

# Low friction data transformation and data movement using Fabric Dataflows

Benni De Jagere





# Benni De Jagere

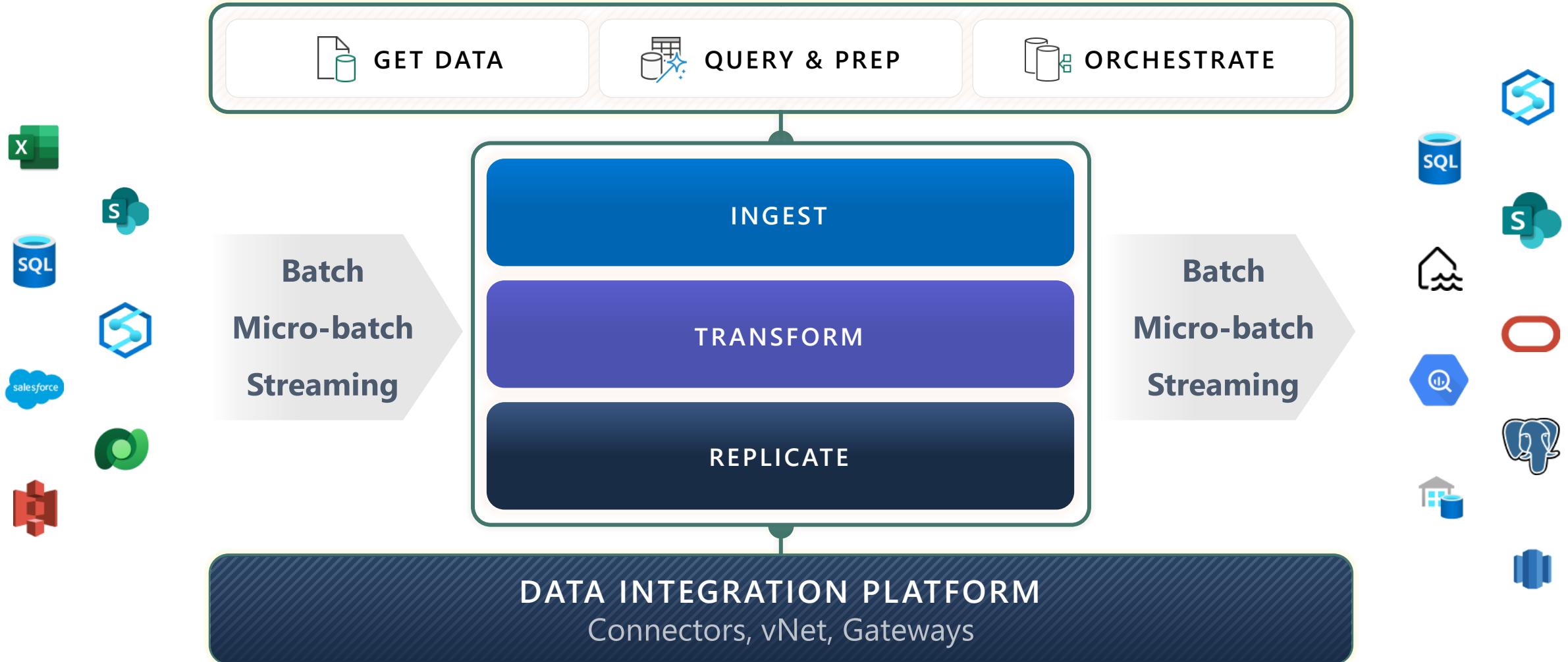
Senior Program Manager | Fabric Customer Advisory Team ( Fabric CAT )



Fabric CAT  
.be Member  
@BenniDeJagere  
/bennidejagere  
/bennidejagere  
/bennidejagere

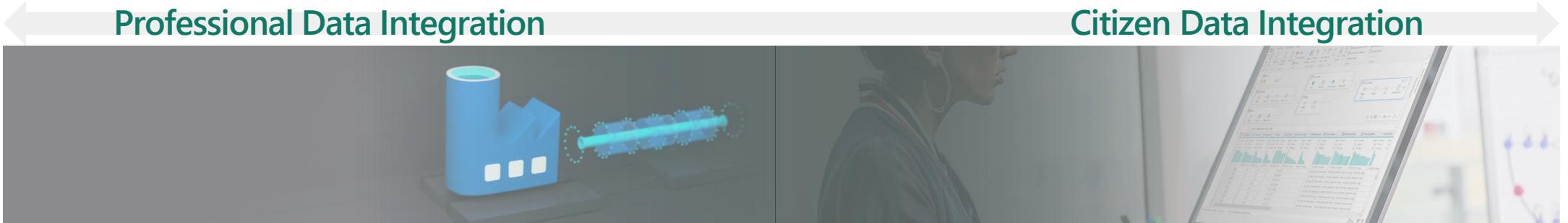


# Microsoft Data Integration



# Microsoft Data Integration

## Products



### 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



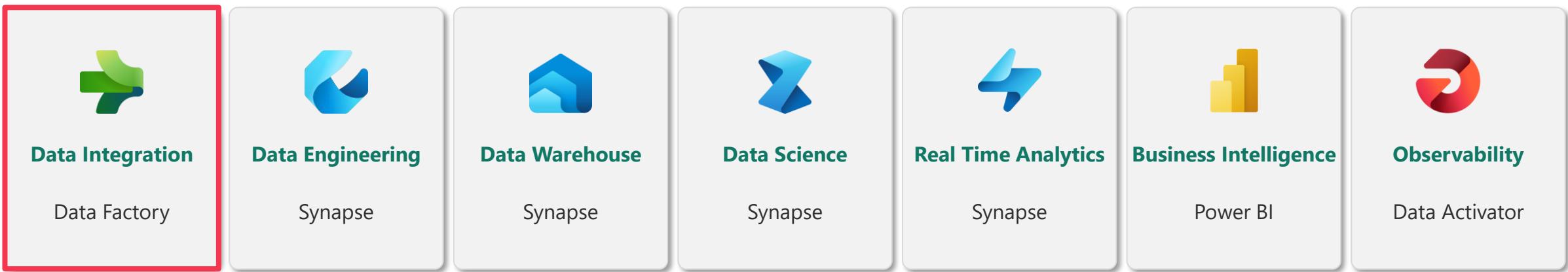
### Data Factory in Microsoft Fabric

- 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



**Unified data foundation**  
OneLake

**UNIFIED**

SaaS product experience

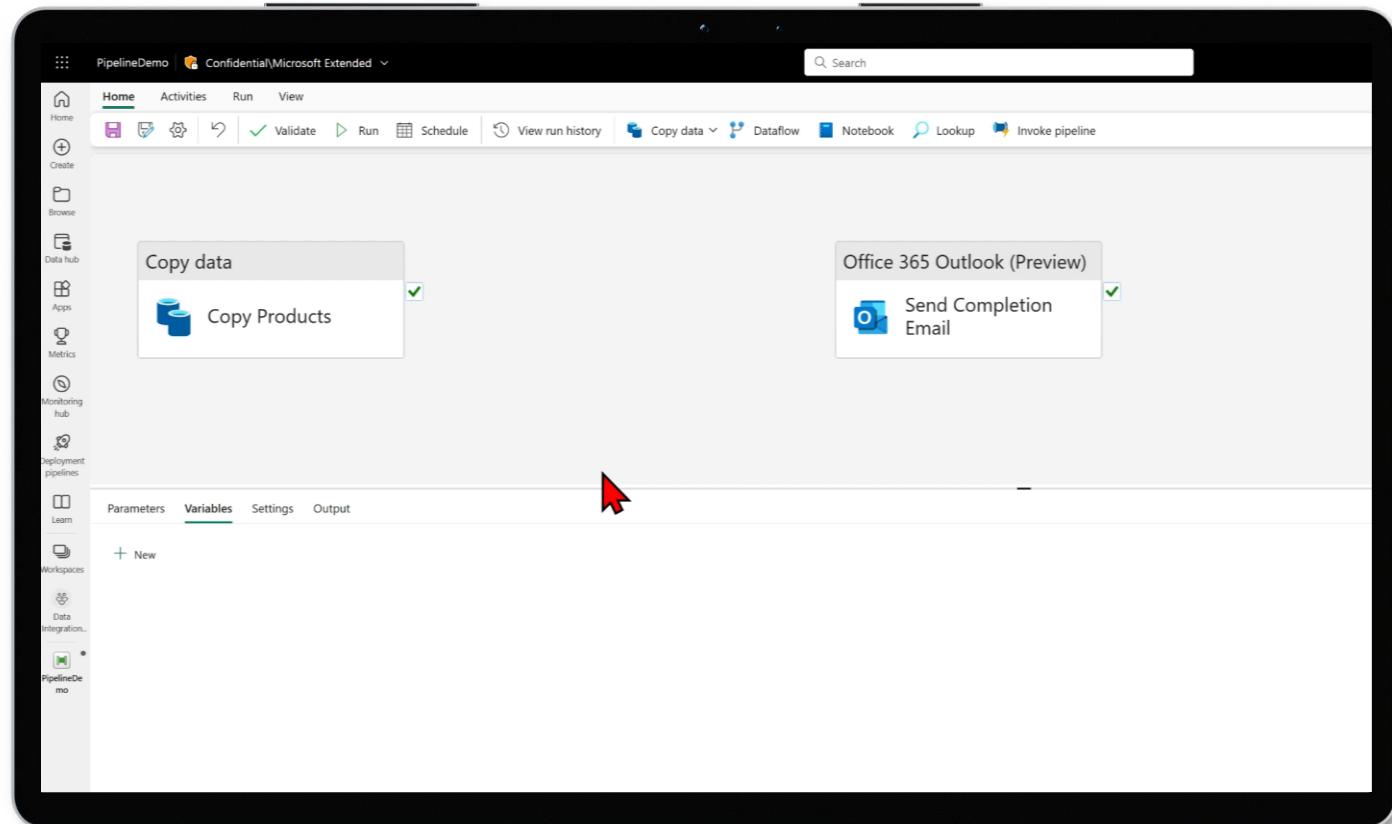
Security and governance

Compute and storage

Business model

# 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

The screenshot shows the Microsoft Fabric Data Pipeline interface. At the top, there's a navigation bar with 'myPipeline' and 'Confidential\Microsoft Extended'. A search bar and a trial status ('Trial: 59 days left') are also at the top. Below the bar is a ribbon menu with 'Home', 'Activities' (which is selected), 'Run', and 'View'. The main area is titled 'Start building your data pipeline' and features three buttons: 'Add pipeline activity' (with a gear icon), 'Copy data' (with a clipboard icon), and 'Choose a task to start' (with a document icon). On the left side, there's a sidebar with icons for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, and specific pipelines like 'myDataIntegratio...'. A large blue arrow points from this interface to the OneLake section.



OneLake



# Data pipeline

## Ingest and orchestrate activities at scale

Empower every person to integrate data

The screenshot shows the Microsoft Fabric Data Pipeline interface. On the left, there's a sidebar with various navigation options like Home, Create, Browse, Data hub, Apps, Metrics, Monitoring hub, Deployment pipelines, Learn, Workspaces, Data Integration..., and PipelineDe... (partially visible). The main area displays two selected activities: 'Copy data' (with 'Copy Products' listed) and 'Office 365 Outlook (Preview)' (with 'Send Completion Email' listed). Below these, tabs for Parameters, Variables, Settings, and Output are visible, along with a '+ New' button. A red arrow points from the bottom right of the pipeline interface towards a white circle containing a black right-pointing arrow. This arrow points to a separate box on the right containing the OneLake logo and text.

OneLake

The OneLake logo consists of three overlapping blue and teal circular shapes forming a stylized 'L' or wave-like pattern. To its right, the word "OneLake" is written in a bold, sans-serif font.



