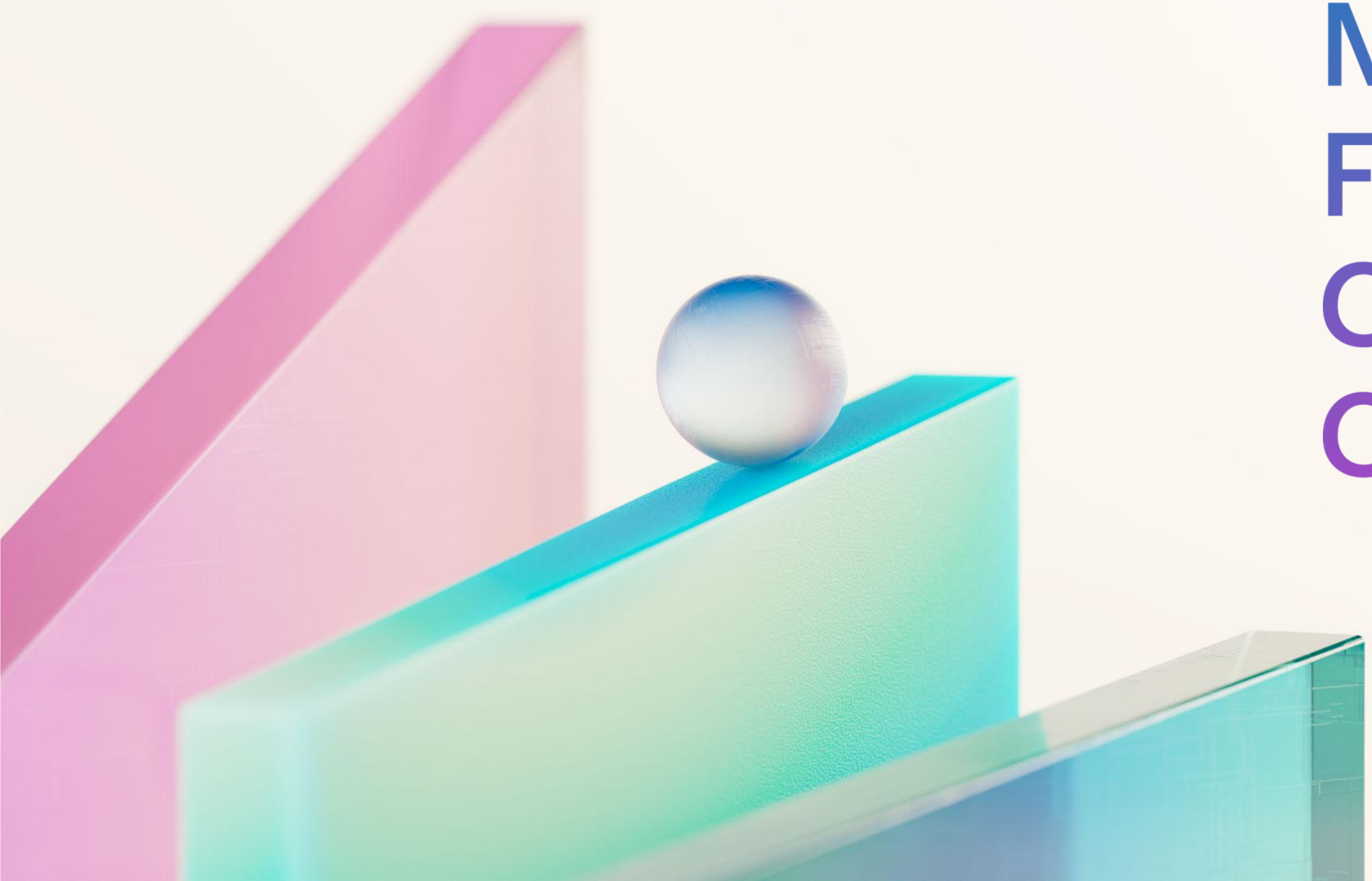
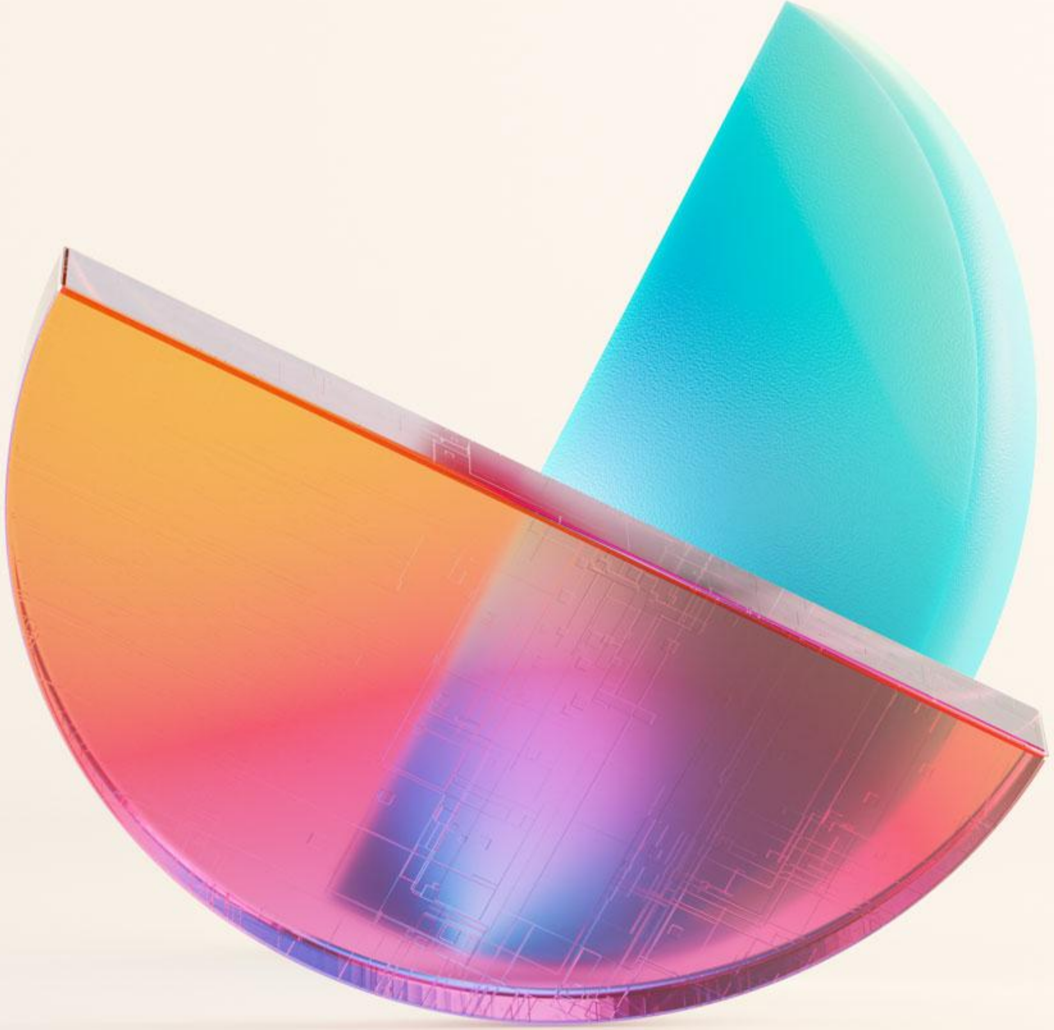
Microsoft



