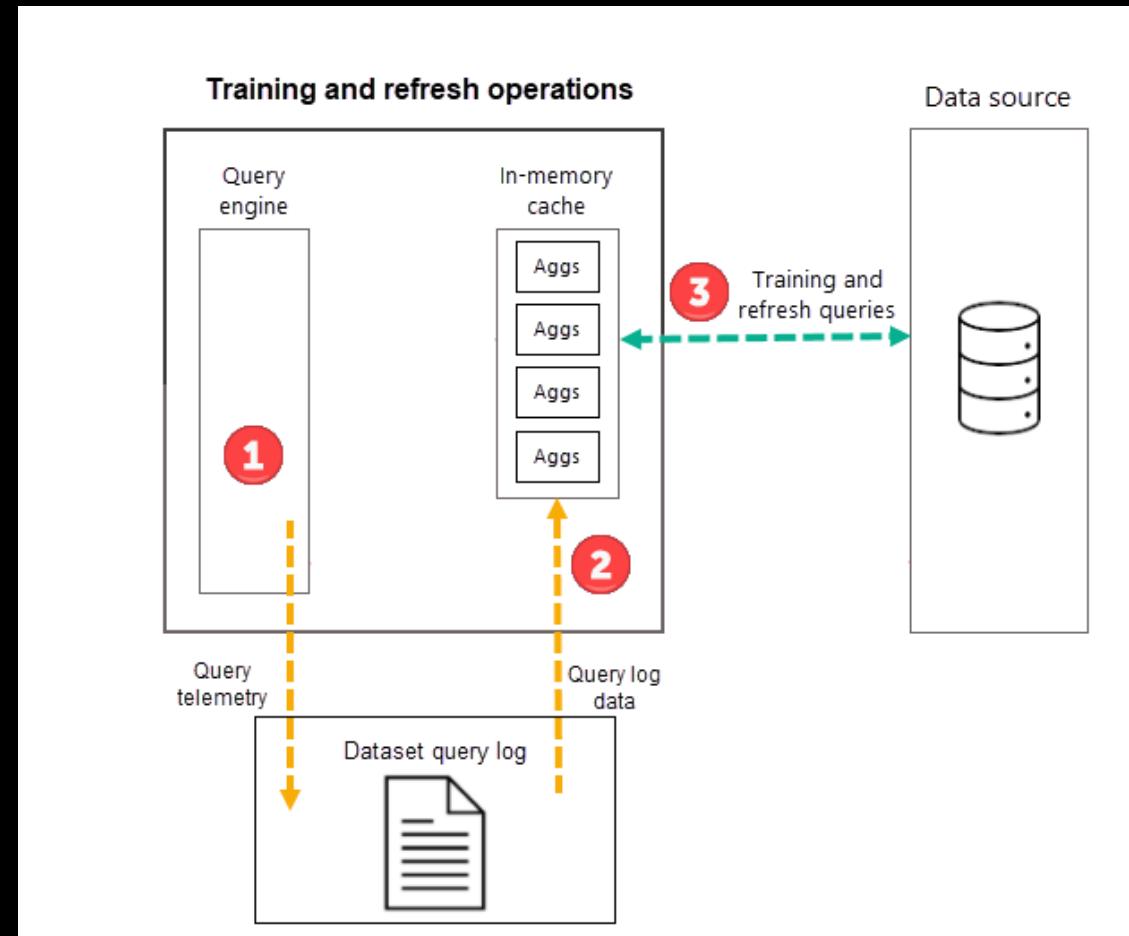
# Automatic Aggregations

**Power BI Premium per User, Premium Capacity and Embedded datasets**

**Automatic aggregations based on Query logs (7 days)**

**Supported sources during preview:**

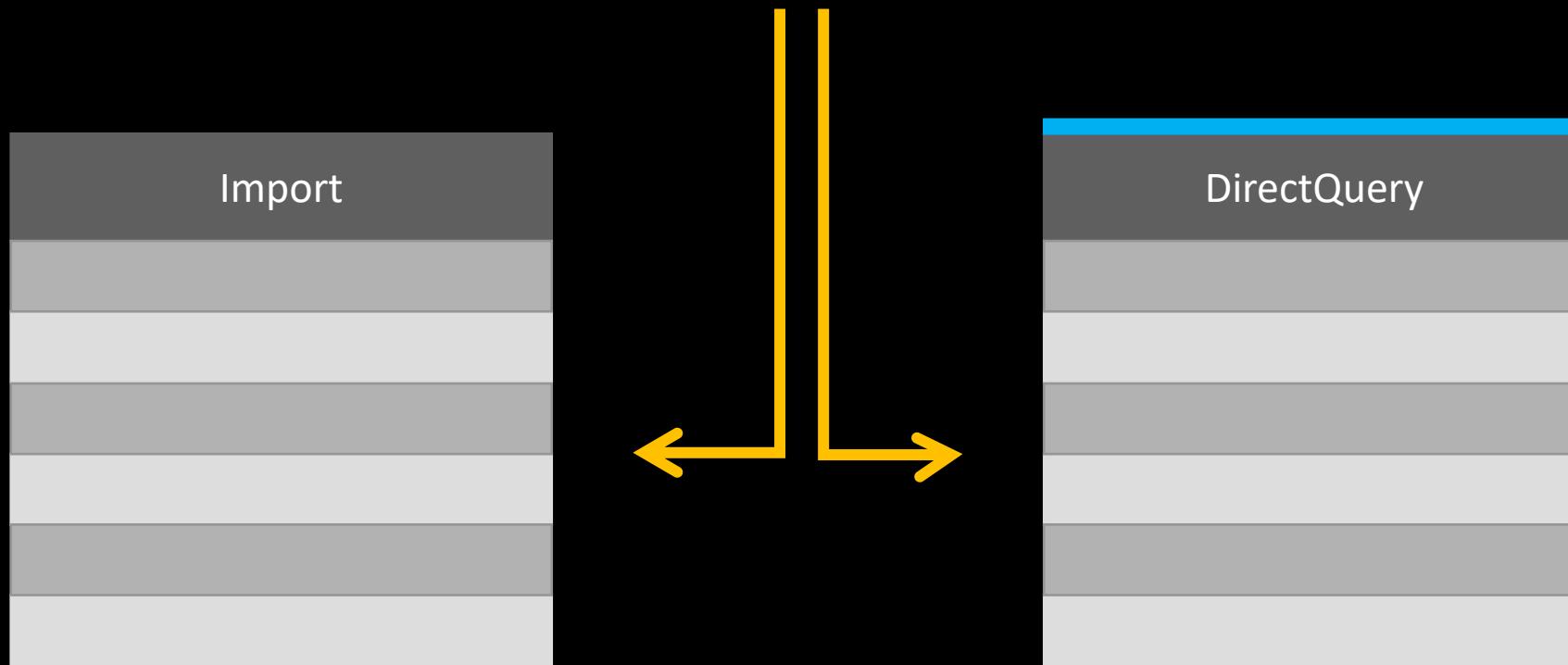
- Azure SQL Database
- Azure Synapse Dedicated SQL pool
- Google BigQuery
- Snowflake

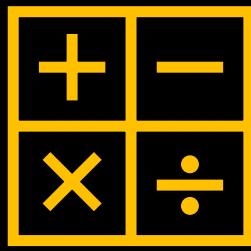


Aggregated data

Detailed data

{ *Query* }





# Demo Aggregations



# Query Folding



Generate SQL queries



Push queries back to  
the source



Improve performance  
in Power Query



# Query folding – supported sources

- Relational data sources like SQL Server, Oracle...
- OData sources (SharePoint lists...)
- Active Directory
- Exchange
- ...



# Query folding – supported operations

- Filtering (both rows and columns)
- Joins
- Aggregation (Group By)
- Pivot and Unpivot
- Numeric Calculations
- Simple transformations (Uppercase / Lowercase)

« »

 Home

 Data

 Develop

 Integrate

 Monitor

 Manage

Analytics pools

 SQL pools

 Apache Spark pools

Data Explorer pools (preview)