Public Preview

# Copy job

## Easily ingest and move data at scale

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

The screenshot shows the Microsoft Fabric Data factory interface. In the top navigation bar, there are links for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, and My workspace. The main area has a "New" section with icons for Dataflow Gen2 (Preview), Data pipeline, Copy job, Apache Airflow project, and Data Factory mount. Below this is a "Recommended" section with cards for "Not sure where to start? Launch Dataflow Copilot", "Not sure where to start? Launch Pipelines Copilot", "Introduction to data integration Watch video", "Getting started with dataflow Watch the tutorial video", and "Getting started with data pipelines Watch the tutorial video". At the bottom, there is a "Quick access" table with columns for Name, Type, Opened, Owner, Endorsement, Sensitivity, and Workspace. The table lists four items: Cool data mart (Datamart, 7m ago, Tim Deboar, —, General, Contoso workspace), Data flow for triggers (Data flow, 7m ago, Tim Deboar, —, —, Contoso workspace), User data (Datamart, 7m ago, Tim Deboar, Certified, —, Contoso workspace), and Copy data pipeline (Pipeline, 7m ago, Tim Deboar, —, —, Contoso workspace).

Name	Type	Opened	Owner	Endorsement	Sensitivity	Workspace
Cool data mart	Datamart	7m ago	Tim Deboar	—	General	Contoso workspace
Data flow for triggers	Data flow	7m ago	Tim Deboar	—	—	Contoso workspace
User data	Datamart	7m ago	Tim Deboar	Certified	—	Contoso workspace
Copy data pipeline	Pipeline	7m ago	Tim Deboar	—	—	Contoso workspace





# Dataflow Gen 2

## Ingest and transform data at scale

Enterprise-scale data ingestion and transformation

The screenshot shows the Microsoft Power Query Editor interface. On the left, the navigation pane includes sections for Home, Queries (13), Data staging, Data load, and Data transformation. A cursor points to the 'DimCustomer' step under 'Data transformation'. The main area displays a table with columns such as CustomerKey, GeographyKey, FirstName, MiddleName, LastName, BirthDate, MaritalStatus, Suffix, Title, EmailAddress, and Education. Each column has a corresponding histogram showing data distribution. To the right, the 'Query settings' pane is open, showing properties like Name (DimCustomer) and Entity type (Custom). Under 'Applied steps', there is a list of steps including 'Source' (tyrone15@adventure-works.com) and 'Merged queries' (Expanded DimGeography.raw). At the bottom, the 'Data destination' section is set to 'No data destination'.

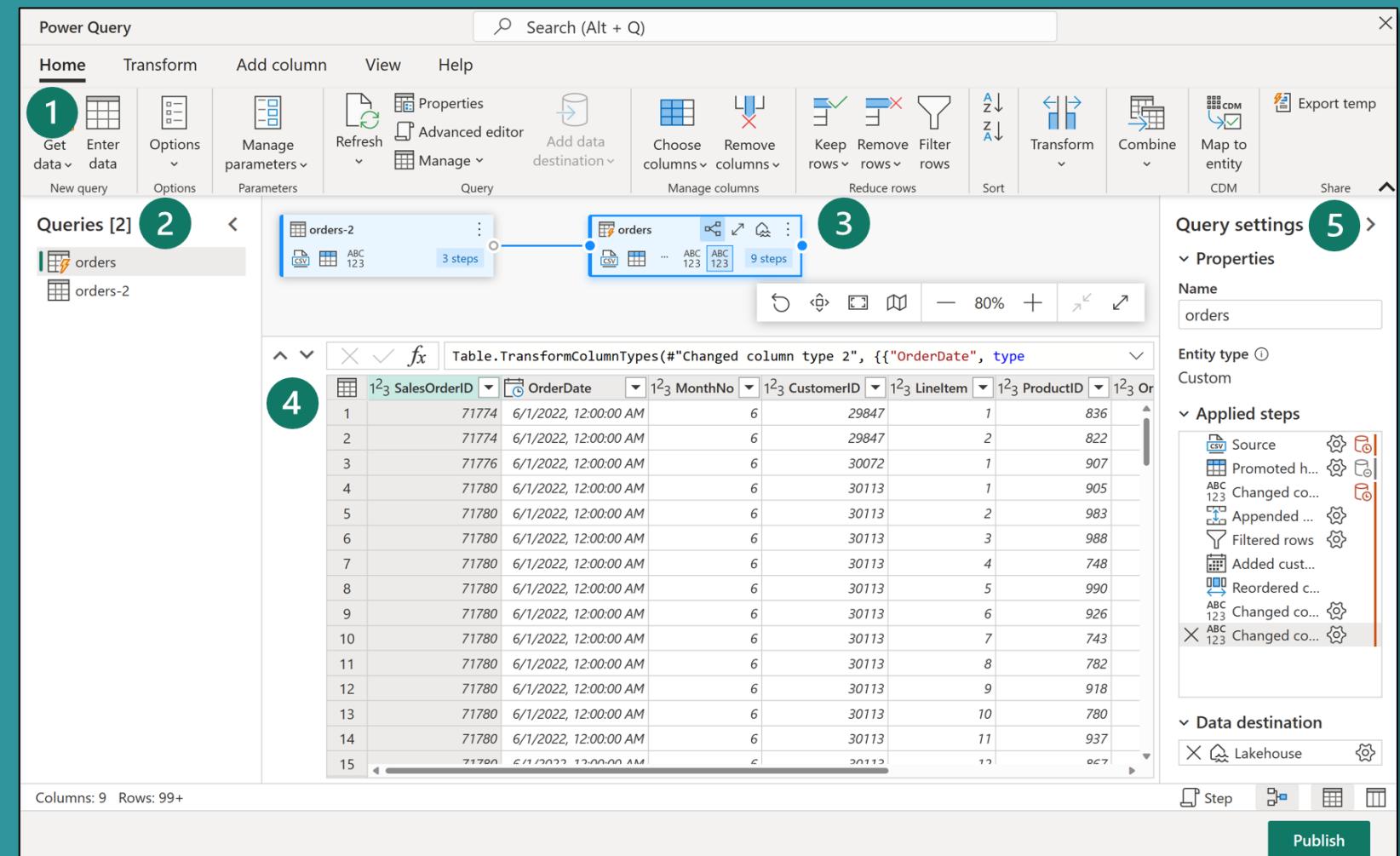


OneLake

# 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

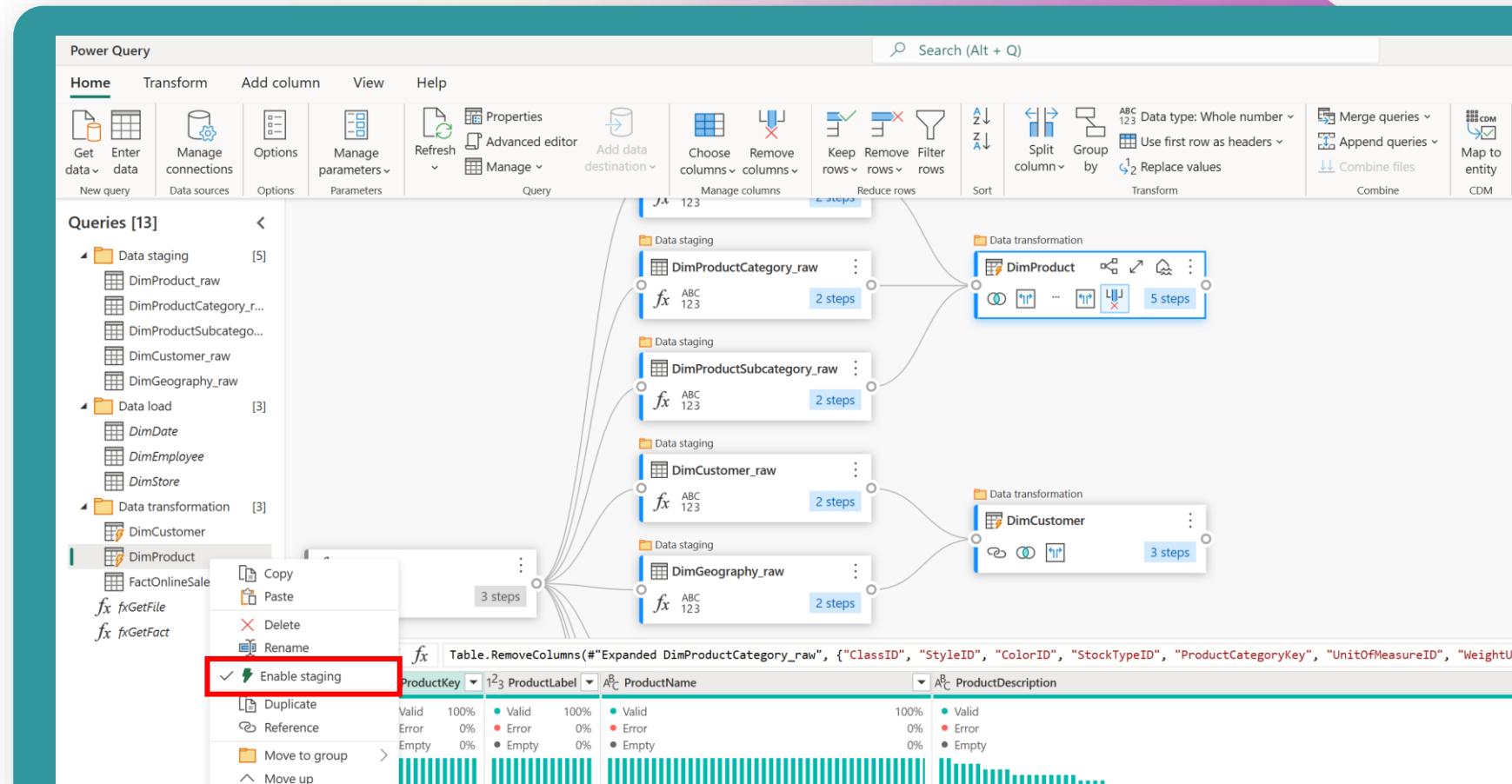
The screenshot shows the Microsoft Power Query Editor interface. On the left, the 'Queries [13]' pane is visible, with a mouse cursor hovering over the 'DimCustomer' query under the 'Data transformation' folder. The main area displays a table with 99+ rows of data from the 'DimCustomer' table. The columns include CustomerKey, GeographyKey, FirstName, MiddleName, LastName, BirthDate, MaritalStatus, Suffix, Title, EmailAddress, YearlyIncome, and Education. Each column has a corresponding histogram showing the distribution of values. The top right corner of the interface shows 'Query settings' and 'Applied steps' sections.

