SQL
KONFERENZ
2026



Warehouse loading – tips and tricks for better performance

Margarita Naumova

CEO, Data Architect,
Inspir-IT AS, Inspirit EOOD

Magi Naumova

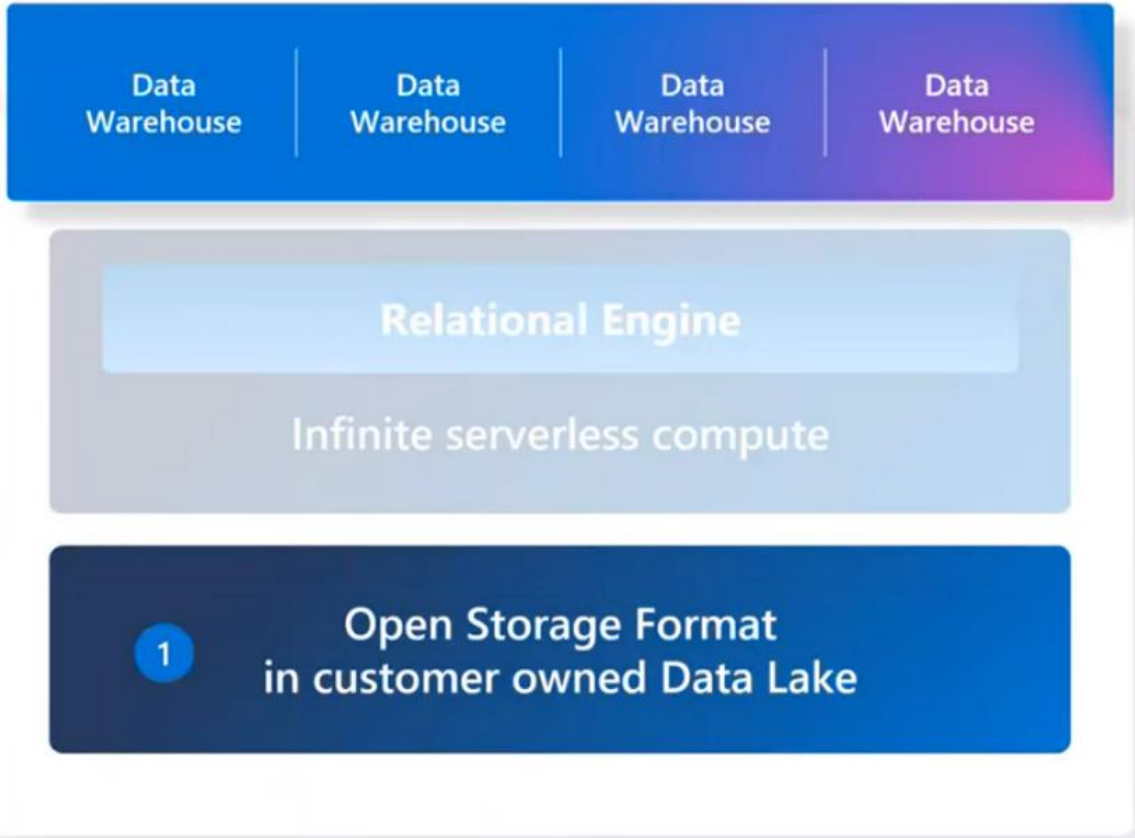
- Founder and CEO of Inspirit LTD and Inspir-IT AS
- Founder of SQL Master Academy
- Consultant, Trainer, Author, Speaker
- 25+ years of SQL Server Experience

- MCM SQL Server, Azure Data Engineer, MCT
- 15 years MVP Data Platform

magi@inspir-it.no
Magi.Naumova@inspirit.bg
https://www.linkedin.com/in/maginaumova/

www.inspir-it.no
www.inspirit.bg
www.sqlmasteracademy.com

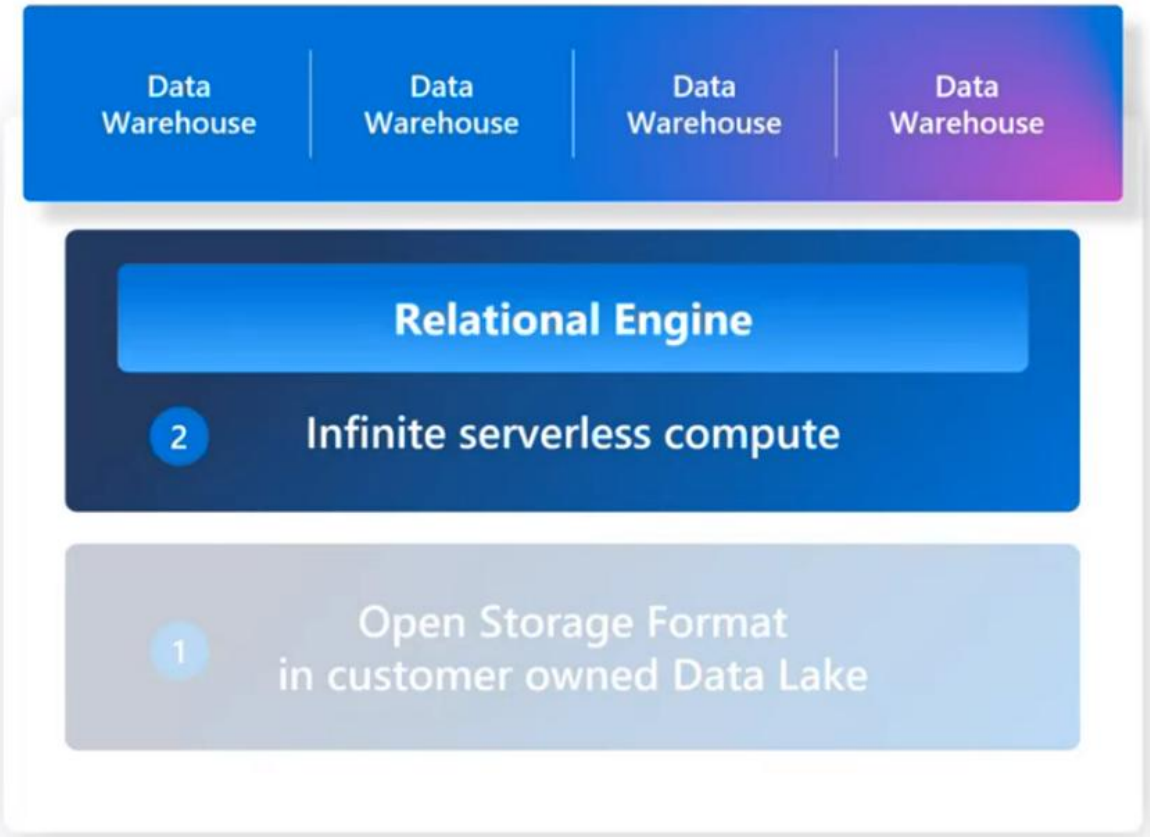
Fabric Warehouse



1 Open standard format in an open data lake replaces proprietary formats as the native storage

- First transactional data warehouse natively embracing an open standard format
- Data is stored in Delta – Parquet with no vendor lock-in
- Is auto-integrated and auto-optimized with minimal knobs
- Extends full SQL ecosystem benefits

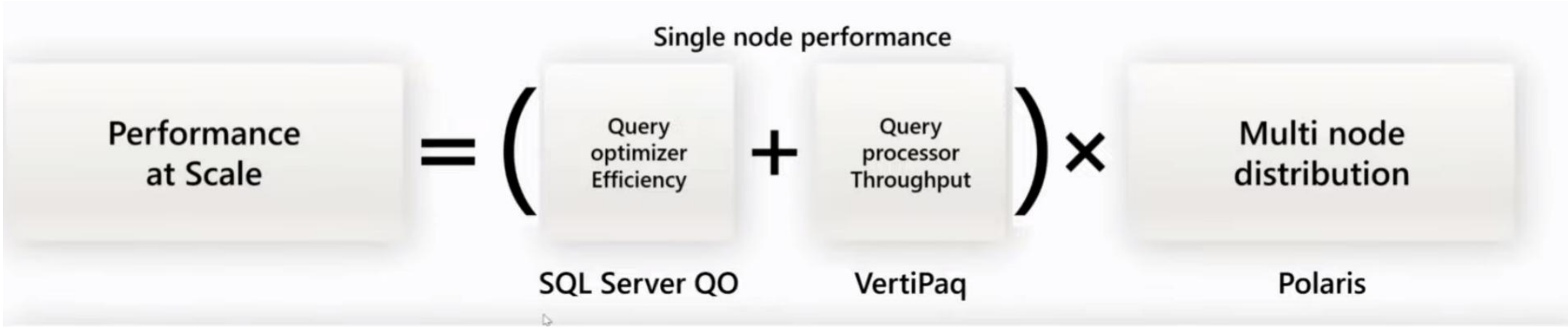
Fabric Warehouse

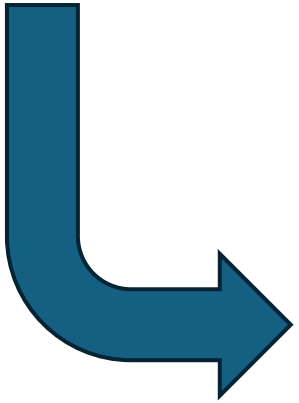


2 Dedicated clusters are replaced by serverless compute infrastructure

- Physical compute resources assigned within milliseconds to jobs
- Infinite scaling with dynamic resource allocation tailored to data volume and query complexity
- Instant scaling up/down with no physical provisioning involved
- Resource pooling providing significant efficiencies and pricing

