Microsoft

Microsoft Fabric
COMMUNITY CONFERENCE

Low friction data transformation and data movement using Fabric Dataflows

Benni De Jagere

Microsoft

Benni De Jagere

Senior Program Manager | Fabric Customer Advisory Team (Fabric CAT)

Fabric CAT

dataMinds .be Member

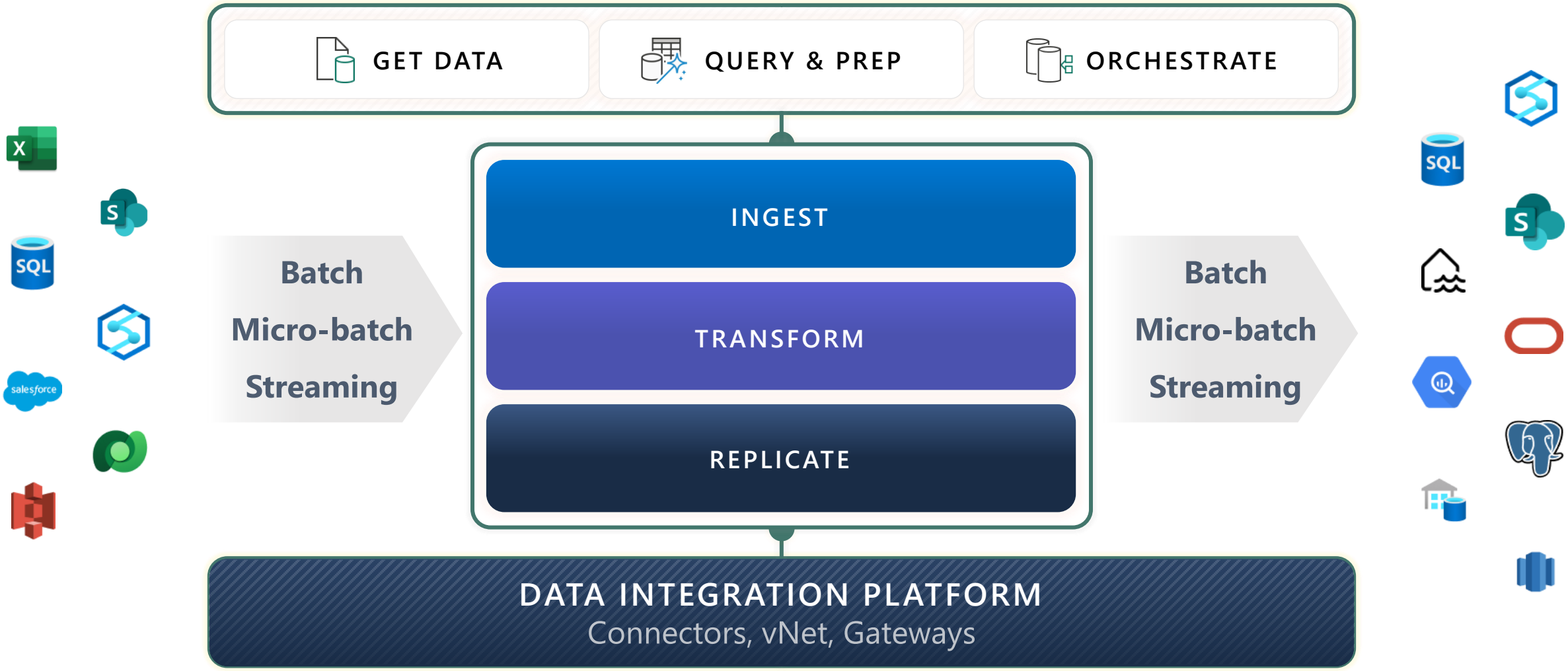
@BenniDeJagere

/bennidejagere

sessionize /bennidejagere

/bennidejagere

Microsoft Data Integration

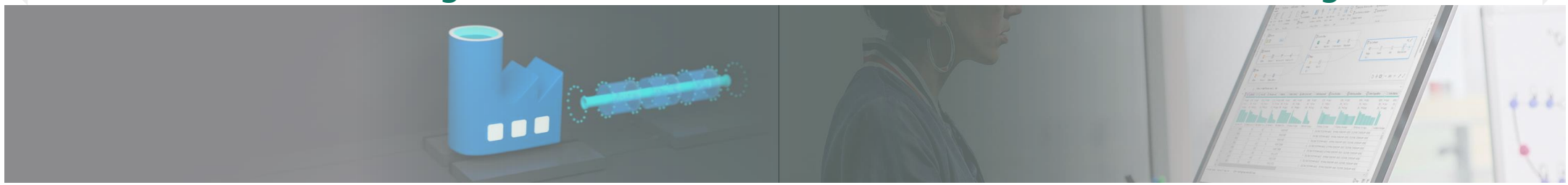


Microsoft Data Integration

Products

Professional Data Integration

Citizen Data Integration



Azure Data Factory, Azure Synapse Analytics, SQL Server Integration Services

- Fully managed, with serverless data integration services
- Visually integrate data sources with more than 100 built-in connectors
- Easily construct ETL and ELT processes code-free

Power Query

- Seamlessly integrated into many popular Microsoft products
- An easy to use, engaging, no-code experience
- Includes powerful and smart AI-based data preparation

Microsoft Data Integration

Unified Product Portfolio

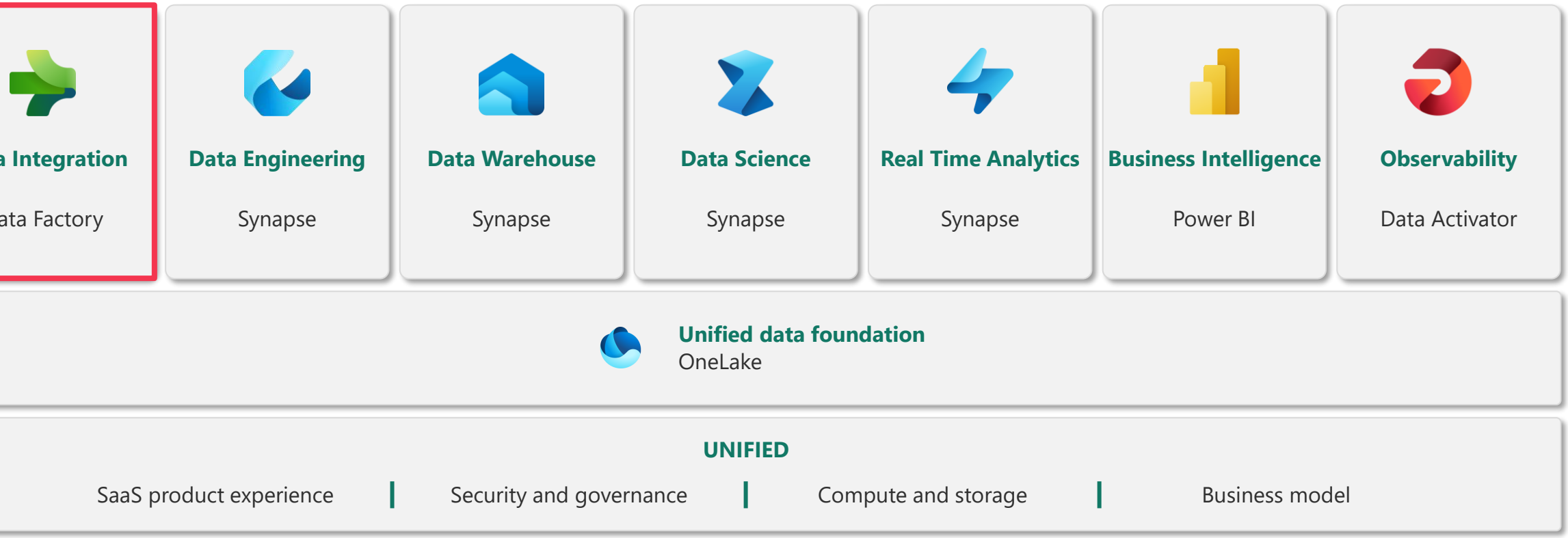
Professional & Citizen Data Integration

Data Factory in Microsoft Fabric

- Brings together the best of Power Query and Azure Data Factory, into a modern data integration experience
- Empowers both professional and citizen developers
- Ingest and transform data as well as orchestrate data workflows
- Data Factory enables everyone to connect to diverse data sources and to bring that data to where it can best help derive insights for better business decisions.

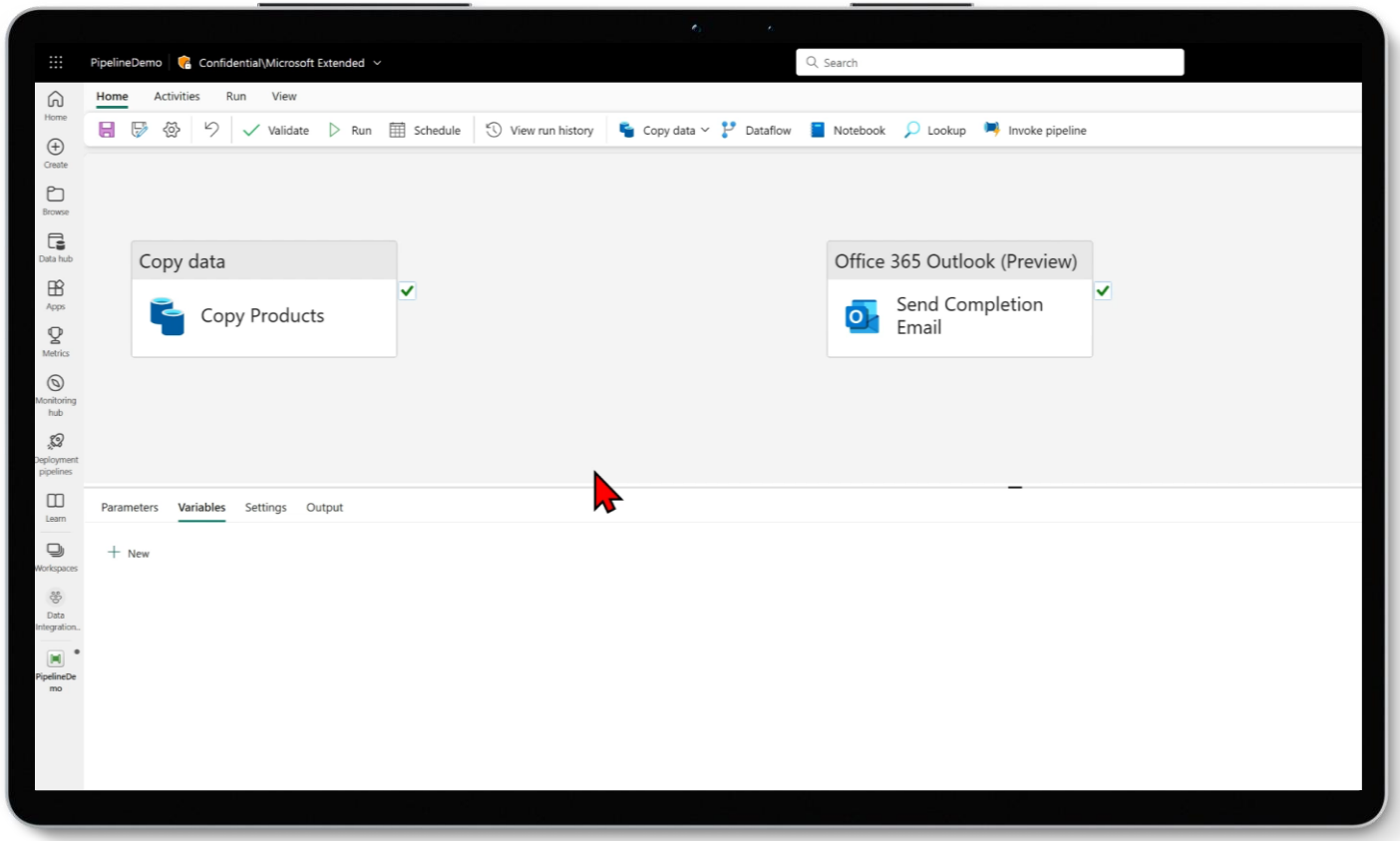
Microsoft Fabric does it all—in a unified solution

An end-to-end analytics platform that brings together all the data and analytics tools that organizations need to go from the data lake to the business user



Data Factory in Microsoft Fabric

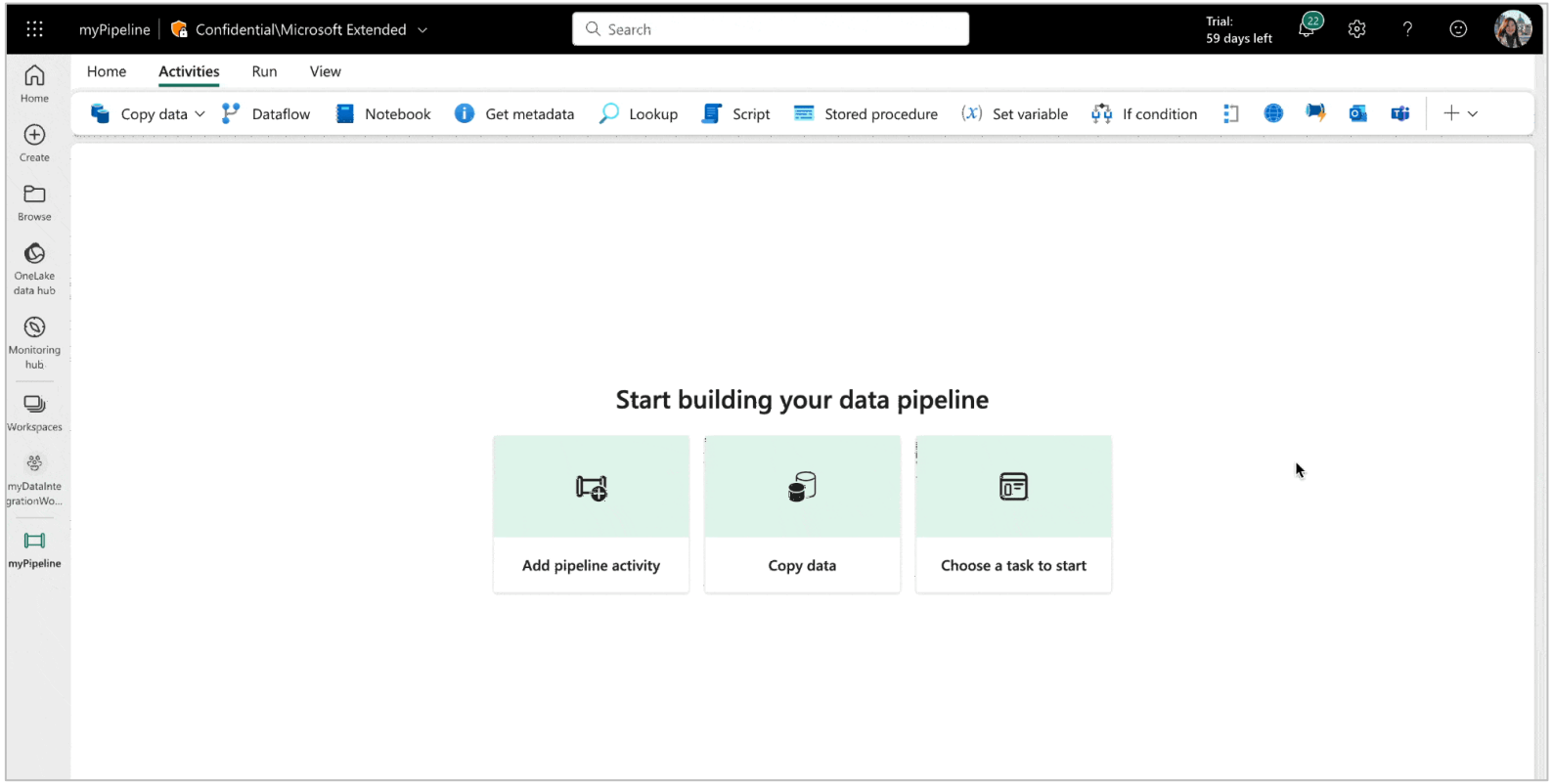
Data Factory converges our data capabilities into a single SaaS interface to provide **the world's most complete data integration experience.**