Microsoft Fabric Community Conference

Microsoft

Microsoft Fabric
Community Conference



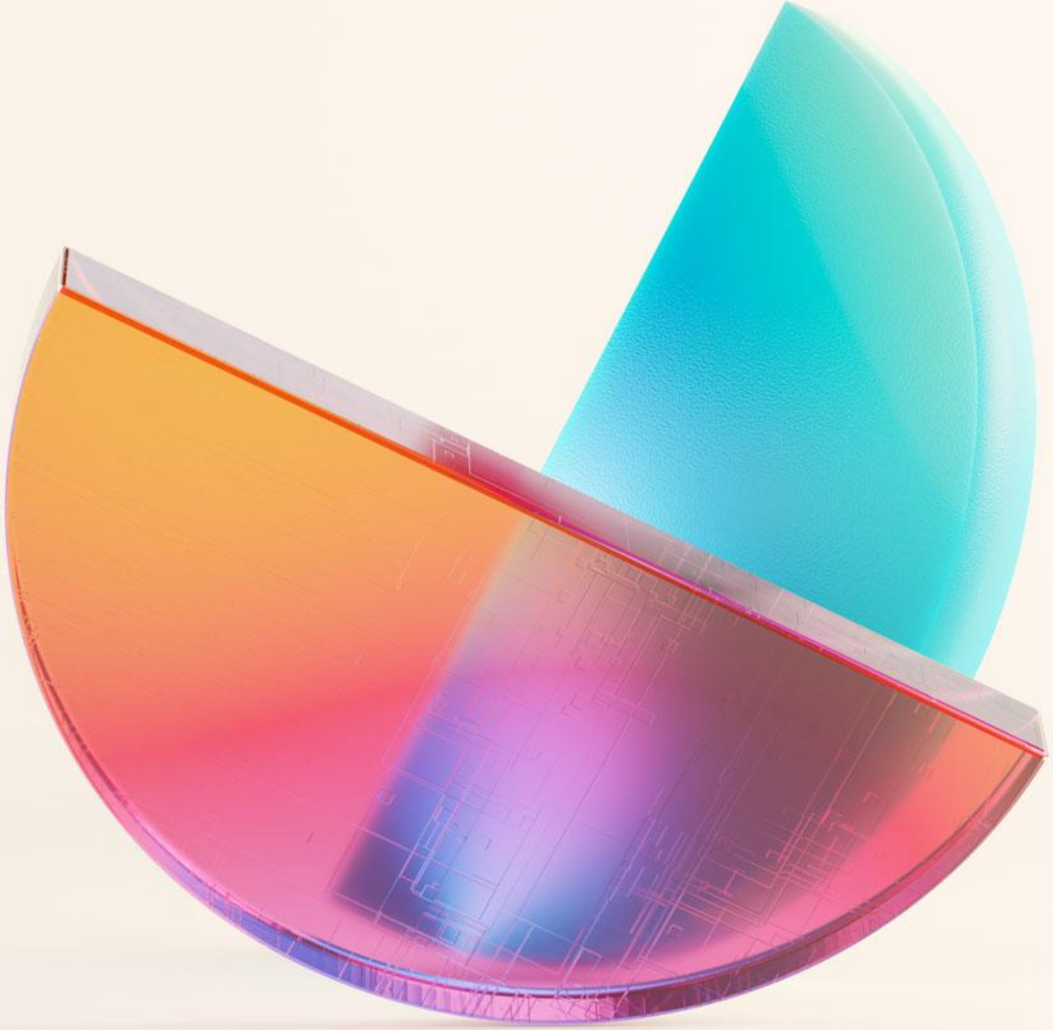
End to End Data Engineering

Modern Data Warehousing on Microsoft Fabric

Wifi Network: MGMResorts-WiFi
No password

Microsoft

Microsoft Fabric
Community Conference



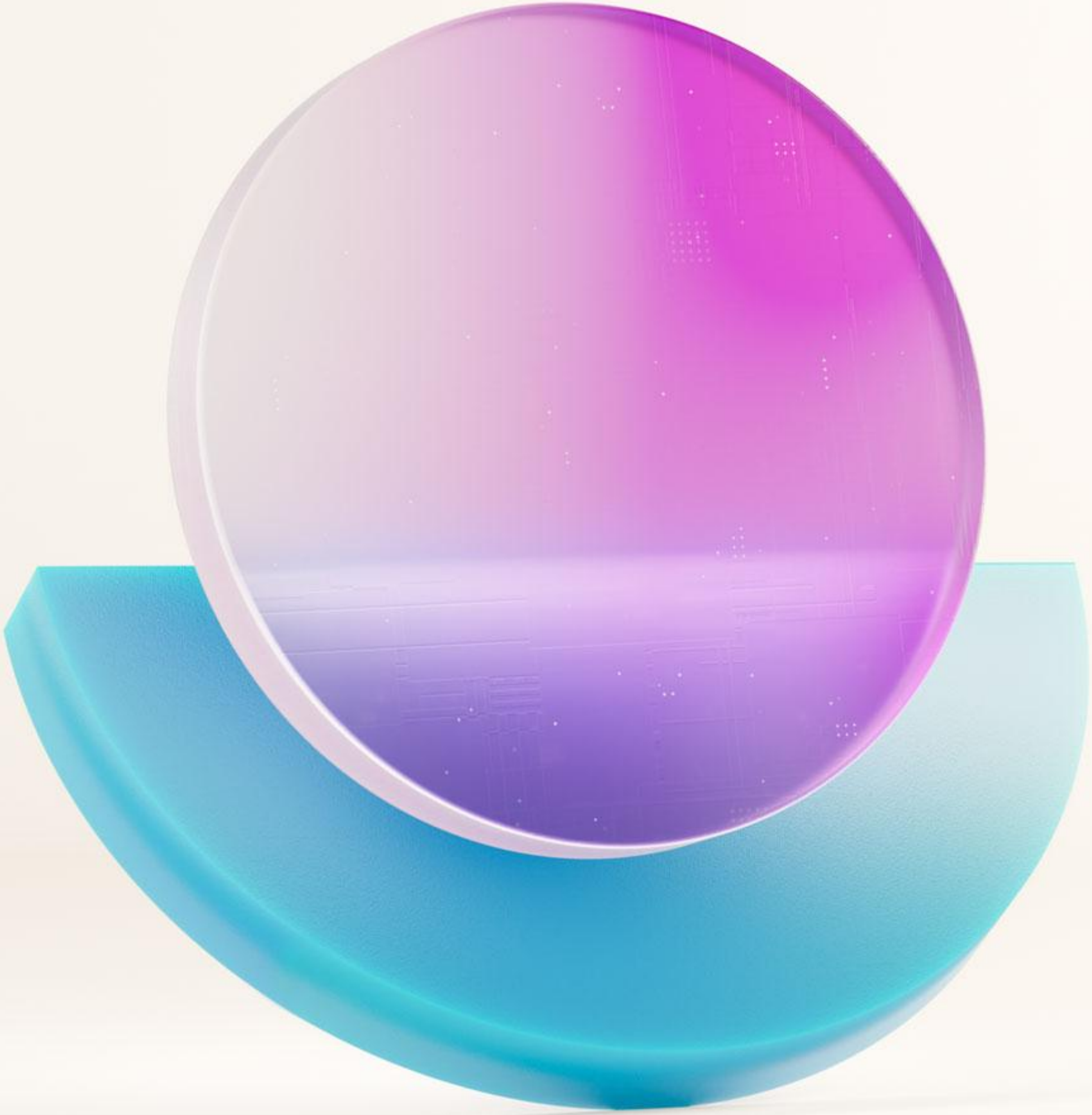
Join Us : Fabric User Panel

JOIN US!