Fabric Warehouse Performance





Best performance with no tuning required

Intelligent Workload Management

Optimized Storage & Automated Maintenance

Intelligent Data Distribution

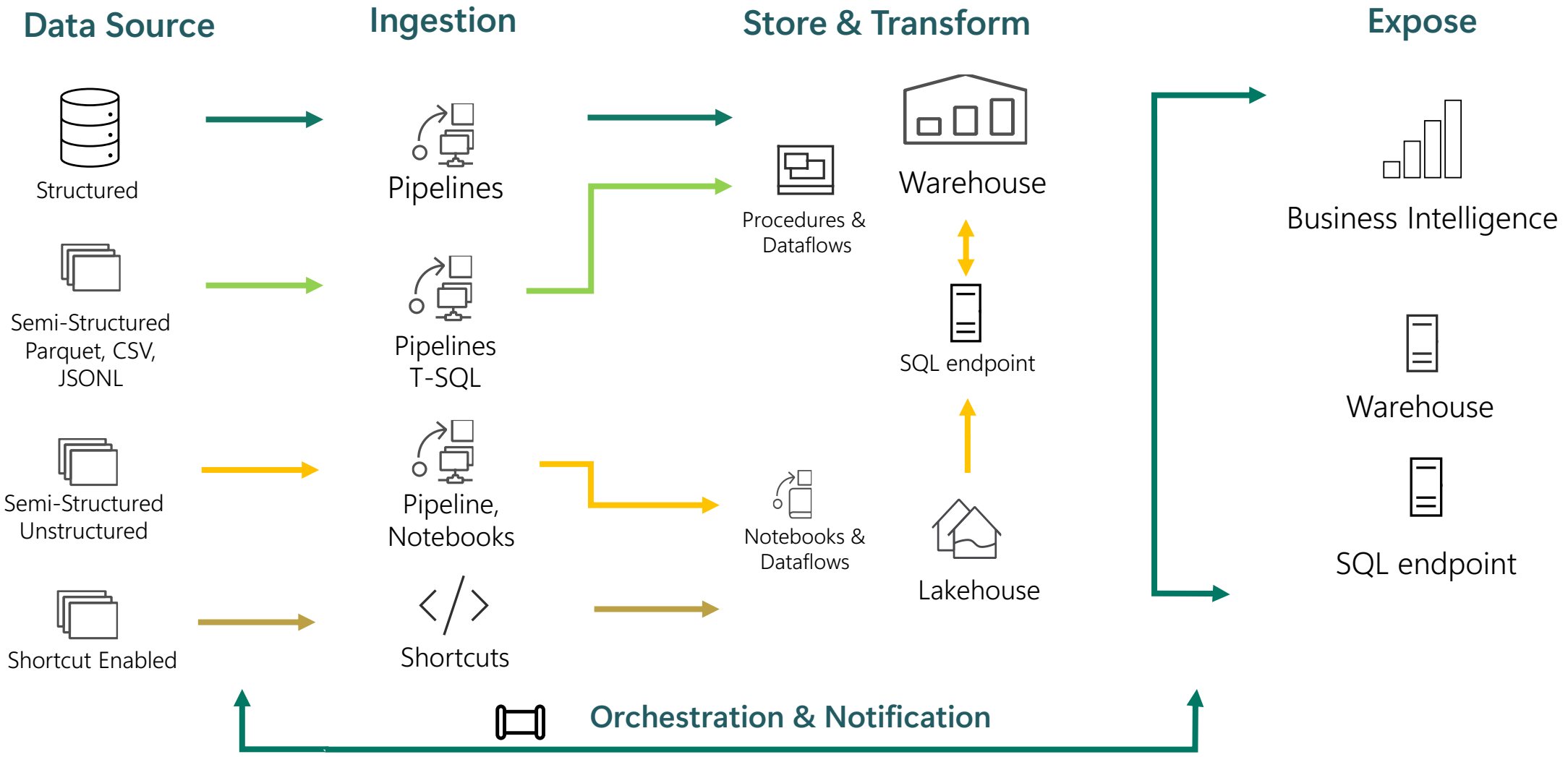
Self-managed performance for scale

What You *can* influence

- **Architecting** the warehouse to gain the best of it!
- **Choose** a Design Scenario:
  - Traditional Warehouse
  - Virtual Warehouses
  - WHS Medalion Architecture
- **Adjust** the design for scale:
  - The loading process and tools
  - The workload and querying
  - Optimizing the costs