Data pipeline

Ingest and orchestrate activities at scale

Familiar authoring canvas experience

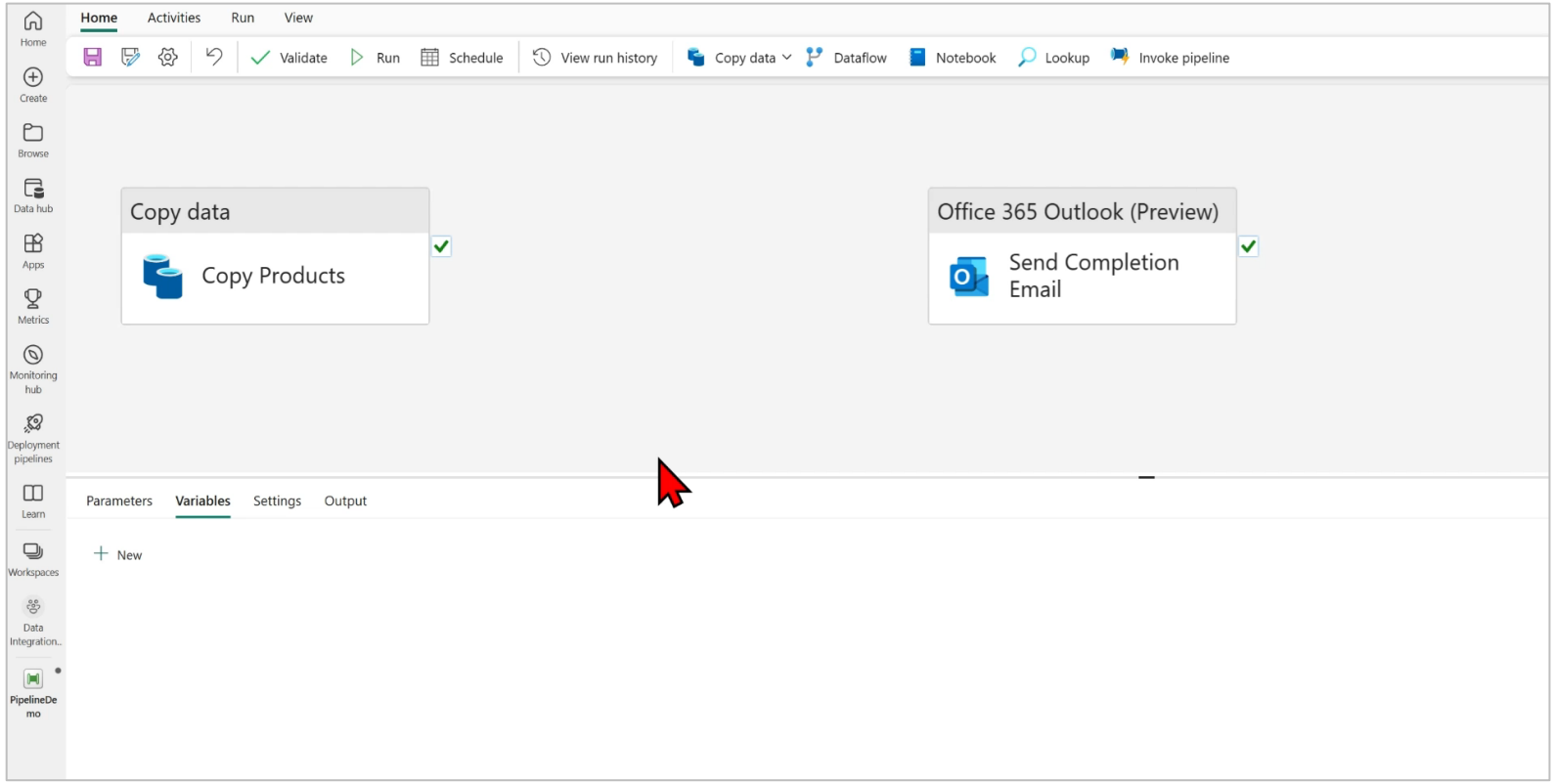


OneLake

Data pipeline

Ingest and orchestrate activities at scale

Empower every person to integrate data



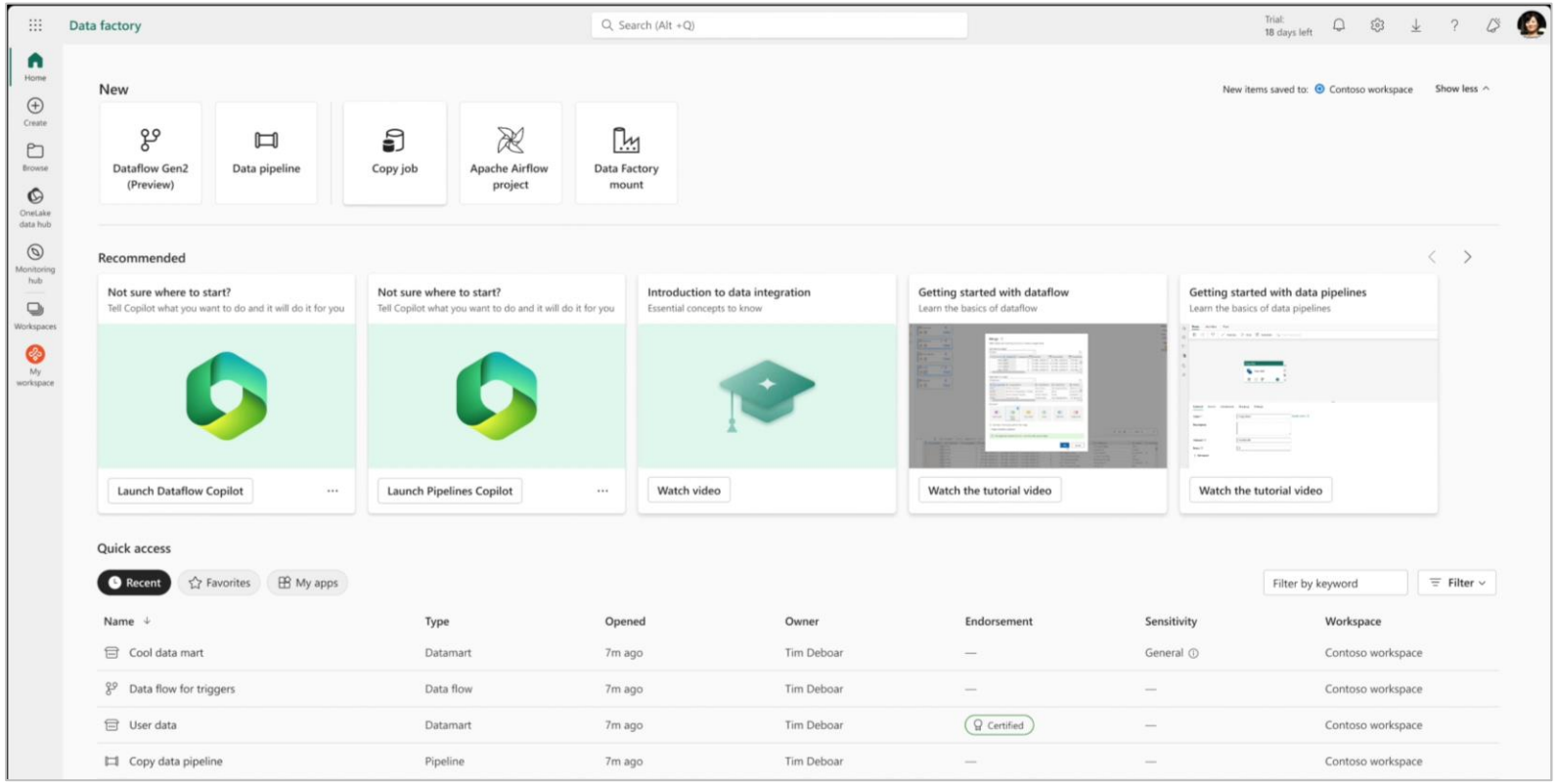
OneLake

Copy job

Easily ingest and move data at scale

Public Preview

Supports both batch and near real-time incremental copy (CDC)

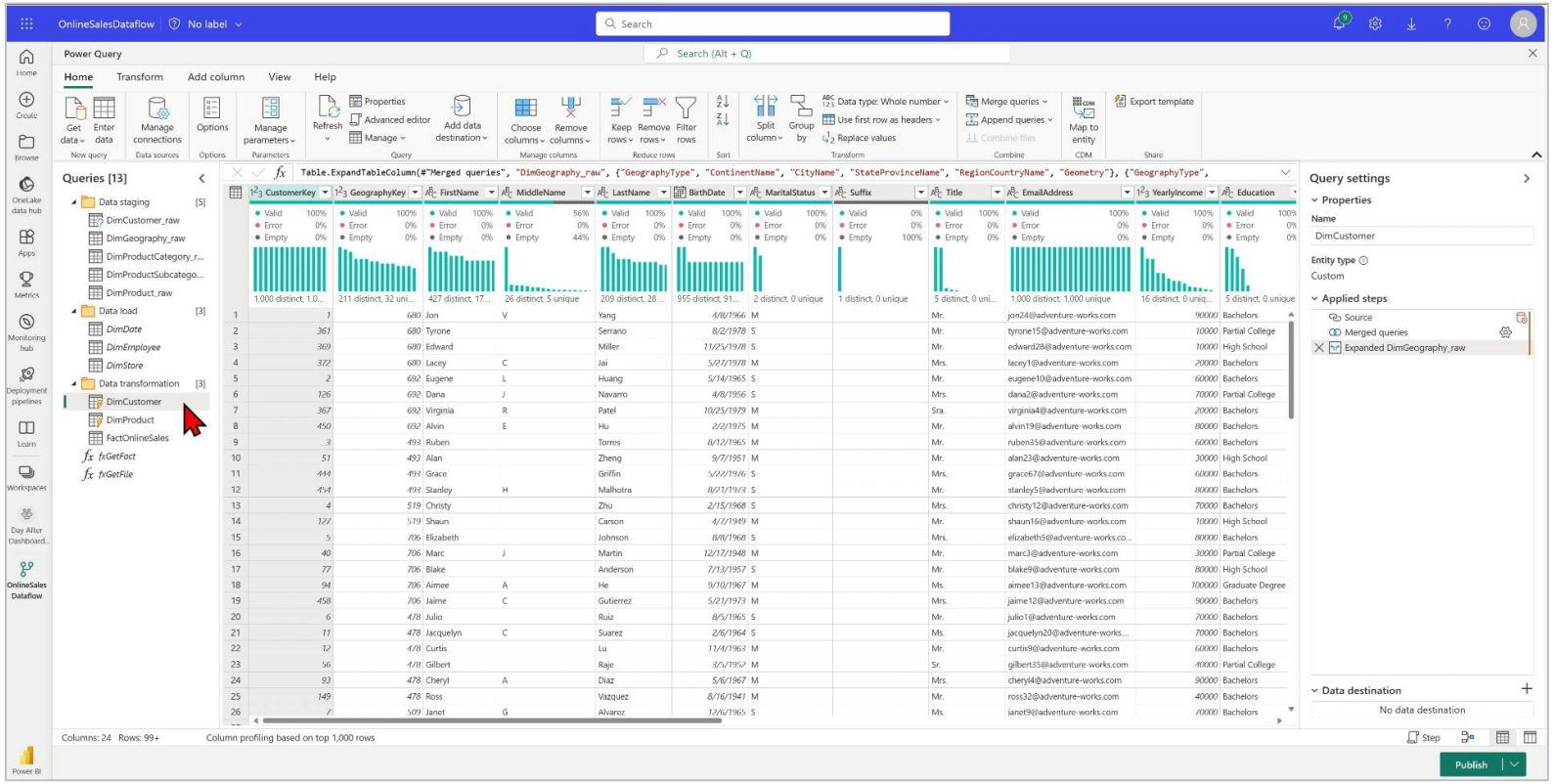


OneLake

Dataflow Gen 2

Ingest and transform data at scale

Enterprise-scale data ingestion and transformation

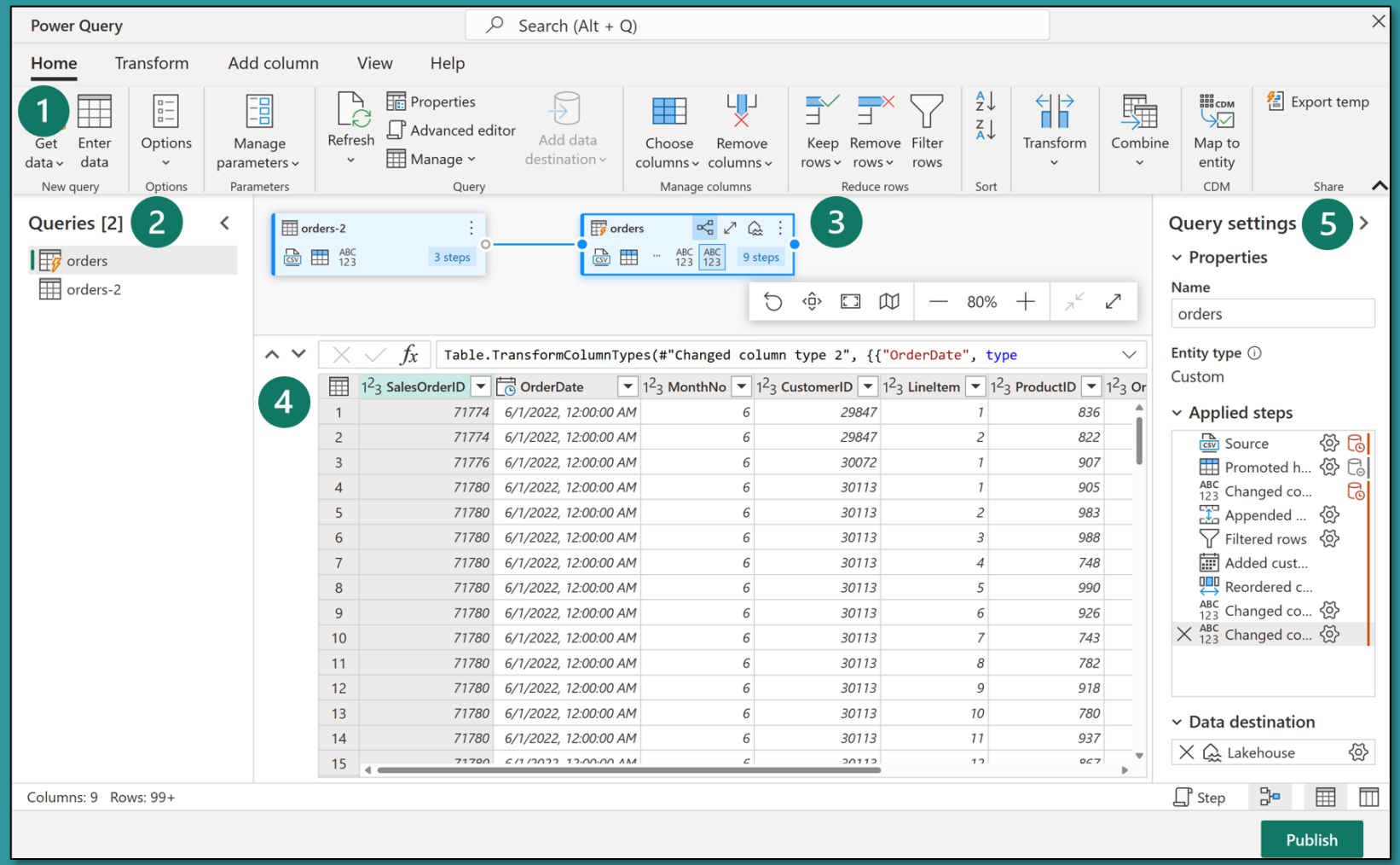


OneLake

Exploring Dataflows Gen2

Navigation

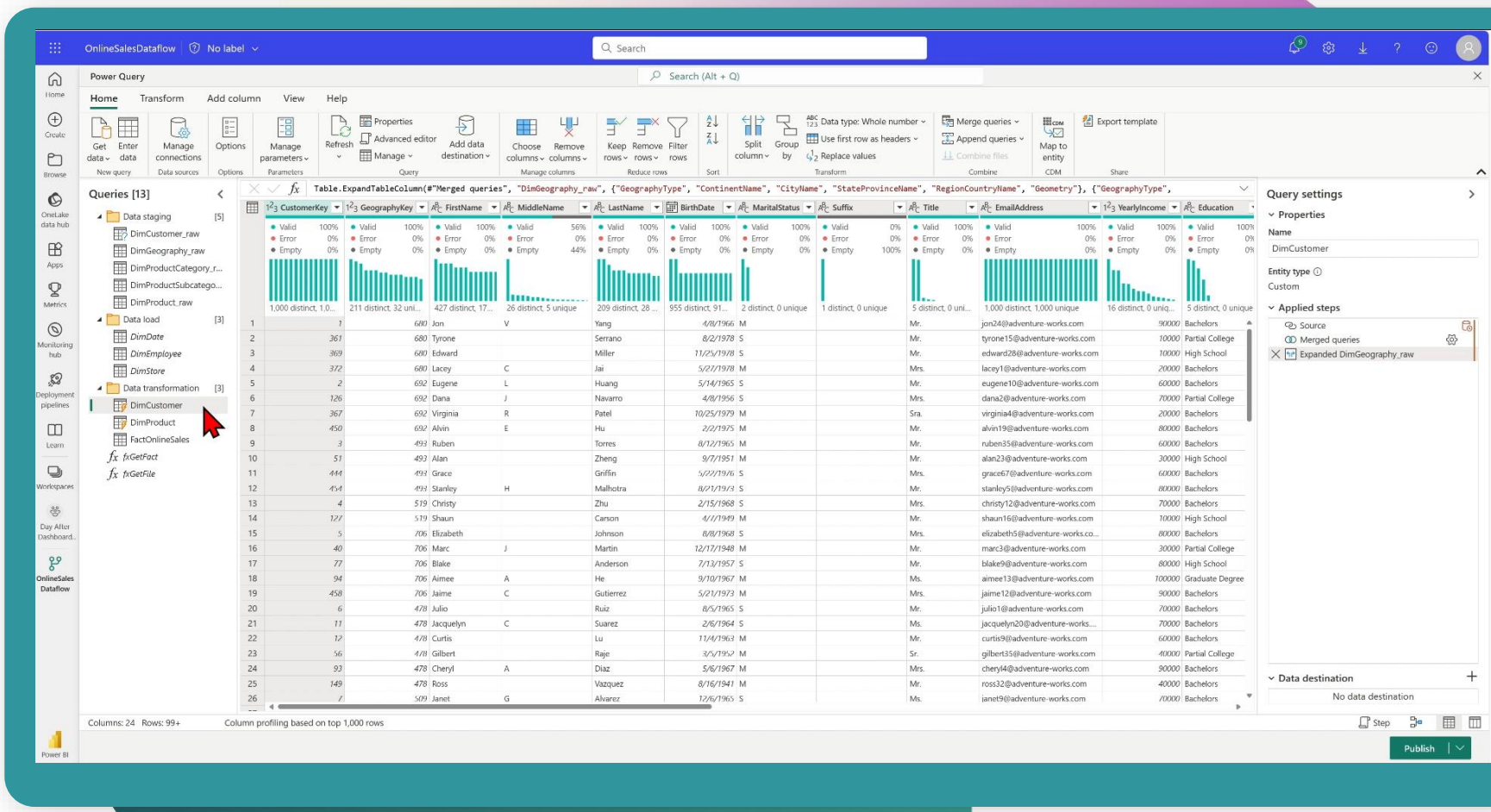
1. Command ribbon
2. Queries pane
3. Visual diagram
4. Data preview grid
5. Query settings



Dataflow Gen2

Next generation of **data preparation**

- **Easy to use**, no-code ETL & ELT
- Includes **smart AI**-based data prep
- More than **300+ transformations**
- **Output data destinations**
  Write output of dataflows to Azure SQL database, Data warehouse, Lakehouse and more