# 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

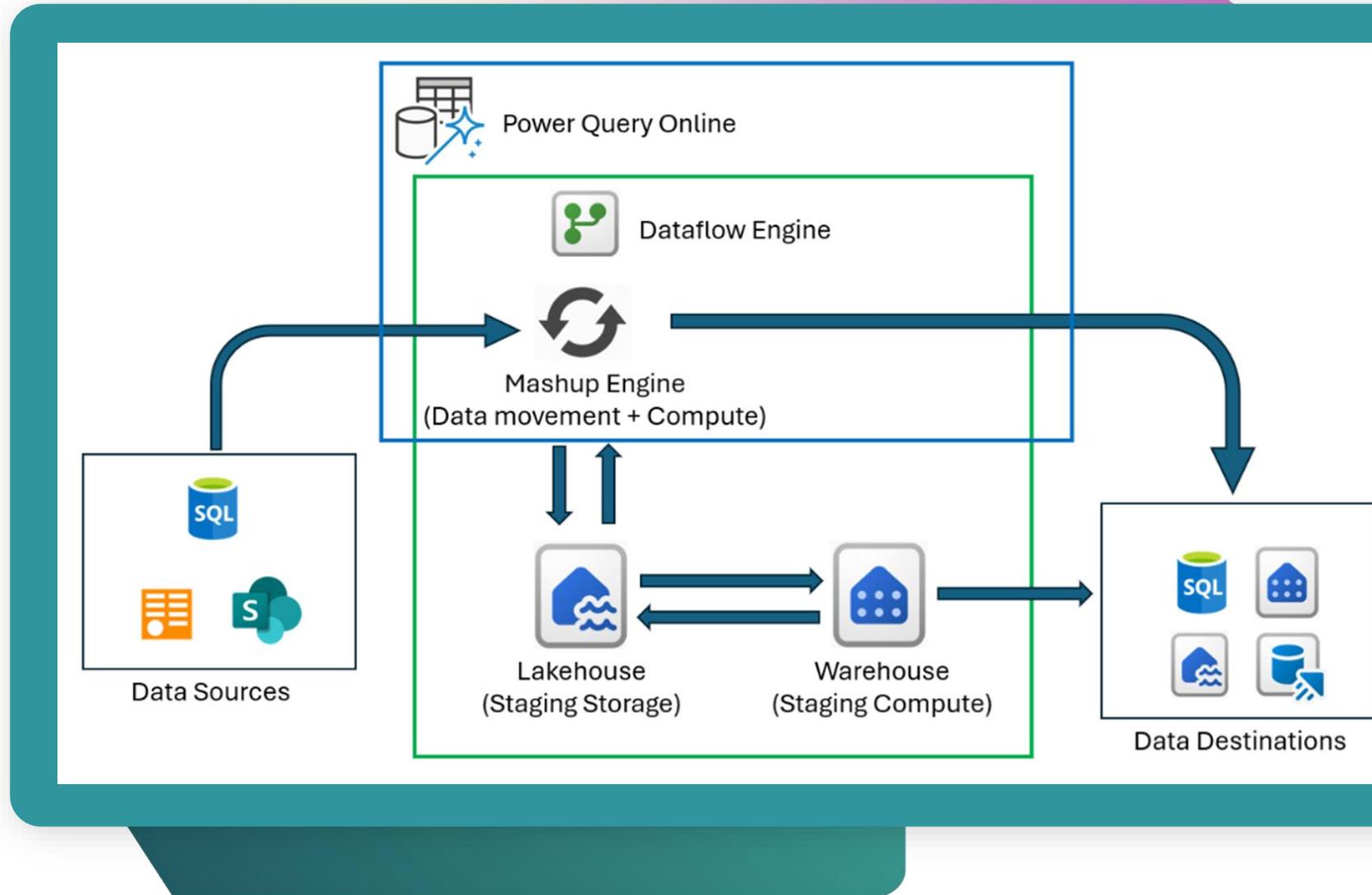


\*Previously titled "Enable load"

# 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.



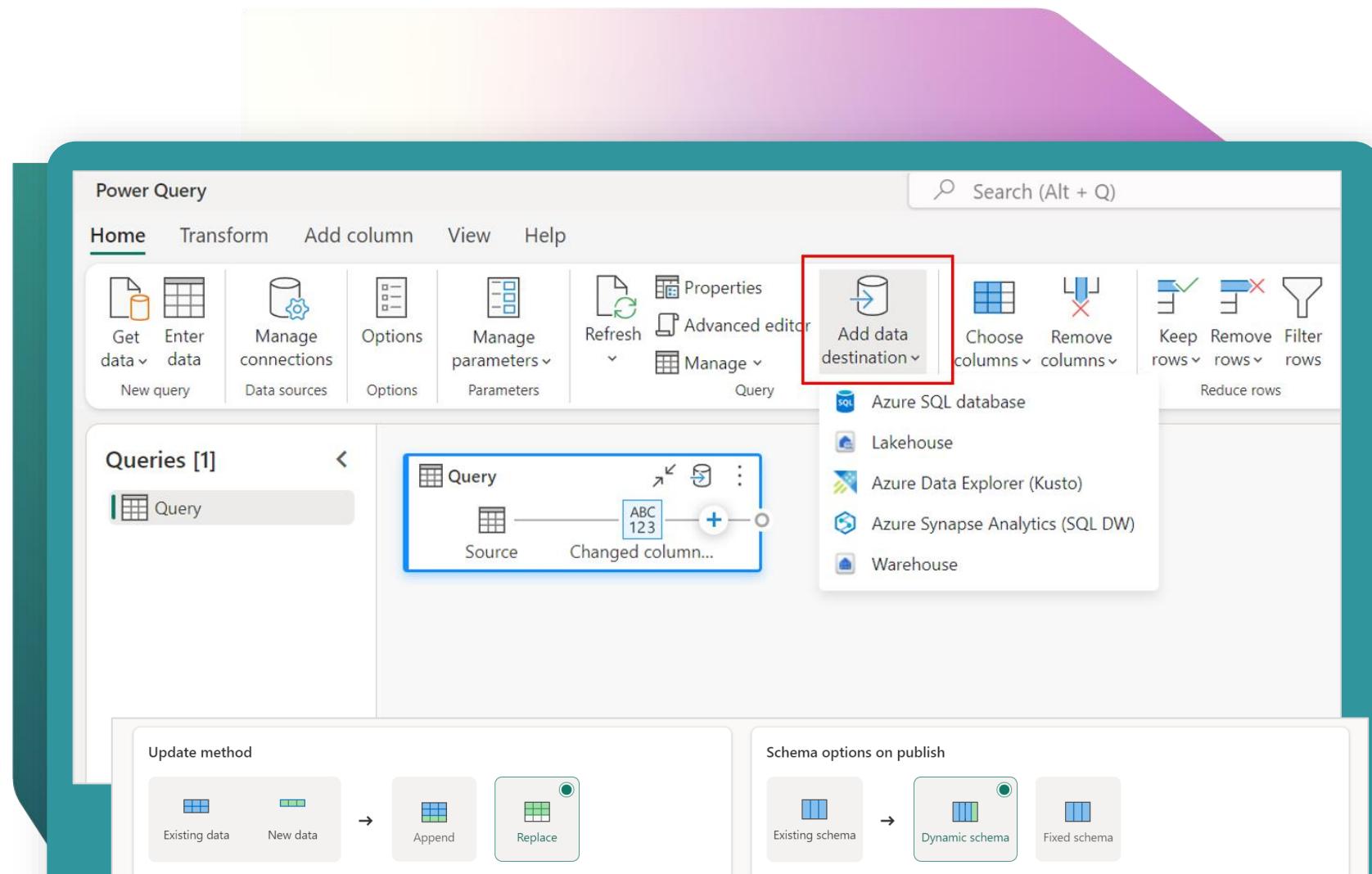
# Data destinations

Supported destinations:

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

Update methods:

- Replace
- Append



# Fast Copy

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

The screenshot shows the Microsoft Dataflow interface. On the left, there's a preview pane displaying a table with five columns and six rows of data. The first column is labeled 'r\_1' and contains values like 27450000, 1080000, etc. The second column is labeled 'r\_2', third 'r\_3', fourth 'r\_4', and fifth 'r\_5'. A tooltip message, 'This step is going to be evaluated with fast copy.', is overlaid on the preview area, highlighted with a red box. To the right of the preview is the 'Applied steps' panel, which lists various transformation steps: 'Filtered hid...', 'Invoke cust...', 'Renamed c...', 'Removed o...', 'Expanded t...', and 'Changed c...'. The 'Changed c...' step is currently selected.

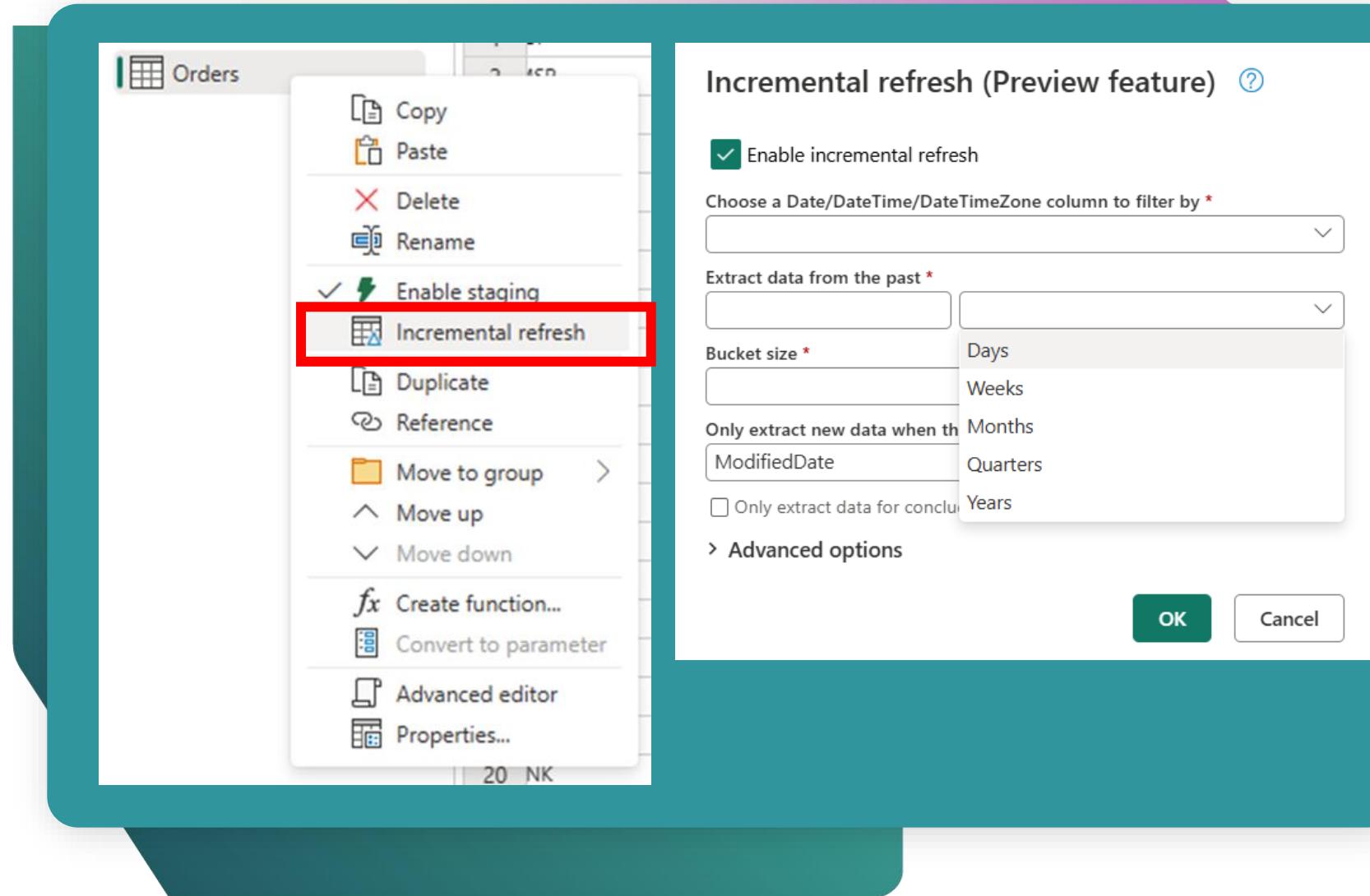
r_1	r_2	r_3	r_4	r_5
27450000	3240000	26640000	22770000	2016
1080000	28800000	29070000	15210000	2547
16020000	24210000	26460000	15750000	846
2070000	24840000	12150000	3150000	1926
30870000	9720000	18270000	8010000	594
4320000	17640000	21420000	18990000	1458