Activities

 SQL requests

 KQL requests

 Apache Spark applications

 Data flow debug

Integration

 Pipeline runs

 Trigger runs

 Integration runtimes

## SQL requests

 Refresh  Edit columns

 Filter by keyword Local time : Last 24 hours Status : All Pool : Built-in 

Showing 1 - 94 of 94 items

Request ID ↑	Request content ↑	Submit time ↑	Duration	Data processed	Submitter ↑
11198235	*** Global stats qu... <a href="#">More</a>	3/1/22, 9:56:40 PM	2 sec	10.00 MB	Marc@Data-marc.com
11198526	SELECT TOP (1000... <a href="#">More</a>	3/1/22, 9:56:40 PM	4 sec	3.29 GB	Marc@Data-marc.com
11196299	SELECT TOP (1000... <a href="#">More</a>	3/1/22, 9:56:09 PM	18 sec	6.64 GB	Marc@Data-marc.com
11033321	SELECT TOP (1000... <a href="#">More</a>	3/1/22, 8:51:57 PM	3 sec	3.29 GB	Marc@Data-marc.com
11031891	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:51:36 PM	2 sec	10.00 MB	Marc@Data-marc.com
11032227	SELECT TOP (1000... <a href="#">More</a>	3/1/22, 8:51:36 PM	5 sec	3.41 GB	Marc@Data-marc.com
11028238	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:51:00 PM	22 sec	1.85 GB	Marc@Data-marc.com
11029080	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:51:00 PM	30 sec	1.62 GB	Marc@Data-marc.com
11030543	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:51:00 PM	30 sec	10.00 MB	Marc@Data-marc.com
11030806	SELECT TOP (1000... <a href="#">More</a>	3/1/22, 8:51:00 PM	36 sec	10.00 MB	Marc@Data-marc.com
11026584	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:50:57 PM	16 sec	351.00 MB	Marc@Data-marc.com
11025192	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:50:47 PM	2 sec	11.00 MB	Marc@Data-marc.com
11025794	SELECT TOP (3502)... <a href="#">More</a>	3/1/22, 8:50:47 PM	3 sec	10.00 MB	Marc@Data-marc.com
11004240	select [].[Customer... <a href="#">More</a>	3/1/22, 8:44:08 PM	1 sec	10.00 MB	Marc@Data-marc.com
11004313	select [].[Employee... <a href="#">More</a>	3/1/22, 8:44:08 PM	1 sec	10.00 MB	Marc@Data-marc.com
11003771	select [rows].[Invoi... <a href="#">More</a>	3/1/22, 8:44:05 PM	15 sec	37.47 GB	Marc@Data-marc.com
10988500	select top 1000 [ro... <a href="#">More</a>	3/1/22, 8:38:51 PM	15 sec	37.41 GB	Marc@Data-marc.com
10987176	select top 1000 [ro... <a href="#">More</a>	3/1/22, 8:38:33 PM	15 sec	37.41 GB	Marc@Data-marc.com
10982220	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:37:42 PM	7 sec	83.00 MB	Marc@Data-marc.com
10982965	*** Global stats qu... <a href="#">More</a>	3/1/22, 8:37:42 PM	15 sec	910.00 MB	Marc@Data-marc.com

Home

Data

Develop

Integrate

Monitor

Manage

Analytics pools

- SQL pools
- Apache Spark pools
- Data Explorer pools (preview)

Activities

- SQL requests
- KQL requests
- Apache Spark applications
- Data flow debug

Integration

- Pipeline runs
- Trigger runs
- Integration runtimes

## SQL requests

 Refresh  Edit columns

Filter by keyword

Local time : Last 24 hours

Status : All

Showing 1 - 94 of 94 items

Request ID ↑	Request content ↑	Submit time ↑
11198235	*** Global stats qu... More	3/1/22, 9:56:40 PM
11198526	SELECT TOP (1000... More	3/1/22, 9:56:40 PM
11196299	SELECT TOP (1000... More	3/1/22, 9:56:09 PM
11033321	SELECT TOP (1000... More	3/1/22, 8:51:57 PM
11031891	*** Global stats qu... More	3/1/22, 8:51:36 PM
11032227	SELECT TOP (1000... More	3/1/22, 8:51:36 PM
11028238	*** Global stats qu... More	3/1/22, 8:51:00 PM
11029080	*** Global stats qu... More	3/1/22, 8:51:00 PM
11030543	*** Global stats qu... More	3/1/22, 8:51:00 PM
11030806	SELECT TOP (1000... More	3/1/22, 8:51:00 PM
11026584	*** Global stats qu... More	3/1/22, 8:50:57 PM
11025192	*** Global stats qu... More	3/1/22, 8:50:47 PM
11025794	SELECT TOP (3502)... More	3/1/22, 8:50:47 PM
11004240	select [__].[CustomerKey] as [Customer Key], [__].[CityKey] as [City Key], [__].[EmployeeKey] as [Employee Key], [__].[SalesTerritoryKey] as [Sales Territory]	3/1/22, 8:44:08 PM
11004313	select [__].[EmployeeKey] as [Employee Key], [__].[SalesTerritoryKey] as [Sales Territory]	3/1/22, 8:44:08 PM
11003771	select [__].[SalesTerritoryKey] as [Sales Territory]	3/1/22, 8:44:05 PM
10988500	select top 1000 [rowguid] as [Row GUID]	3/1/22, 8:38:51 PM
10987176	select top 1000 [rowguid] as [Row GUID]	3/1/22, 8:38:33 PM
10982220	*** Global stats qu... More	3/1/22, 8:37:42 PM
10982965	*** Global stats qu... More	3/1/22, 8:37:42 PM

## Request content

11198526

```
SELECT
TOP (1000001) *
FROM
(
SELECT [t5].[Sales Territory],SUM([t4].[Profit])
AS [a0]
FROM
(
select [__].[SaleKey] as [Sale Key],
[__].[CityKey] as [City Key],
[__].[CustomerKey] as [Customer Key],
[__].[BillToCustomerKey] as [Bill To Customer Key],
[__].[StockItemKey] as [Stock Item Key],
[__].[InvoiceDateKey] as [Invoice Date Key],
[__].[DeliveryDateKey] as [Delivery Date Key],
[__].[SalespersonKey] as [Salesperson Key],
[__].[WWIInvoiceID] as [WWIInvoice ID],
[__].[Description] as [Description],
[__].[Package] as [Package],
[__].[Quantity] as [Quantity],
[__].[UnitPrice] as [Unit Price],
[__].[TaxRate] as [Tax Rate],
[__].[TotalExcludingTax] as [Total Excluding Tax],
[__].[TaxAmount] as [Tax Amount],
[__].[Profit] as [Profit],
[__].[TotalIncludingTax] as [Total Including Tax],
[__].[TotalDryItems] as [Total Dry Items],
[__].[TotalChillerItems] as [Total Chiller Items],
[__].[LineageKey] as [Lineage Key]
from [sales_model].[Sales] as [__]
where [__].[InvoiceDateKey] >= convert(date, '2013-01-01') and [__].[InvoiceDateKey]
) AS [t4]
INNER JOIN

(
select [__].[CityKey] as [City Key],
[__].[WWICityID] as [WWICity ID],
[__].[City] as [city],
```

# DAX Studio trace

The screenshot shows the DaxStudio interface version 2.17.3. The top ribbon includes File, Home, Advanced, Help, Layout, Traces, and a toolbar with various icons for running queries, clearing cache, and connecting to servers. The main window has a title bar "Query1.dax\*". On the left, there's a sidebar with "Metadata" dropdowns for "Test with partitions" and "Model", and a search bar. Below the sidebar is a tree view of database objects: City, Customer, Date, Employee, Sales, Sales Agg, Sales Hybrid, and StockItem. The central pane displays XML code for a DAX batch transaction, starting with a <Batch Transaction="true" ...> tag. The bottom pane shows a table of tracing information with columns: StartTime, Type, Duration, User, Database, and Query. The table lists numerous entries from different users (Power BI Service, DemoUser@Data-marc.com) performing various DAX operations like XMLA requests, Xaml, and DAX, along with their corresponding SQL queries and execution details.

StartTime	Type	Duration	User	Database	Query
10:11:51	Xmla	9.262	Power BI Service	Test with partitions	<Batch Transaction="true" xmlns="http://schemas.microsoft.com/analysisservices/2003/engine..."
10:11:42	Xmla	16	Power BI Service	Test with partitions	<Batch Transaction="false" xmlns="http://schemas.microsoft.com/analysisservices/2003/engine..."
10:10:55	Xmla	527	DemoUser@Data-marc.com	Test with partitions	<Batch Transaction="true" xmlns="http://schemas.microsoft.com/analysisservices/2003/engine..."
10:09:22	DAX	2.127	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOCore = TREATAS("2022", [Date].[Year]) VAR __DSOCore = SUMMARIZ...
10:08:49	DAX	9.957	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOFilterTable = SUMMARIZECOLUMNS(ROLLUPPADDISSTOTAL(ROLLUPGROUP(...
10:08:49	DAX	16	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOCore = ADDCOLUMNS(KEEPFILTERS(VALUES([Date].[Year])), "M0", CALCUL...
10:07:47	DAX	2.236	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOFilterTable = TREATAS("2022", [Date].[Year]) VAR __DSOCore = SUMMARIZ...
10:07:10	DAX	14.613	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOCore = SUMMARIZECOLUMNS(ROLLUPPADDISSTOTAL(ROLLUPGROUP(...
10:05:58	Xmla	11.027	Power BI Service	Test with partitions	<Batch Transaction="true" xmlns="http://schemas.microsoft.com/analysisservices/2003/engine..."
10:05:47	Xmla	31	Power BI Service	Test with partitions	<Batch Transaction="false" xmlns="http://schemas.microsoft.com/analysisservices/2003/engine..."
10:04:16	Xmla	515	DemoUser@Data-marc.com	Test with partitions	<Batch Transaction="true" xmlns="http://schemas.microsoft.com/analysisservices/2003/engine..."
09:57:45	Xmla	47	DemoUser@Data-marc.com	Test with partitions	<Batch Transaction="false" xmlns="http://schemas.microsoft.com/analysisservices/2003/engine..."
09:51:41	DAX	10.685	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOCore = SUMMARIZECOLUMNS(ROLLUPPADDISSTOTAL([City].[Sales Territ...)
09:51:34	DAX	4.215	DemoUser@Data-marc.com	Test with partitions	EVALUATE ROW( [v_Sales_Hybrid], Sales Hybrid!\$ Sales Hybrid! )
09:50:41	DAX	4.603	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOFilterTable = TREATAS("2022", [Date].[Year]) VAR __DSOCore = SUMMARIZ...
09:50:36	Xmla	0	Power BI Service (DemoUser@Data-m...	Test with partitions	<Cancel xmlns="http://schemas.microsoft.com/analysisservices/2003/engine" xmlns:soap="h...
09:50:28	Xmla	8.376	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOCore = SUMMARIZECOLUMNS(ROLLUPPADDISSTOTAL([Date].[Date], "Is...)
09:50:32	DAX	16	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOCore = ADDCOLUMNS(KEEPFILTERS(VALUES([Date].[Year])), "M0", CALCUL...
09:49:58	DAX	2.345	DemoUser@Data-marc.com	Test with partitions	DEFINE VAR __DSOFilterTable = FILTER(KEEPFILTERS(VALUES([Sales Hybrid].[Invoice Date Key]))...