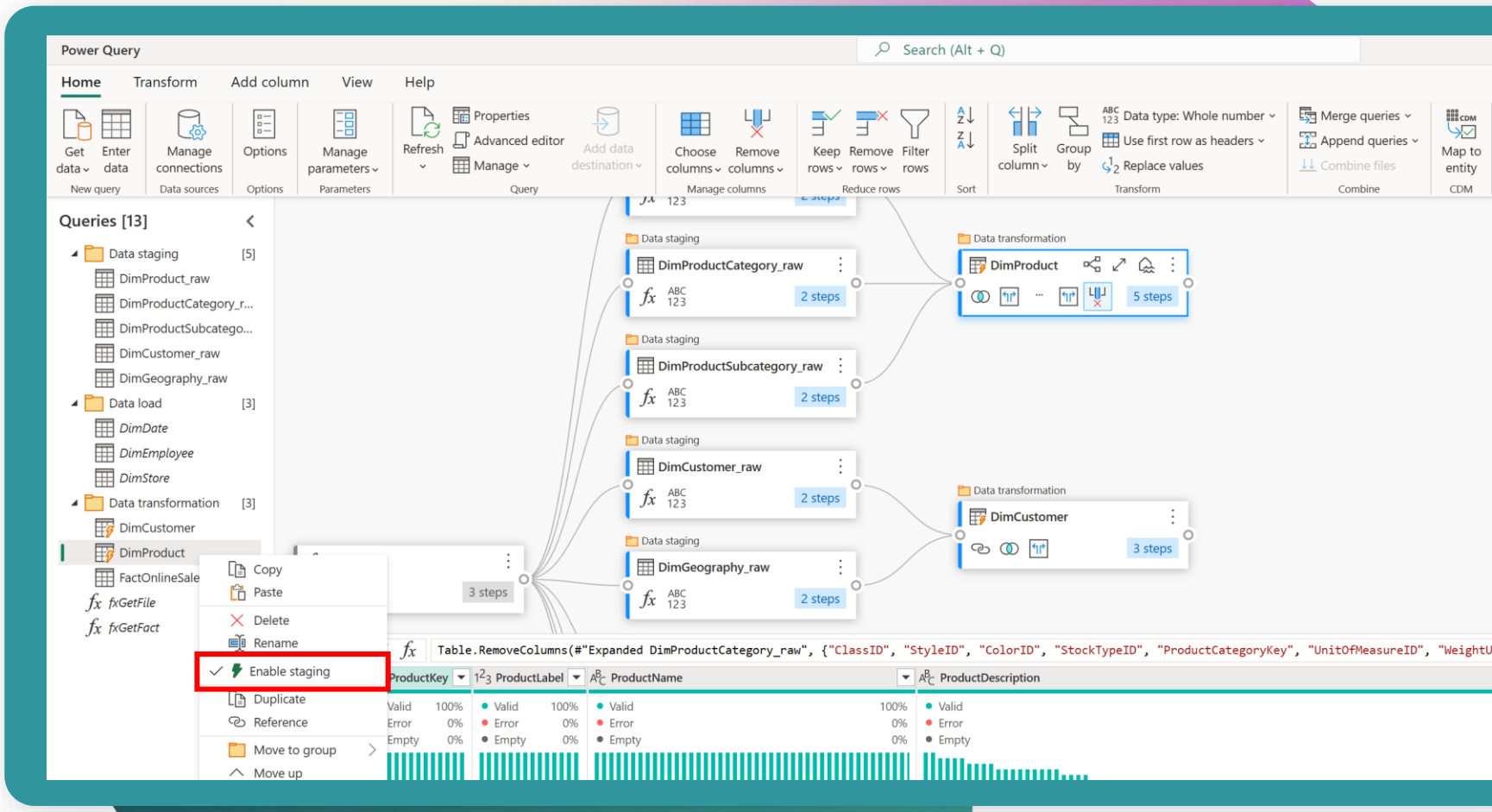


Dataflow Gen2 staging

Highly scalable using **Fabric compute**

A **seamless experience** - yielding fast, easy and powerful results

Abstracts away the complexities of traditional ETL and ELT



*Previously titled “Enable load”

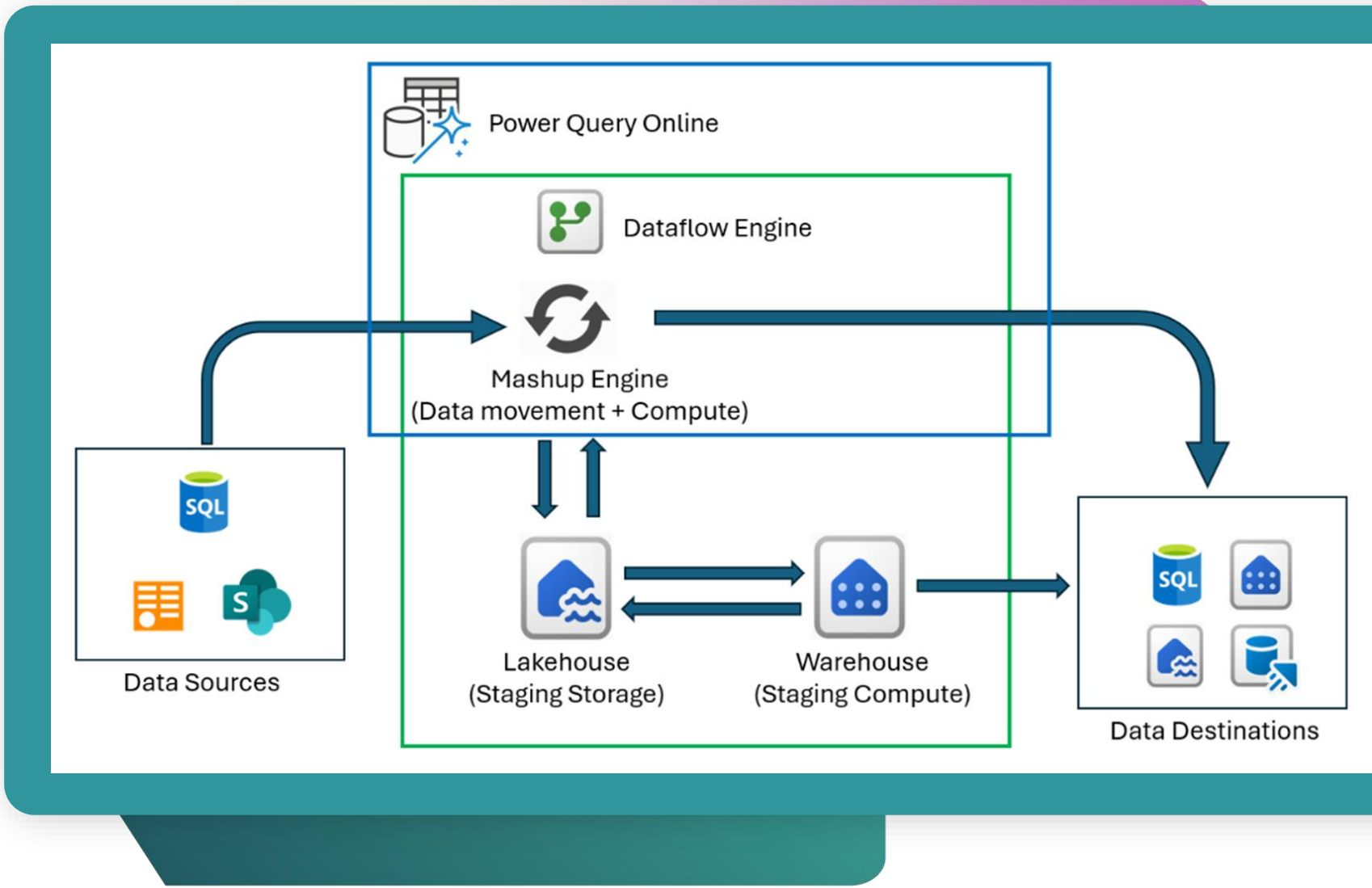
Dataflow Gen2 staging

Optimize the use of dataflows with Fabric compute

1. Connect to your data and **copy** it into Fabric using the ***Enable staging** option
   (***On by default**)

2. Create a **reference** query in a new query.

3. Apply transformation steps to the **computed** table for complex ETL operations such as join, distinct, filter and group by – leveraging Fabric compute.



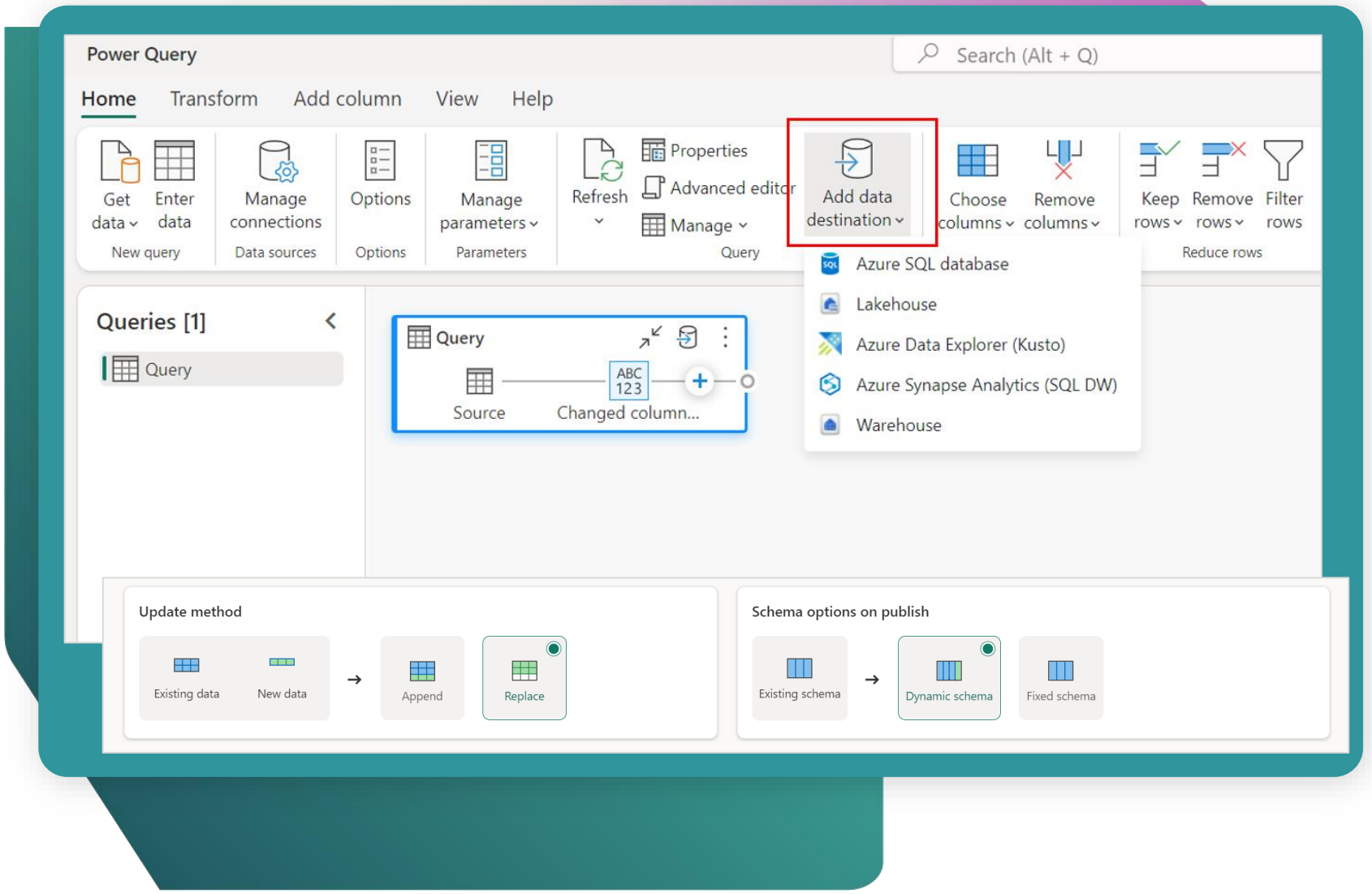
Data destinations

Supported destinations:

- Lakehouse
- Warehouse
- Eventhouse
- Azure SQL database
- Azure Synapse Analytics

Update methods:

- Replace
- Append



MS Learn | Dataflow Gen2 data destinations and managed settings

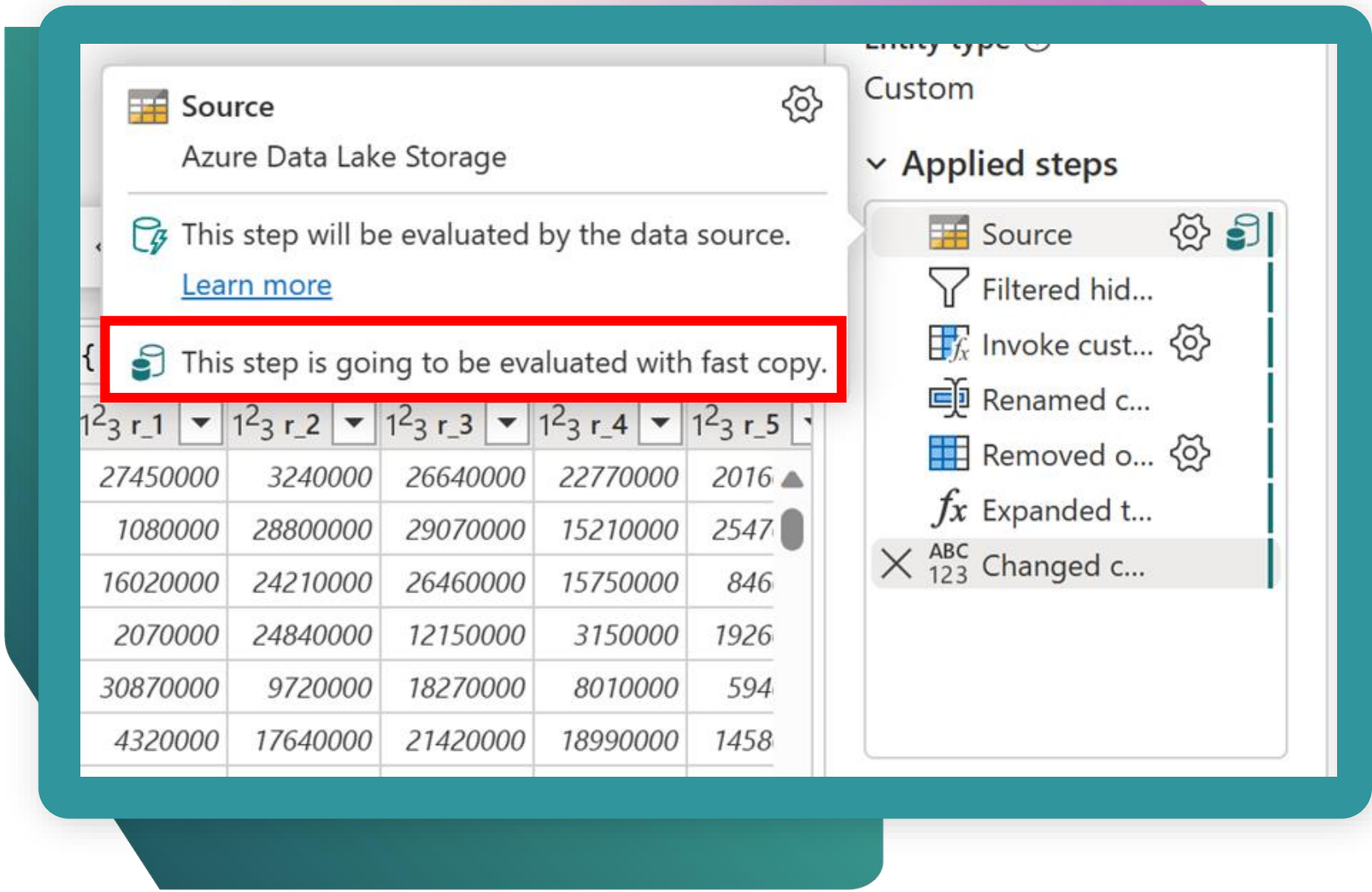
Fast Copy

General Availability

Ingest gigabytes of data **effortlessly** with dataflows, powered by the scalable backend of the Copy activity

A limited set of transformations are supported:

- Combine files
- Select or remove columns
- Change data types
- Rename a column



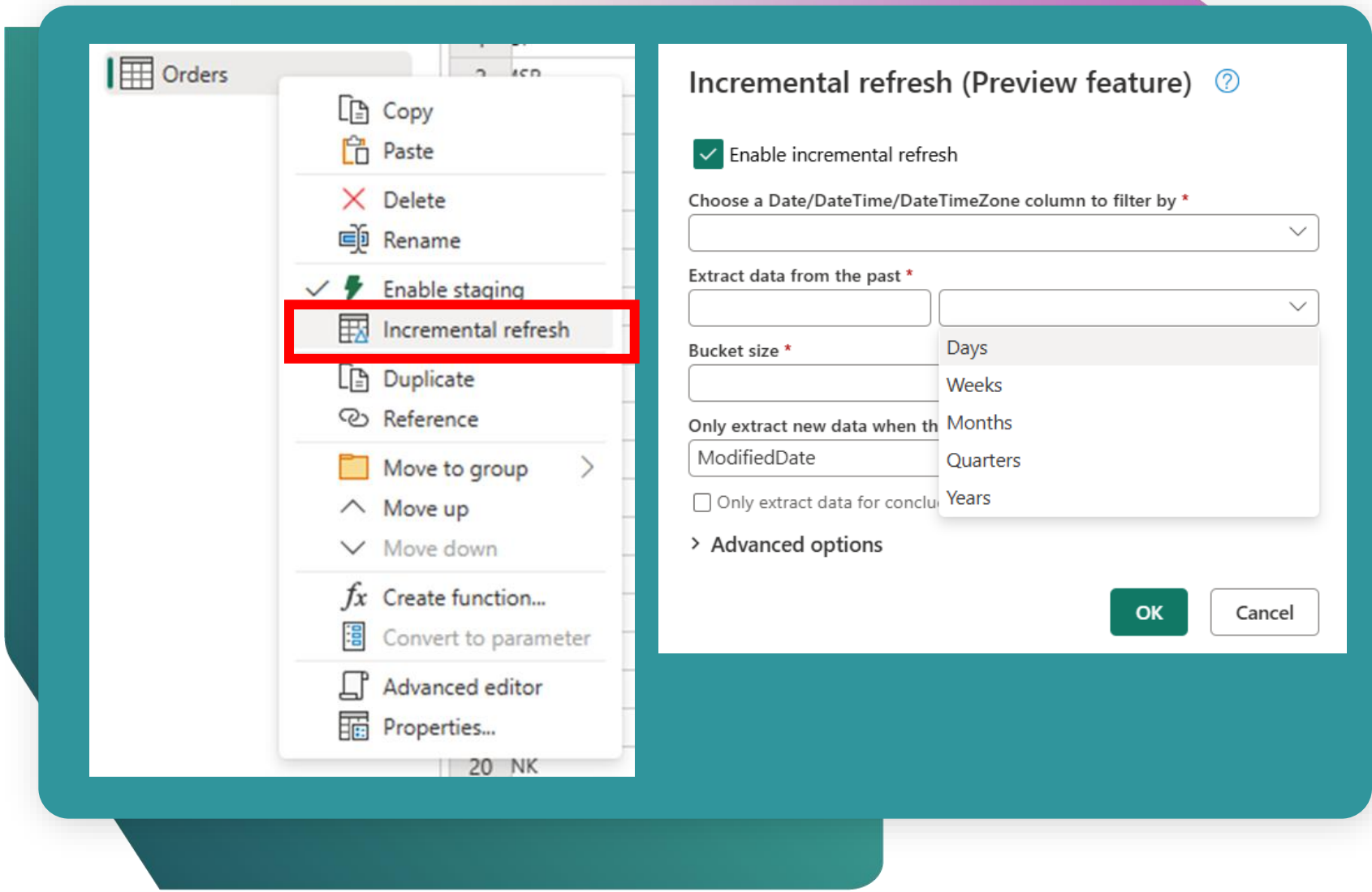
MS Learn | Fast Copy in Dataflows Gen2

Incremental refresh

Public Preview

Process only changed data since the last refresh to **save time and resources**

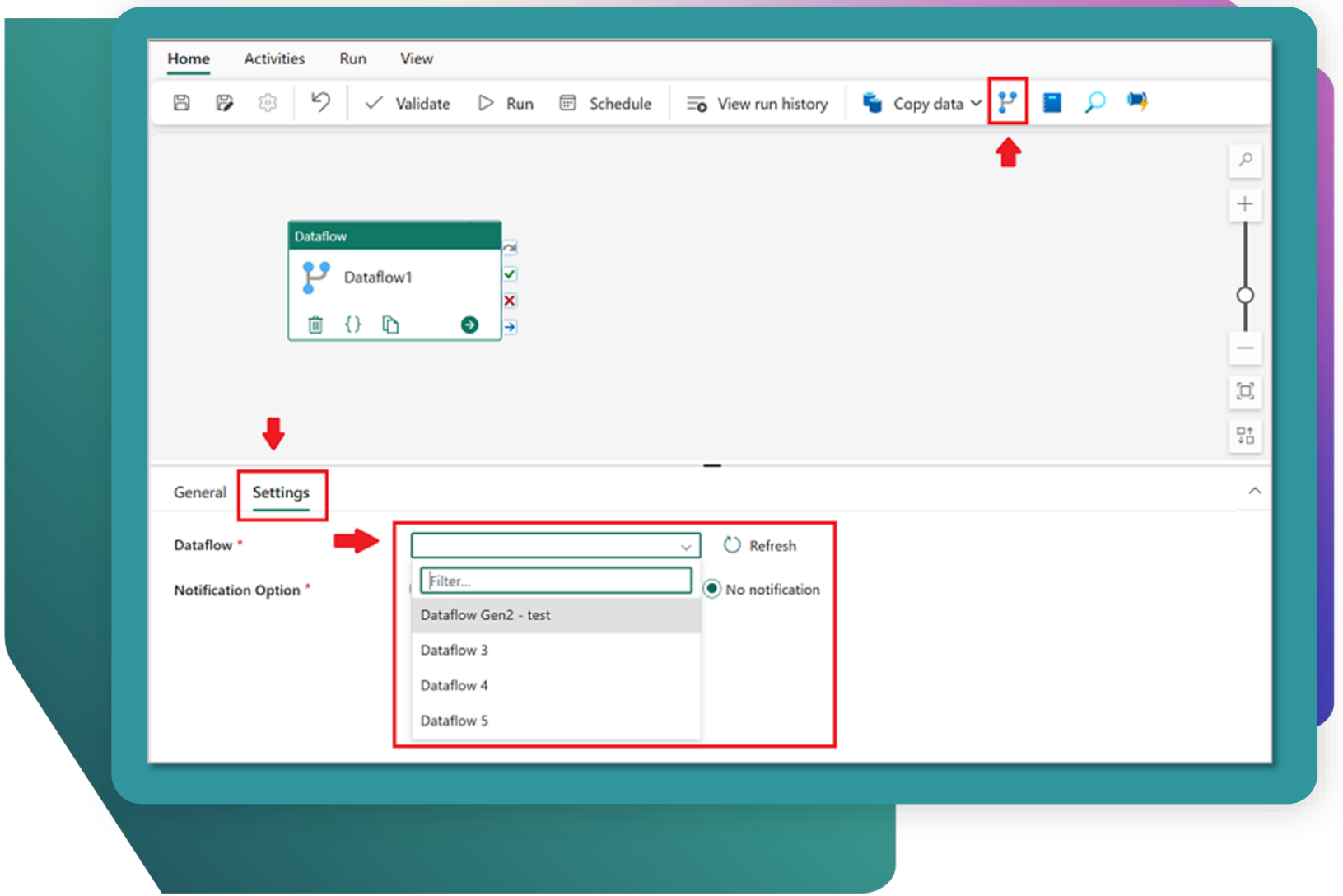
- Evaluate changes by comparing the maximum **DateTime** value with the previous refresh.
- Retrieve and load data for changed buckets in parallel, loaded to staging
- Replaces the destination data with the new data, affecting only updated buckets.



Orchestrating dataflows gen2 with data pipelines

A dataflow for data ingestion and transformation, and landing into a lakehouse using dataflows

Then incorporate the dataflow into a pipeline to orchestrate additional activities



Dataflow Gen2 scale recommendations

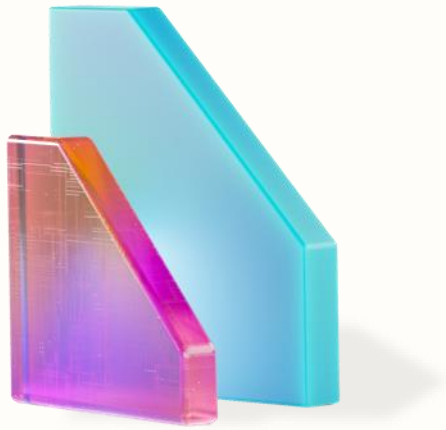
Separate **dimension** tables and **fact** tables into separate dataflows based on update method

Separate tables **with staging** and **without staging** into separate dataflows

Separate tables **with fast copy** support and **without fast copy** support into separate dataflows

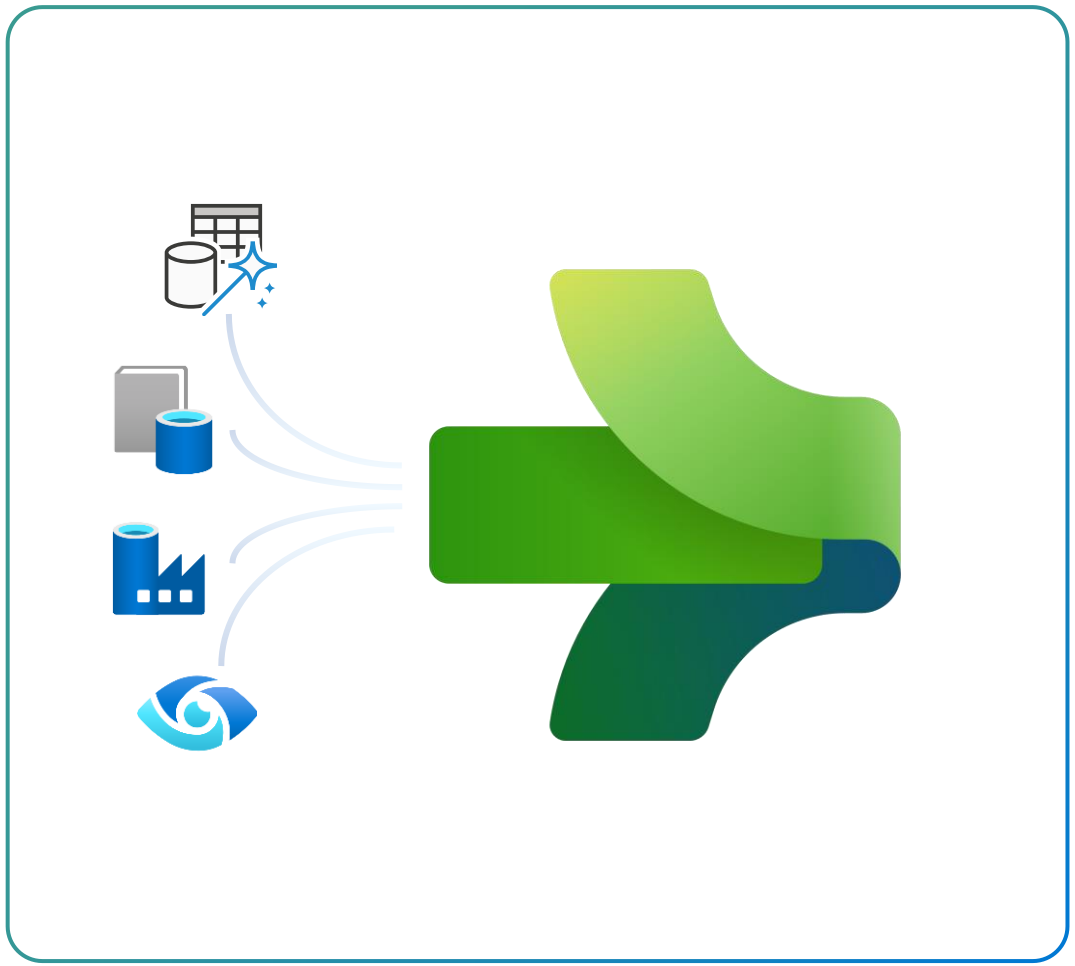
Separate tables based on the data source and if they **support query folding** or do **not support query folding**

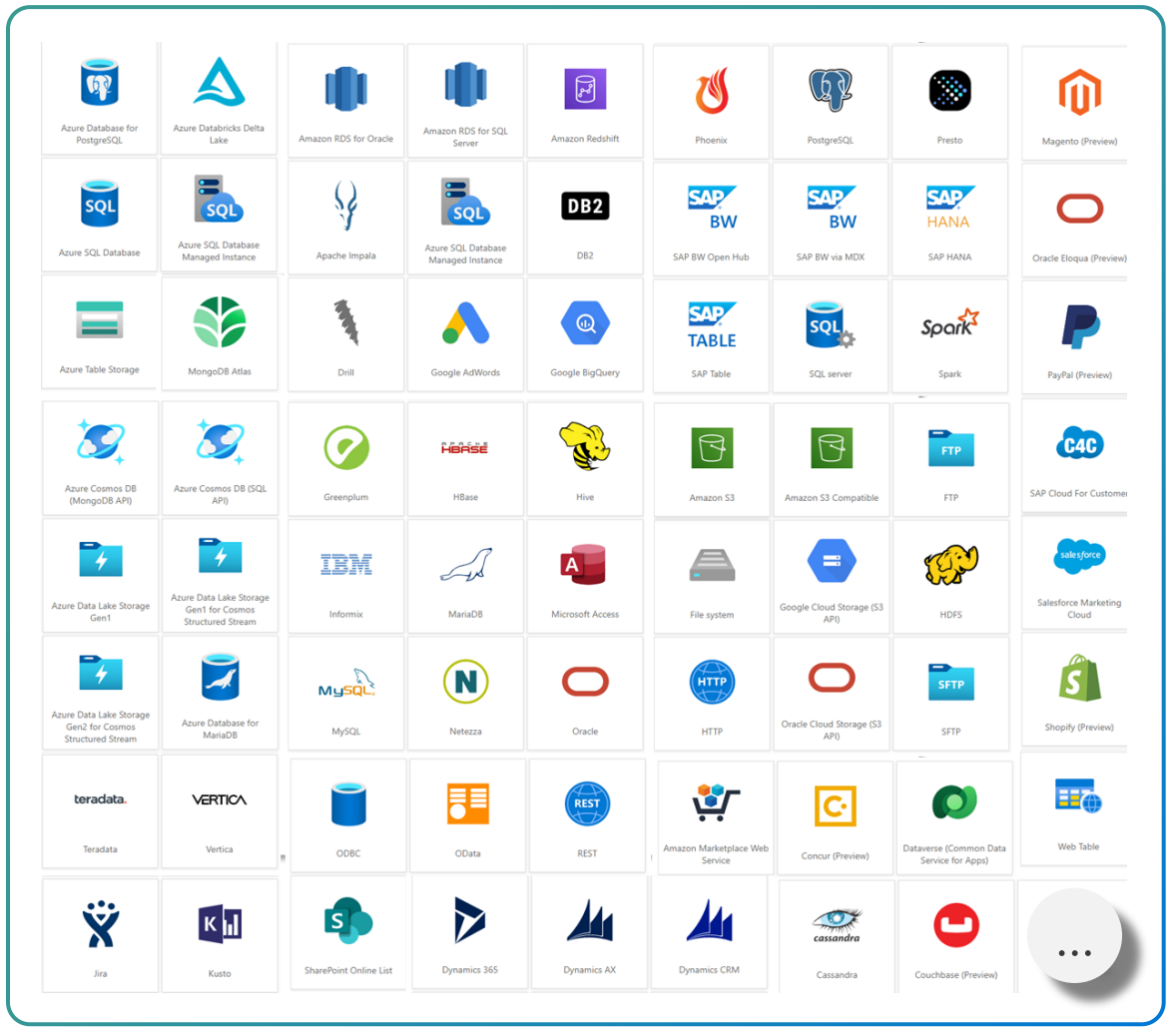
Use **copy job, mirroring** or **data pipelines** to ingest data and dataflows to transform



Unifying data in OneLake

Seamlessly connect to more than 170+ data sources

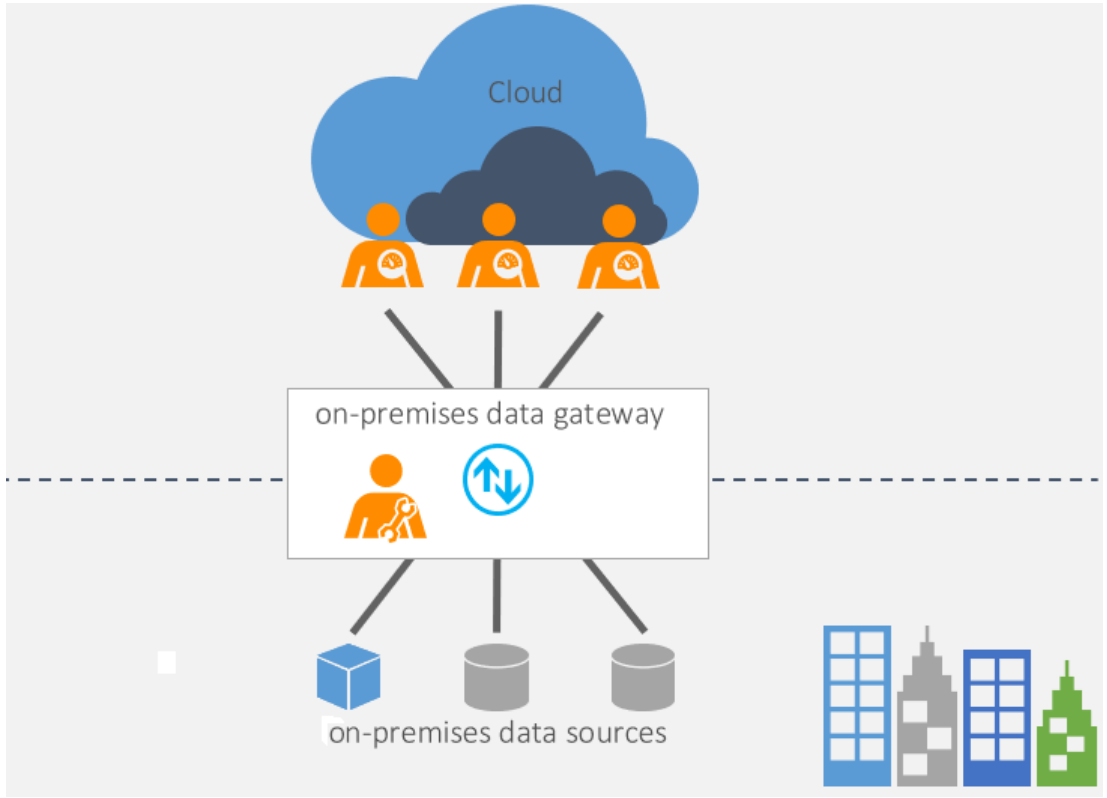




On-premises data gateway

Secure connectivity to on-premises and private endpoints

Bridge between your secure data sources and the Microsoft cloud

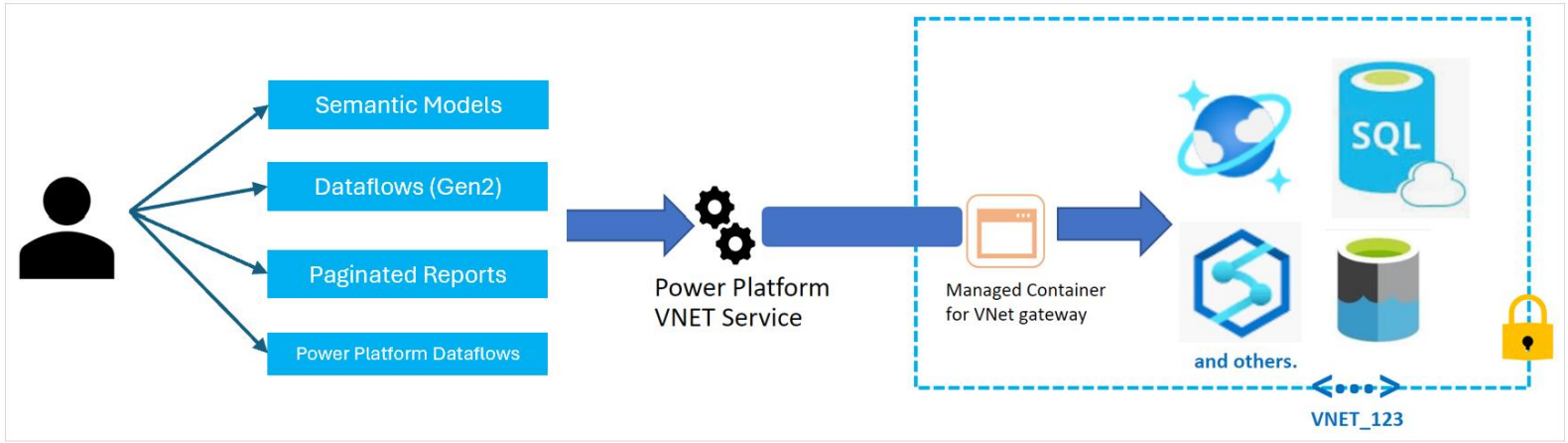


OneLake

VNet data gateway

Secure connectivity to your Azure and private data services

Bridge between your secure data sources and the Microsoft cloud



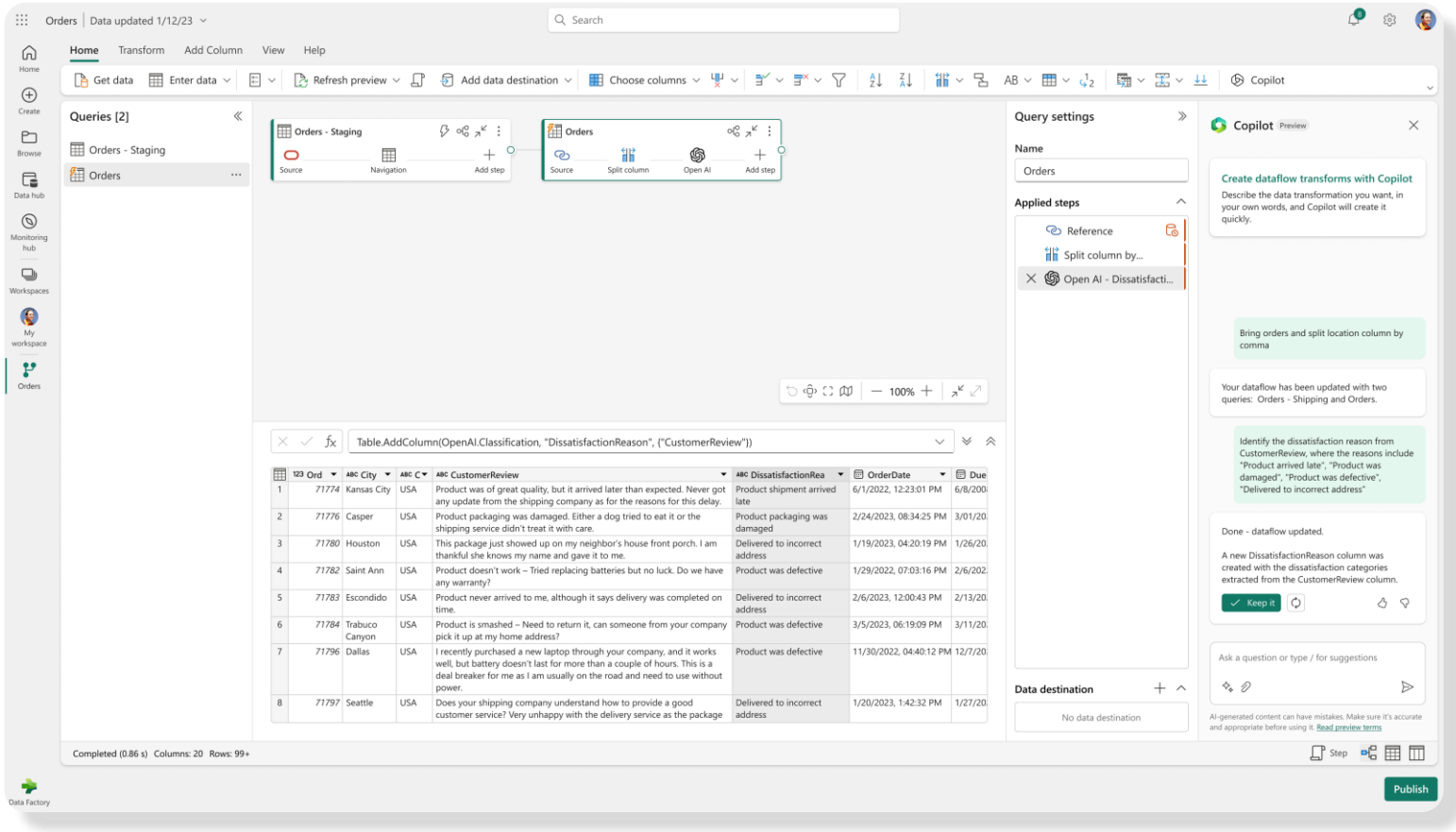
OneLake

Data integration + AI

Copilot in Data Factory

Easily integrate generative AI into your dataflows and *pipelines using Copilot

- Chat with **Copilot** to transform data using natural language
- Tap into generative AI capabilities from **Azure Open AI** as data transformation steps
- Use **Copilot** to describe your data transformations