# Incremental refresh

Public Preview

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

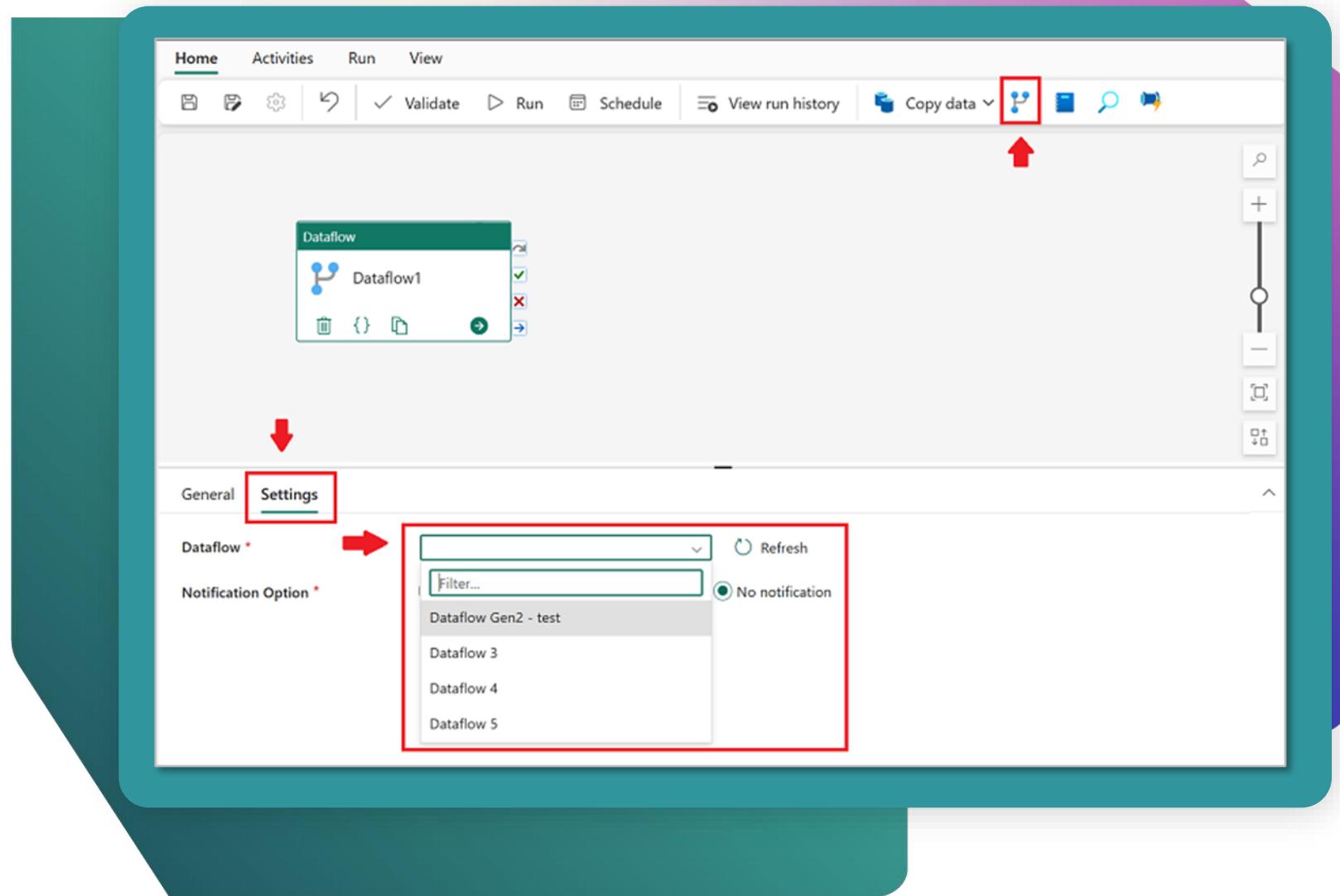
- Evaluate changes by comparing the maximum **DateTime** value with the previous refresh.
- Retrieve and load data for changed buckets in parallel, loaded to staging
- Replaces the destination data with the new data, affecting only updated buckets.



# 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



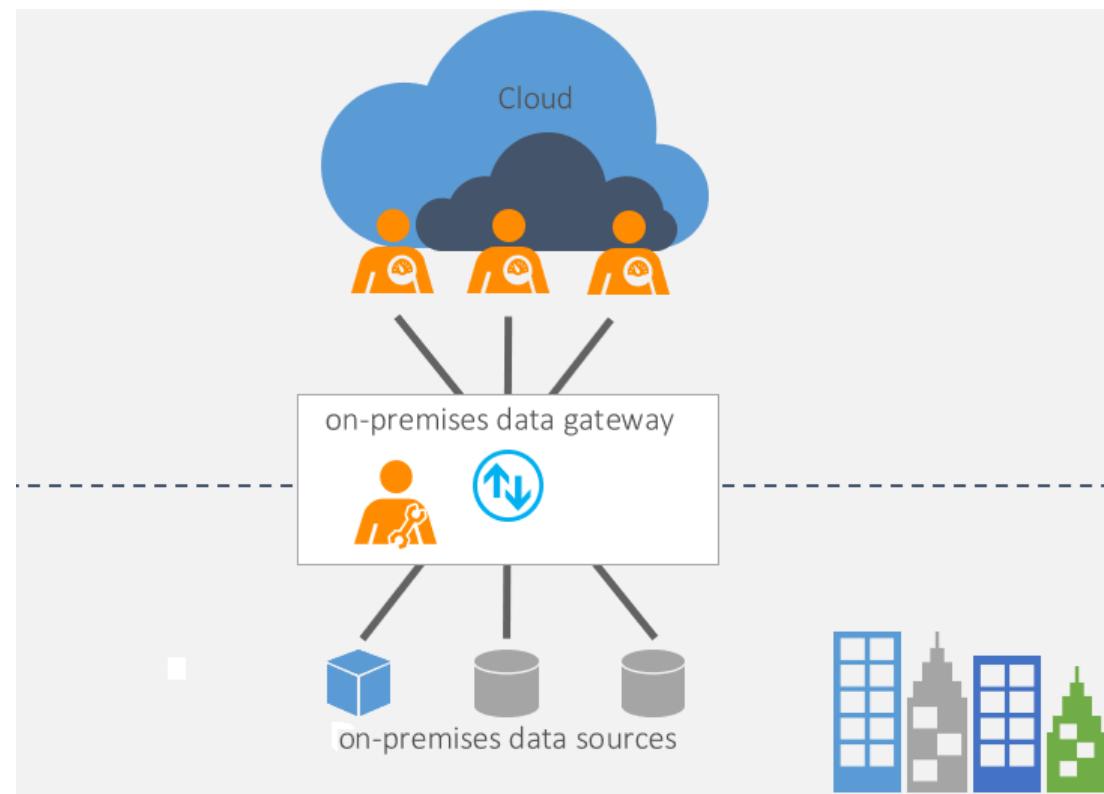
Azure Database for PostgreSQL	Azure Databricks Delta Lake	Amazon RDS for Oracle	Amazon RDS for SQL Server	Amazon Redshift	Phoenix	PostgreSQL	Presto	Magento (Preview)
Azure SQL Database	Azure SQL Database Managed Instance	Apache Impala	Azure SQL Database Managed Instance	DB2	SAP BW	SAP BW	SAP HANA	Oracle Eloqua (Preview)
Azure Table Storage	MongoDB Atlas	Drill	Google AdWords	Google BigQuery	SAP TABLE	SQL server	Spark	PayPal (Preview)
Azure Cosmos DB (MongoDB API)	Azure Cosmos DB (SQL API)	Greenplum	HBase	Hive	Amazon S3	Amazon S3 Compatible	FTP	SAP Cloud For Customer
Azure Data Lake Storage Gen1	Azure Data Lake Storage Gen1 for Cosmos Structured Stream	Informix	MariaDB	Microsoft Access	File system	Google Cloud Storage (S3 API)	HDFS	Salesforce Marketing Cloud
Azure Data Lake Storage Gen2 for Cosmos Structured Stream	Azure Database for MariaDB	MySQL	Netezza	Oracle	HTTP	Oracle Cloud Storage (S3 API)	SFTP	Shopify (Preview)
Teradata	Vertica	ODBC	OData	REST	Amazon Marketplace Web Service	Concur (Preview)	Dataverse (Common Data Service for Apps)	Web Table
Jira	Kusto	SharePoint Online List	Dynamics 365	Dynamics AX	Dynamics CRM	cassandra	Couchbase (Preview)	...



# On-premises data gateway

Secure connectivity to on-premises and private endpoints

Bridge between your secure data sources and the Microsoft cloud



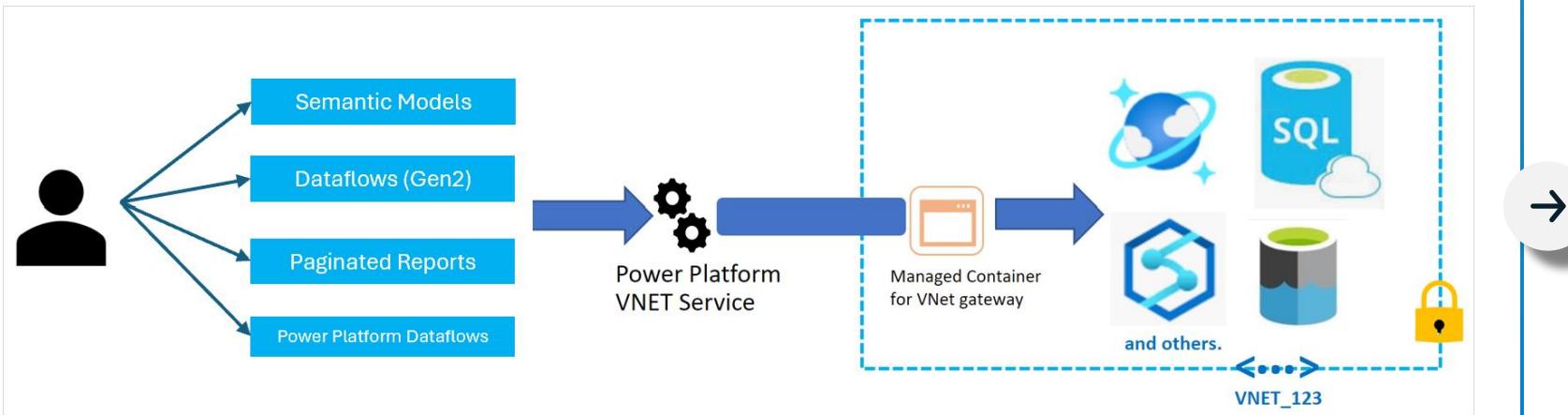
OneLake



# VNet data gateway

Secure connectivity to your Azure and private data services

Bridge between your secure data sources and the Microsoft cloud





# 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

The screenshot shows the Microsoft Data Factory interface. On the left, there's a sidebar with options like Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and My workspace. The main area shows a dataflow named 'Orders' with two stages: 'Orders - Staging' and 'Orders'. The 'Orders' stage has a 'Source' step and an 'Open AI' step. A 'Copilot' button is visible in the top right of the interface. To the right, there's a 'Copilot Preview' panel where users can describe data transformations using natural language. Below the dataflow, a preview table shows rows of data with new columns added by AI, such as 'ABC CustomerReview' and 'ABC DissatisfactionReason'. The bottom right corner has a 'Publish' button.

# 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	<b>YES!</b>
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





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.

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table with two columns: "ContactName" and "ContactTitle". The table contains 22 rows of data. The "ContactName" column lists names like Maria Anders, Ana Trujillo, Antonio Moreno, etc., and the "ContactTitle" column lists titles like Sales Representative, Owner, Marketing Manager, etc. The Power Query ribbon is visible at the top, and the "Applied steps" pane on the right shows the "Removed other columns" step.