Influence our product roadmap and ensure Fabric meets your real-life needs

Join us at aka.ms/FabricUserPanel

Microsoft Fabric
Community Conference



Fast-track your career in data and AI

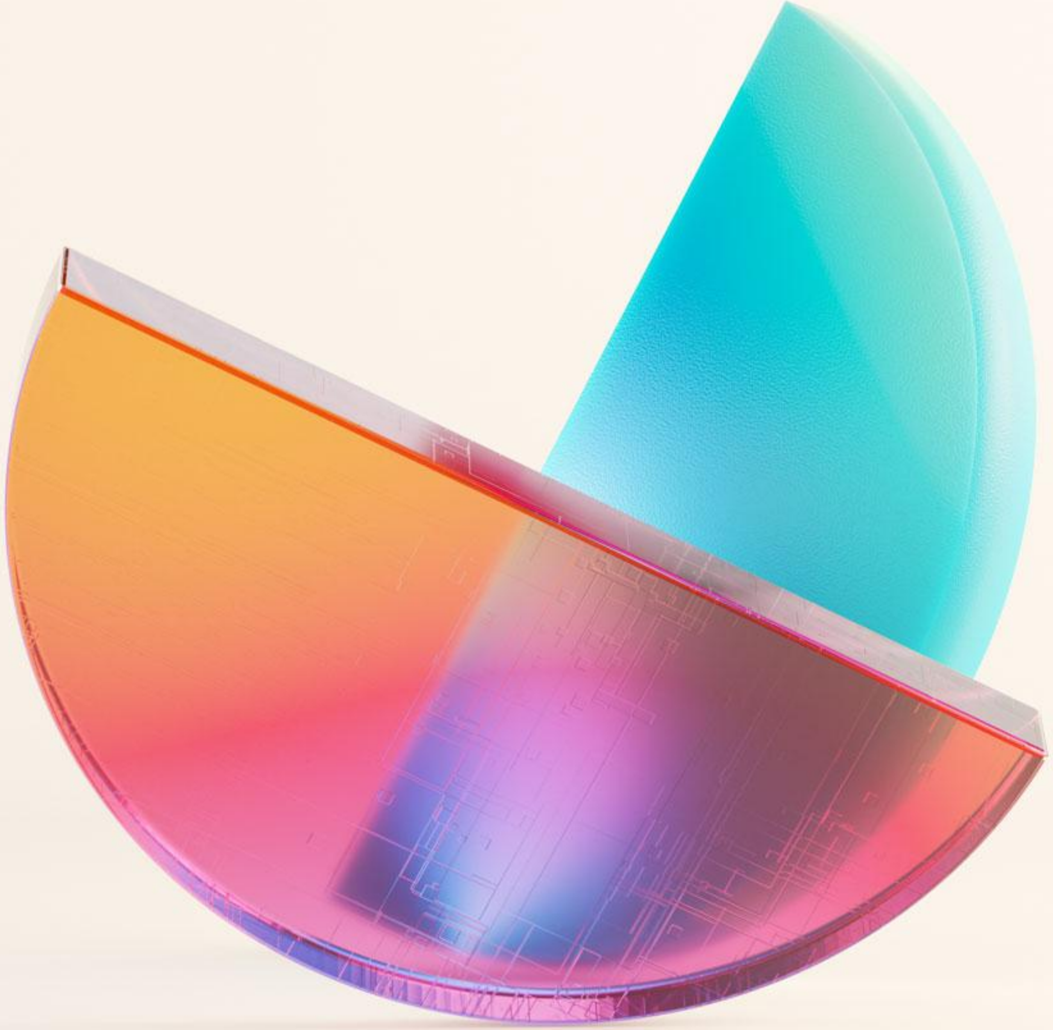
Become a Microsoft Certified Fabric Data Engineer Associate – take Exam DP-700 for free!

Visit the Fabric Community Lounge to learn more about this *limited-time offer*.

aka.ms/fabcon/dp700

Microsoft

Microsoft Fabric
Community Conference



Learn more about Microsoft Fabric

Power your AI transformation with a complete data platform

Microsoft

Get Involved in the Fabric Community

aka.ms/FabricCommunity
Connect with community members, ask questions, and learn more about Fabric

aka.ms/FabricUserGroups
Find a user group that matches your interests in your area or online

aka.ms/SuperUsers
Spread your Fabric knowledge, insights, and best practices with others

aka.ms/MVP
Technology experts that share their knowledge and passion with the community

Restrooms

Breaks and lunch

Questions

Workshop content

What is your learning style?