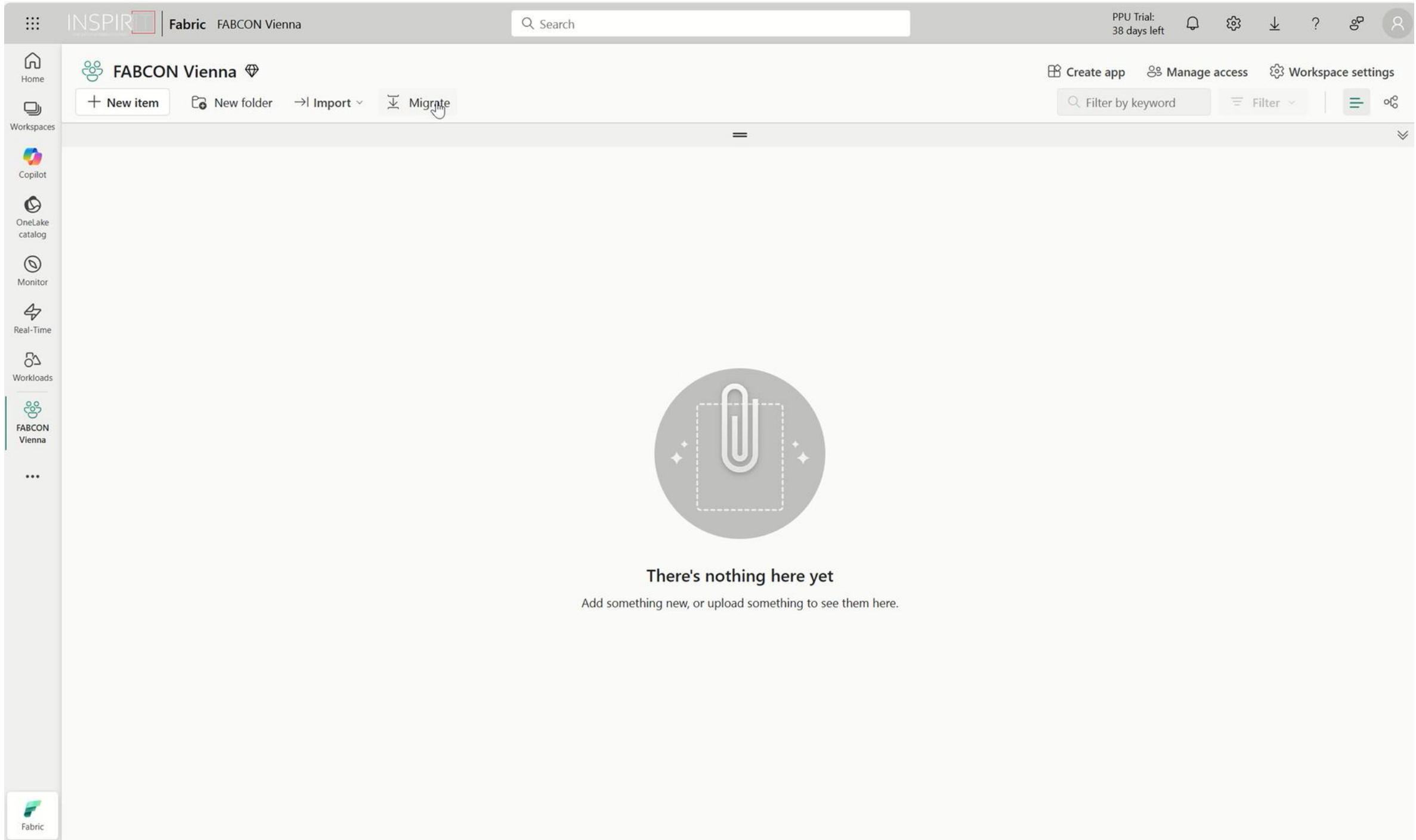


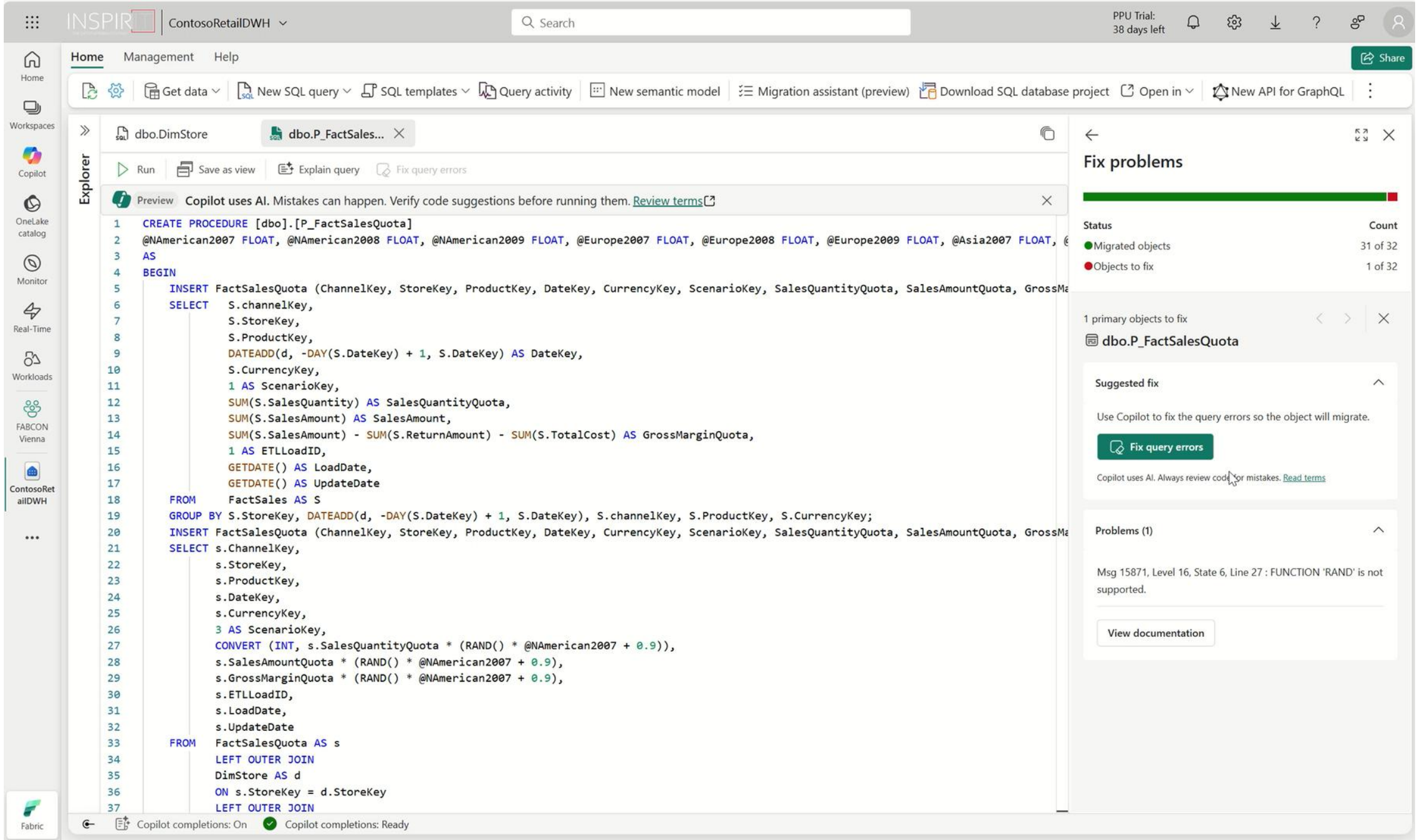
A Modernized Warehouse in MS Fabric

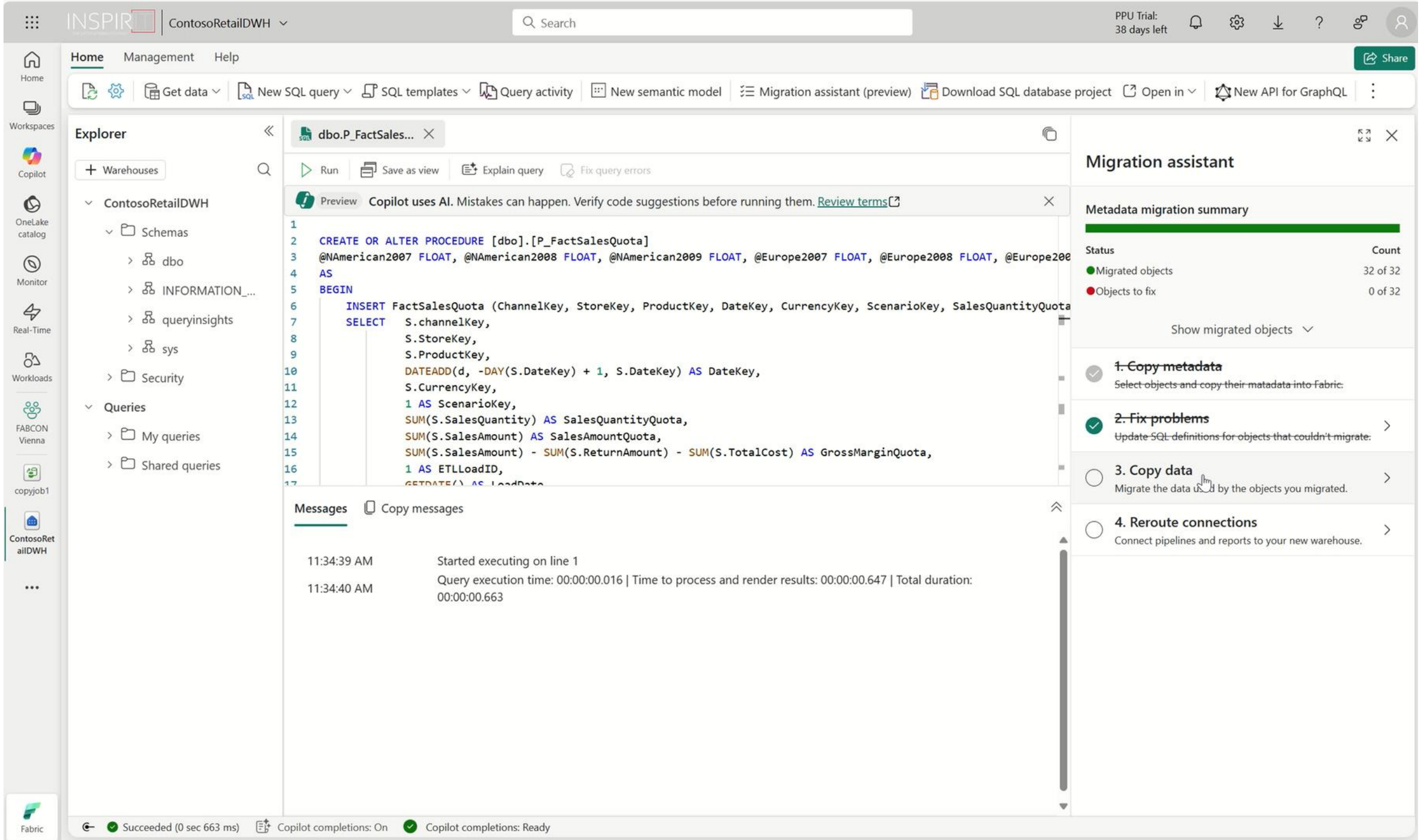


DEMO

Migrate and load an existing Azure SQL warehouse in Fabric Warehouse


Fabric FABCON Vienna
Search
PPU Trial:
38 days left
Home
Workspaces
Copilot
OneLake catalog
Monitor
Real-Time
Workloads
FABCON Vienna
Fabric
FABCON Vienna
Create app
Manage access
Workspace settings
New item
New folder
Import
Migrate
Filter by keyword
Filter
There's nothing here yet
Add something new, or upload something to see them here.