ContactName	ContactTitle
Maria Anders	Sales Representative
Ana Trujillo	Owner
Antonio Moreno	Owner
Thomas Hardy	Sales Representative
Christina Berglund	Order Administrator
Hanna Moos	Sales Representative
Frédérique Citeaux	Marketing Manager
Martín Sommer	Owner
Laurence Lebihan	Owner
Elizabeth Lincoln	Accounting Manager
Victoria Ashworth	Sales Representative
Patricia Simpson	Sales Agent
Francisco Chang	Marketing Manager
Yang Wang	Owner
Pedro Afonso	Sales Associate
Elizabeth Brown	Sales Representative
Sven Ottlieb	Order Administrator
Janine Labrune	Owner
Ann Devon	Sales Agent
Roland Mendel	Sales Manager
Aria Cruz	Marketing Assistant
Diego Roel	Accounting Manager

# 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.

## Out-of-the-box connectivity to hundreds of sources

We're building the connectors data analysts want most into a lot of Microsoft products you use every day. And if you can't find what you're looking for, we've made it easy to build your own. Even [get them certified by Microsoft](#).

Choose a connector category:

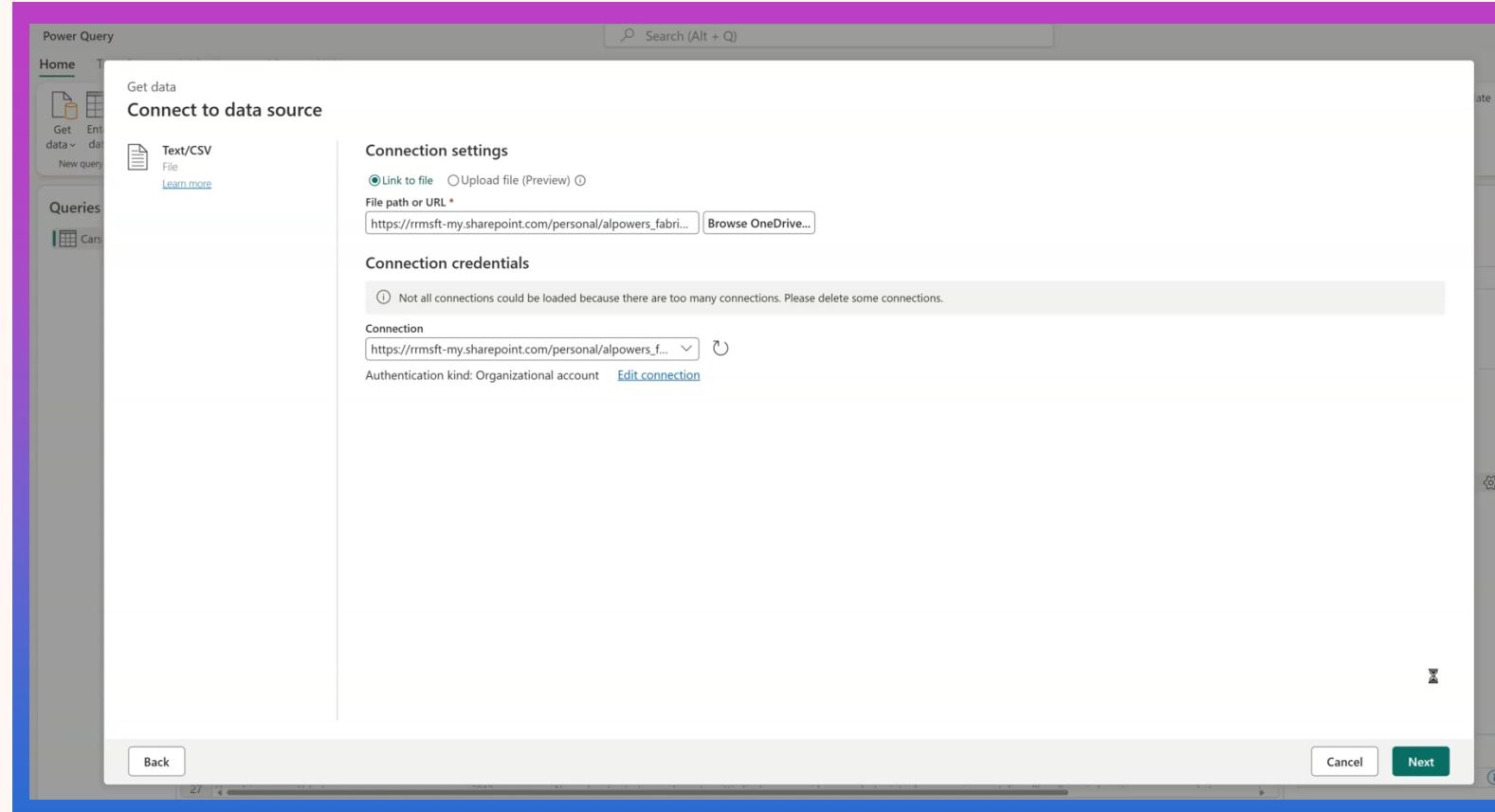
All connectors

Category	Connector	By	For
Database	Access database	By Microsoft	Power BI, Excel, Analysis Services
Other	Acterys (Beta)	By Acterys	Power BI
Database	Action (Beta)	By Action	Power BI
Other	Active Directory	By Microsoft	Power BI, Excel, Analysis Services
Online Services	Adobe Analytics	By Microsoft	Power BI
Database	Amazon Athena (Beta)	By Amazon	Power BI
Other	Amazon OpenSearch Service (Beta)	By Amazon	Power BI
Database	Amazon Redshift	By Microsoft	Power BI
Other	Anaplan	By Anaplan	Power BI
Online Services	appFigures (Beta)	By Microsoft	Power BI
Online Services	Asana (Beta)	By Asana	Power BI
Online Services	Assemble Views	By Autodesk	Power BI

# 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.

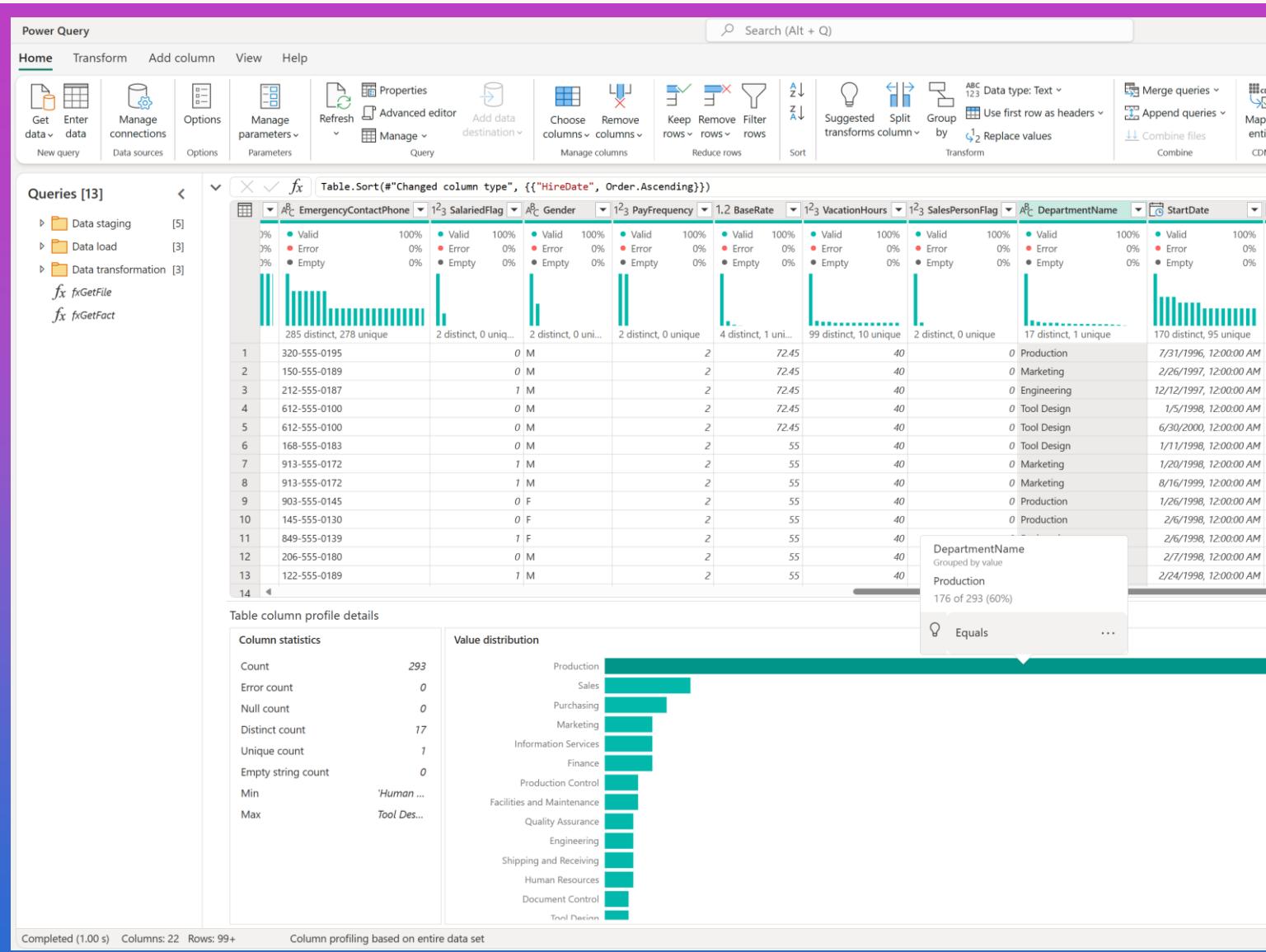
The screenshot shows the Microsoft Power Query interface. The ribbon at the top includes Home, Transform, Add column, View, and Help tabs. The Home tab is selected. Below the ribbon are various icons for data management, such as Get data, Manage connections, Options, and Refresh. The main area displays a table titled "Question" with the following data:

	Question
1	Apple
2	Aple
3	Pineapple
4	Water melon
5	watermln
6	watermleon
7	banana
8	Bananans
9	apls

The "Transform" tab is active, showing a formula bar with the code: `Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText("i45WciwoyElvitUBsaCMgMy81ES4chhiSWqRQm5qTn4emF804ufmoHBSoXJJix1ACGY6gZl5xwBOYkE0kBELAA==", ...`. The interface also includes sections for Queries [3], Manage columns, and various transform tools like Split column, Group by, and Replace values.

# 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.

Merge [?](#)

Select a table and matching columns to create a merged table.

Left table for merge \*

1 <sup>2</sup> ProductKey	1 <sup>2</sup> ProductLabel	A <sup>B</sup> ProductName	A <sup>B</sup> ProductDescription
1	101001	Contoso 512MB MP3 Player E51 Silver	512MB USB driver p
2	101002	Contoso 512MB MP3 Player E51 Blue	512MB USB driver p
3	101003	Contoso 1G MP3 Player E100 White	1GB flash memory a
4	101004	Contoso 2G MP3 Player E200 Silver	2GB flash memory, l

Right table for merge \*

1 <sup>2</sup> ProductSubcategoryKey	1 <sup>2</sup> ProductSubcategoryLabel	A <sup>B</sup> ProductSubcategoryName	A <sup>B</sup> ProductSubcategoryDescription
1	101	MP4&MP3	MP4&MP3
2	102	Recorder	Recorder
3	103	Radio	Radio
4	104	Recording Pen	Recording I

Join kind

Left outer

Right outer

Full outer

Inner

Left anti

Right anti

Use fuzzy matching to perform the merge

› Fuzzy matching options

 The selection matches 2,517 rows from both the tables

[OK](#) [Cancel](#)

Suggestions [Learn more](#)

Select any of the suggested column-pair mappings for the selected tables.