Agenda

01

Introduction to Fabric

02

Introduction to Data Warehouse

03

Creating Tables

04

Data Ingestion

05

Data Transformation and T-SQL

06

Monitoring Warehouse Activity

07

Statistics and V-Order

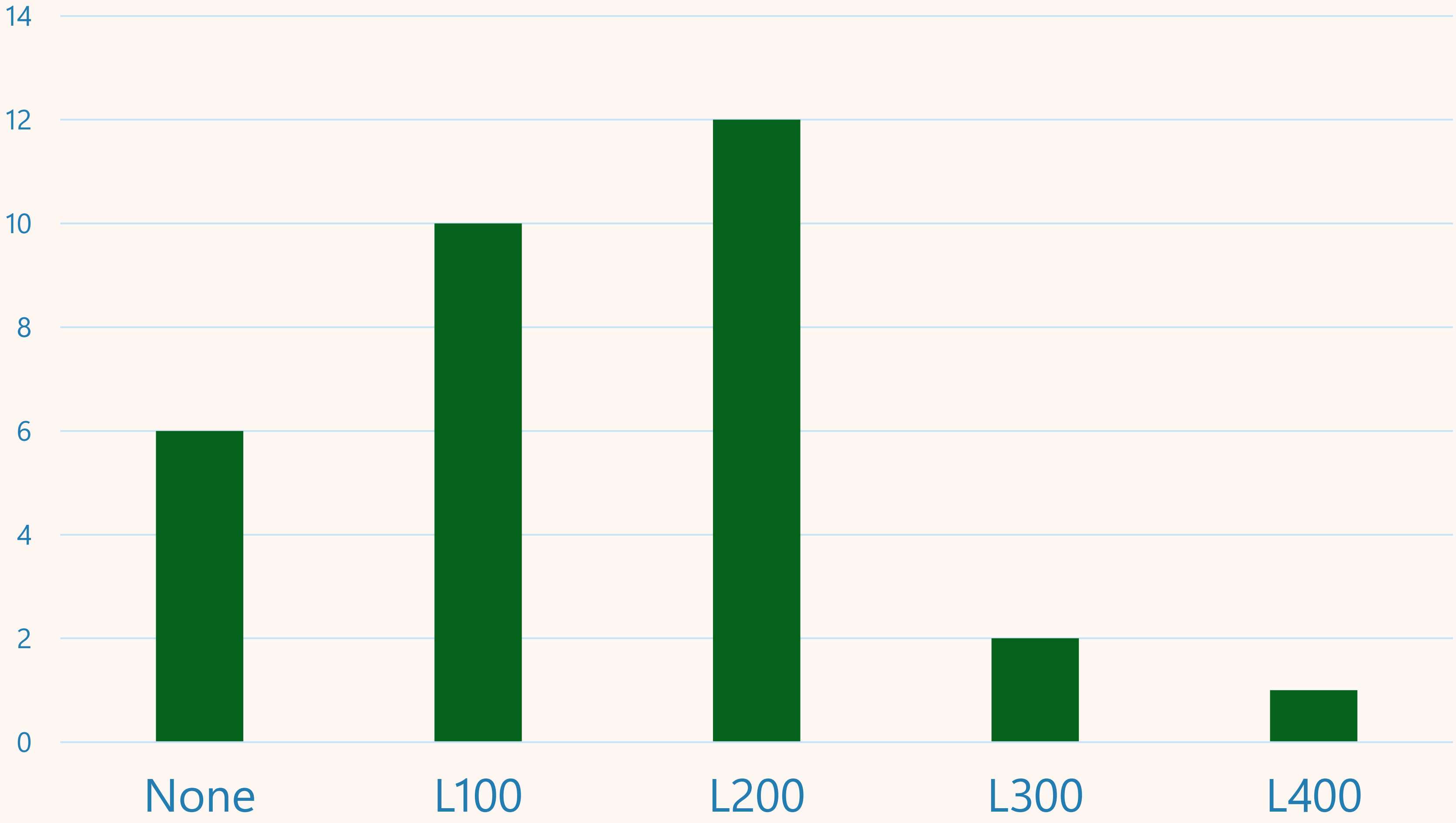
08