ContosoRetailDWH
Search
PPU Trial:
38 days left
Home
Management
Help
Share
Get data
New SQL query
SQL templates
Query activity
New semantic model
Migration assistant (preview)
Download SQL database project
Open in
New API for GraphQL
Explorer
dbo.DimStore
dbo.P_FactSales...
Run
Save as view
Explain query
Fix query errors
Preview
Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. Review terms
1 CREATE PROCEDURE [dbo].[P_FactSalesQuota]
2 @NAmerican2007 FLOAT, @NAmerican2008 FLOAT, @NAmerican2009 FLOAT, @Europe2007 FLOAT, @Europe2008 FLOAT, @Europe2009 FLOAT, @Asia2007 FLOAT,
3 AS
4 BEGIN
5 INSERT FactSalesQuota (ChannelKey, StoreKey, ProductKey, DateKey, CurrencyKey, ScenarioKey, SalesQuantityQuota, SalesAmountQuota, GrossMa
6 SELECT S.channelKey,
7 S.StoreKey,
8 S.ProductKey,
9 DATEADD(d, -DAY(S.DateKey) + 1, S.DateKey) AS DateKey,
10 S.CurrencyKey,
11 1 AS ScenarioKey,
12 SUM(S.SalesQuantity) AS SalesQuantityQuota,
13 SUM(S.SalesAmount) AS SalesAmount,
14 SUM(S.SalesAmount) - SUM(S.ReturnAmount) - SUM(S.TotalCost) AS GrossMarginQuota,
15 1 AS ETLLoadID,
16 GETDATE() AS LoadDate,
17 GETDATE() AS UpdateDate
18 FROM FactSales AS S
19 GROUP BY S.StoreKey, DATEADD(d, -DAY(S.DateKey) + 1, S.DateKey), S.channelKey, S.ProductKey, S.CurrencyKey;
20 INSERT FactSalesQuota (ChannelKey, StoreKey, ProductKey, DateKey, CurrencyKey, ScenarioKey, SalesQuantityQuota, SalesAmountQuota, GrossMa
21 SELECT s.ChannelKey,
22 s.StoreKey,
23 s.ProductKey,
24 s.DateKey,
25 s.CurrencyKey,
26 3 AS ScenarioKey,
27 CONVERT (INT, s.SalesQuantityQuota * (RAND() * @NAmerican2007 + 0.9)),
28 s.SalesAmountQuota * (RAND() * @NAmerican2007 + 0.9),
29 s.GrossMarginQuota * (RAND() * @NAmerican2007 + 0.9),
30 s.ETLLoadID,
31 s.LoadDate,
32 s.UpdateDate
33 FROM FactSalesQuota AS s
34 LEFT OUTER JOIN
35 DimStore AS d
36 ON s.StoreKey = d.StoreKey
37 LEFT OUTER JOIN
Fix problems
Status
Count
Migrated objects
31 of 32
Objects to fix
1 of 32
1 primary objects to fix
dbo.P_FactSalesQuota
Suggested fix
Use Copilot to fix the query errors so the object will migrate.
Fix query errors
Copilot uses AI. Always review code for mistakes. Read terms
Problems (1)
Msg 15871, Level 16, State 6, Line 27 : FUNCTION 'RAND' is not supported.
View documentation
Copilot completions: On
Copilot completions: Ready
Home
Workspaces
Copilot
OneLake catalog
Monitor
Real-Time
Workloads
FABCON Vienna
ContosoRetailDWH
Fabric


ContosoRetailDWH
Search
PPU Trial:
38 days left
Home
Management
Help
Share
Get data
New SQL query
SQL templates
Query activity
New semantic model
Migration assistant (preview)
Download SQL database project
Open in
New API for GraphQL
Explorer
Warehouses
ContosoRetailDWH
Schemas
dbo
INFORMATION_...
queryinsights
sys
Security
Queries
My queries
Shared queries
dbo.P_FactSales...
Run
Save as view
Explain query
Fix query errors
Preview
Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. Review terms
CREATE OR ALTER PROCEDURE [dbo].[P_FactSalesQuota]
@NAmerican2007 FLOAT, @NAmerican2008 FLOAT, @NAmerican2009 FLOAT, @Europe2007 FLOAT, @Europe2008 FLOAT, @Europe200
AS
BEGIN
INSERT FactSalesQuota (ChannelKey, StoreKey, ProductKey, DateKey, CurrencyKey, ScenarioKey, SalesQuantityQuota
SELECT S.channelKey,
S.StoreKey,
S.ProductKey,
DATEADD(d, -DAY(S.DateKey) + 1, S.DateKey) AS DateKey,
S.CurrencyKey,
1 AS ScenarioKey,
SUM(S.SalesQuantity) AS SalesQuantityQuota,
SUM(S.SalesAmount) AS SalesAmountQuota,
SUM(S.SalesAmount) - SUM(S.ReturnAmount) - SUM(S.TotalCost) AS GrossMarginQuota,
1 AS ETLLoadID,
Messages
Copy messages
11:34:39 AM Started executing on line 1
11:34:40 AM Query execution time: 00:00:00.016 | Time to process and render results: 00:00:00.647 | Total duration: 00:00:00.663
Migration assistant
Metadata migration summary
Status Count
Migrated objects 32 of 32
Objects to fix 0 of 32
Show migrated objects
1. Copy metadata
Select objects and copy their matadata into Fabric.
2. Fix problems
Update SQL definitions for objects that couldn't migrate.
3. Copy data
Migrate the data used by the objects you migrated.
4. Reroute connections
Connect pipelines and reports to your new warehouse.
Succeeded (0 sec 663 ms)
Copilot completions: On
Copilot completions: Ready



Results

Refresh

Status: In progress | Rows read: 0 | Rows written: 0 | Throughput: 0 byte/s | More

Filter | Column Options

Source → Destination	Status	Rows read	Rows written	Duration	Run start	Run end
dbo.DimDate → dbo.DimDate	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.FactITMachine → dbo.FactIT...	Queued	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimMachine → dbo.DimMac...	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimEmployee → dbo.DimEm...	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimCurrency → dbo.DimCurr...	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.FactInventory → dbo.FactInve...	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimOutage → dbo.DimOutage	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimProduct → dbo.DimProd...	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimStore → dbo.DimStore	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.FactExchangeRate → dbo.Fac...	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimProductCategory → dbo....	In progress	-	-	21 sec	9/14/2025, 12:19:40 PM	-
dbo.DimChannel → dbo.DimChan...	Not started	-	-	-	-	-



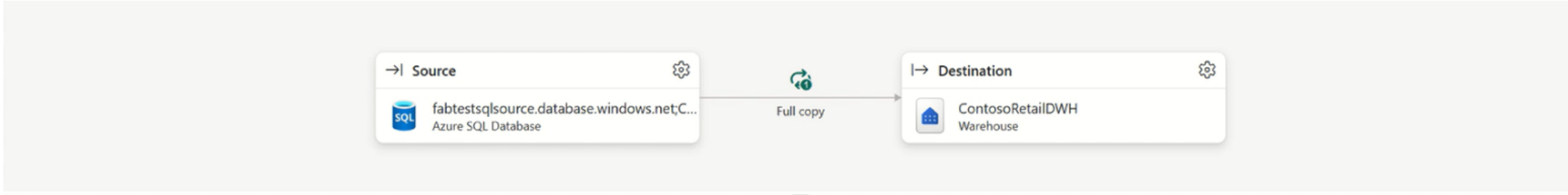
Results

Refresh

Status: In progress | Rows read: 58,215 | Rows written: 58,215 | Throughput: 931.1 KB/s | More