Copilot skills

Create new transformations

- Add a single transformation step to an existing query
- Chain multiple transformation steps to an existing query

Create a new query

- Create a new query from scratch or by referencing existing data

Explain a single step

- Generates a description of a transformation step

Explain the full query

- Generates a description of a complete query and its steps

DEMO

Creating a…

- Lakehouse
- Dataflow Gen2

Migrate from Power BI dataflow to Fabric dataflow gen2

Feature	Power BI Dataflow	Dataflow Gen2
Incremental refresh	Yes	**YES!**
Accessible file outputs	*No	Yes
Fast copy	No	Yes
Data destination output	No	Yes
Premium capacity required	No	Yes
AI Insights	Yes	*No
AutoML	Deprecated	Deprecated
Attach Common Data Model (CDM) folder	Yes	No
Linked Tables	Yes	No (use Shortcuts)
On-premises data gateway	Yes	Yes
Vnet data gateway	No	Yes

* Supported by external services / REST APIs

Microsoft Fabric
COMMUNITY CONFERENCE

Smart data preparation and deeply integrated experiences

Add column from examples

Transform, combine, extract or enrich your data, in one or more columns within the Power Query Editor tables.

Simply specify a few output values for your new columns and Power Query generates the right column generation logic for you.

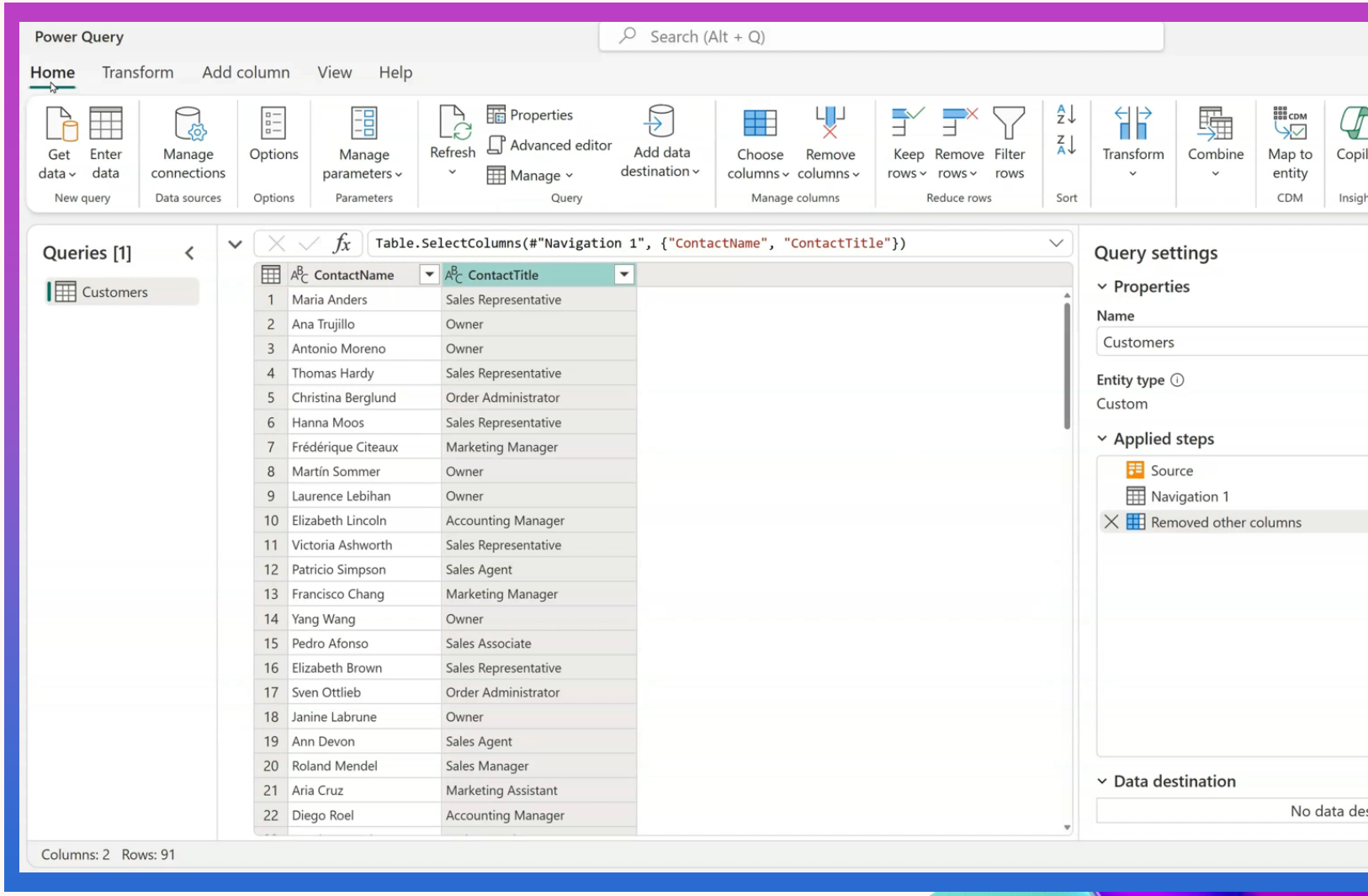


Table by example (Web)

Extract any data from any HTML page in the world. Just preview the source page and specify output values.

Power Query will do the time-consuming work and extract all the appropriate data for you.

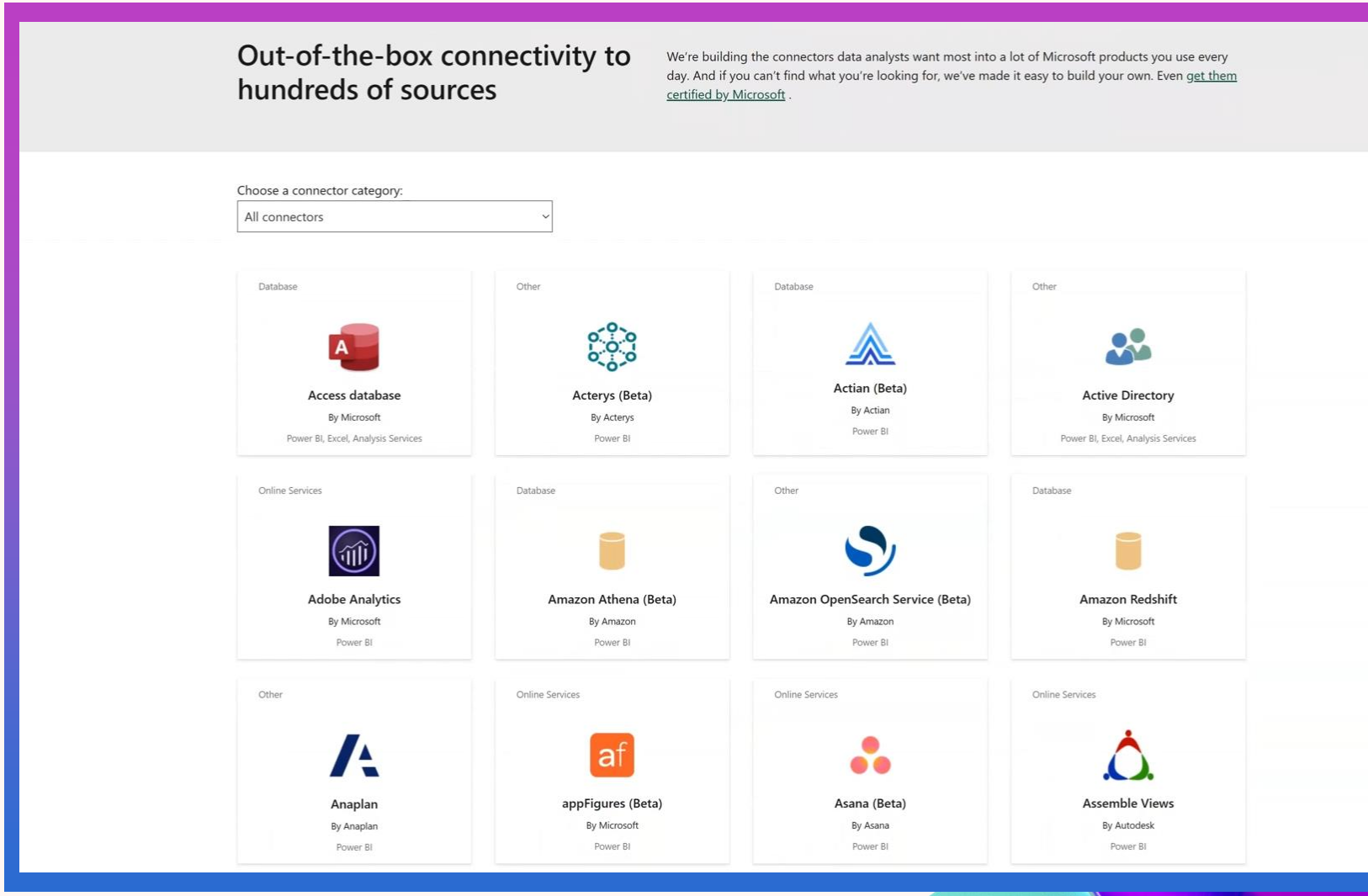
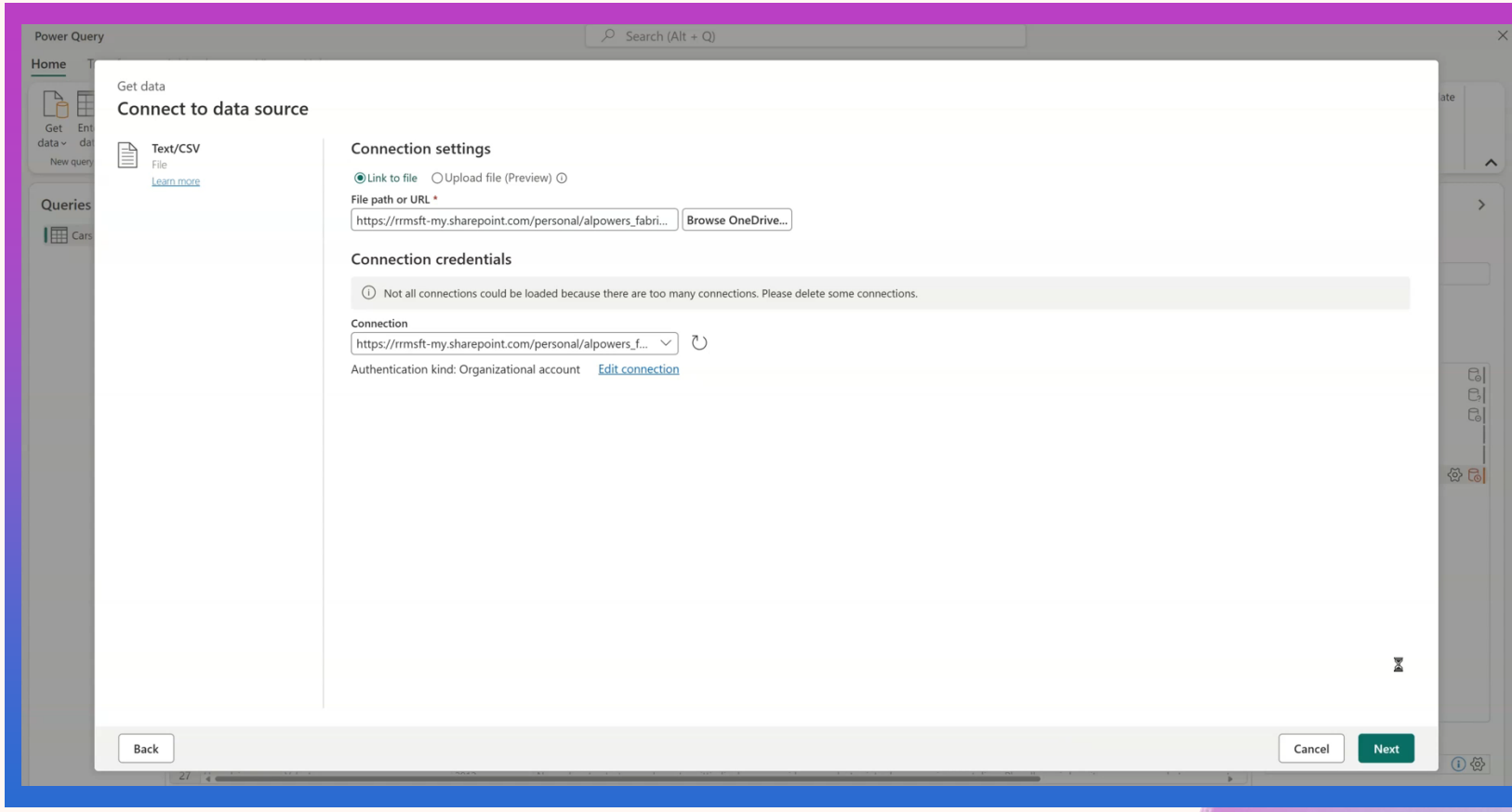


Table by example (Text/CSV)

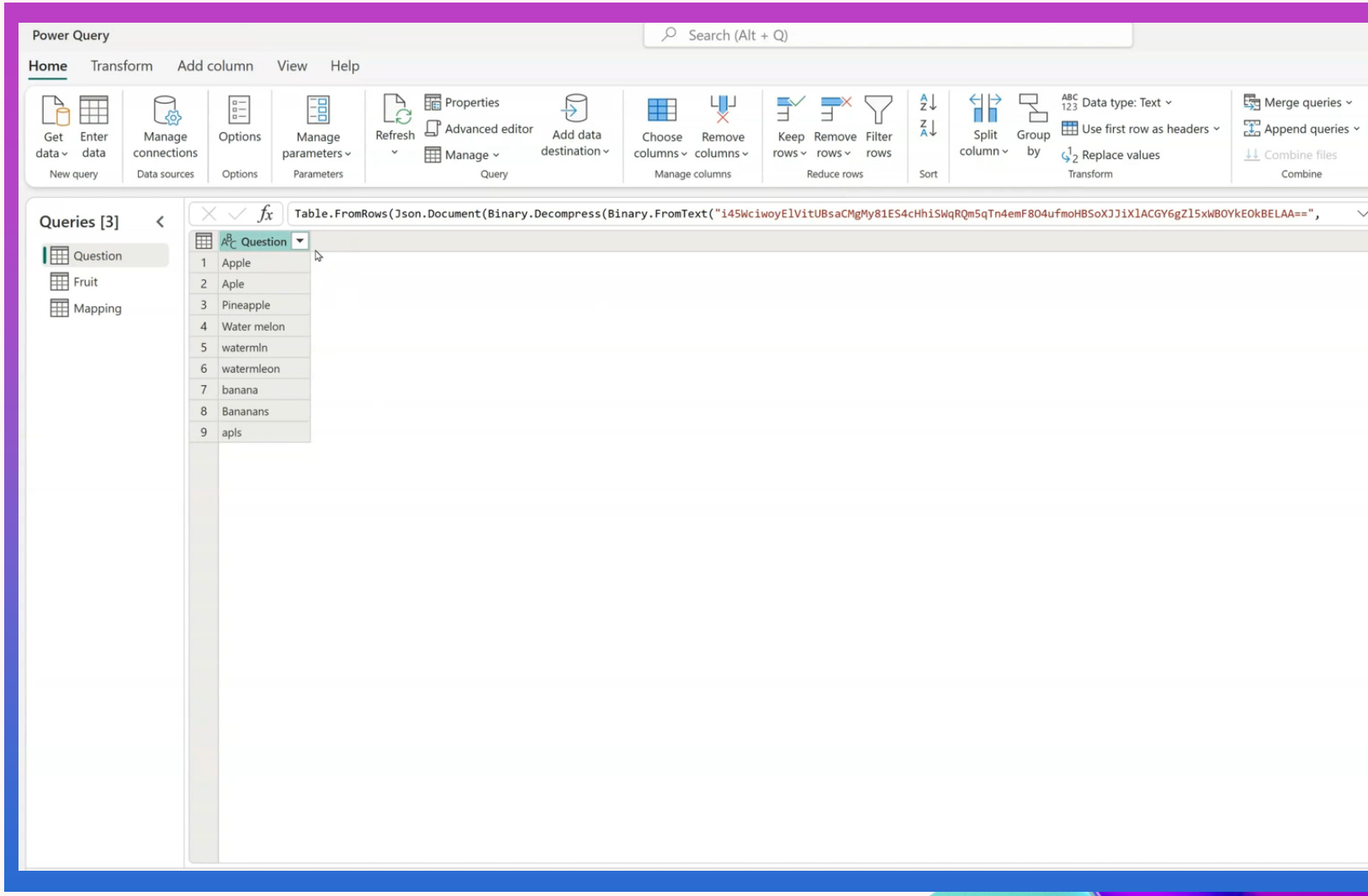
Extract data from a Text/CSV file. Just specify sample output values from the source.