# Demo Query Folding



# Premium specific features

- Paginated Reports
- Share with free users  
(P sku only! Not in A sku!)
- Auto-scale
- Support for larger dataset sizes
- 48x daily refresh  
(and Automatic Page Refresh)
- Extra dataflows features
- Enhanced embedding scenarios
- Bring your own key (BYOK)
- Hybrid tables
- Multi-geo support
- XMLA endpoints
- Paginated Reports
- Deployment Pipelines (extra slide on this)
- AI workloads

# Hybrid tables & incremental refresh

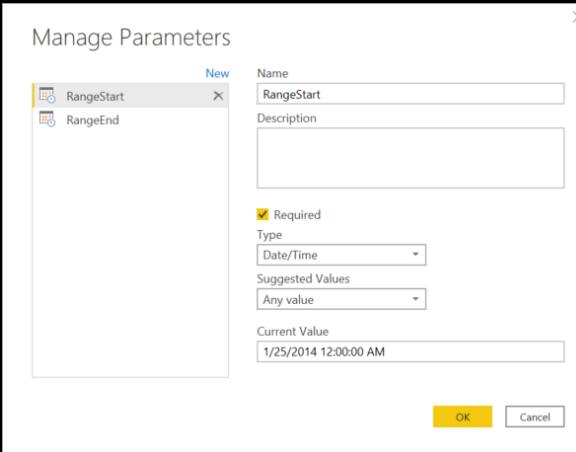
# Incremental refresh

- Fewer refresh cycles for fast-changing data – DirectQuery mode gets the latest data updates as queries are processed without requiring a high refresh cadence.
- Refreshes are faster - Only the most recent data that has changed needs to be refreshed.
- Refreshes are more reliable - Long-running connections to volatile data sources aren't necessary. Queries to source data run faster, reducing potential for network problems to interfere.
- Resource consumption is reduced - Less data to refresh reduces overall consumption of memory and other resources in both Power BI and data source systems.
- Enables large datasets - Datasets with potentially billions of rows can grow without the need to fully refresh the entire dataset with each refresh operation.
- Easy setup - Incremental refresh policies are defined in Power BI Desktop with just a few tasks. When published, the service automatically applies those policies with each refresh.

# Incremental refresh

- Incremental refresh is supported for Power BI Premium, Premium per user, **Power BI Pro**, and Power BI Embedded datasets.
- Getting the latest data in **real time** with DirectQuery is **only supported for Power BI Premium**, Premium per user, and Power BI Embedded datasets.

# Incremental refresh config



CustomerPONumber

OrderDate

DueDate

ShipDate

RangeStart

RangeEnd

Name: RangeStart

Description:

Required

Type: Date/Time

Suggested Values

Any value

Current Value: 1/25/2014 12:00:00 AM

OK Cancel

Sort Ascending

Sort Descending

Clear Sort

Clear Filter

Remove Empty

Date/Time Filters

Search

(Select All)

12/29/2010 12:00:00 AM

12/30/2010 12:00:00 AM

12/31/2010 12:00:00 AM

1/1/2011 12:00:00 AM

1/2/2011 12:00:00 AM

1/3/2011 12:00:00 AM

1/4/2011 12:00:00 AM

1/5/2011 12:00:00 AM

1/6/2011 12:00:00 AM

1/7/2011 12:00:00 AM

1/8/2011 12:00:00 AM

1/9/2011 12:00:00 AM

1/10/2011 12:00:00 AM

1/11/2011 12:00:00 AM

1/12/2011 12:00:00 AM

1/13/2011 12:00:00 AM

1/14/2011 12:00:00 AM

List may be incomplete. Load more

OK Cancel Custom Filter...

## Incremental refresh and real-time data

Refresh large tables faster with incremental refresh. Plus, get the latest data in real time with DirectQuery (Premium only). [Learn more](#)

These settings will apply when you publish the dataset to the Power BI service. Once you do that, you won't be able to download it back to Power BI Desktop. [Learn more](#)

- 1. Select table**  
FactInternetSales
- 2. Set import and refresh ranges**

Incrementally refresh this table

Archive data starting 5 Years before refresh date  
Data imported from 12/21/2016 to 12/18/2021.

Incrementally refresh data starting 3 Days before refresh date  
Data will be incrementally refreshed from 12/18/2021 to 12/21/2021.
- 3. Choose optional settings**

Get the latest data in real time with DirectQuery (Premium only) [Learn more](#)

Only refresh complete days [Learn more](#)

Detect data changes [Learn more](#)
- 4. Review and apply**

Archived Incremental Refresh Real time

5 years before refresh date 3 days before refresh date Refresh date

Apply Cancel



# Hybrid tables

- Live / Realtime data in Power BI
- Combines different storage modes on partition level in a single table
- Goes hand-in-hand with Incremental Refresh

Granularity	Name	Row Count
Year	2011	295,489,717
Year	2012	297,678,498
Year	2013	295,575,442
Year	2014	292,477,875
Year	2015	297,780,469
Year	2016	294,060,081
Year	2017	300,419,682
Year	2018	296,541,108
Year	2019	292,787,420
Year	2020	299,273,979
Quarter	2021Q1	74,135,277
Month	2021Q104	24,939,498
Day	2021Q10501	820,805
Day	2021Q10502	826,885
Day	2021Q10503	821,043
Day-DirectQuery	2021Q10504-DQ	271,110
Total		3,063,898,887

Archived: **Import**

Incremental refresh: **Import**

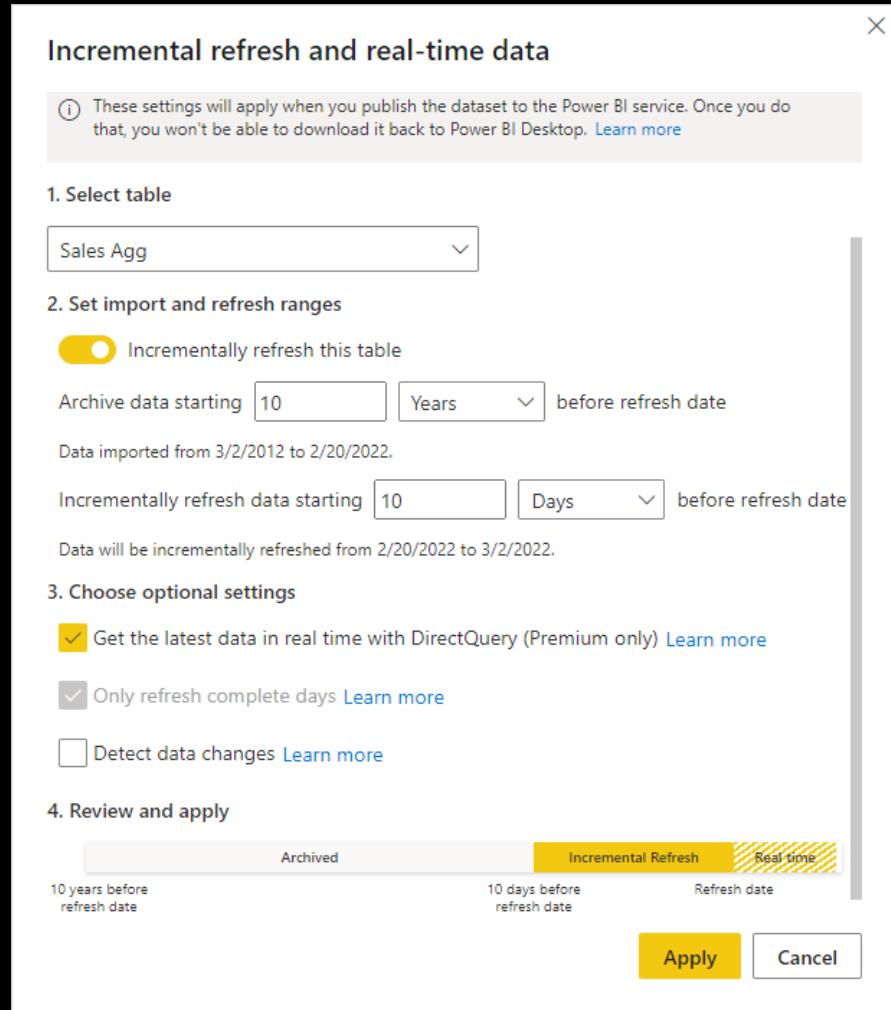
Real time: **DirectQuery**



# Hybrid tables

- Implementation with Incremental Refresh
- Customizable via 3rd party tooling like Tabular Editor