Workload Management

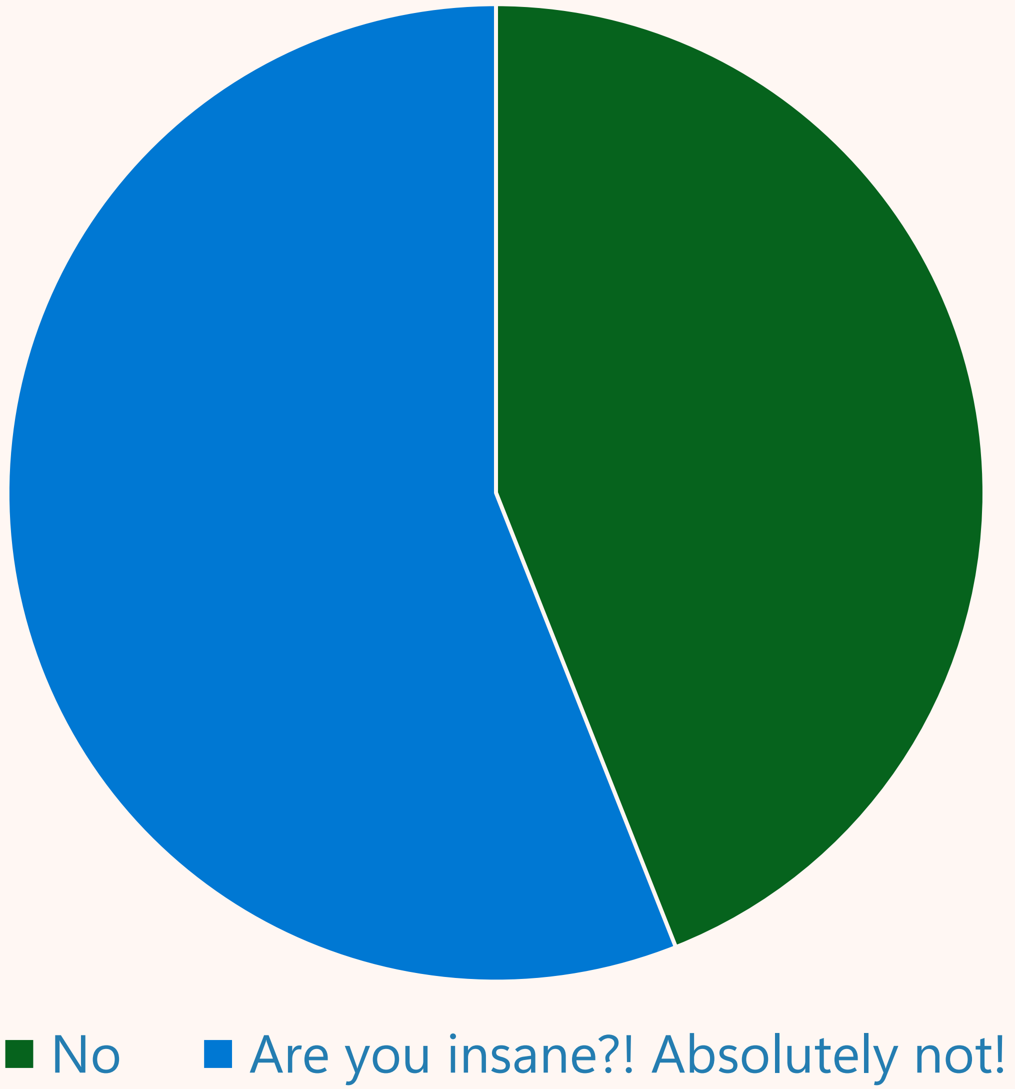
09

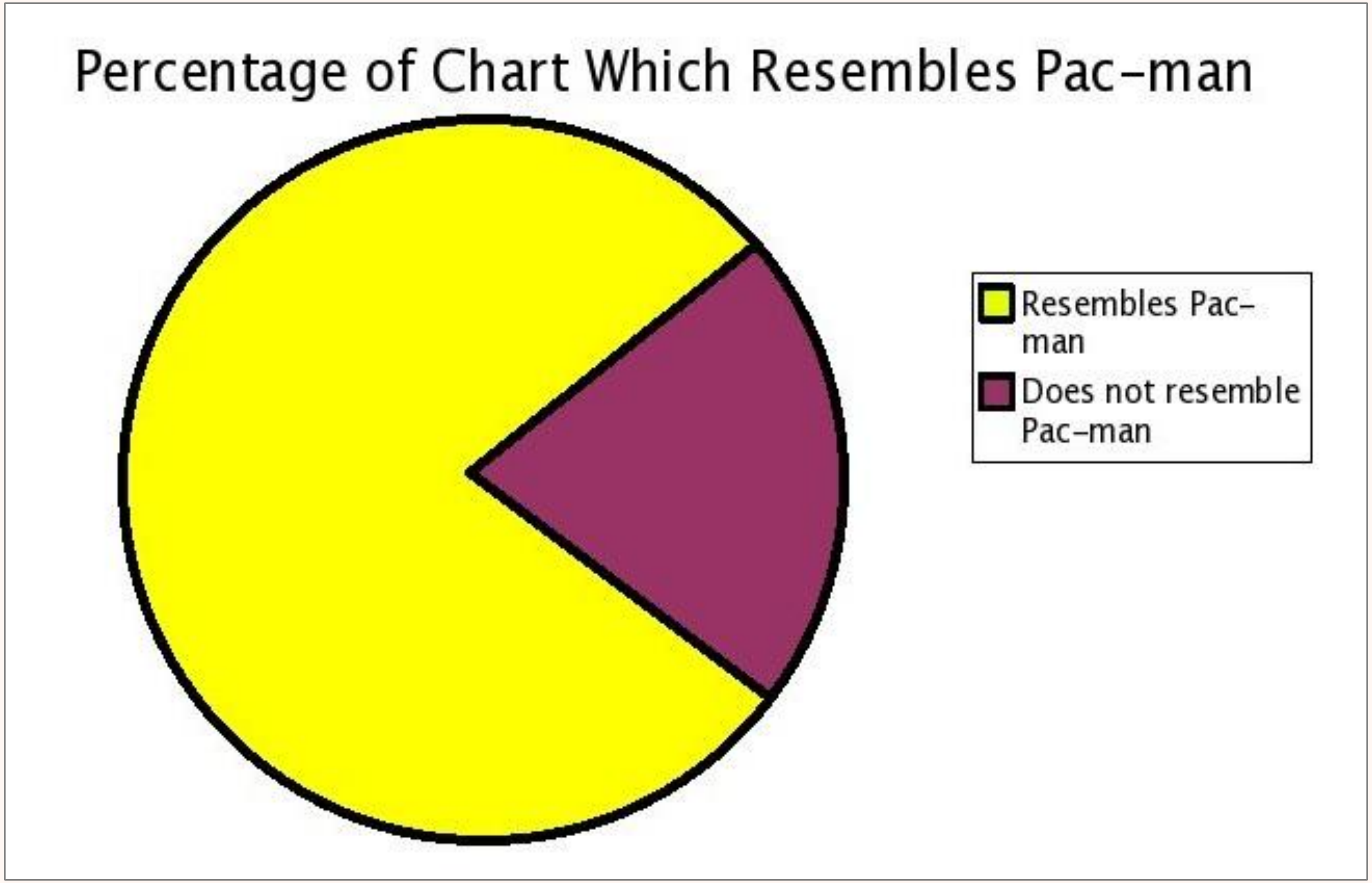
Security

Poll: What level of experience do you have with Fabric Data Warehouse?



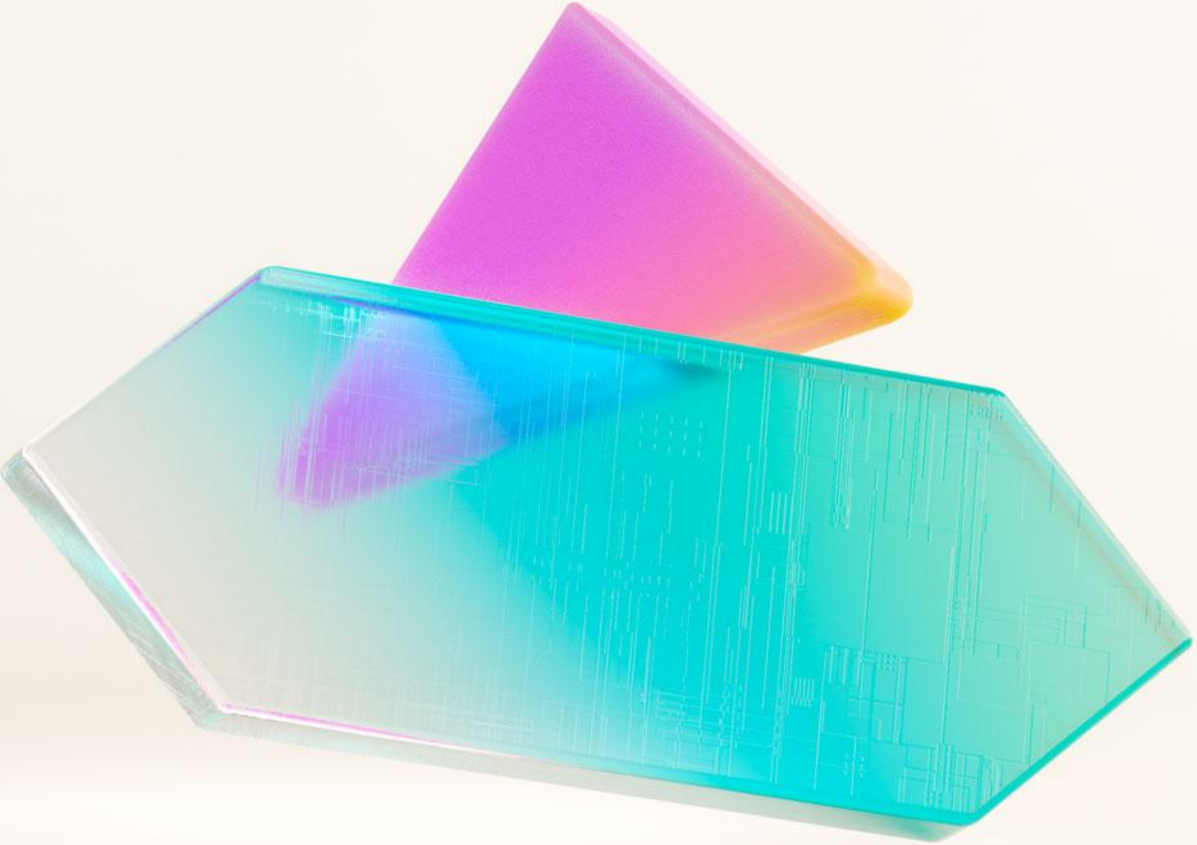
Poll: Should pineapple be on pizza?