Power Query will do the time-consuming work and extract all the appropriate data for you.



Fuzzy merge

Power Query's built-in Fuzzy Matching algorithm lets you to merge multiple tables using an approximate match to correlate things like slightly different versions of product names, customer names or address information—to name a few examples.



Data profiling

- Inline Column Quality bars

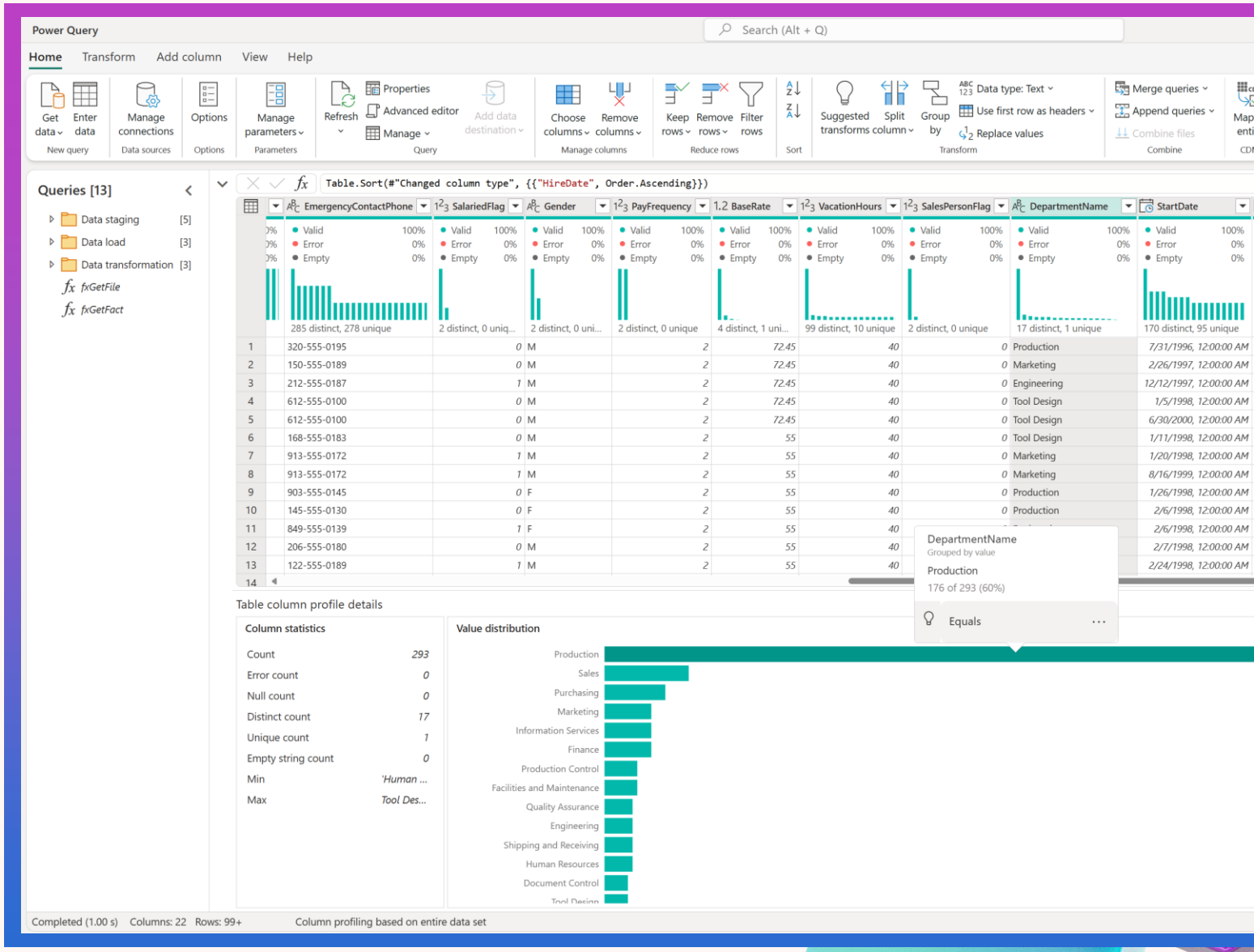
  Quickly spot erroneous or empty values across all your columns.

- Inline Value Distribution histograms

  Understand the number of different values, which are unique, and most/least common.

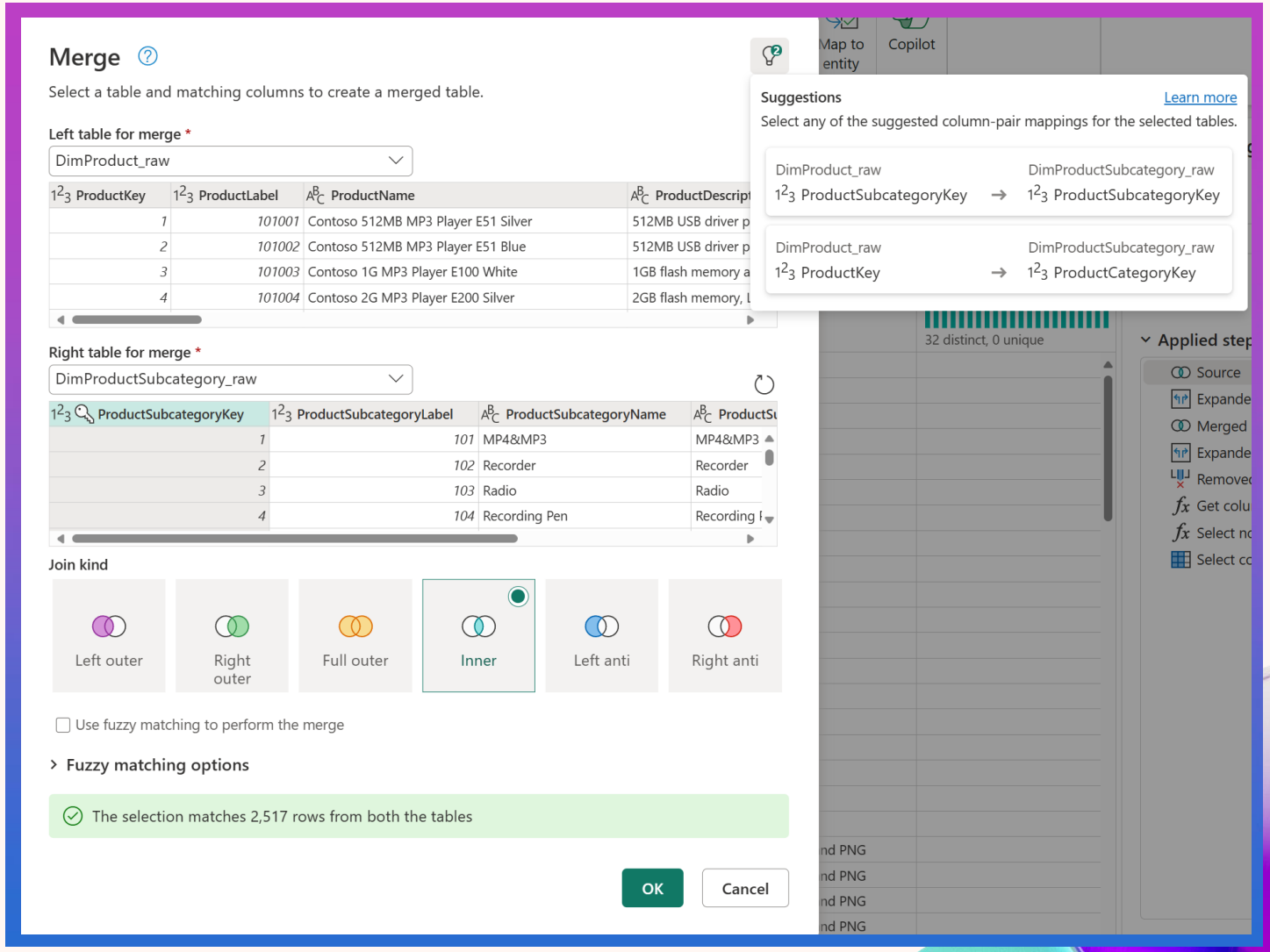
- Detailed Column Profiles pane

  Select a specific column and dig deeper into a profile to fully understand what's in the data.

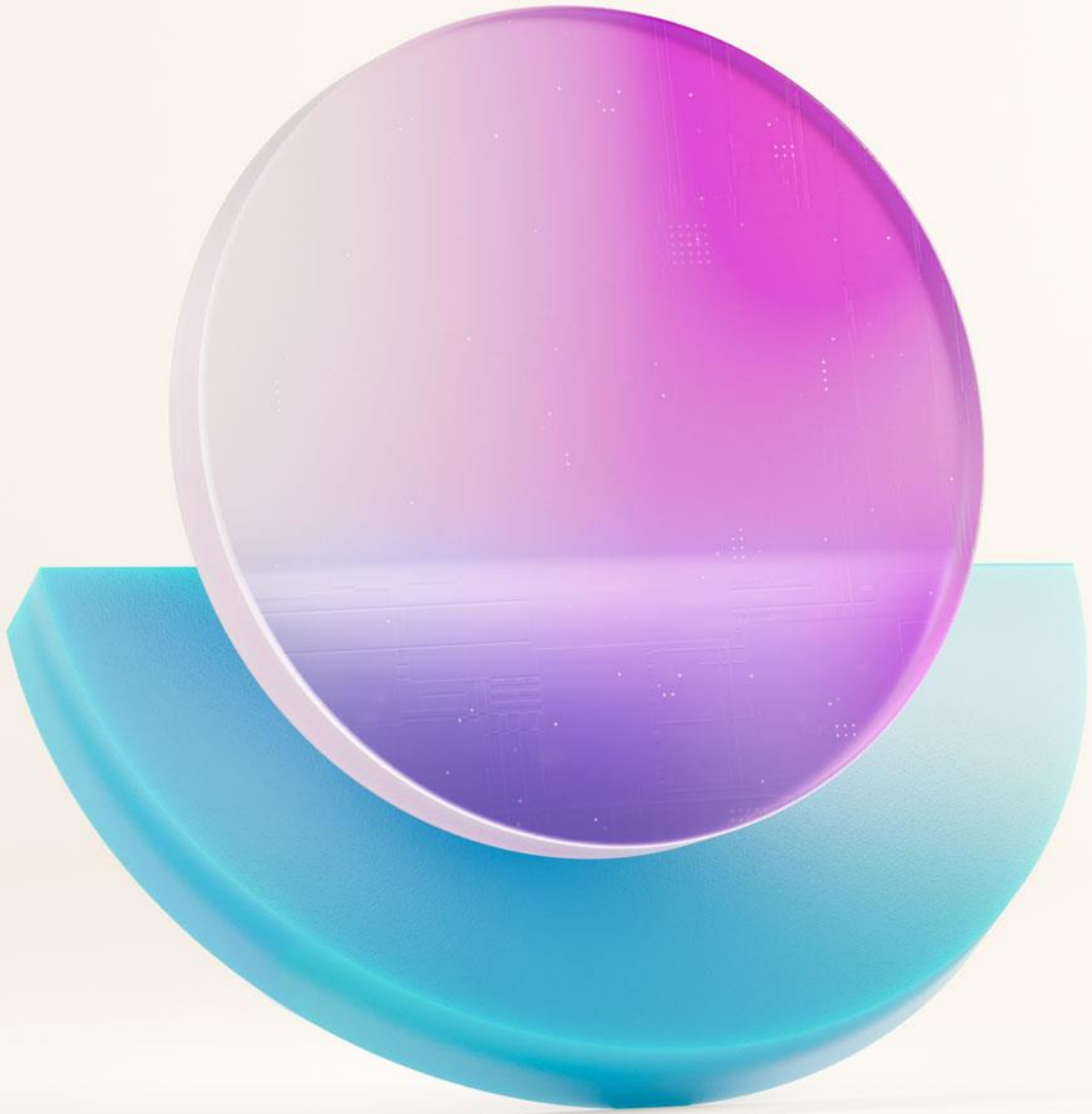


Column-pair suggestions

- The Merge dialog now intelligently detects matching columns from both the left and right tables, displayed in a convenient lightbulb icon in the top right corner.
  - Precise or approximate column title matches trigger these helpful suggestions.

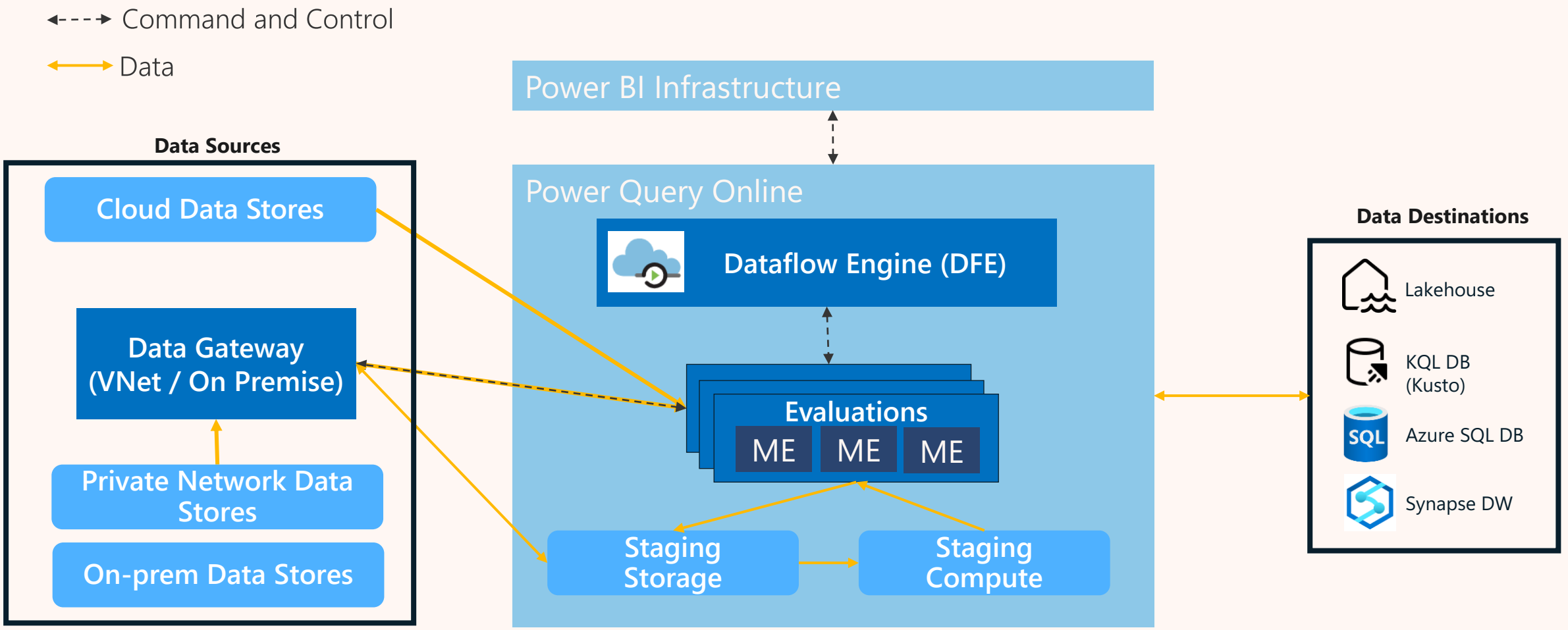


Microsoft Fabric
COMMUNITY CONFERENCE

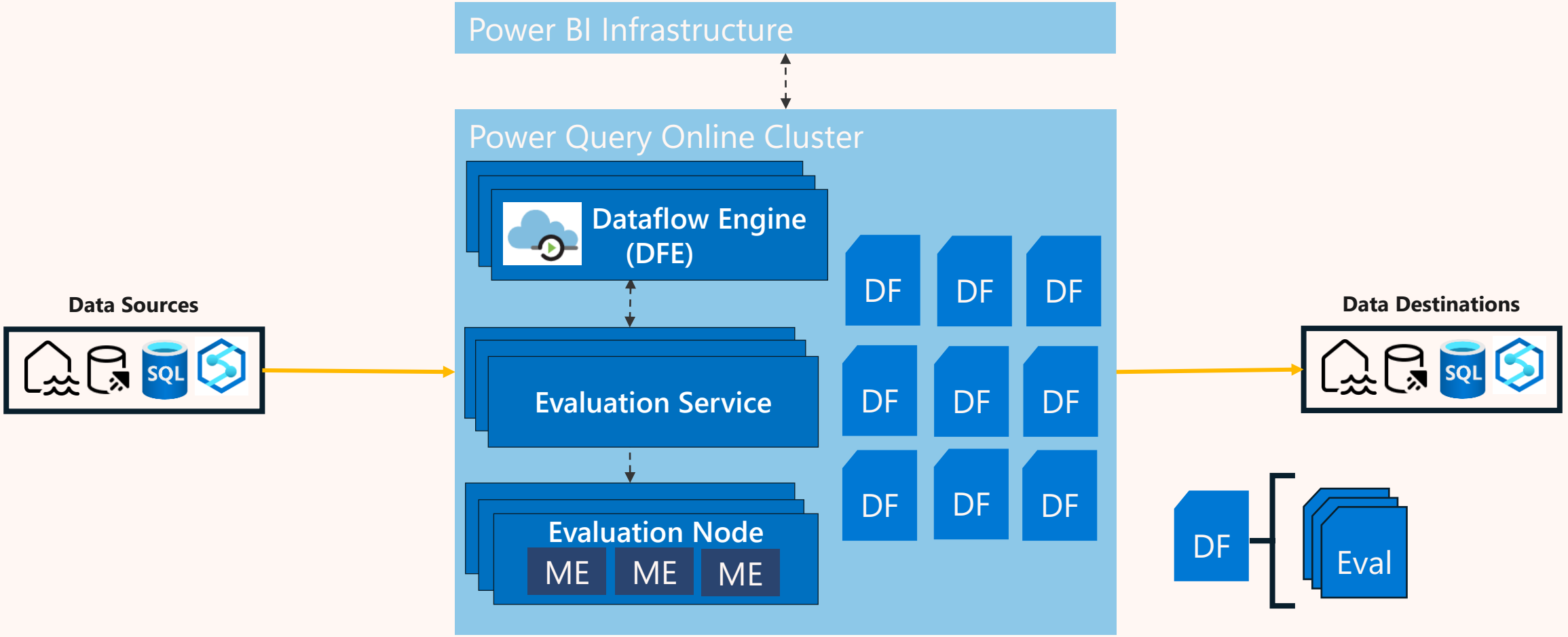


Dataflows In-Depth

Fabric Dataflows (aka Gen2)