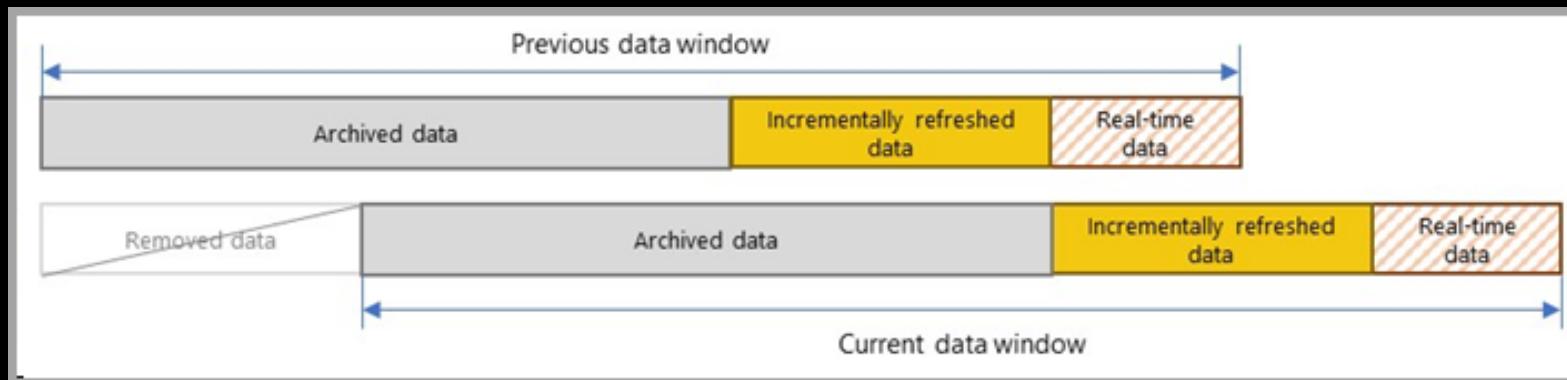
>> Limitation: Only 1 DQ partition per table allowed at the moment.





# Hybrid tables – what challenge does it solve?

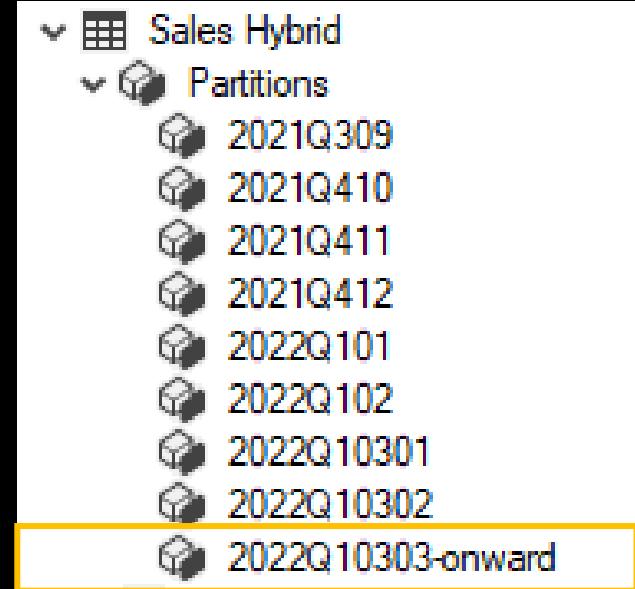
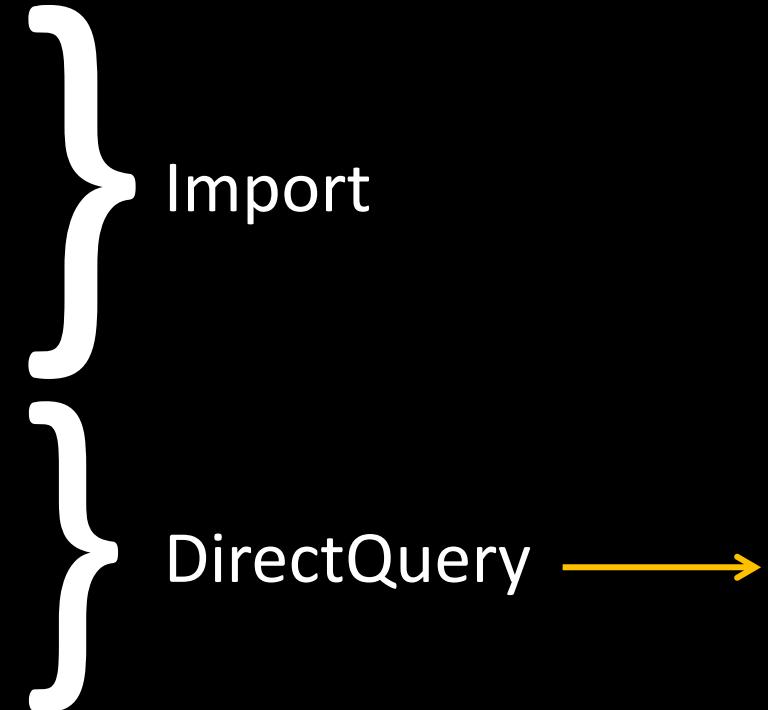
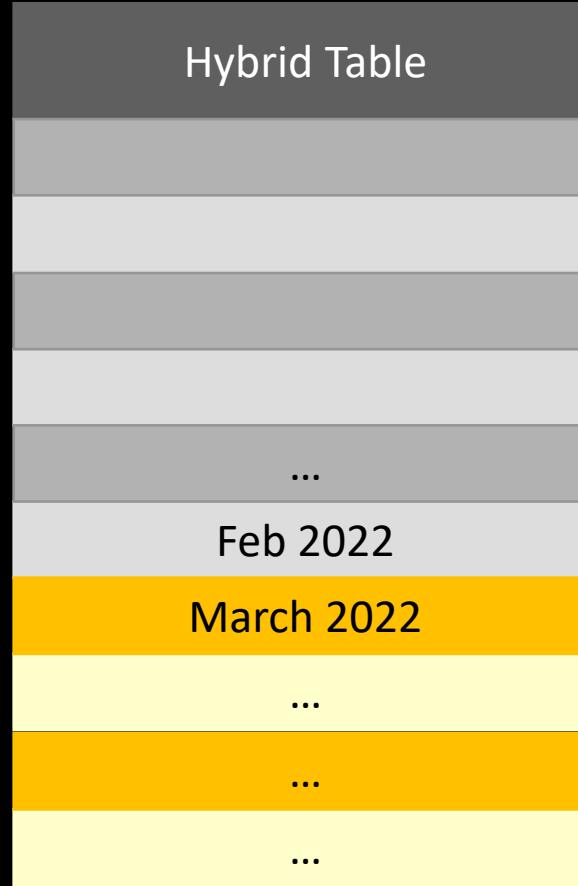
- Realtime scenarios without full tables on DQ mode
- No complex refresh mechanisms needed with partition refresh and queries over XMLA
- No more multiple tables and complex DAX to combine to achieve the same goal





# Hybrid tables – Keep in mind that...

- Premium feature
- DAX restrictions for DirectQuery apply
- Limited Power Query capabilities (due to DQ)
- Requires Large Dataset Format (storage) in workspace
- Performance hit on upstream data sources





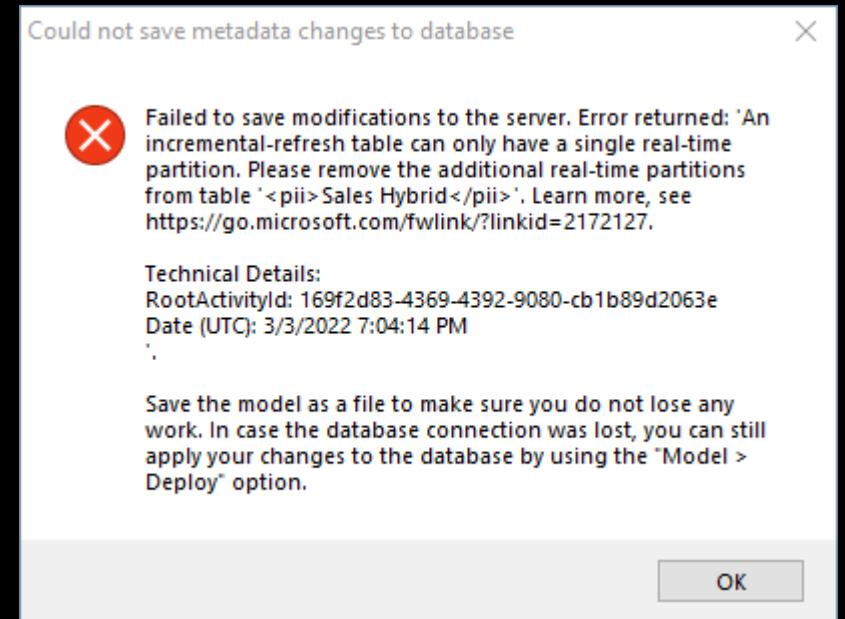
Demo Hybrid Tables  
Latest data in real time



# Can I change partition storage modes?

You cannot adjust tables with incremental refresh policies applied. However, there are other options to consider.