Percentage of Chart Which Resembles Pac-man

Community Conference



Introduction to Fabric

Module 1

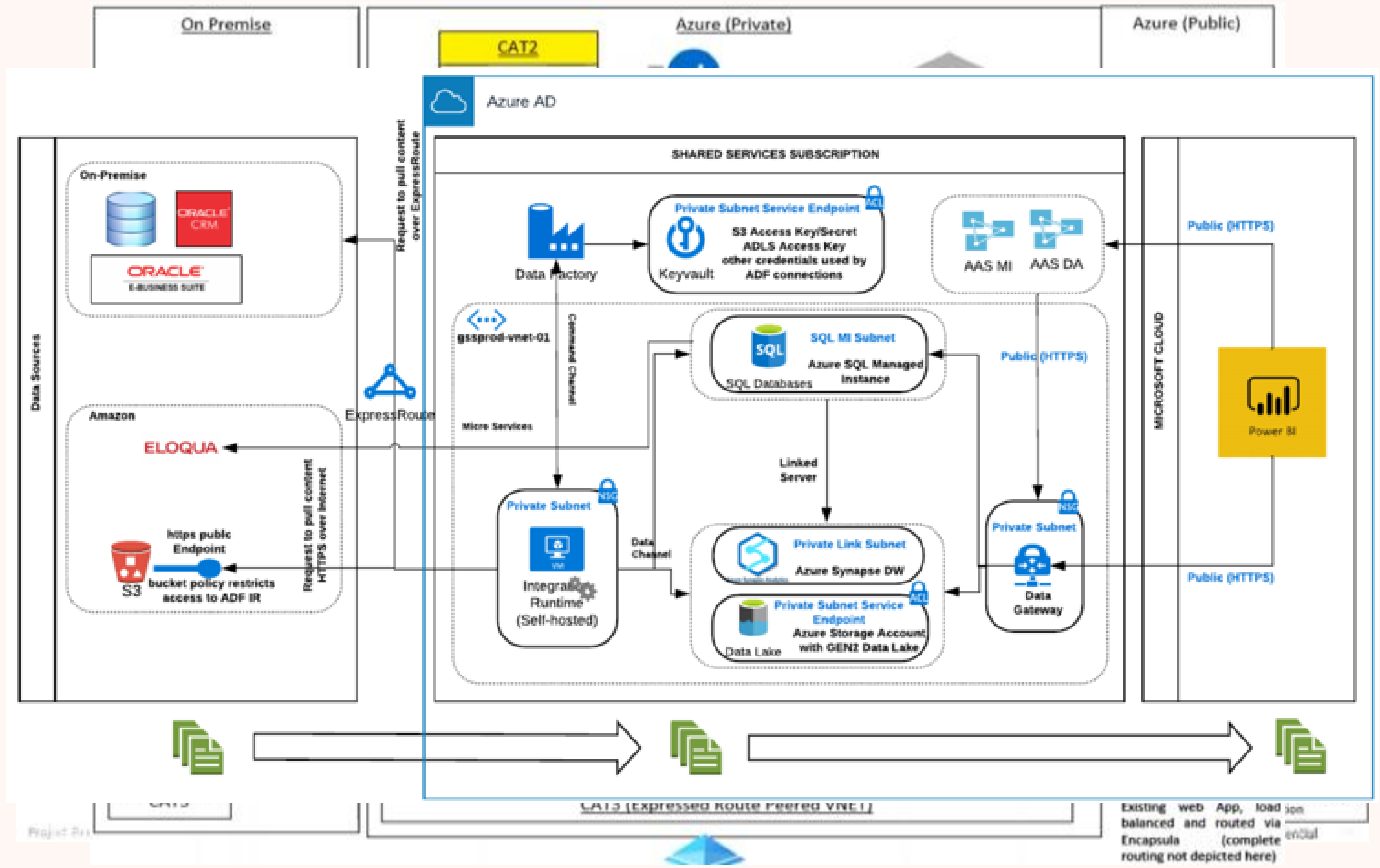
Analytics is complex and fragmented

Every project has many subsystems

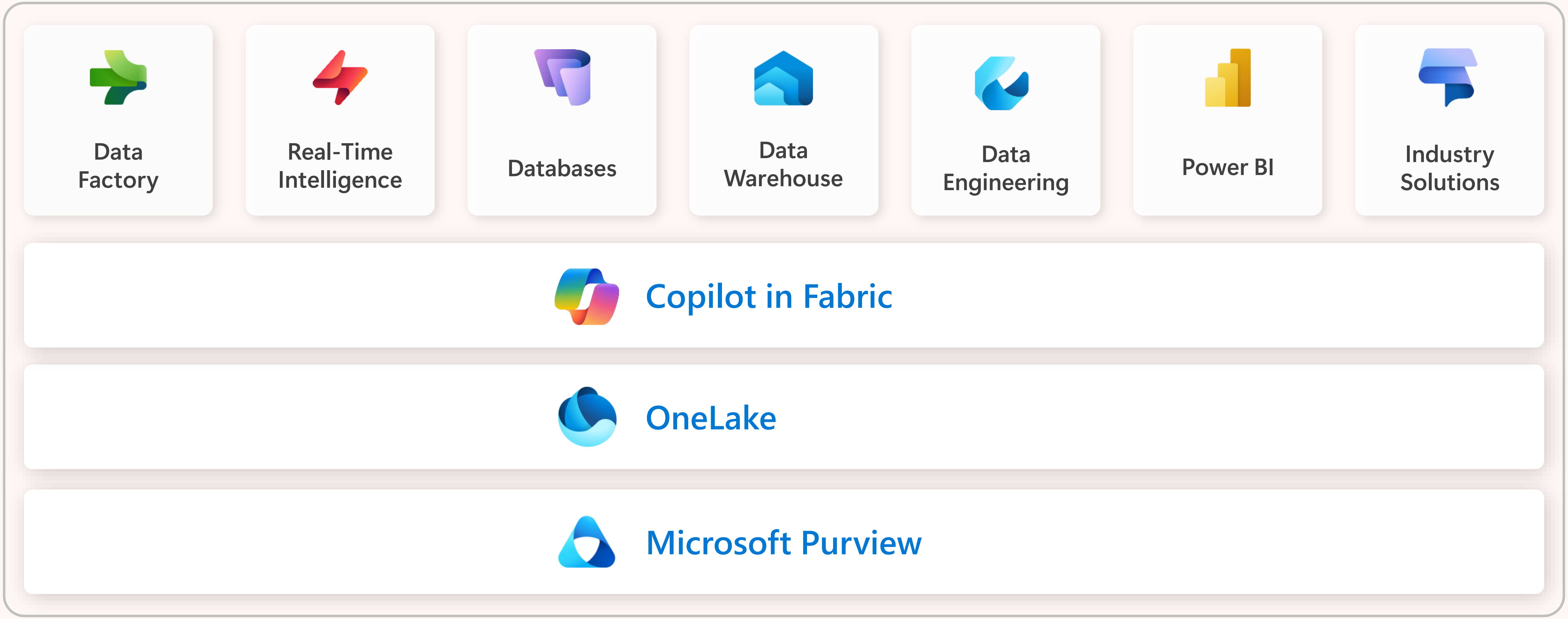
Every subsystem need a different class of product

Products often comes from multiple vendors

Integration is complex, fragile and expensive


On Premise
CAT2
Azure (Private)
Azure (Public)
Azure AD
Data Sources
On-Premise
ORACLE CRM
ORACLE E-BUSINESS SUITE
Amazon
ELOQUA
https public Endpoint
S3
bucket policy restricts access to ADF IR
Request to pull content HTTPS over Internet
Request to pull content over ExpressRoute
ExpressRoute
SHARED SERVICES SUBSCRIPTION
Data Factory
Private Subnet Service Endpoint
Keyvault
S3 Access Key/Secret
ADLS Access Key
other credentials used by ADF connections
AAS MI
AAS DA
gssprod-vnet-01
Command Channel
Micro Services
SQL MI Subnet
SQL Databases
Azure SQL Managed Instance
Linked Server
Private Subnet
Integration Runtime (Self-hosted)
Data Channel
Private Link Subnet
Azure Synapse DW
Private Subnet Service Endpoint
Data Lake
Azure Storage Account with GEN2 Data Lake
Private Subnet
Data Gateway
Public (HTTPS)
Public (HTTPS)
Public (HTTPS)
MICROSOFT CLOUD
Power BI
CAT3
CAT3 (Expressed Route Peered VNET)
Existing web App, load balanced and routed via Encapsula (complete routing not depicted here)