Fabric Dataflows Scale Out

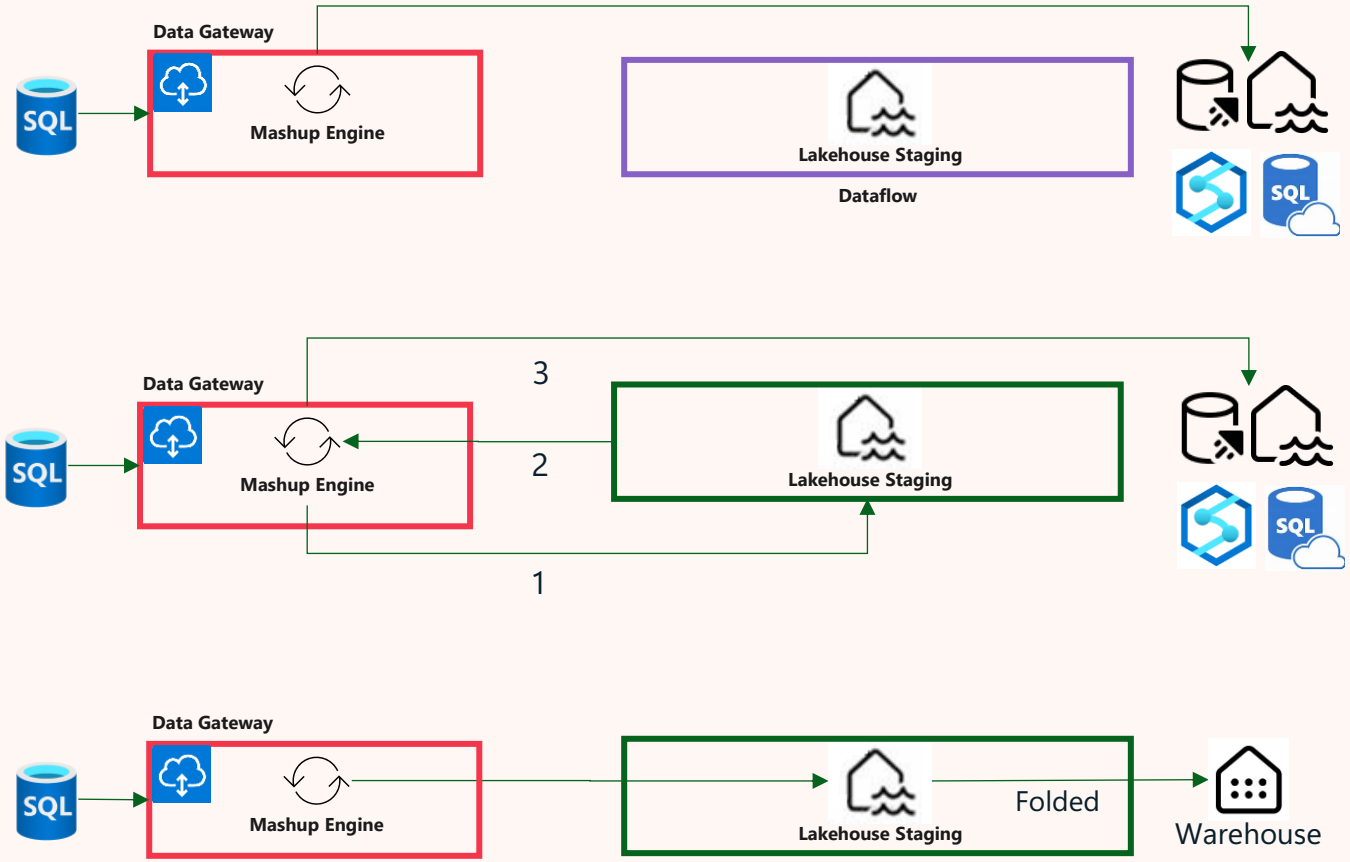


Data Gateway (VNET / On Premise)

- Capabilities
  - Enables access to data sources without exposing them to the internet
  - Data is processed locally

- When to use it
  - Data source is inside your private network
  - Move compute closer to source
  - Control where evaluations happen

- When to skip it
  - Entities that stage data and write to destination (excluding WH)
    - Split Ingest and Destination into separate dataflows (consider where to Transform)
    - Don’t stage



Dataflows Performance Principles

Four Principles

- Delegate to the most capable resource
- Do the expensive work first
- Divide and conquer
- Do as little work as possible

Dataflows Performance Capabilities

Principle	Capability	High-Level Approach
Delegate	Query Folding	Send as much of the query as possible to the underlying data source
Expensive work first	Staging	For large volumes of data reused by multiple query, first land the shared data in staging
Divide & conquer	Fast Copy	Leverage the Copy Activity for parallelized, high-scale data movement prior to transformation (ELT)
	Partitioning	Divide expensive queries into sub-queries that can be executed in parallel
Be ~~lazy~~ efficient	Incremental Refresh	During refreshes, process only the data that has changed

Query Folding – What it is

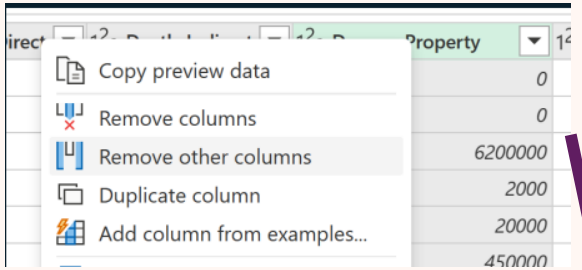
- Terms that are synonymous to Query Folding
  - Query Delegation
  - Query Push Down
  - Remote/Distributed Query Evaluation

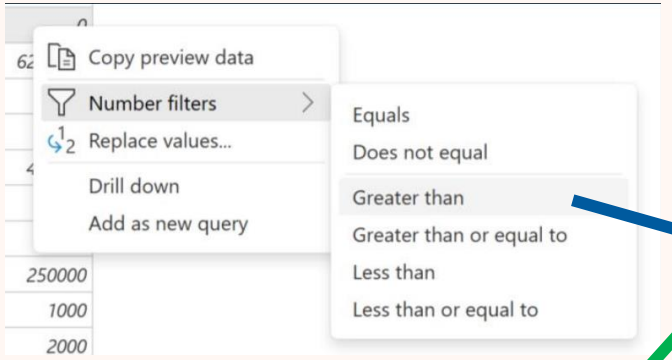
- Wherever possible, the script in the Power Query Editor (“M”) is translated to a “native query”

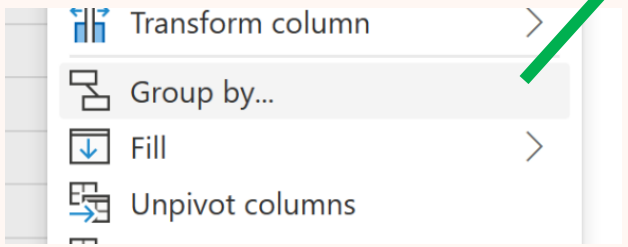
- The native query is then executed by the underlying data source.

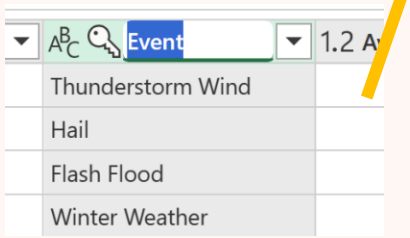
https://learn.microsoft.com/en-us/power-bi/guidance/power-query-folding

Query Folding - Example










```
let
  Source = Sql.Databases("server"),
  Navigation1 = Source{[Name = "Samples"]}[Data],
  Navigation2 = Navigation1{[Name = "StormEvents"]}[Data],
  SelectColumns = Table.SelectColumns(Navigation2, {"State", "EventType", "DamageProperty"}),
  Filter = Table.SelectRows(SelectColumns, each [DamageProperty] > 0),
  Group = Table.Group(Filter,{"State","EventType"},{{"AvgDmg",each List.Average([DamageProperty])}}),
  Rename = Table.RenameColumns(Group, {{"EventType", "Event"}})
in
  Rename
```

Query Folding – SQL native queries

The Dataflow runtime translates the M query to the data source's native query language, pushing down as much work as possible to the backend. In this example, all transformations steps can be folded to SQL.

```
let
  Source = Sql.Databases("server"),
  Navigation1 = Source{[Name = "Samples"]}[Data],
  Navigation2 = Navigation1{[Name = "StormEvents"]}[Data],
  SelectColumns = Table.SelectColumns(Navigation2, {"State", "EventType", "DamageProperty"}),
  Filter = Table.SelectRows(SelectColumns, each [DamageProperty] > 0),
  Group = Table.Group(Filter,{"State","EventType"},{{"AvgDmg",each List.Average([DamageProperty])}}),
  Rename = Table.RenameColumns(Group, {{"EventType", "Event"}})
in
  Rename
```

```
SELECT      [State], [EventType] as "Event", AVG([DamageProperty]) as AvgDmg
FROM        [StormEvents]
WHERE       [DamageProperty] > 0
GROUP BY    [State], [Event]
```

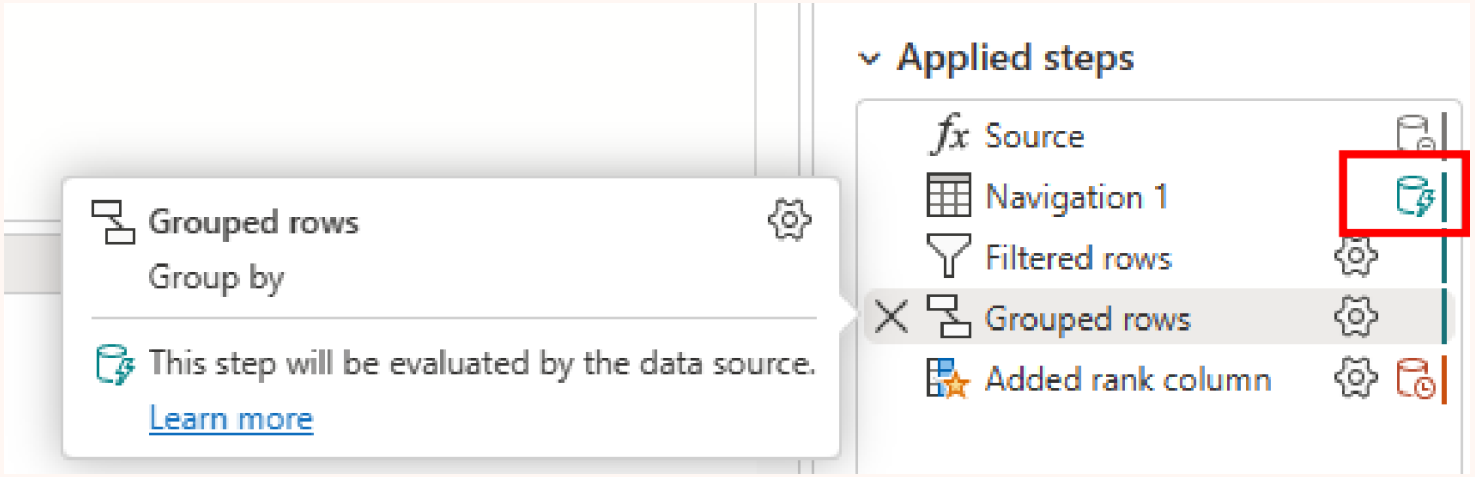
Query Folding – Indicators

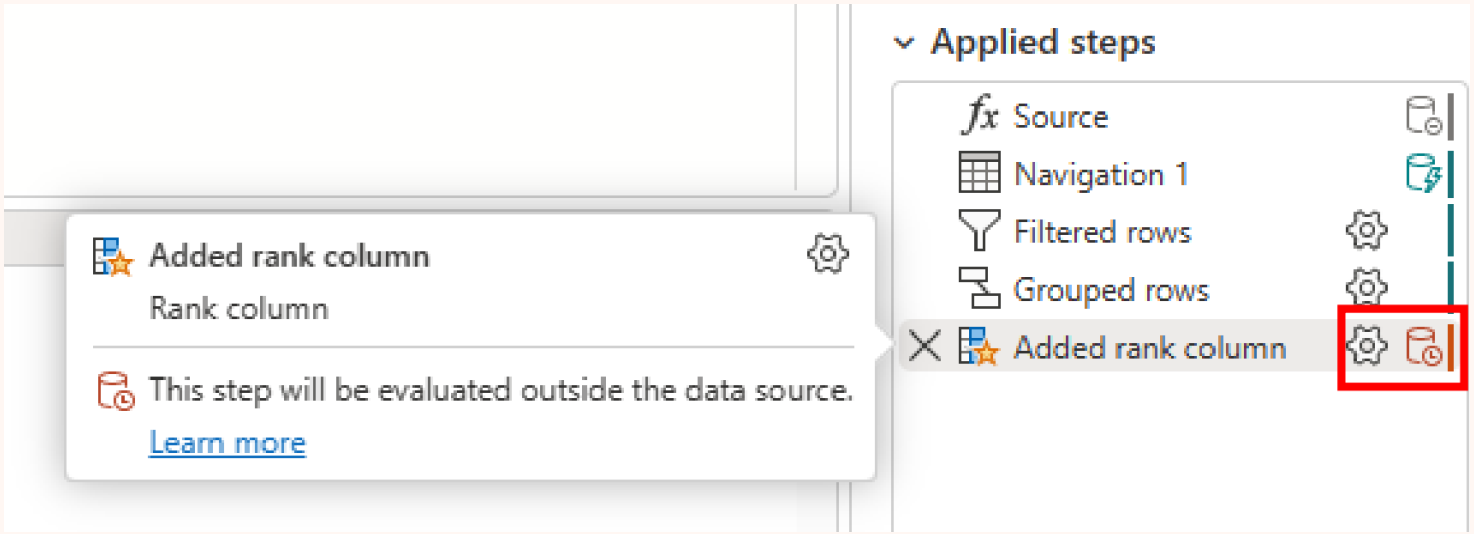
Green “lightning bolt” indicator for folded steps

(typically, faster)

Red “clock” indicator for inmemory steps evaluated outside the data source

(typically, slower)





Query Folding – Why it's Faster

- Reduced data movement
  - In-memory evaluation engine would have to first transfer the data from the source
  - A filter that selects 10 rows from a 10M row table, can avoid transferring 10M rows

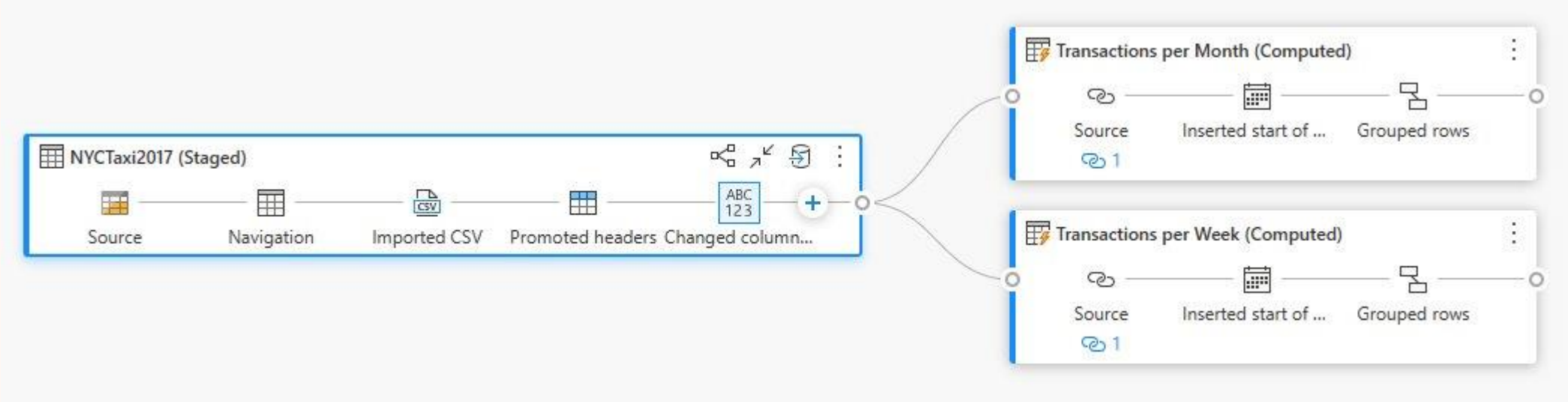
- Many data sources have highly efficient "query processors"
  - Optimized query plans
  - Ability to leverage indices, keys, etc.

Tips

1. Perform foldable operations early on
2. Do data type conversions last as they frequently break folding
3. Use the Table.StopFolding function to force local evaluation of subsequent steps

Staging – What it is

- Loading data into Fabric storage (Staging Lakehouse) as a first step
- The staged data can then be referenced by downstream queries that benefit from SQL Compute over the staged data



Staging - Tips

- **Data sources without rich query folding (e.g. files) are good candidates for staging**, especially when the files are large
- Data sources with rich query folding (e.g. databases) often do not require staging
- For a faster authoring experience create **separate dataflows for staging and transformation**
  - The transformation dataflow references the staging dataflow's data via the Dataflows connector

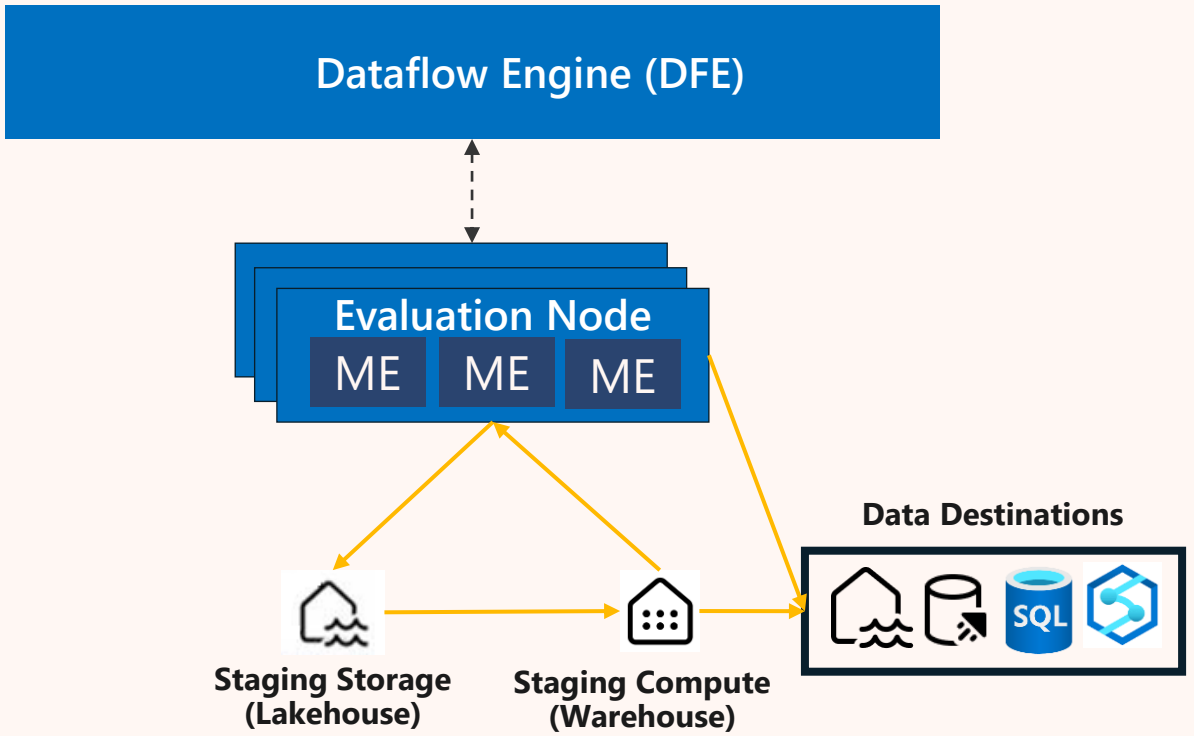
https://blog.crossjoin.co.uk/2023/09/03/fabric-dataflows-gen2-to-stage-or-not-to-stage/

Staging – Why it's Faster

- In Fabric (Gen 2) Dataflows, staged data is stored in a Staging Lakehouse
- Queries against the Staging Lakehouse benefit from a SQL analytics endpoint
- SQL analytics endpoint queries can be orders of magnitude faster than directly querying "slow sources" like files (or SharePoint..)