Options	Default
Data View	DirectQuery
Mode	PolicyRange
Source Type	





# Manually setup partitioning

The screenshot shows the Tabular Editor interface with the following details:

- Left pane (Model View):** Shows the project structure with nodes like Model, Data Sources, Perspectives, Relationships, Roles, Shared Expressions, Tables, and Partitions.
- Middle pane (Expression Editor):** Displays DAX code for a partition:

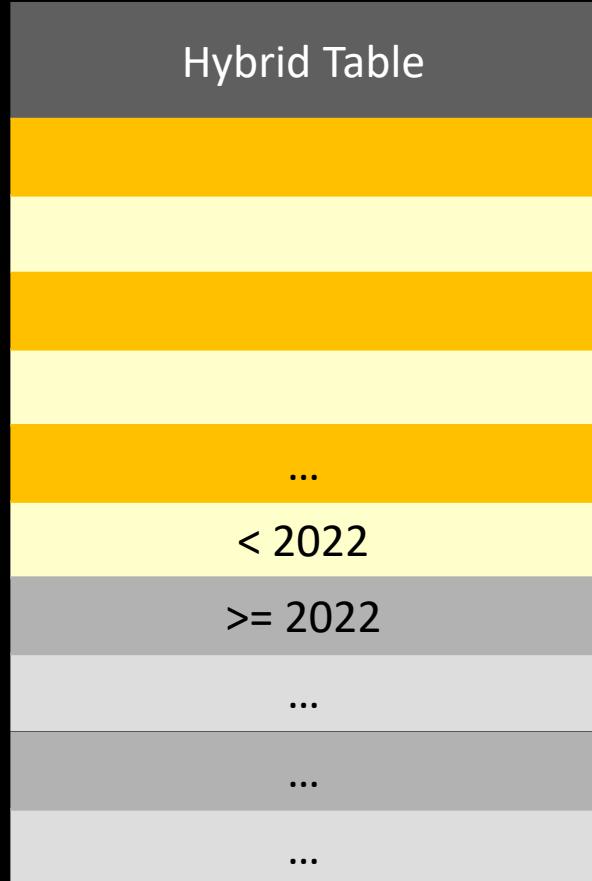
```
let
    Source = Sql.Databases((SynapseEndpoint)),
    #"WideWorldImportersDW-Standard" = Source[[Name=(DatabaseName)]][Data],
    Fact_Sale = #"WideWorldImportersDW-Standard"[[Schema=(SynapseSchema), Item="Sales"]][Data],
    #"Filtered Rows" = Table.SelectRows(#"Fact_Sale", each [InvoiceDateKey] < #date(2022, 01, 01) ),
    SplitByCharacter = Table.TransformColumnNames(#"Filtered Rows", each
        Text.Combine(
            Splitter.SplitTextByCharacterTransition({{"a".."z"}, {"A".."Z"}})(_)
            , " ")
        )
    in
        SplitByCharacter
```

A red box highlights the part of the code where rows are filtered based on the date key.
- Bottom pane (Properties):** Shows the properties for the selected partition "Sales Hybrid-DQ".

Basic	let Source = Sql.Databases((SynapseEndpoint)). #"WideWorldImportersDW
Metadata	Sales Hybrid-DQ
Annotations	0 annotations
Last Processed	31-12-1699
Object Type	Partition (M - DirectQuery)
State	Ready
Options	
Attributes	
Data View	Default
Mode	DirectQuery
Source Type	M

A red box highlights the "Mode" field set to "DirectQuery".
- Bottom status bar:** Shows "1 partition (m - directquery) selected." and "0 BP issues".

Create your own hybrid table setup, using 3rd party tooling like Tabular Editor



} DirectQuery  
}

} Import

### DirectQuery Partition definition

```
let
    Source = Sql.Databases((SynapseEndpoint)),
    #"WideWorldImportersDW-Standard" = Source{[Name=(DatabaseName)][Data],
    Fact_Sale = #"WideWorldImportersDW-
Standard"[{[Schema=(SynapseSchema),Item="Sales"]}[Data],
    #"Filtered Rows" = Table.SelectRows(#"Fact_Sale", each
        [InvoiceDateKey] < #date(2022, 01, 01) and
        [InvoiceDateKey] >= #date(2013, 01, 01)),
        SplitByCharacter = Table.TransformColumnNames(#"Filtered Rows", each
            Text.Combine(
                Splitter.SplitTextByCharacterTransition({"a".."z"}, {"A".."Z"})(_)
                , " ")
            )
        in
            SplitByCharacter
```

### Import Partition definition

```
let
    Source = Sql.Databases((SynapseEndpoint)),
    #"WideWorldImportersDW-Standard" = Source{[Name=(DatabaseName)][Data],
    Fact_Sale = #"WideWorldImportersDW-
Standard"[{[Schema=(SynapseSchema),Item="Sales"]}[Data],
    #"Filtered Rows" = Table.SelectRows(#"Fact_Sale", each
        [InvoiceDateKey] >= #date(2022, 01, 01) and
        [InvoiceDateKey] < #date(2022, 03, 31)),
        SplitByCharacter = Table.TransformColumnNames(#"Filtered Rows", each
            Text.Combine(
                Splitter.SplitTextByCharacterTransition({"a".."z"}, {"A".."Z"})(_)
                , " ")
            )
        in
            SplitByCharacter
```



Demo Hybrid Tables  
Historic data via DirectQuery

# Refresh challenges



# Refresh options

- Scheduled in the service
- Manual trigger
- Power Automate
- PowerShell
- API



# Effective refreshing

## Considerations

- Refreshing the entire model takes too long with high load on sources
- Can we only refresh certain tables?
- Can we only refresh certain partitions?
- Can we use DQ tables/partitions (Hybrid Tables)

## What do we need?

- Async refresh API
- XMLA Endpoints



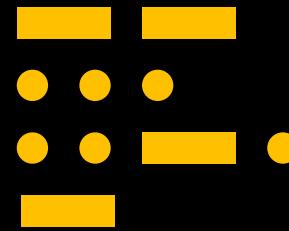
# Async refresh API (preview)

- Specify the objects to refresh

POST

*https://api.powerbi.com/v1.0/myorg/groups/f089354e-8366-4e18-aea3-4cb4a3a50b48/datasets/cfafbeb1-8037-4d0c-896e-a46fb27ff229/refreshes*

```
{  
  "type": "Full",  
  "commitMode": "transactional",  
  "maxParallelism": 2,  
  "retryCount": 2,  
  "objects": [  
    {  
      "table": "DimCustomer",  
      "partition": "DimCustomer"  
    },  
    {  
      "table": "DimDate"  
    }  
  ]  
}
```

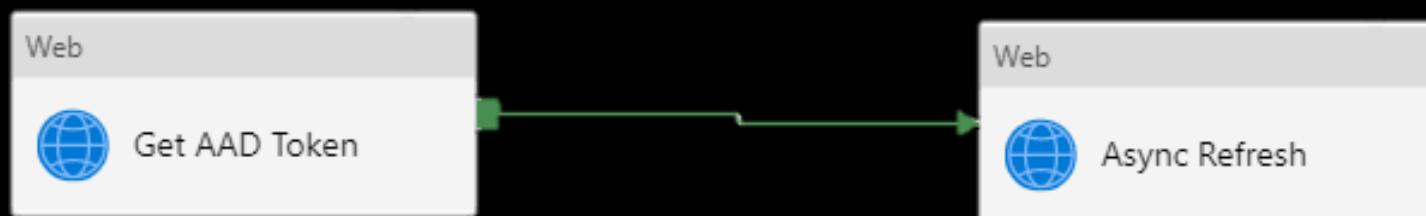


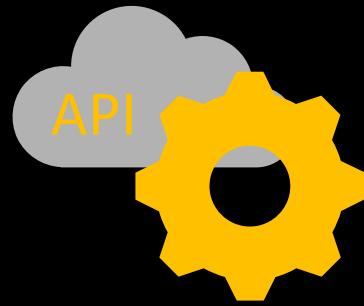
# — Demo Asynchronous refresh API



# Central E2E orchestration

- Combine pipelines from Data Platform with Power BI
- Lowest latency between source and report
- Consider including backup operations for Power BI
- Incremental loading where possible