Microsoft Fabric



Data Factory

Real-Time Intelligence

Databases

Data Warehouse

Data Engineering

Power BI

Industry Solutions

Copilot in Fabric

OneLake

Microsoft Purview

The data platform for the era of AI

Complete Analytics Platform

Unified product, experience, and architecture

Delivered as SaaS

Lake Centric and Open

Common SaaS data lake shared by all compute engines

Deep commitment for open formats and APIs

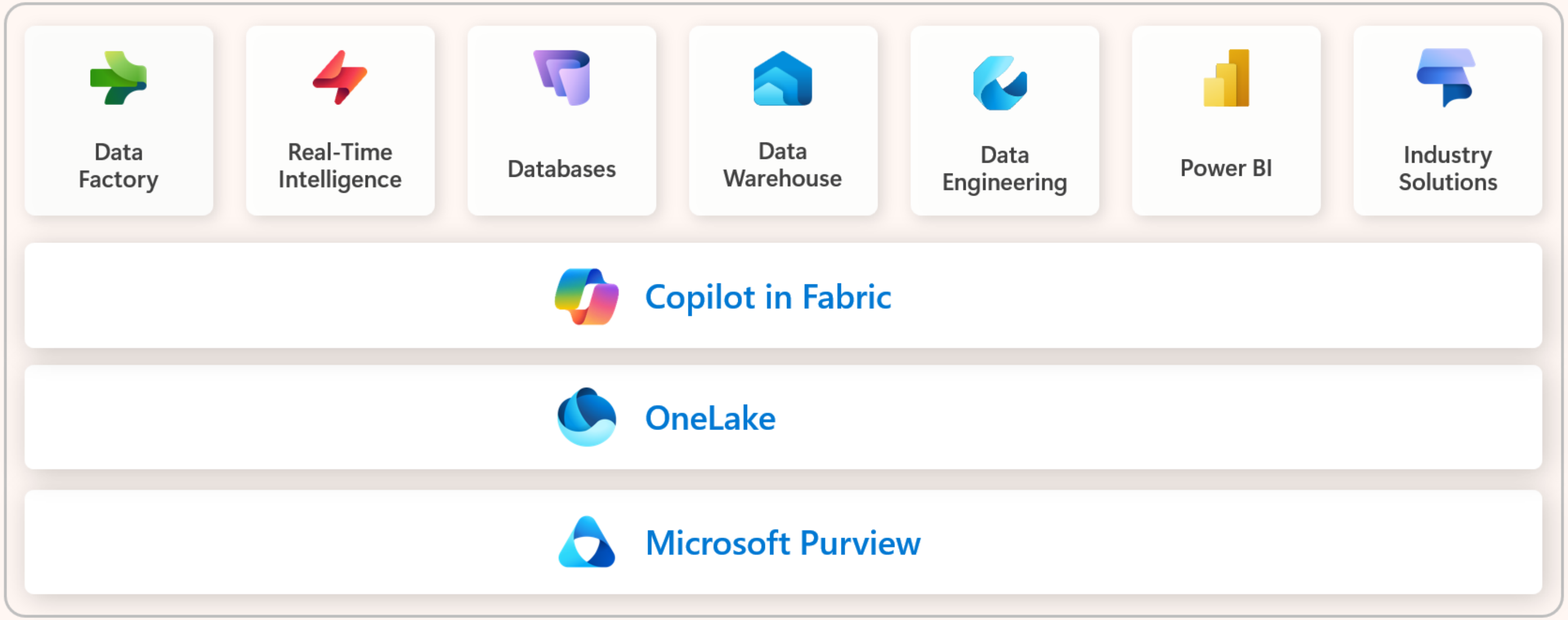
Empower Every Business User

Deliver data directly to the users in their favorite Office applications

AI Powered

Designed from the ground up for AI

Microsoft Fabric



Data Factory

Real-Time Intelligence

Databases

Data Warehouse

Data Engineering

Power BI

Industry Solutions

Copilot in Fabric

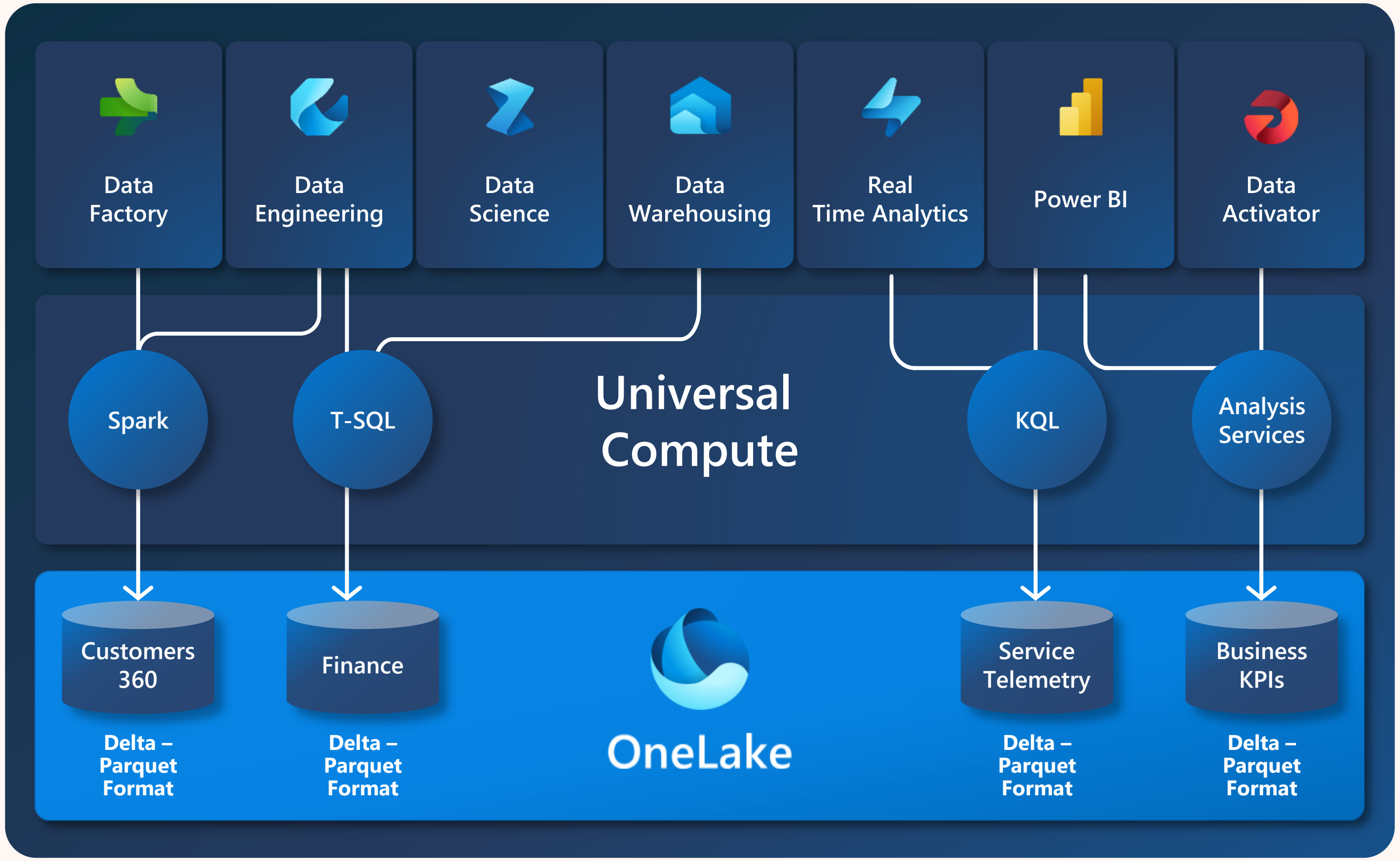
OneLake

Microsoft Purview

Single…

- Onboarding and trials
- Sign-on
- Navigation model
- UX model
- Workspace organization
- Collaboration experience
- Data Lake
- Storage format
- Data copy for all engines
- Security model
- CI/CD
- Monitoring hub
- Data Hub
- Governance & compliance