File	Size	Direct Grouping Query	Staged Grouping Query
2020 Yellow Taxi Trip Data (CSV)	2.2 GB	4 mins 12 secs	2.8 secs
2017 Yellow Taxi Trip Data (CSV)	9.8 GB	Timeout (> 10 mins)	4.6 secs

Staging - Architecture

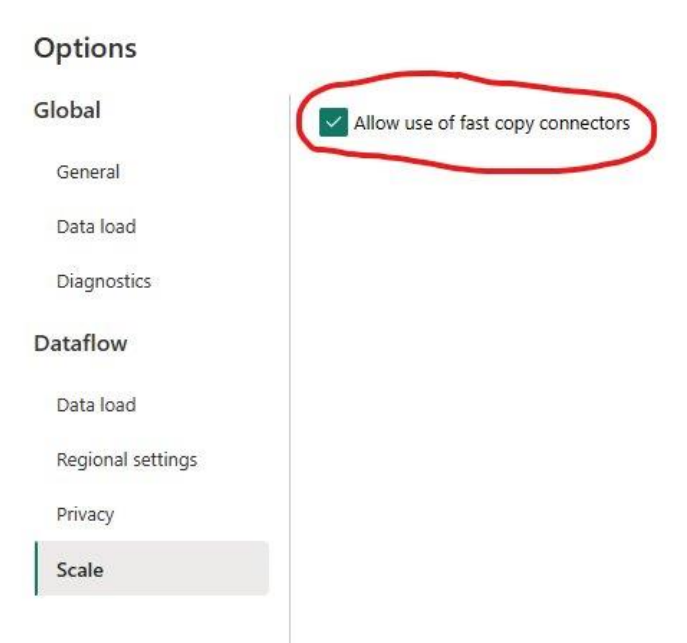


Fast Copy – What it is

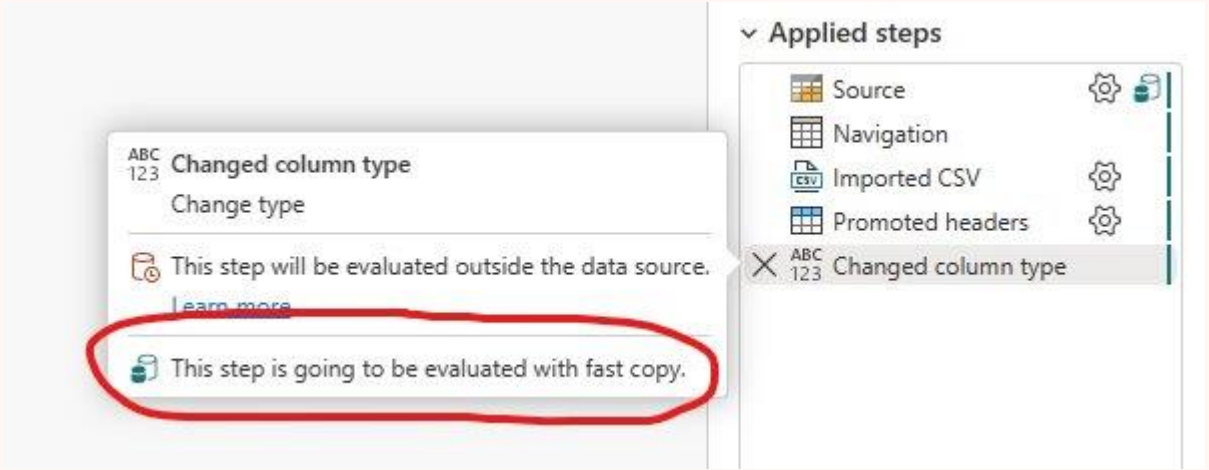
Highly parallelized data movement

- Powered by Azure Data Factory Copy Tasks
- Standard Power Query experience

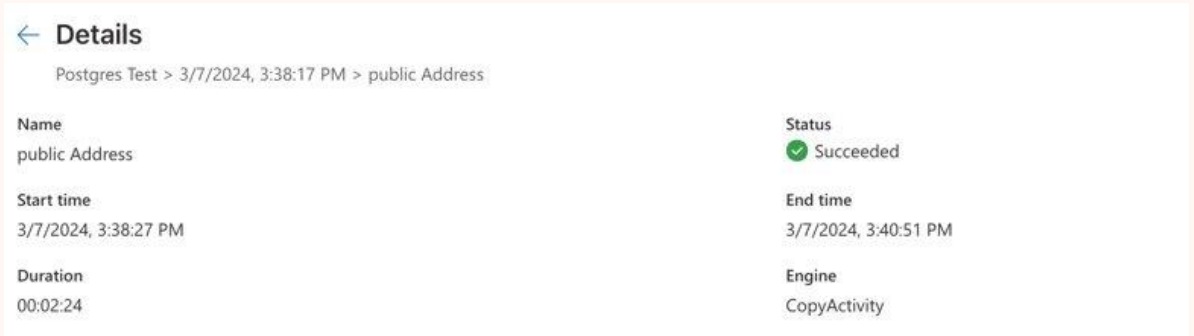
Step 1: Enable Fast Copy



Step 2: Use Indicators in Steps pane to verify use of Fast Copy



Step 3: Win



Fast Copy – Keep this in mind

Only for limited number of data sources (today)

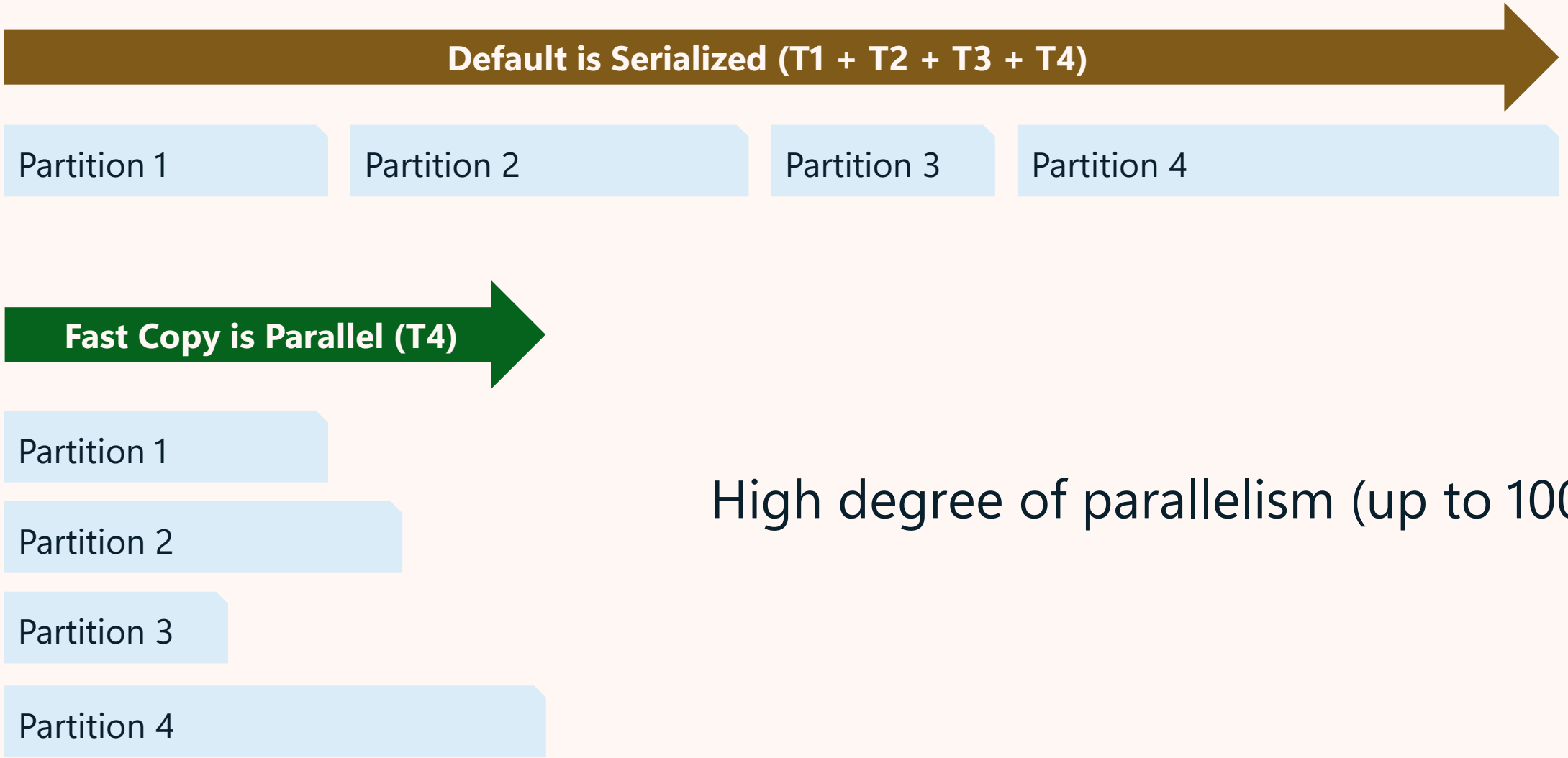
- Azure Data Lake Storage Gen 2
- Azure Blob Storage
- Azure SQL DB
- Fabric Lakehouse
- PostgreSQL

Only for limited number of transformations (today)

- Combine files
- Select or remove columns
- Change data types
- Rename a column

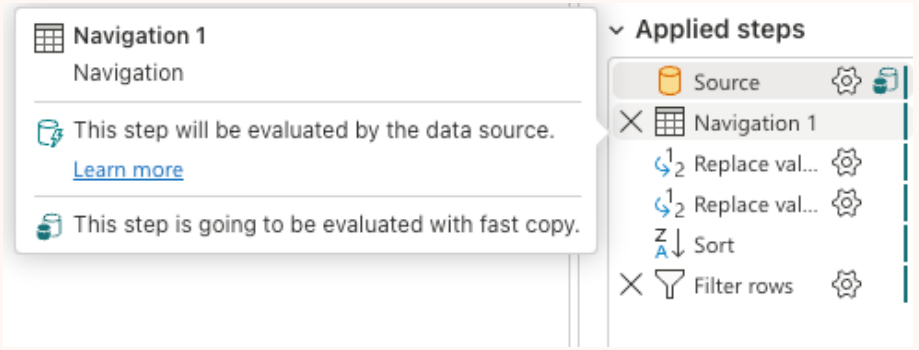
https://blog.fabric.microsoft.com/en-us/blog/fast-copy-in-dataflows-gen-2

Fast Copy – Why it’s Faster

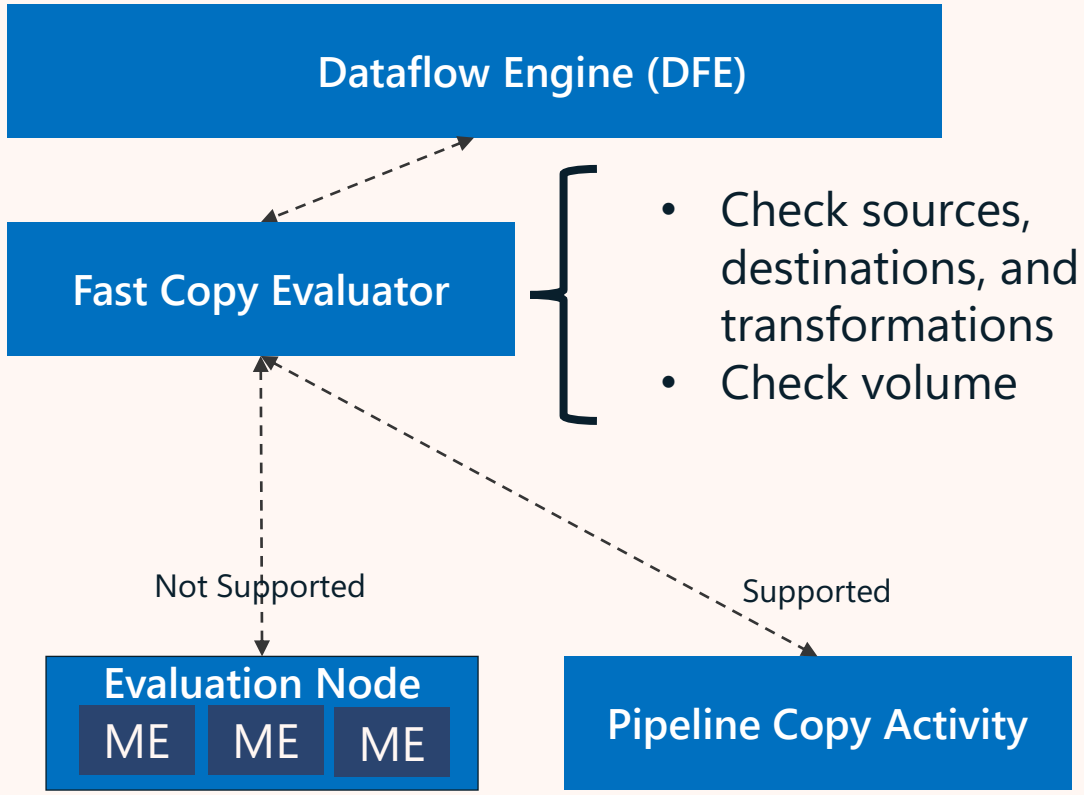


High degree of parallelism (up to 100)

Fast Copy - Architecture



- Capabilities
  - Leverages Pipeline Copy Activity for large performance boost in ingest
  - Automatically used based on pattern matching and volume
  - Transparent (no pipeline to manage)
- When to use it
  - Whenever possible
    - Defer transformations to post ingest if they affect Fast Copy use
  - Enable in Options..Scale..Allow use of fast copy connectors
- When to skip it
  - Don't – if it's an option, use it
  - Mark as "Require fast copy" to enforce



Partitioning – What it is

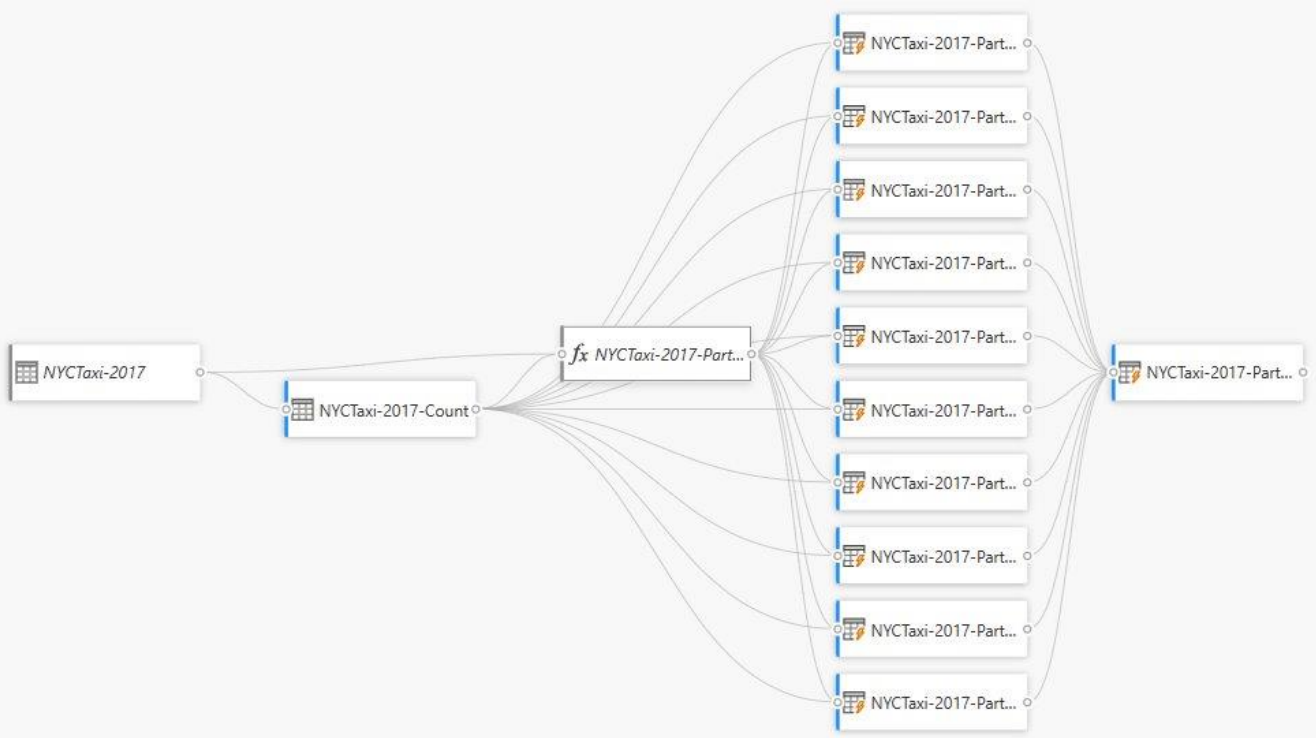
- Breaking long-running queries into smaller queries that can be run in parallel
- Explicit specification of what Fast Copy does implicitly
- Faster for the same reason as Fast Copy – parallelism

Partitioning – Sample

2017 Yellow Taxi Trip Data (9.8GB)

1. Calculate count (~113M rows)
2. Divide into 10 dynamic buckets using Table.Range (~11.3M rows per bucket)
3. Union the buckets

7x faster load time!



Leveraging the Architecture

- Carefully consider ETL vs ELT when laying out dataflows
- Know why you are using staging
  - Accelerates some operations, but adds no value to others
- Avoid “double-hops” with Data Gateways
- Take advantage of Fast Copy wherever possible
  - As patterns are added, dataflows can automatically benefit
  - Enable “use fast copy connectors”
  - Ensure that queries fully fold to maximize fast copy usage
- Build for parallel processing
  - Evaluations, dataflows
  - Incremental refresh when available

Slides

https://github.com/BenniDeJagere/Presentations/{Year}/{YYYYMMDD}_{Event}