— Demo end-2-end orchestration

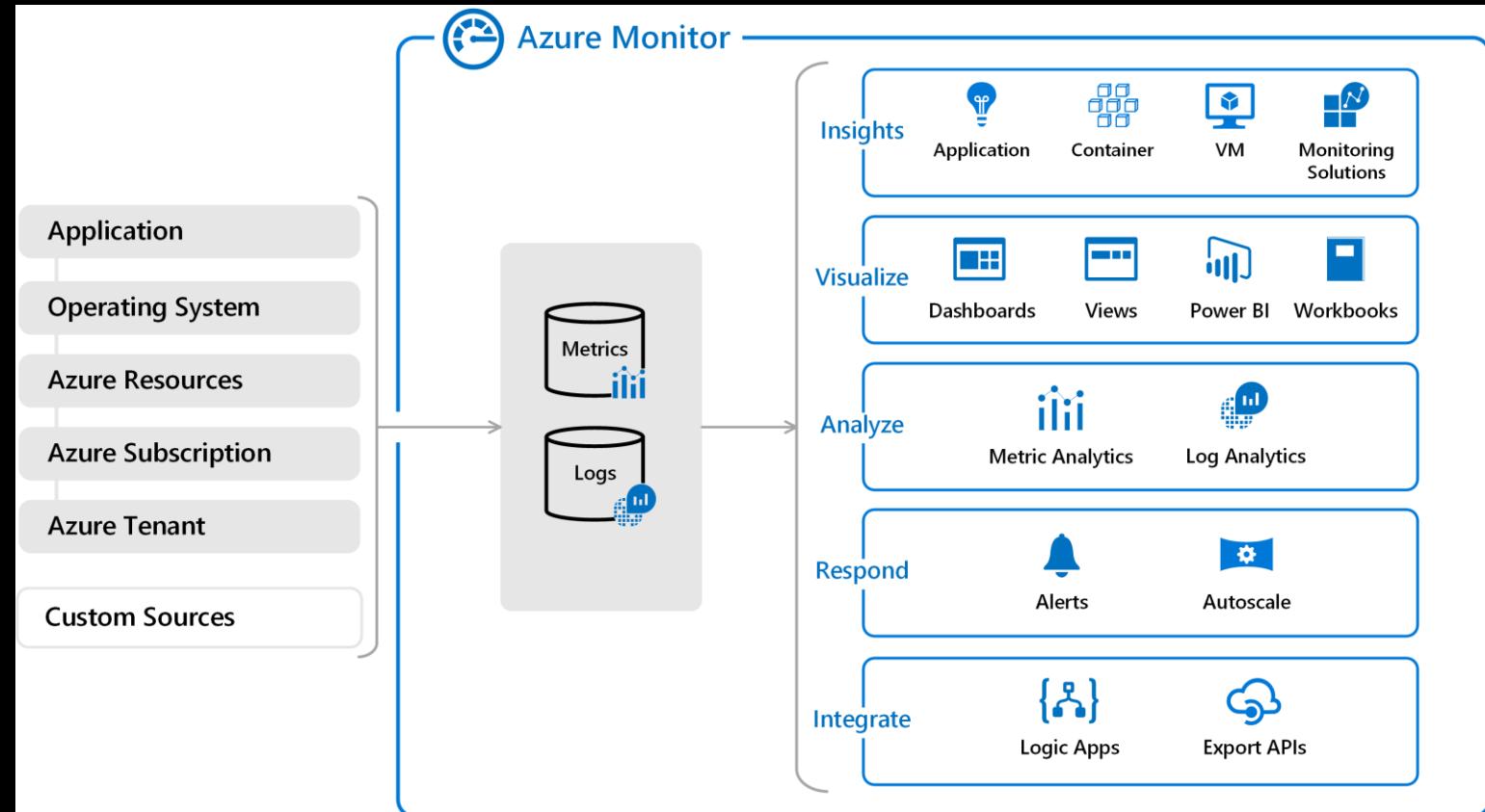
# Monitoring - revisited

# Data platform monitoring



# Log Analytics Workspace / Azure Monitor

Azure Monitor delivers a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. It helps you understand how your applications are performing and proactively identifies issues affecting them and the resources they depend on.





# Scenarios

- Pipelines:
  - Identify (unusual) long durations
  - Identify unusual amount of runs
  - Trend analysis for durations
- Spark cluster:
  - Analyse utilization of driver and worker nodes
- SQL Serverless:
  - Data scanned by DirectQuery & import to datasets
- Storage:
  - Analyse trend in amount of data stored
  - Analyse trend in amount/duration of reads and writes
- Cost:
  - Analyse Resource Costs



# Azure Synapse monitoring

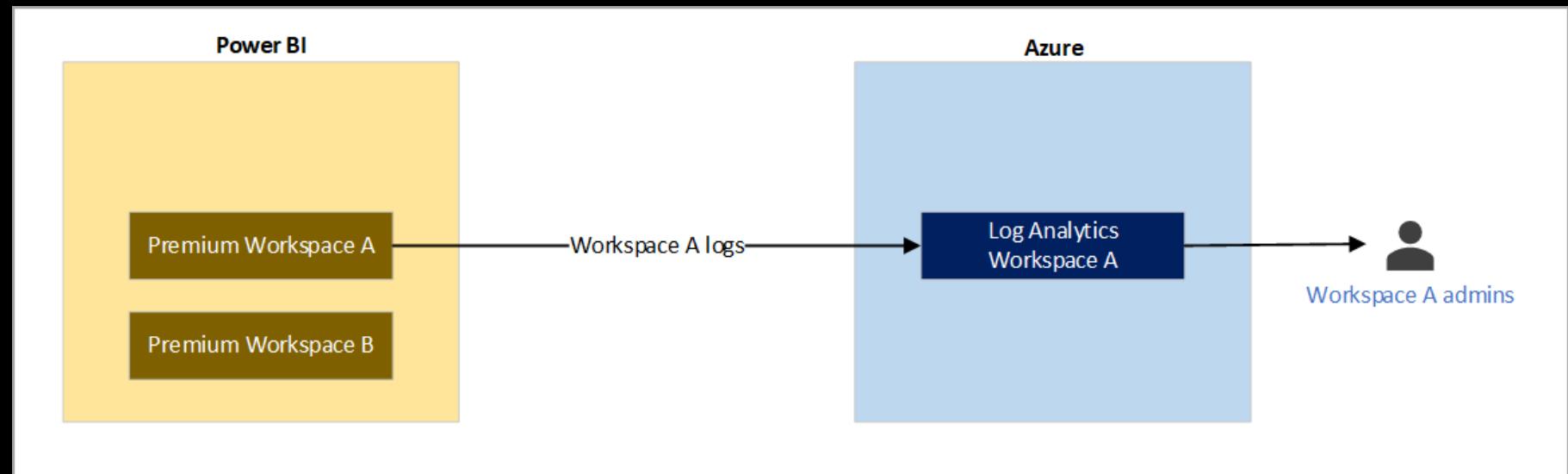
Pipeline runs  
Spark cluster  
Log Analytics

# Power BI monitoring



# Power BI Azure Log Analytics (ALA) integration

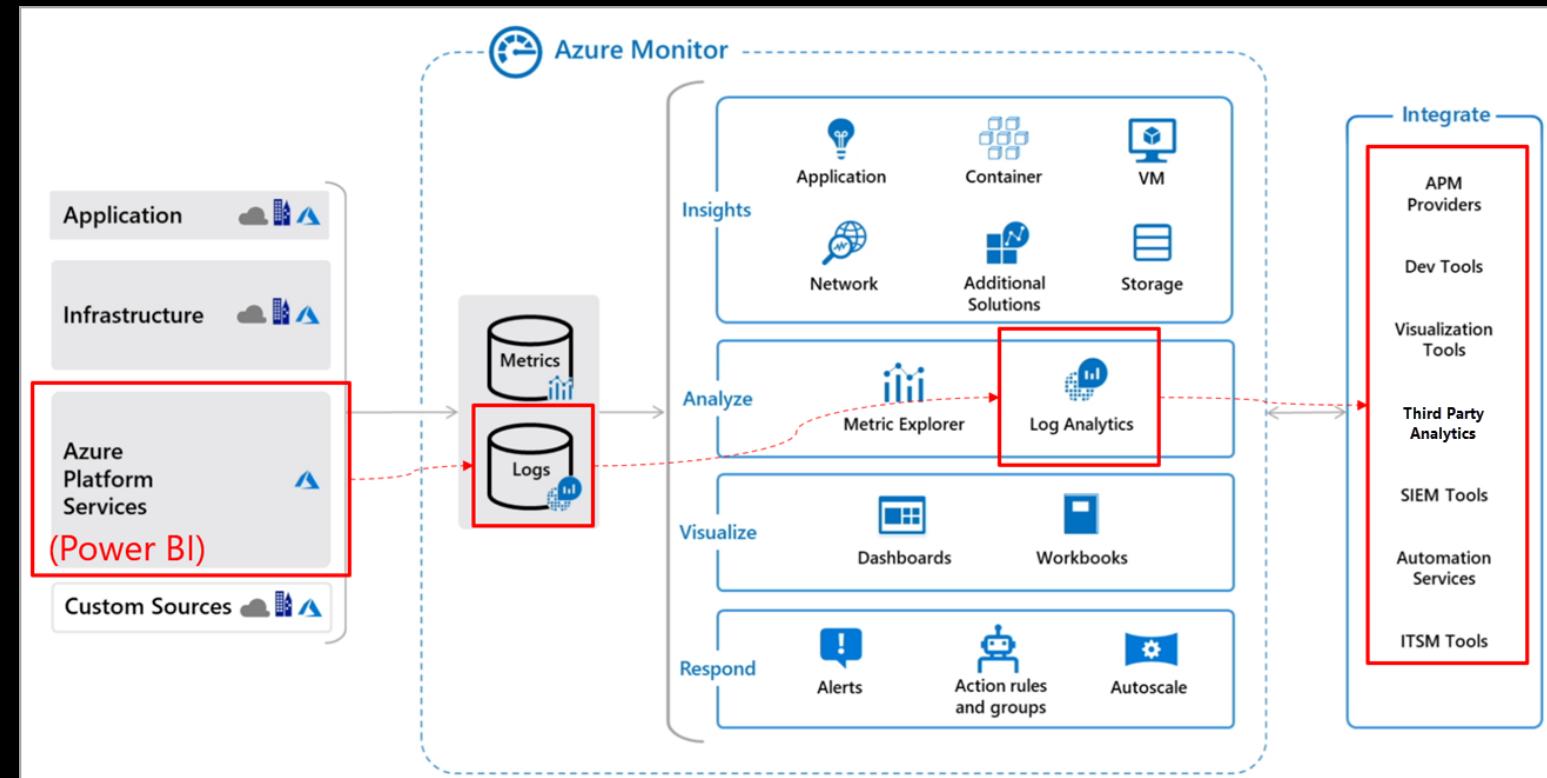
Configure Azure Monitor integration on Power BI workspace level to allow workspace administrators to collect, analyze and act on telemetry data. PBI integration with ALA exposes events from the Analysis Services engine.