Filter | Column Options

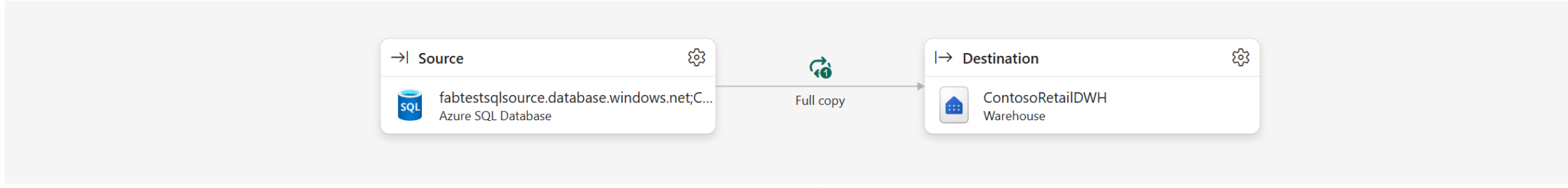
Source → Destination	Status	Rows read	Rows written	Duration	Run start	Run end
dbo.FactSalesQuota → dbo.FactSa...	In progress	-	-	17 sec	9/14/2025, 12:20:43 PM	-
dbo.FactITSLA → dbo.FactITSLA	In progress	-	-	17 sec	9/14/2025, 12:20:42 PM	-
dbo.FactOnlineSales → dbo.FactO...	In progress	-	-	20 sec	9/14/2025, 12:20:40 PM	-
dbo.FactStrategyPlan → dbo.FactS...	In progress	-	-	27 sec	9/14/2025, 12:20:33 PM	-
dbo.FactSales → dbo.FactSales	In progress	-	-	29 sec	9/14/2025, 12:20:31 PM	-
dbo.DimCustomer → dbo.DimCus...	Succeeded	18,869	18,869	1 min 2 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:42 PM
dbo.DimSalesTerritory → dbo.Dim...	Succeeded	265	265	1 min 0 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:40 PM
dbo.DimEntity → dbo.DimEntity	Succeeded	421	421	52 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:32 PM
dbo.DimScenario → dbo.DimScen...	Succeeded	3	3	1 min 15 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:55 PM
dbo.DimGeography → dbo.DimG...	Succeeded	674	674	1 min 17 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:58 PM
dbo.DimProductSubcategory → d...	Succeeded	44	44	52 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:32 PM



Results

Refresh | Status: In progress | Rows read: 57,538 | Rows written: 57,538 | Throughput: 927.63 KB/s | More | Filter | Column Options

Source → Destination	Status	Rows read	Rows written	Duration	Run start	Run end
dbo.DimMachine → dbo.DimMac...	Succeeded	7,816	7,816	54 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:34 PM
dbo.DimEmployee → dbo.DimEm...	Succeeded	293	293	52 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:32 PM
dbo.DimCurrency → dbo.DimCurr...	Succeeded	28	28	49 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:29 PM
dbo.FactInventory → dbo.FactInve...	In progress	-	-	1 min 5 sec	9/14/2025, 12:19:40 PM	-
dbo.DimOutage → dbo.DimOutage	Succeeded	303	303	49 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:29 PM
dbo.DimChannel → dbo.DimChan...	Succeeded	4	4	58 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:38 PM
dbo.DimPromotion → dbo.DimPr...	Succeeded	28	28	1 min 2 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:43 PM
dbo.DimProduct → dbo.DimProd...	Succeeded	2,517	2,517	53 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:33 PM
dbo.DimStore → dbo.DimStore	Succeeded	306	306	53 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:33 PM
dbo.FactExchangeRate → dbo.Fac...	Succeeded	773	773	51 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:31 PM
dbo.DimProductCategory → dbo....	Succeeded	8	8	51 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:31 PM



Results

Refresh

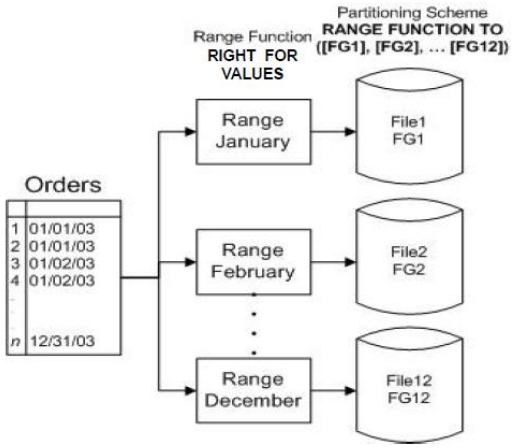
Status: Succeeded | Rows read: 34,326,475 | Rows written: 34,326,475 | Throughput: 36.441 MB/s | More

Filter | Column O

Source → Destination	Status	Rows read	Rows written	Duration	Run start	Run end
dbo.FactSalesQuota → dbo.FactSa...	Succeeded	7,465,911	7,465,911	2 min 41 sec	9/14/2025, 12:20:43 PM	9/14/2025, 12:23:24 PM
dbo.FactITSLA → dbo.FactITSLA	Succeeded	4,925	4,925	45 sec	9/14/2025, 12:20:42 PM	9/14/2025, 12:21:28 PM
dbo.FactOnlineSales → dbo.FactO...	Succeeded	12,627,608	12,627,608	5 min 3 sec	9/14/2025, 12:20:40 PM	9/14/2025, 12:25:44 PM
dbo.FactStrategyPlan → dbo.FactS...	Succeeded	2,750,628	2,750,628	1 min 13 sec	9/14/2025, 12:20:33 PM	9/14/2025, 12:21:46 PM
dbo.FactSales → dbo.FactSales	Succeeded	3,406,089	3,406,089	2 min 28 sec	9/14/2025, 12:20:31 PM	9/14/2025, 12:23:00 PM
dbo.DimCustomer → dbo.DimCus...	Succeeded	18,869	18,869	1 min 2 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:42 PM
dbo.DimSalesTerritory → dbo.Dim...	Succeeded	265	265	1 min 0 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:40 PM
dbo.DimEntity → dbo.DimEntity	Succeeded	421	421	52 sec	9/14/2025, 12:19:40 PM	9/14/2025, 12:20:32 PM

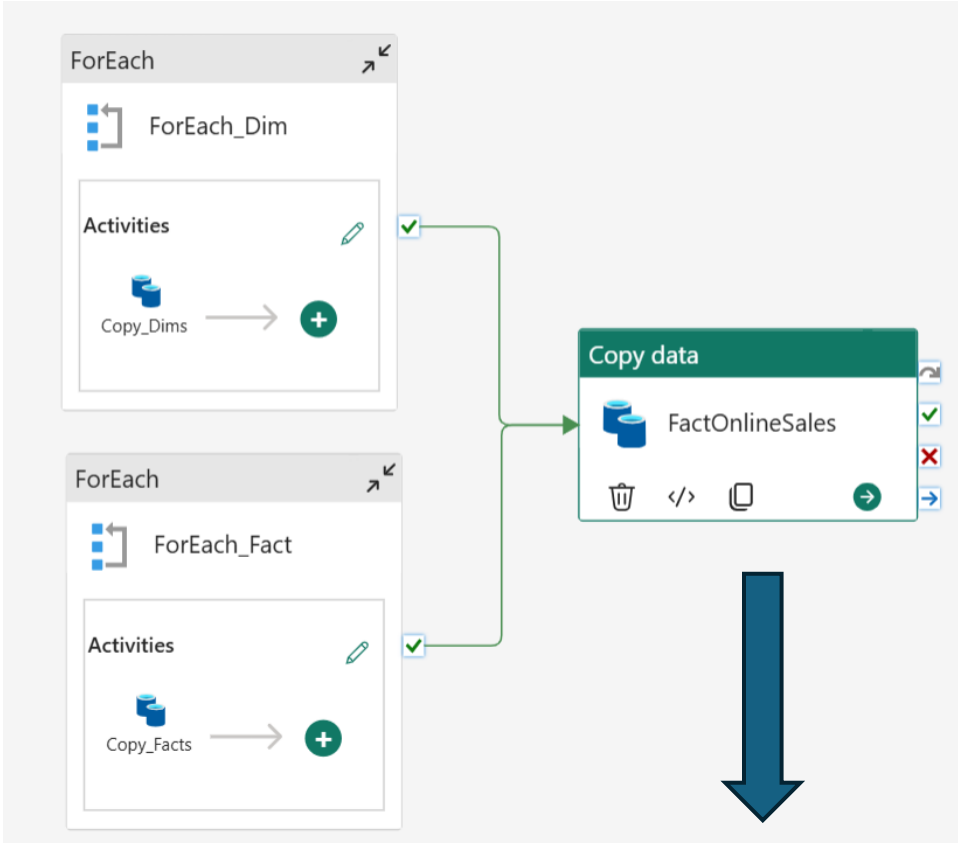
Pipelines opt

Design approaches for scalability and ingestion performance with a main goal: **Achieve parallel load!**

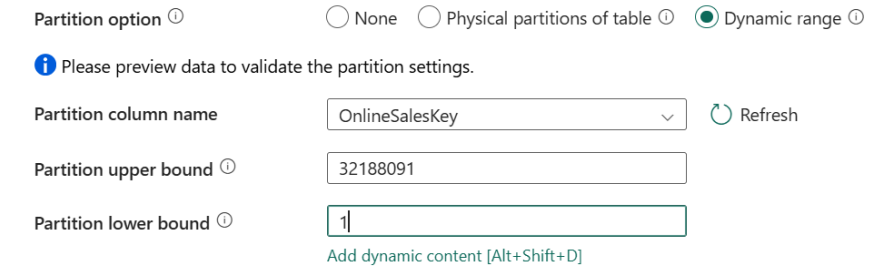


Source Physical Partitions

Logical partition design:



Dynamic range:



Pipelines opt

Adjusting Resources at the source and Capacity at Target WH, and the cost to support the design

AZURE SQL

Service Tier: General Purpose

- Compute Tier: Serverless
- Standard-series (Gen5)
    - Max vCores: 80
    - Min vCores 20
- Record Count: 1,500,000,000
- Region: East US 2

Trial capacity (F64) in Microsoft Fabric (West US 2).