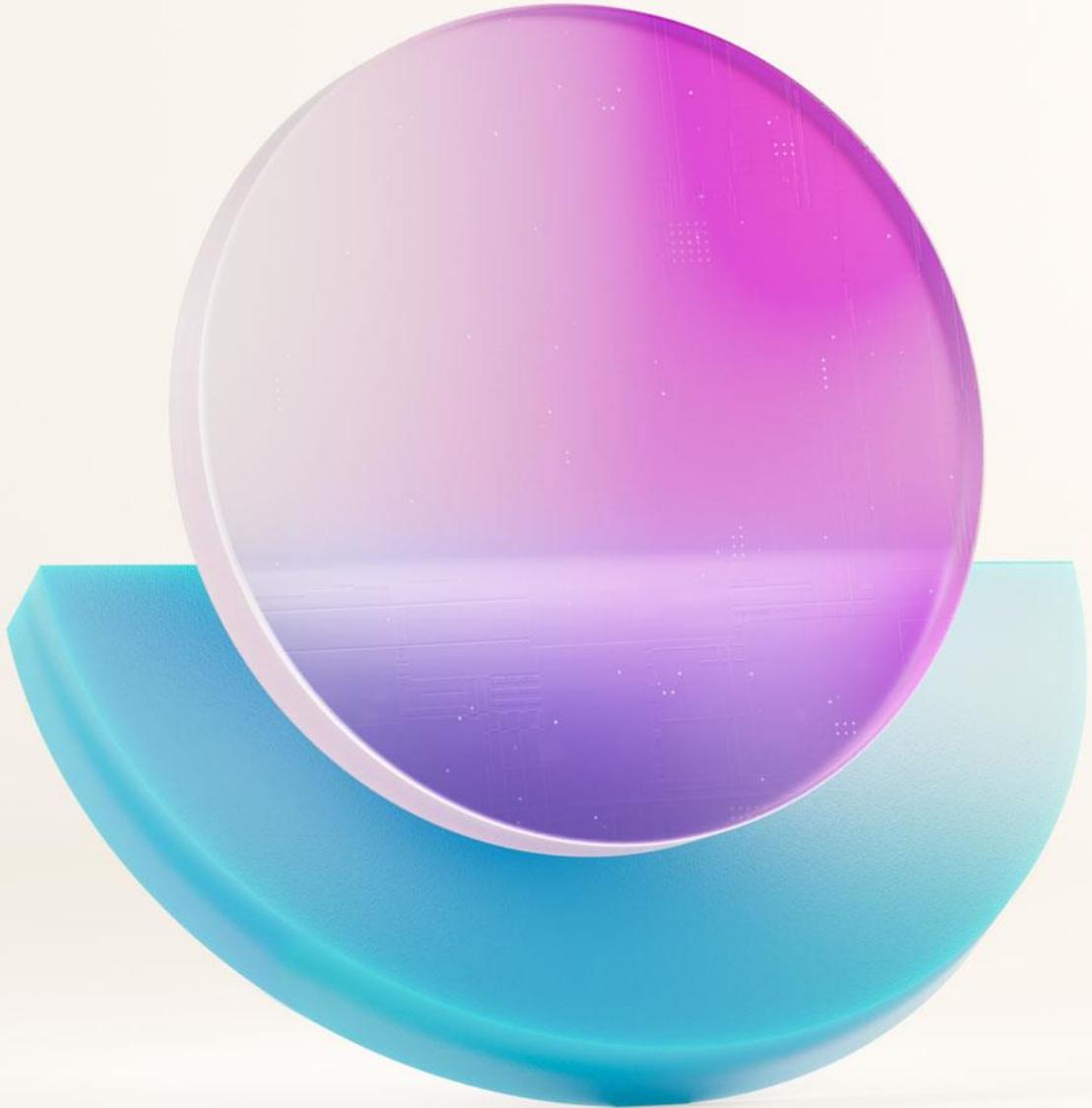
DimProduct_raw	DimProductSubcategory_raw
1 <sup>2</sup> ProductSubcategoryKey	→ 1 <sup>2</sup> ProductSubcategoryKey

DimProduct_raw	DimProductSubcategory_raw
1 <sup>2</sup> ProductKey	→ 1 <sup>2</sup> ProductCategoryKey

Applied steps

- Source
- Expanded
- Merged
- Expanded
- Removed
- Get column
- Select none
- Select column

32 distinct, 0 unique

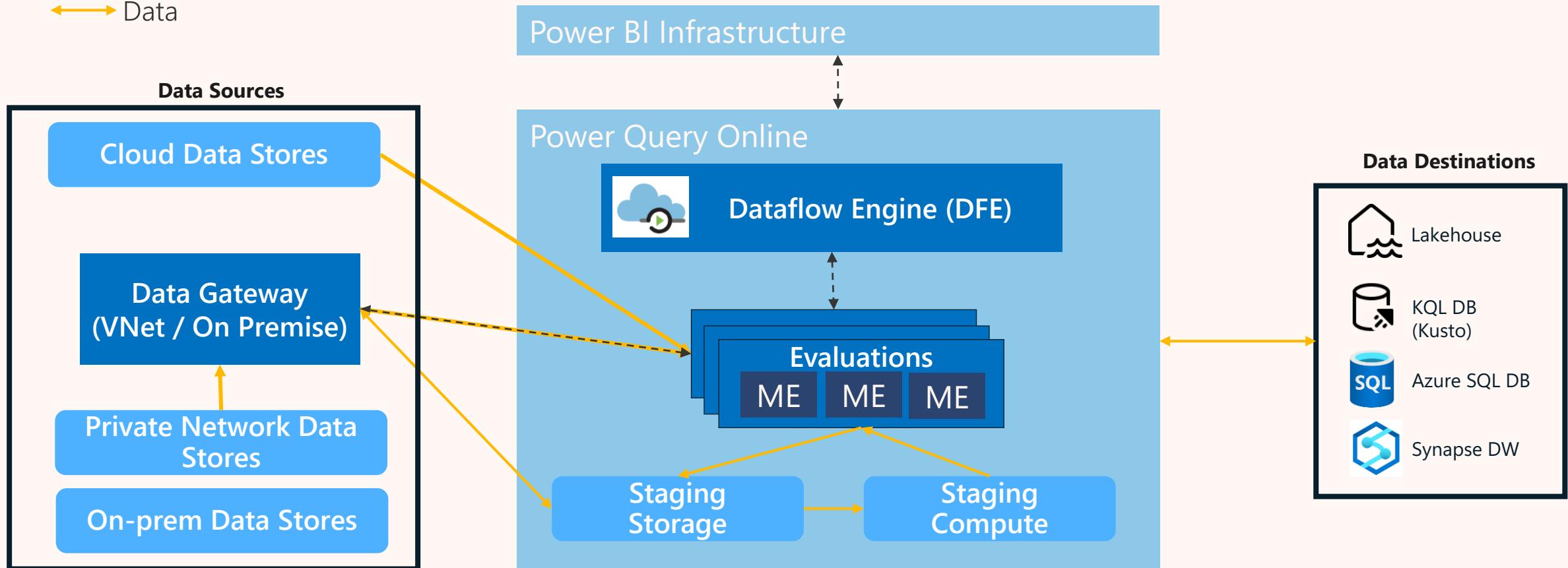


## Dataflows In-Depth

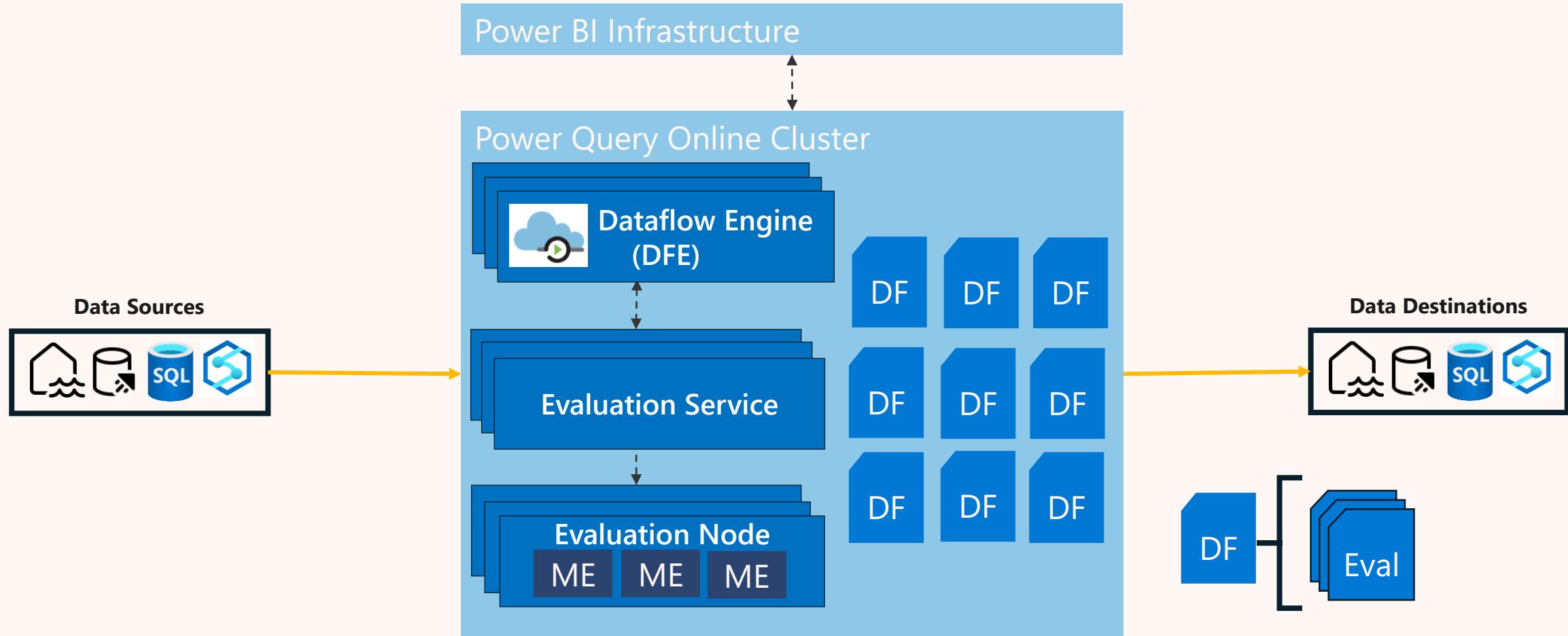
# Fabric Dataflows (aka Gen2)