# Log Analytics Workspace / Azure Monitor

Azure Monitor delivers a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. It helps you understand how your applications are performing and proactively identifies issues affecting them and the resources they depend on.



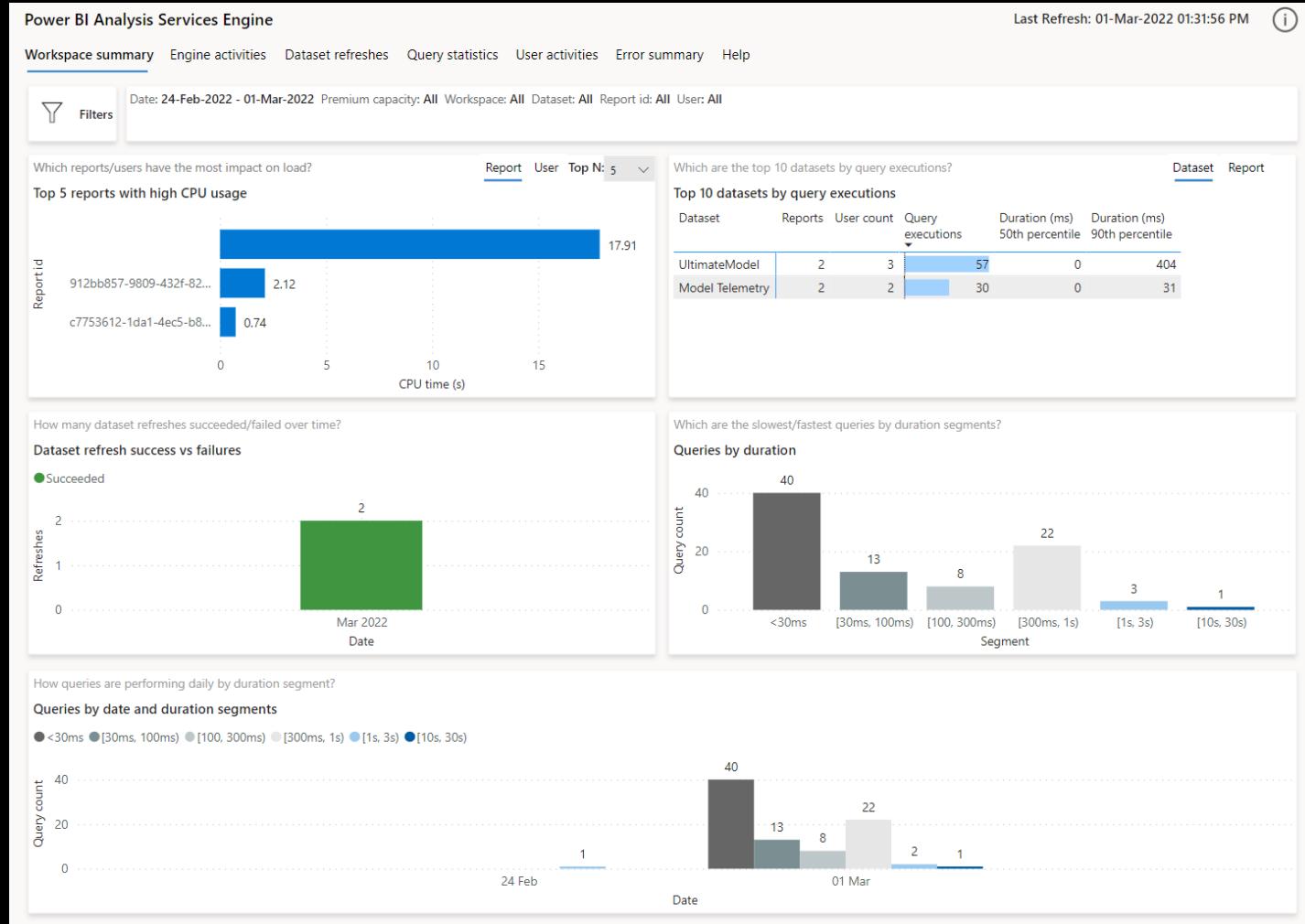


# Scenarios

- Identify high or unusual engine activity by capacity / ws / user
- Analyze query performance
- Analyze dataset refresh durations and engine operations
- Investigate impact of custom operations via Premium XMLA



# Template report to get insights



- Workspace summary
- Engine activities
- Dataset refreshes
- Query statistics
- Error summary



# Power BI Analysis Services Engine

Power BI Analysis Services Engine

Last Refresh: 01-Oct-2021 01:30:46 PM

Workspace summary Engine activities Dataset refreshes Query statistics **User activities** Error summary Help

Filters Date: 26-Sep-2021 - 01-Oct-2021 Premium capacity: All Wokspace: All Dataset: All Report id: All User: All

**Operations** Queries Top N: 5

**Top 5 users by operation**

User	Operations
3f7ecc014cc48b5cf2...	222
ea1a7a2ca8ae278b6...	198
335989582a9c58801...	110
27f4484675f0aaaf256...	107
d661cf332ee8e0cbd...	107

**Queries vs CPU time by users**

**Daily user and operation count**

Date	Users	Operations
Sep 26	0.00K	0.10K
Sep 27	0.46K	0.01K
Sep 28	0.71K	0.02K
Sep 29	0.01K	0.47K
Sep 30	0.68K	0.01K
Oct 01	0.00K	0.02K

**Hourly user and operation count**

Hour	Users	Operations
0	0.00K	0.02K
1	0.00K	0.00K
2	0.00K	0.00K
3	0.19K	0.00K
4	0.00K	0.00K
5	0.09K	0.00K
6	0.23K	0.00K
7	0.35K	0.00K
8	0.00K	0.00K
9	0.00K	0.00K
10	0.00K	0.00K
11	0.00K	0.00K
12	0.00K	0.00K
13	0.00K	0.00K
14	0.00K	0.00K
15	0.00K	0.00K
16	0.00K	0.00K
17	0.00K	0.00K
18	0.00K	0.00K
19	0.00K	0.00K
20	0.00K	0.00K
21	0.00K	0.00K
22	0.00K	0.00K
23	0.00K	0.00K
24	0.00K	0.00K

**User details**

User	Last active date	Dataset	Report	Operations	Refreshes	Queries	Avg CPU time	Avg duration (ms)
3f7ecc014cc48b5cf2...	01-Oct-2021	2	2	222		115	219	228
ea1a7a2ca8ae278b6...	01-Oct-2021	2	2	198		43	133	132
335989582a9c58801...	01-Oct-2021	2	2	110		58	72	104
27f4484675f0aaaf256...	01-Oct-2021	2	3	107		32	316	340
d661cf332ee8e0cbd...	01-Oct-2021	2	2	107		49	75	70
f22a8aef8a9630d9a...	01-Oct-2021	2	2	106		77	156	166
f6c9179e9875704f...	01-Oct-2021	2	2	90		64	186	181
<b>Total</b>	01-Oct-2021	4	6	2,454		959	196	178



# Configuration

## Admin portal

Usage metrics

Users

Premium Per User

Audit logs

Tenant settings

Capacity settings

Refresh summary

Embed Codes

Organizational visuals

Azure connections

Workspaces

Custom branding

Protection metrics

Featured content

### Connect to Azure resources (preview)

- ▶ Tenant-level storage (preview)
- ▶ Workspace-level storage permissions (preview)

#### ◀ Workspace-level Log Analytics permissions (preview)



Allow workspace admins to connect their own Log Analytics workspace

Save

Cancel



# Configuration

The screenshot shows the Microsoft Power BI Admin Center. At the top, there's a navigation bar with a user icon, the name "admin" (redacted), a diamond icon, and a "Create app" button. Below the navigation bar are buttons for "New", "Create a pipeline", "View", "Filters", "Settings" (which is highlighted with a red box), "Access", and "Search". Underneath these buttons, there are tabs for "All", "Content", and "Datasets + dataflows", with "All" being the active tab. A table below lists three items: a Report named "Customer360 TEST", a Dataset named "Customer360 TEST", and a Dashboard named "Customer360 TEST.pbix". Each item has columns for Name, Type, Owner, Refreshed, and Next refresh.