Partition option	Degree of copy parallelism	Used parallel copies	Total duration	Capacity Units	Max CPU %	Max Session Count

```
Is your table large (100M+ rows)?
├── YES → Enable partitioning
│         ├── Has physical partitions? → Use "Physical partition"
│         └── No partitions? → Use "Dynamic range" on an INT/DATE column
│         Then set Degree of parallelism (start at 4-8, increase if source can
handle it)
└── NO  → Leave partition as None

Do you need speed above all else?
├── YES → ITO = Maximum
└── NO  → ITO = Balanced or Auto

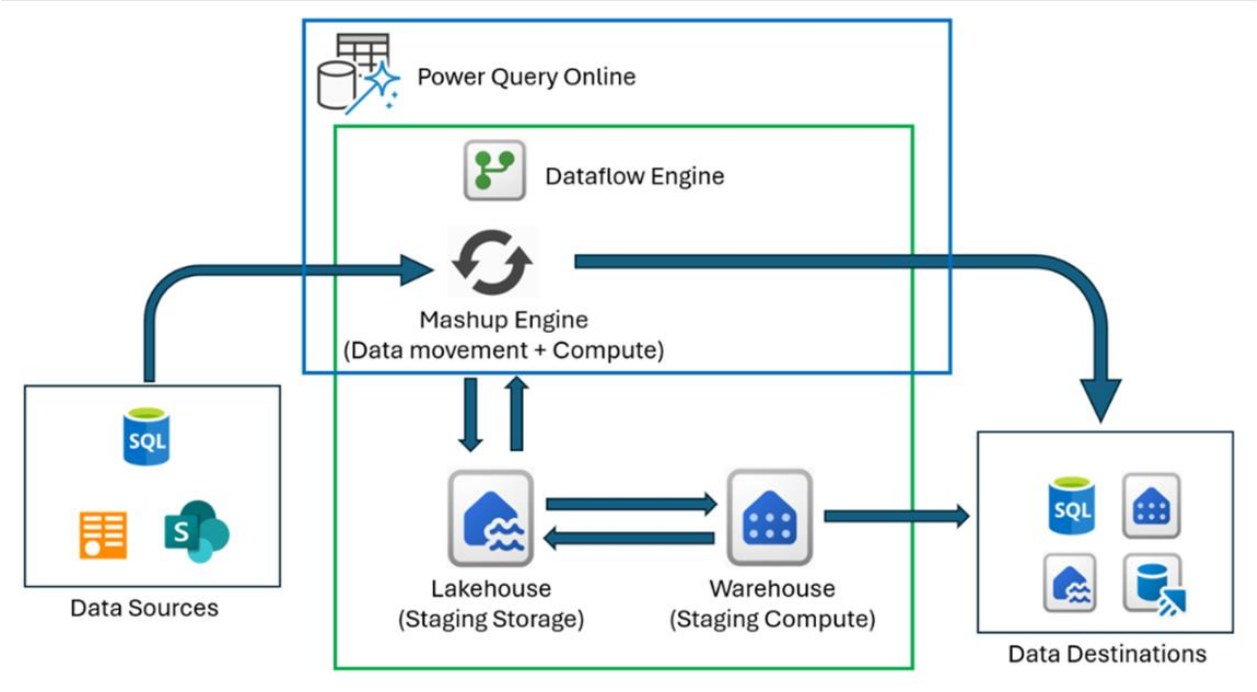
Is source DB live during copy?
├── YES → Set appropriate Isolation level (Snapshot is safest)
└── NO  → Isolation level = None
```

Copy activity performance with SQL databases - Microsoft Fabric | Microsoft Learn

When LOW CODE & Transformations are needed

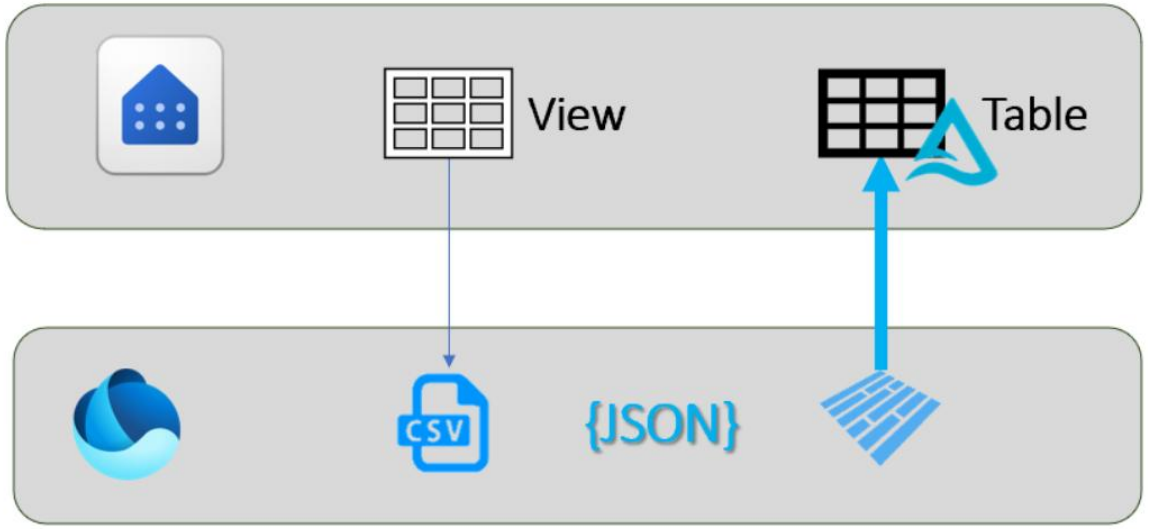
Data Flow Gen 2 opt

- Key areas to focus on for performance optimization
  - Using Fast Copy to **Improve data movement**
  - Check Query Folding to **reduce the amount of data transfer**
  - Using Staging to **Improve execution time for complex transformations**



Best practices for getting the best performance with Dataflow Gen2 in Fabric Data Factory - Microsoft Fabric | Microsoft Learn

Warehouse specific toolset (T-SQL)



VIEWS

OPENROWSET

Cross Database Queries

Virtualize or Materialize

SProc

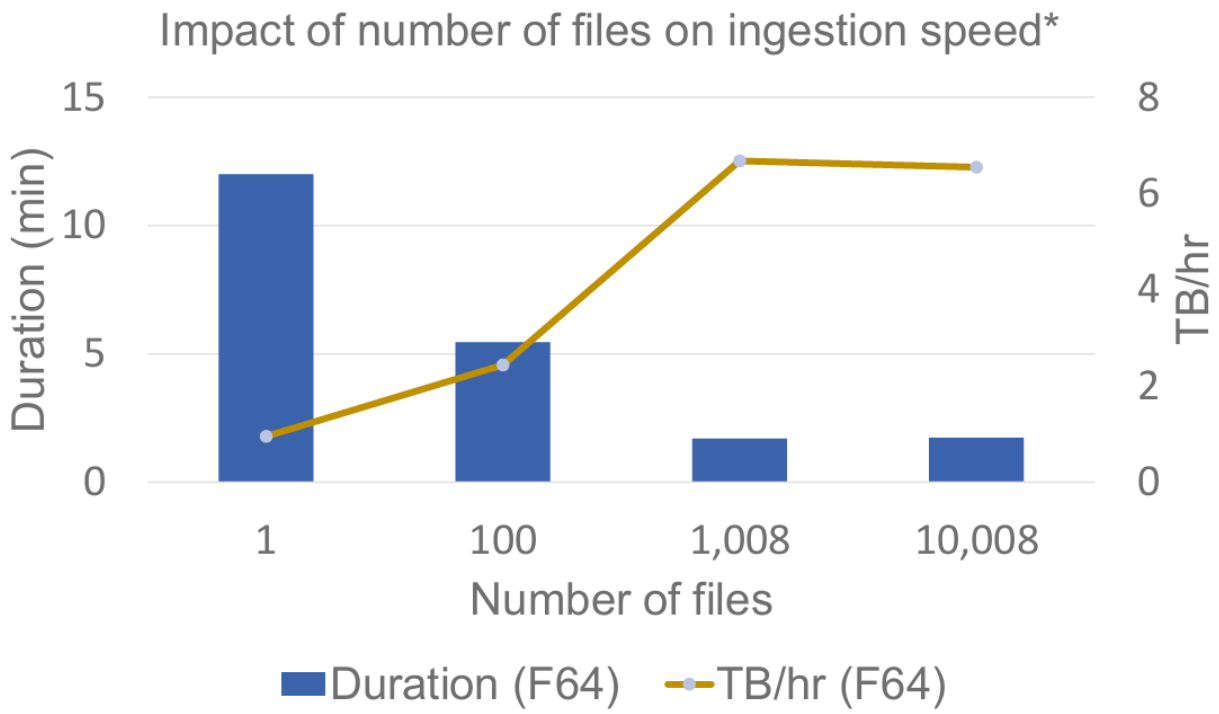
COPY INTO

CTAS

Ingestion best practices
Size and number of files

COPY INTO:

- More small files > few large files
- At least 4MB file size
- Keep the number of files to be at least 1,000
- Best throughput using 1,000 files is 7x faster than single large file

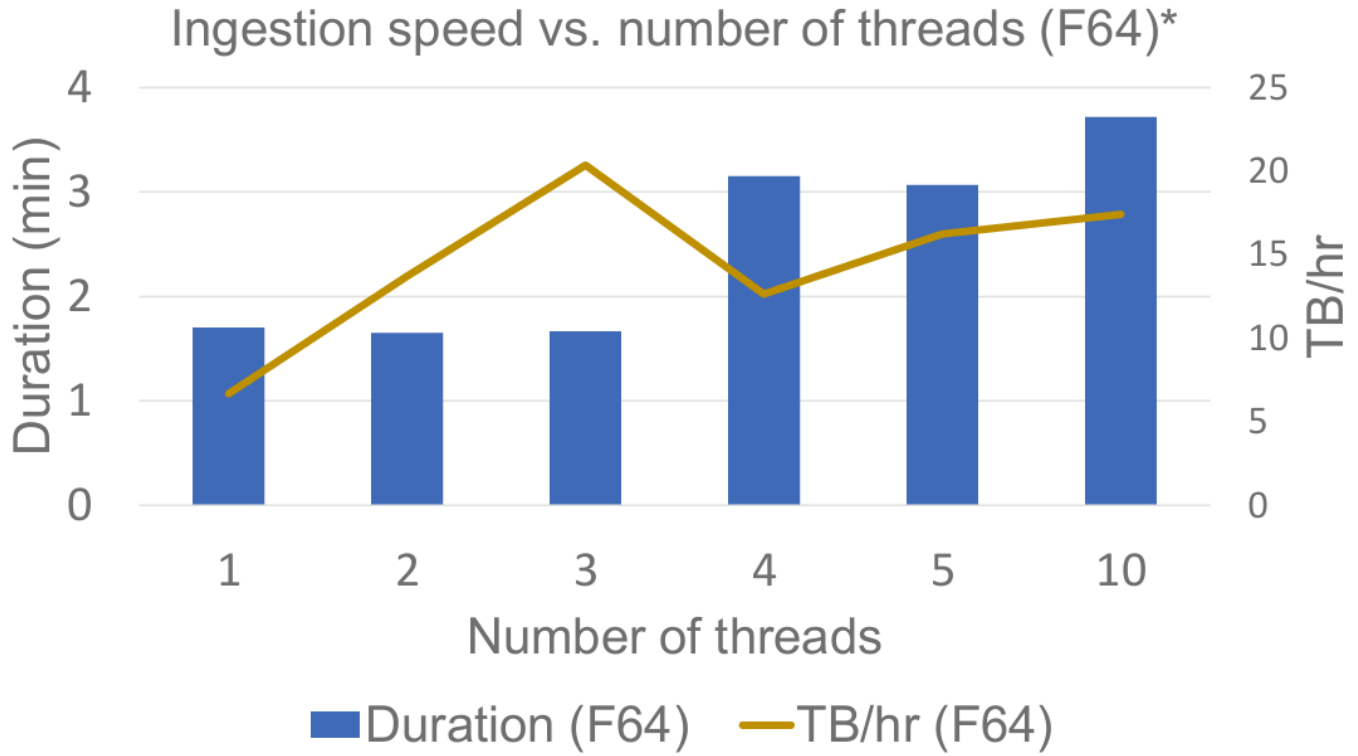


*TPCH 1TB, Orders table

Ingestion best practices
Parallelize

COPY INTO

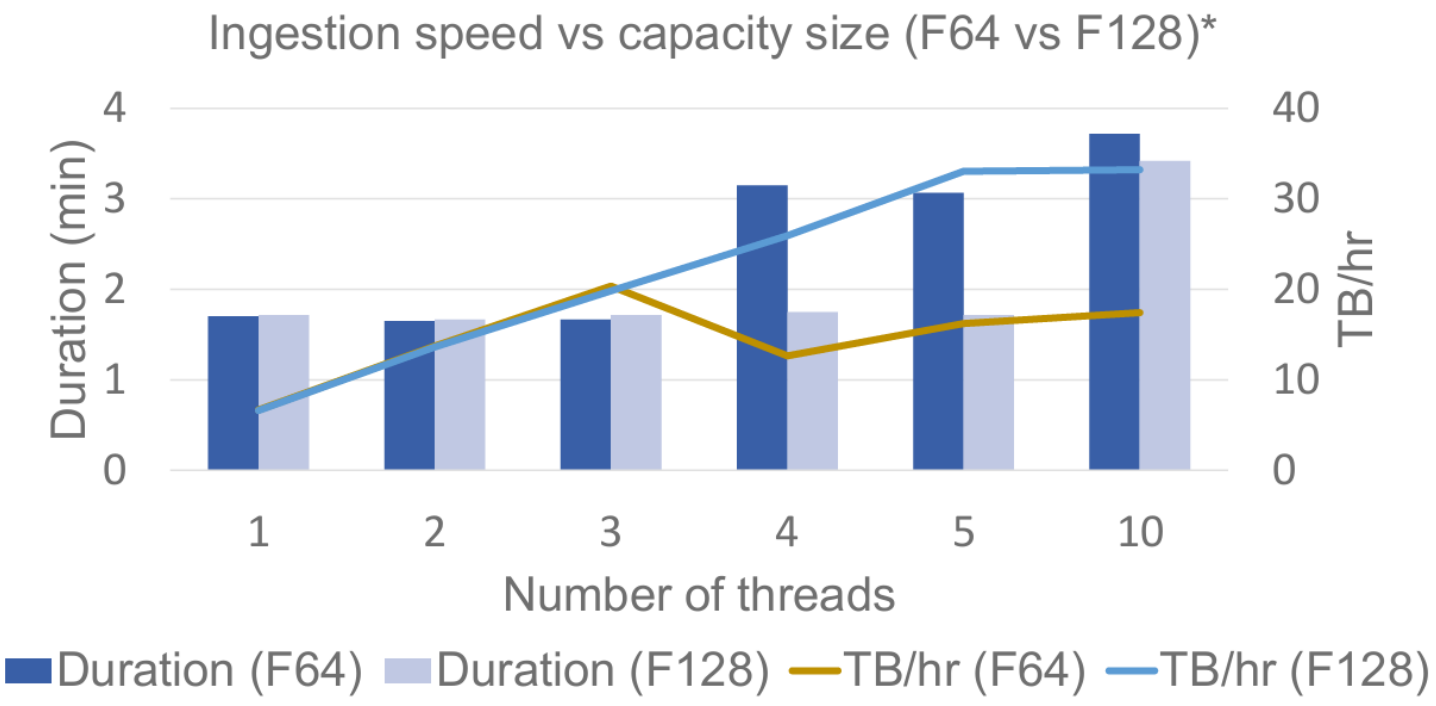
- Parallel loads achieve higher throughput



*TPCH 1TB, Orders table, 170GB each thread

Ingestion best practices
Increase throughput

- Increase throughput with larger Fabric capacities
- **ALL ingestion** (Pipelines, Datalows, SQL) scale with larger Fabric capacities