↔ Command and Control

↔ Data

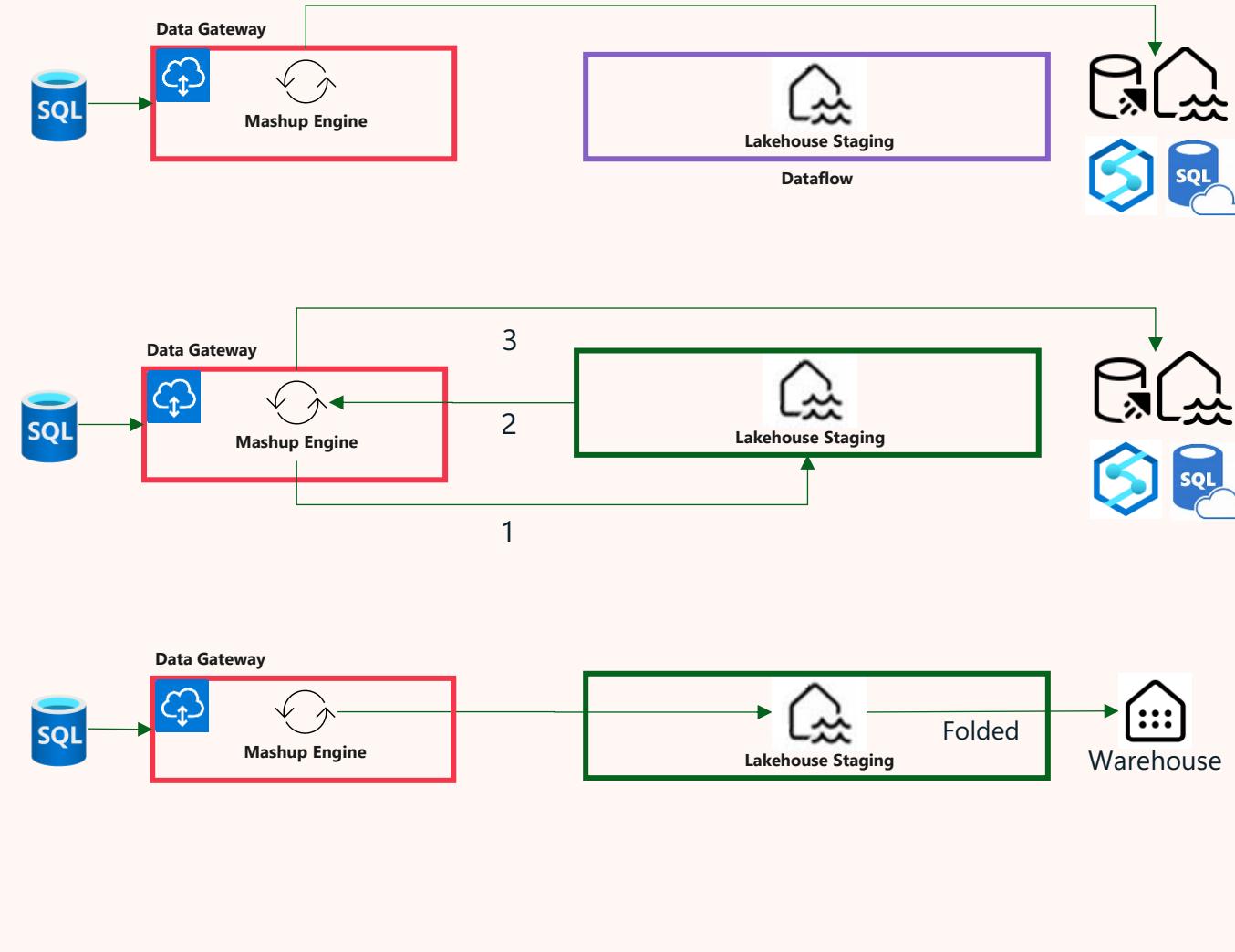


# Fabric Dataflows Scale Out



# 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 <del>lazy</del> 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

The screenshot shows the Power Query Editor interface with a query step highlighted. The step contains the following M code:

```
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
```

Annotations with arrows point to specific parts of the code:

- A purple arrow points from the "Remove other columns" menu item in the context menu to the "Remove other columns" part of the code.
- A blue arrow points from the "Number filters" menu item in the context menu to the "Filter" part of the code.
- A green arrow points from the "Transform column" menu item in the context menu to the "Group by..." part of the code.
- An orange arrow points from the "Unpivot columns" menu item in the context menu to the "Rename" part of the code.

The bottom part of the screenshot shows a preview of the "Event" column, which lists "Thunderstorm Wind", "Hail", "Flash Flood", and "Winter Weather".