Name	Type	Owner	Refreshed	Next refresh
Customer360 TEST	Report	admin (redacted)	10/12/20, 11:05:34 AM	—
Customer360 TEST	Dataset	admin (redacted)	10/12/20, 11:05:34 AM	N/A
Customer360 TEST.pbix	Dashboard	admin (redacted)	—	—



# Configuration

**Settings**

About   Premium   Azure connections (preview) **Azure connections (preview)**

▶ Storage

◀ Log Analytics

Connect an Azure Log Analytics workspace to collect usage and performance logs for this workspace. [Learn more](#)

**Connect to Azure**

Subscription  
Select an Option

Resource group  
Select an Option

Log Analytics workspace  
Select an Option

**Save**   **Cancel**

**Settings**  
admin [REDACTED]

About   Premium   Azure connections (preview) **Azure connections (preview)**

▶ Storage

◀ Log Analytics

Connect an Azure Log Analytics workspace to collect usage and performance logs for this workspace. [Learn more](#)

**Subscription** a1 [REDACTED] 7e

**Resource group** rg-byola

**Log Analytics workspace** admir [REDACTED]

Configured by Admin Admin on 2020-10-12T12:00:00Z

**Disconnect from Azure**

Disconnecting will stop usage and performance data from flowing into your Azure Log Analytics workspace.

**Disconnect from Azure**



# Configuration

Go to [AppSource > Power BI Log Analytics for Analysis Services Engine](#)

and select **Get it now.**

The screenshot shows the Microsoft AppSource page for the "Power BI Log Analytics for AS Engine (Preview)" app. The page includes the app's logo, a "Get it now" button, and several screenshots of the Power BI dashboards it generates. Below the main app listing, there are other related Microsoft apps like "Power BI Premium Capacity Utilization" and "Power BI Premium Capacity Metrics App".

**Power BI Log Analytics for AS Engine (Preview)**

**Overview**   **Ratings + reviews**   **Details + support**

The app provides admins with insights into dataset usage and performance from AS engine traces. This app will connect to the pre-configured Azure Log Analytics workspace where you are collecting Analysis Services engine log data. The app can be used to monitor engine load, engine usage trends, and identify slow queries and dataset refreshes. It provides summary and trend information at Workspace level with the ability to drill to detail about any individual query or dataset refresh. The app allows you to group or slice the data by useful properties such as Premium capacity, dataset, report, and user.

**At a glance**

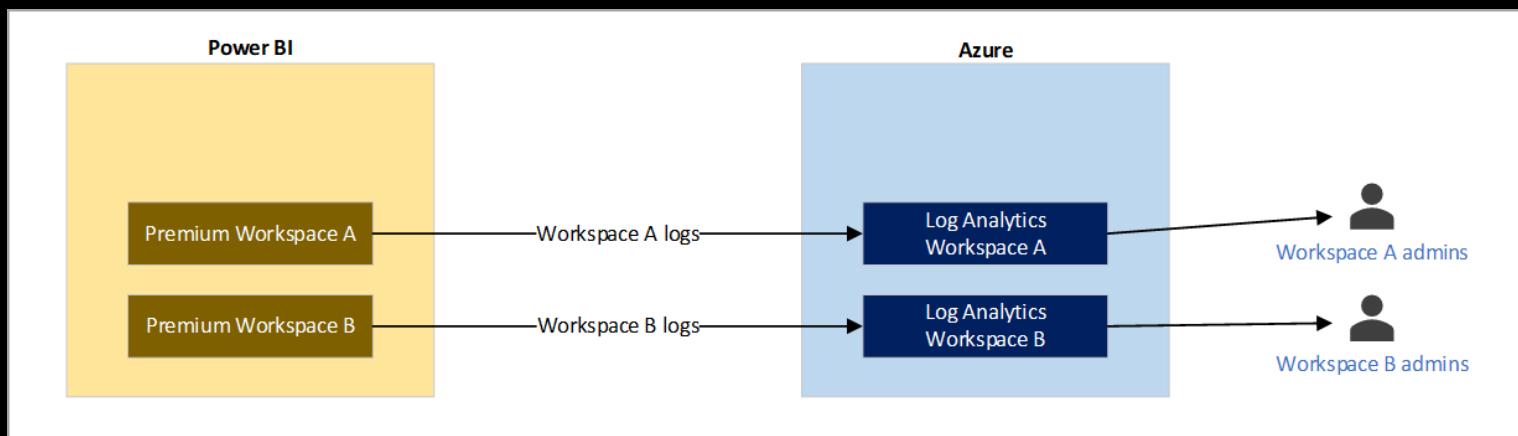
**Other apps from Microsoft**

- Power BI Premium Capacity Utilization app...**
- Power BI Premium Capacity Metrics App**



# Configuration pre-requisites – Generic

- Only Premium
- Only v2 workspaces
- Only for the datasets in the workspace (no shared datasets)
- 1:1 connection between Power BI workspace and ala workspace





# Configuration pre-requisites – Azure

- Register '**microsoft.insights**' resource provider in subscription
- Create Log Analytics Workspace
- **Owner** role on the Log Analytics Workspace for:
  - The user who will set up Log Analytics integration in Power BI
  - The service principal '**Power BI Service**'
- Note: change the **retention period**



# Ad-blocker?

The screenshot shows a dark-themed web interface, likely a dashboard or activity log. At the top, there are several icons: a square, a document, a bell, a gear, a question mark, and a magnifying glass. To the right of these icons are the email address [inbox@daveruijter.nl](mailto:inbox@daveruijter.nl) and the name MARC LELIUVELD next to a blue profile picture. Below this header, the word "Notifications" is displayed in large, bold, blue and red letters, followed by a close button (X). Underneath, there is a link "More events in the activity log →" and a "Dismiss all" button with a dropdown arrow.

**! Failed to retrieve schema. Please try to refresh the page. ×**

We have experienced a connection issue while retrieving data. This is usually an indication that the network is down or a firewall or browser extension (such as an ad blocker) is mistakenly preventing access.

Connection Error

If issue persists, please open a support ticket. Request id: 256ea54b-65a6-4257-b248-a000e699c24e

a few seconds ago



# Example KQLs

```
// log count per day for last 30d  
PowerBIDatasetsWorkspace  
| where TimeGenerated > ago(30d)  
| summarize count() by format_datetime(  
TimeGenerated, 'yyyy-MM-dd')
```

```
// average query duration by day for last 30d  
PowerBIDatasetsWorkspace  
| where TimeGenerated > ago(30d)  
| where OperationName == 'QueryEnd'  
| summarize avg(DurationMs) by  
format_datetime(TimeGenerated, 'yyyy-MM-dd')
```

```
//query duration percentiles for a single day in 1 hour bins  
PowerBIDatasetsWorkspace  
| where TimeGenerated >= todatetime('2021-04-28') and  
TimeGenerated <= todatetime('2021-04-29')  
| where OperationName == 'QueryEnd'  
| summarize percentiles(DurationMs, 0.5, 0.9) by  
bin(TimeGenerated, 1h)
```

```
// refresh durations by workspace and dataset for last 7d  
PowerBIDatasetsWorkspace  
| where TimeGenerated > ago(30d)  
| where OperationName == 'CommandEnd'  
| where ExecutingUser contains 'system'  
| where EventText contains 'refresh'  
| project WorkspaceName, DatasetName = ArtifactName,  
DurationMs
```



# Using the template report

Power BI Log Analytics for Analysis Services Engine

This template will connect to the pre-configured Azure Log Analytics workspace where you are collecting Analysis Services engine log data. The template can be used to monitor engine load, engine usage trends, and identify slow queries and dataset refreshes. It provides summary and trend information at Workspace level with the ability to drill to detail about any individual query or dataset refresh. The template allows you to group or slice the data by useful properties such as Premium capacity, dataset, report, and user.

Days Ago To Start ⓘ:

Days Ago To Finish ⓘ:

Log Analytics Table ⓘ:

Log Analytics WorkspaceId ⓘ:

UTC Offset Hours ⓘ:

Pagination Hours ⓘ:



# GitHub location for .pbix

<https://github.com/microsoft/PowerBI-LogAnalytics-Template-Reports>



# Customizations

- Visual ID
- Report Name
- DirectQuery



# DirectQuery on the trace info

- Which datasets are refreshing right now?
- Which recent datasets refreshes succeeded, and which ones failed?
- If a dataset refresh failed, why did it fail?
- How long did these refreshes take?



# Roadmap

- GA → March?
- Removing the 1:1 workspace mapping restriction
- No tenant level configuration to be expected
- Capacity level configuration to be expected
- API control to configure multiple workspaces at once



# — Demo Log Analytics Integration

# Wrap-up

Design and implement	Orchestrate	Performance	Cost
Better design and implement complex data models, including hybrid tables, aggregations, and combined storage modes (import, DirectQuery , dual).	Orchestrate the end-to-end data processing, with a pipeline chain from data ingest in the data lake house to the incremental Power BI dataset refresh.	Use different techniques to identify performance bottlenecks in your solutions and how to solve those ("does it fold"?).	Implement a cost-efficient solution, that still meets the scalability demands.

# Any bonus questions?



Vul alsjeblieft  
de evaluatie in

Bedankt!  
Thank you!



Please fill in  
the evaluation



Power BI Gebruikersdag 2022

#pbig2022