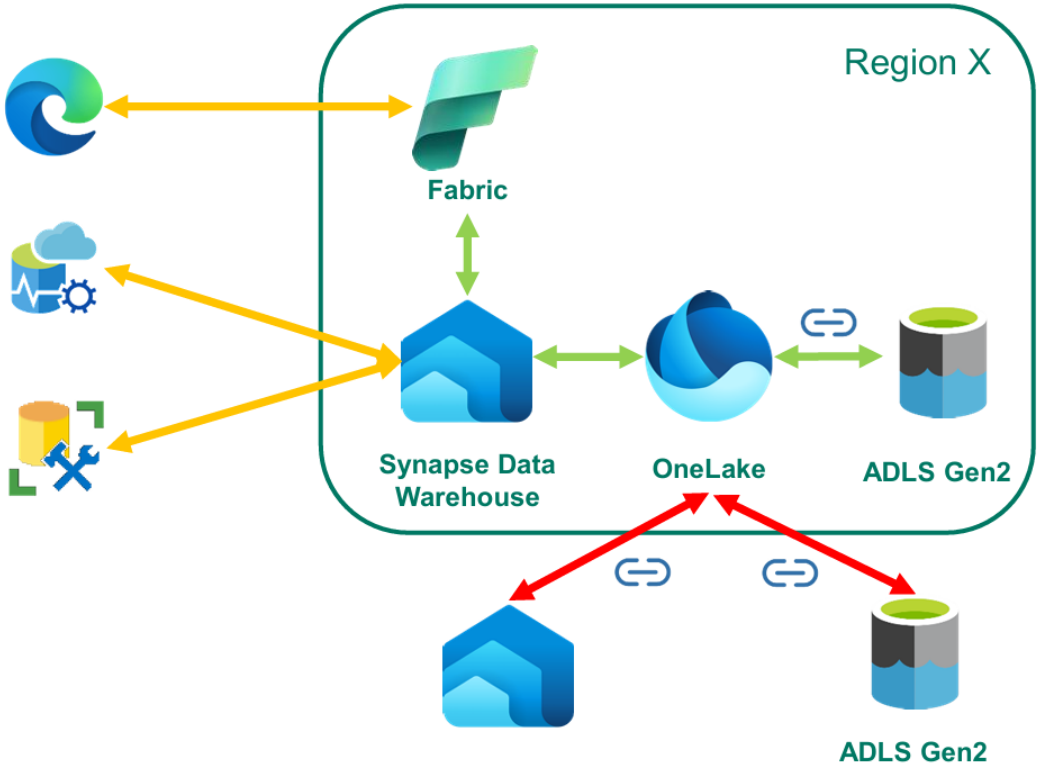


*TPCH 1TB, Orders table, 170GB each thread

Load or Shortcut?

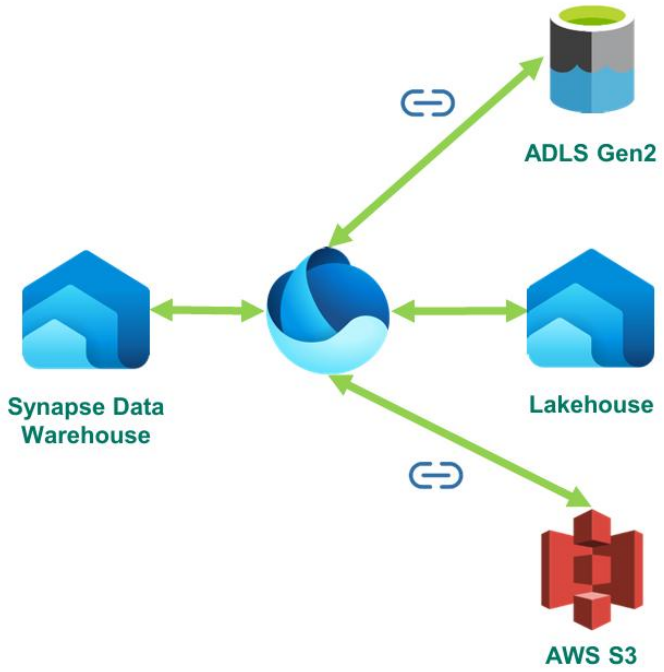
- WHO owns the data?
- WHERE the data resides?

- Collocate resources
  - Data as close to Capacity as possible
  - Mind the network latency between:
    - The client and the endpoint
    - The engine and the data
- In case collocation is not possible, better ingest



Optimize Lakehouse Tables

- Fabric Warehouse optimizes only warehouse tables!
- LKH tables need special attention



Check data types, particularly string lengths, decimal precision and scale and use NOT NULL if possible

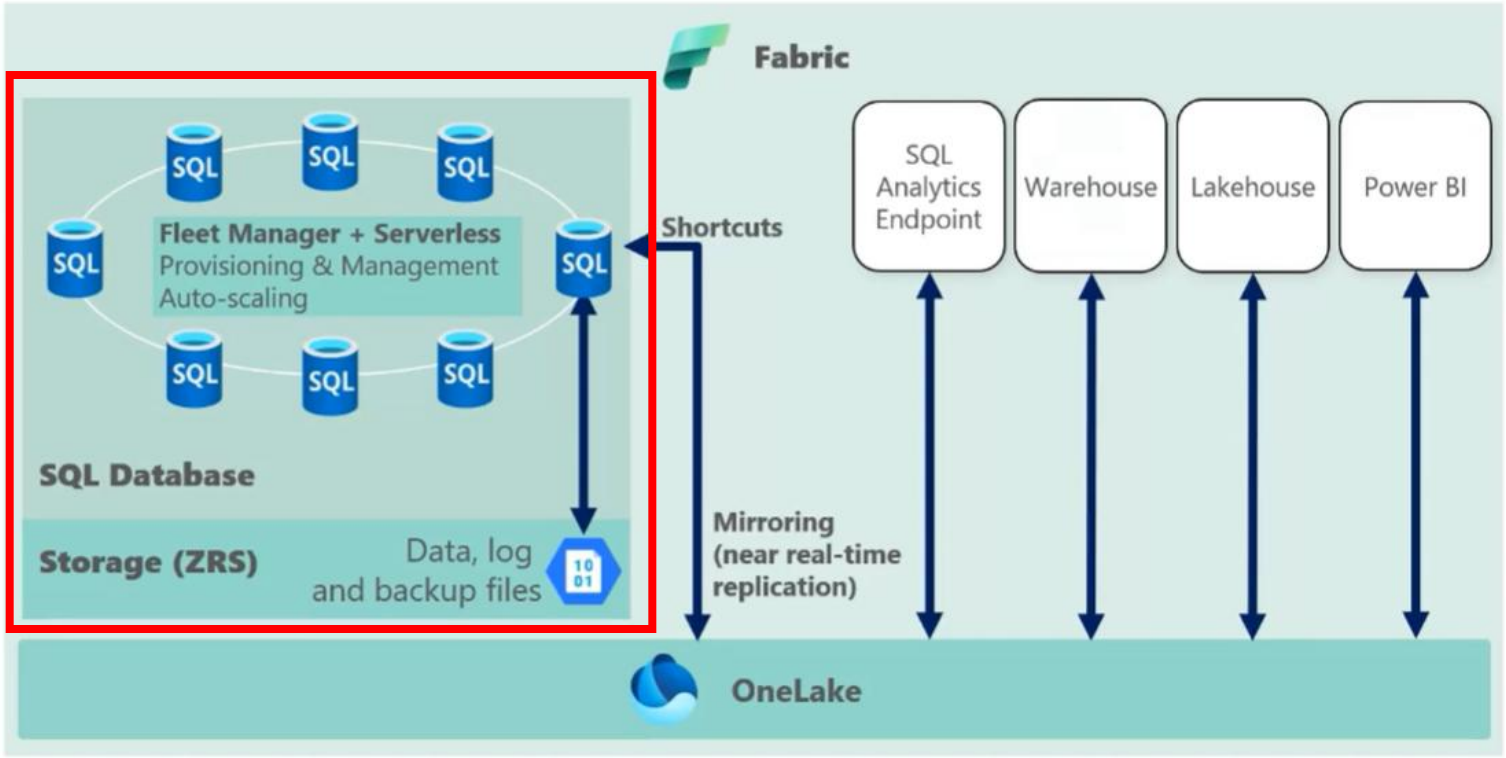
OPTIMIZE table to get optimal number and sizes of files

Partition table by columns commonly used for filtering

Z-ORDER table by non-partitioning columns commonly used for filtering.

Ingest or Mirror?

- For frequent analysis when the source systems shouldn't be directly queries
- IF ETL is still needed for star schema, you can use stored proc on SQL Endpoint
- Use case: Operational reporting & WH



Snowflake				
Mirroring for Snowflake	Mirroring for Azure SQL DB	Mirroring for Cosmos DB	Mirroring for Azure SQL MI	Mirrored Azure Databricks Catalog

Summarize

- Know your data source
- Know the target scale you can have for the load
- Balance between the workload to the source, the parallelism of the loading and the target (i.e. price)

- Use the great flexibility of COPY INTO; OPENROWSET and CTAS. You can build super cool loading process. Pipelines can orchestrate

THANK YOU!

Questions:

Magi.Naumova@inspir-it.no

Scan for feedback! I really appreciate it! ☺