# 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]
```

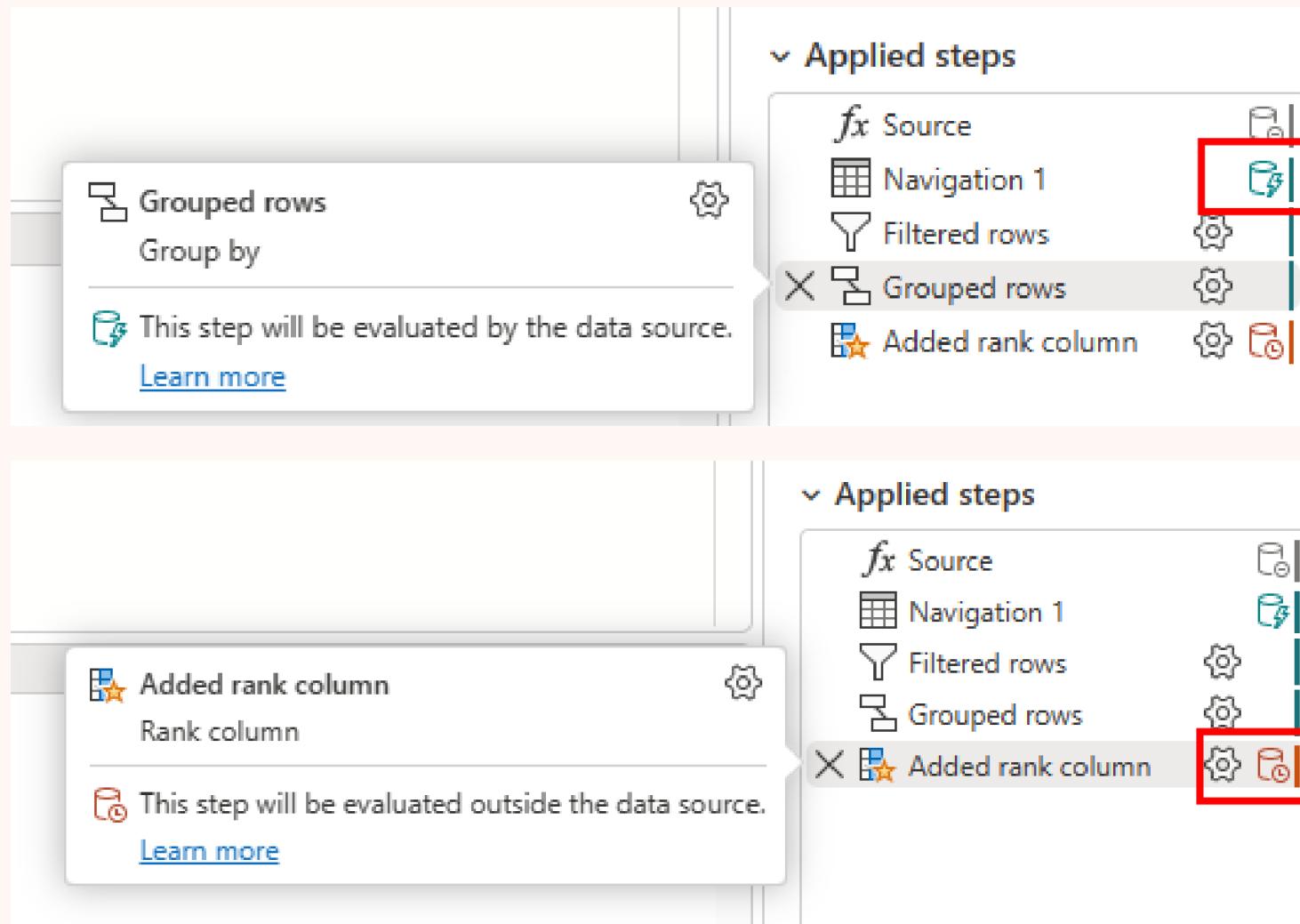
# Query Folding – Indicators

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

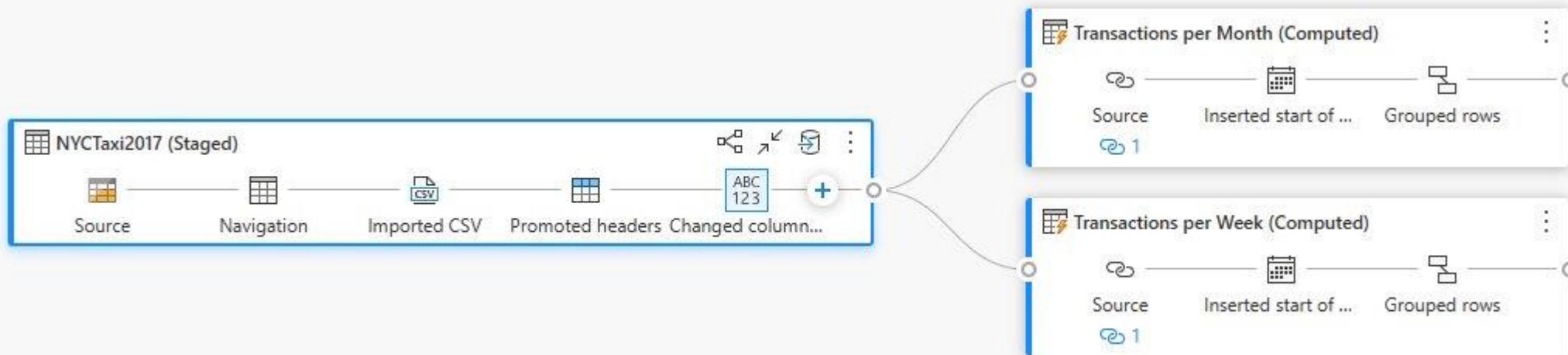
- Reduced data movement
  - In-memory evaluation engine would have to first transfer the data from the source
  - A filter that selects 10 rows from a 10M row table, can avoid transferring 10M rows
- Many data sources have highly efficient “query processors”
  - Optimized query plans
  - Ability to leverage 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

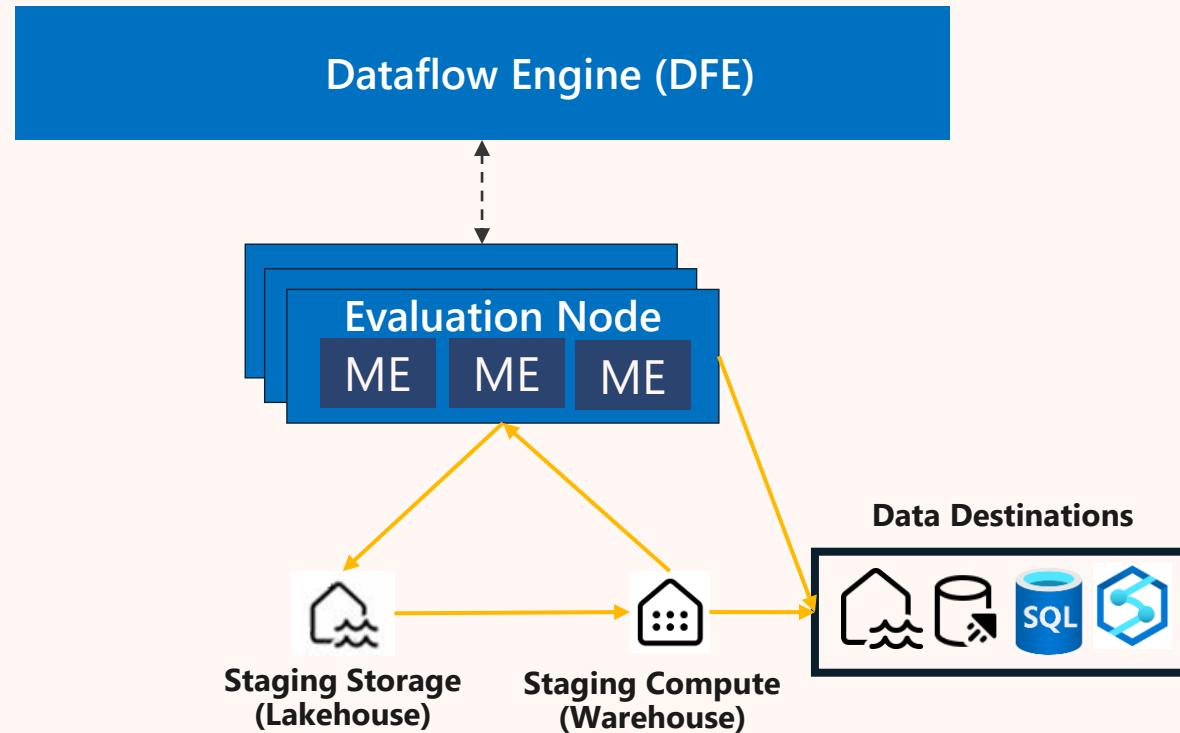
- **Data sources without rich query folding (e.g. files) are good candidates for staging**, especially when the files are large
- Data sources with rich query folding (e.g. databases) often do not require staging
- For a faster authoring experience create **separate dataflows for staging and transformation**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

# 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



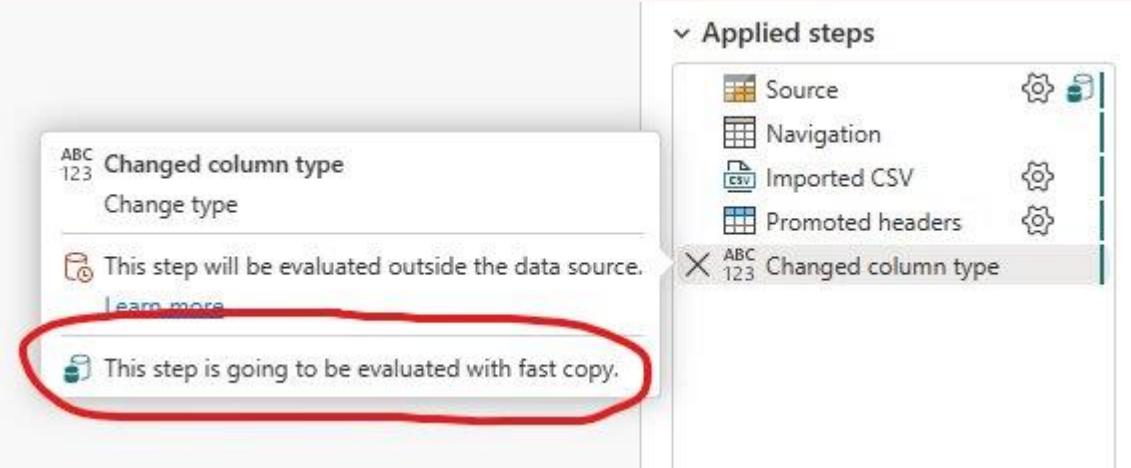
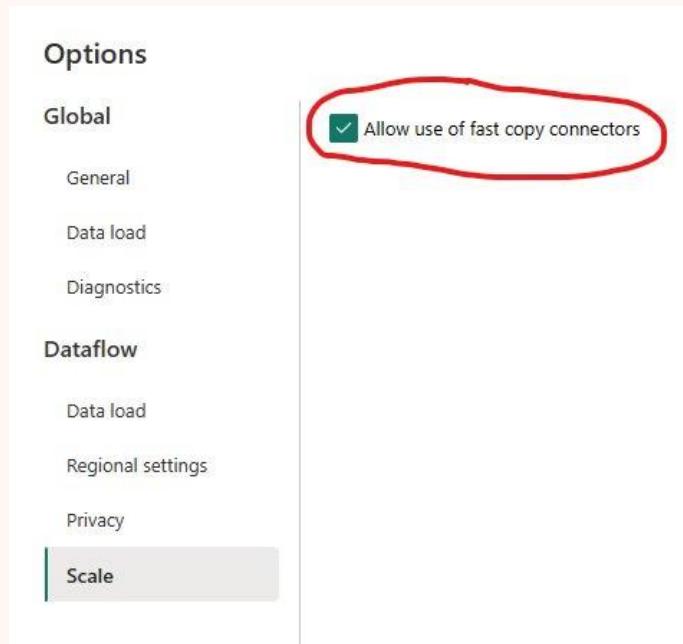
# Fast Copy – What it is

## Highly parallelized data movement

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

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

### Step 1: Enable Fast Copy



### Step 3: Win

Details	
Postgres Test > 3/7/2024, 3:38:17 PM > public Address	
Name	Status
public Address	Succeeded
Start time	End time
3/7/2024, 3:38:27 PM	3/7/2024, 3:40:51 PM
Duration	Engine
00:02:24	CopyActivity

# 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

**Default is Serialized ( $T_1 + T_2 + T_3 + T_4$ )**

Partition 1

Partition 2

Partition 3

Partition 4

**Fast Copy is Parallel ( $T_4$ )**

Partition 1

Partition 2

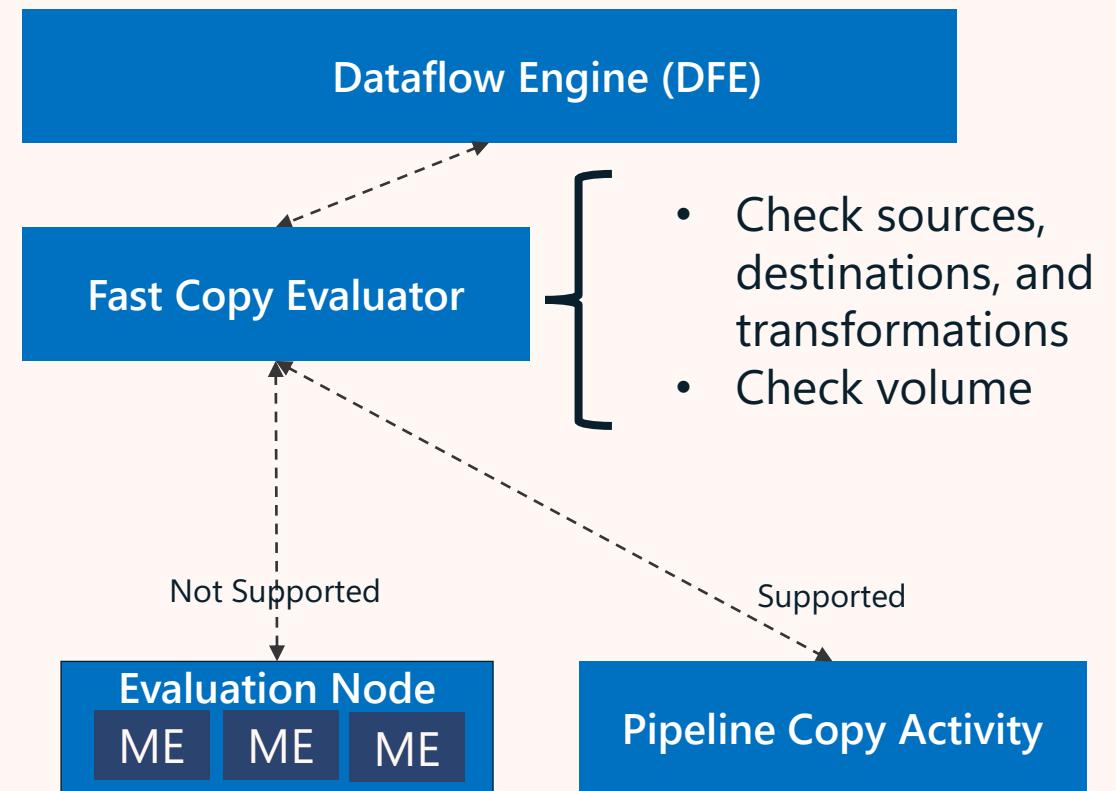
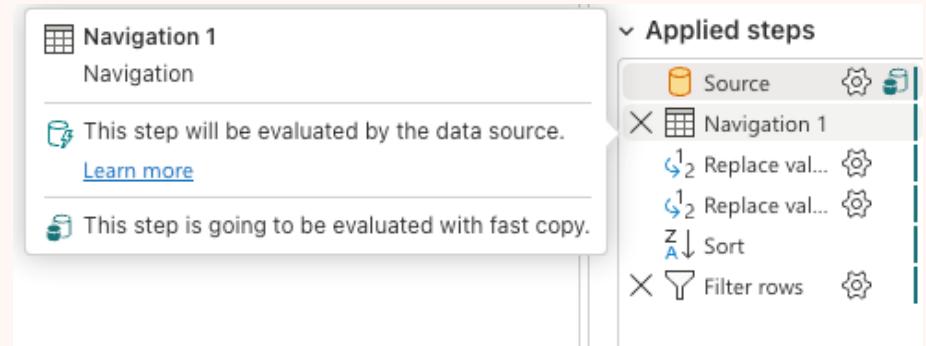
Partition 3

Partition 4

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

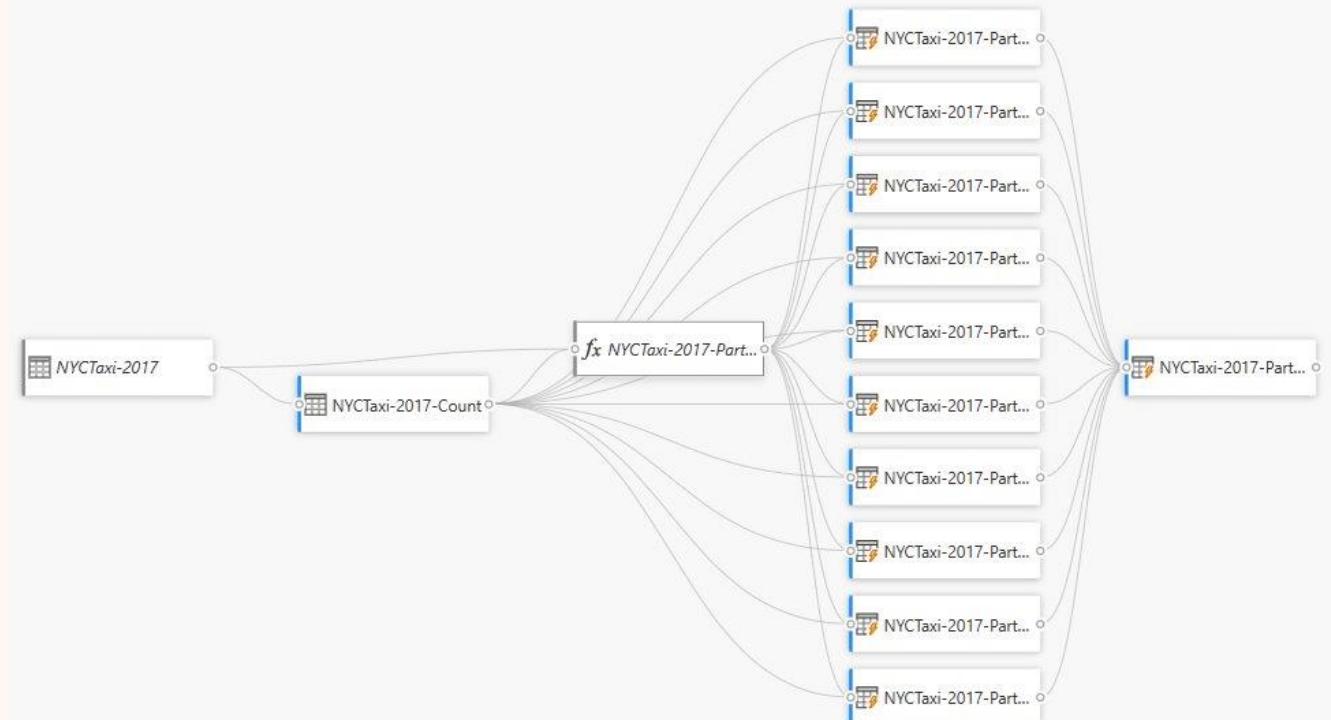
- Breaking long-running queries into smaller queries that can be run in parallel
- Explicit specification of what Fast Copy does implicitly
- Faster for the same reason as Fast Copy – parallelism

# 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}](https://github.com/BenniDeJagere/Presentations/{Year}/{YYYYMMDD}_{Event})