Microsoft Fabric



All the compute engines store their data automatically in OneLake

The data is stored in a single common format

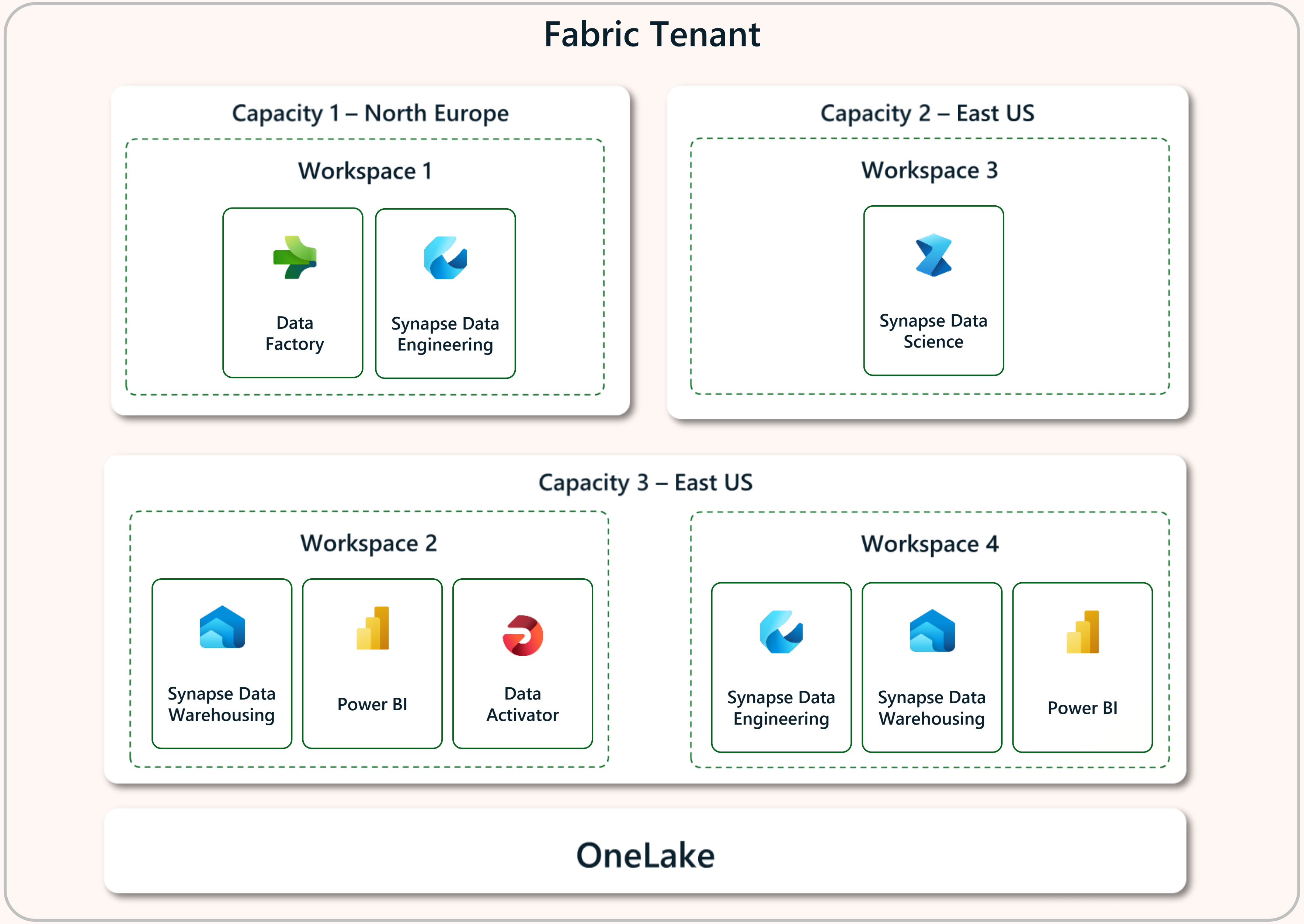
Delta – Parquet, an open standards format, is the storage format for all tabular data in Analytics vNext

Once data is stored in the lake, it is directly accessible by all the engines without needing any import/export

All the compute engines have been fully optimized to work with Delta Parquet as their native format

Shared universal security model is enforced across all the engines

Fabric topology and organization



Fabric Tenant

Capacity 1 – North Europe

Workspace 1

Data Factory

Synapse Data Engineering

Capacity 2 – East US

Workspace 3

Synapse Data Science

Capacity 3 – East US

Workspace 2

Synapse Data Warehousing

Power BI

Data Activator

Workspace 4

Synapse Data Engineering

Synapse Data Warehousing

Power BI

OneLake

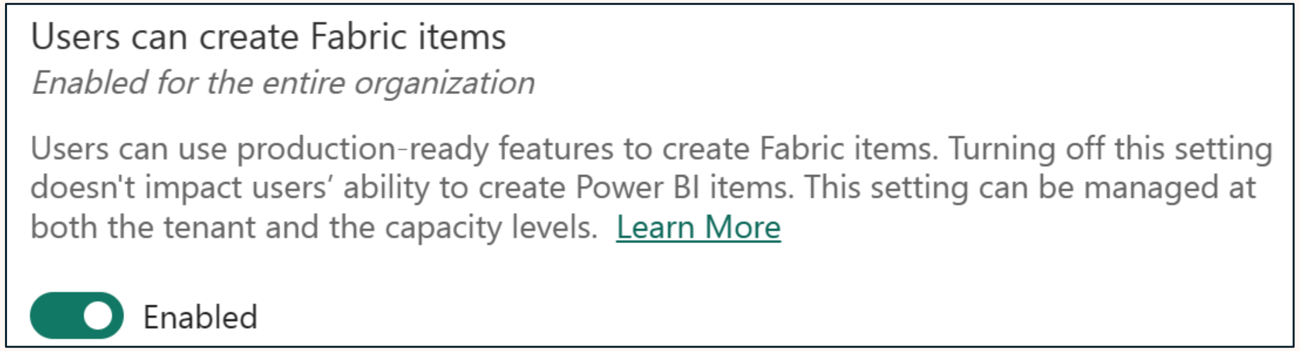
Comparing P and F SKUs

SKU	Capacity Units (CU)	Power BI SKU	Power BI v-cores
F2	2	-	0.25
F4	4	-	0.5
F8	8	EM/A1	1
F16	16	EM2/A2	2
F32	32	EM3/A3	4
F64	**64**	**P1/A4**	**8**
F128	128	P2/A5	16
F256	256	P3/A6	32
F512	512	P4/A7	64
F1024	1024	P5/A8	128
F2048	2048	-	256

Getting started with Fabric

Getting started in your environment

1. Enable Fabric at the tenant level.
   - Override the tenant level setting at the capacity
   - Enable for only specific security groups
2. Get capacity
   - Start a 60-day free trial
   - Use existing P SKU
   - Provision an F SKU
3. Create a workspace and assign a capacity.
4. Build something cool!

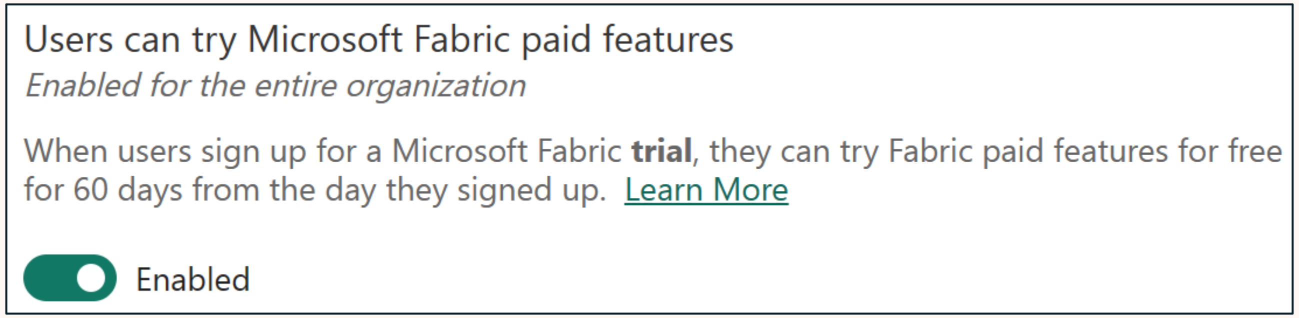


Users can create Fabric items

Enabled for the entire organization

Users can use production-ready features to create Fabric items. Turning off this setting doesn't impact users' ability to create Power BI items. This setting can be managed at both the tenant and the capacity levels. Learn More

Enabled



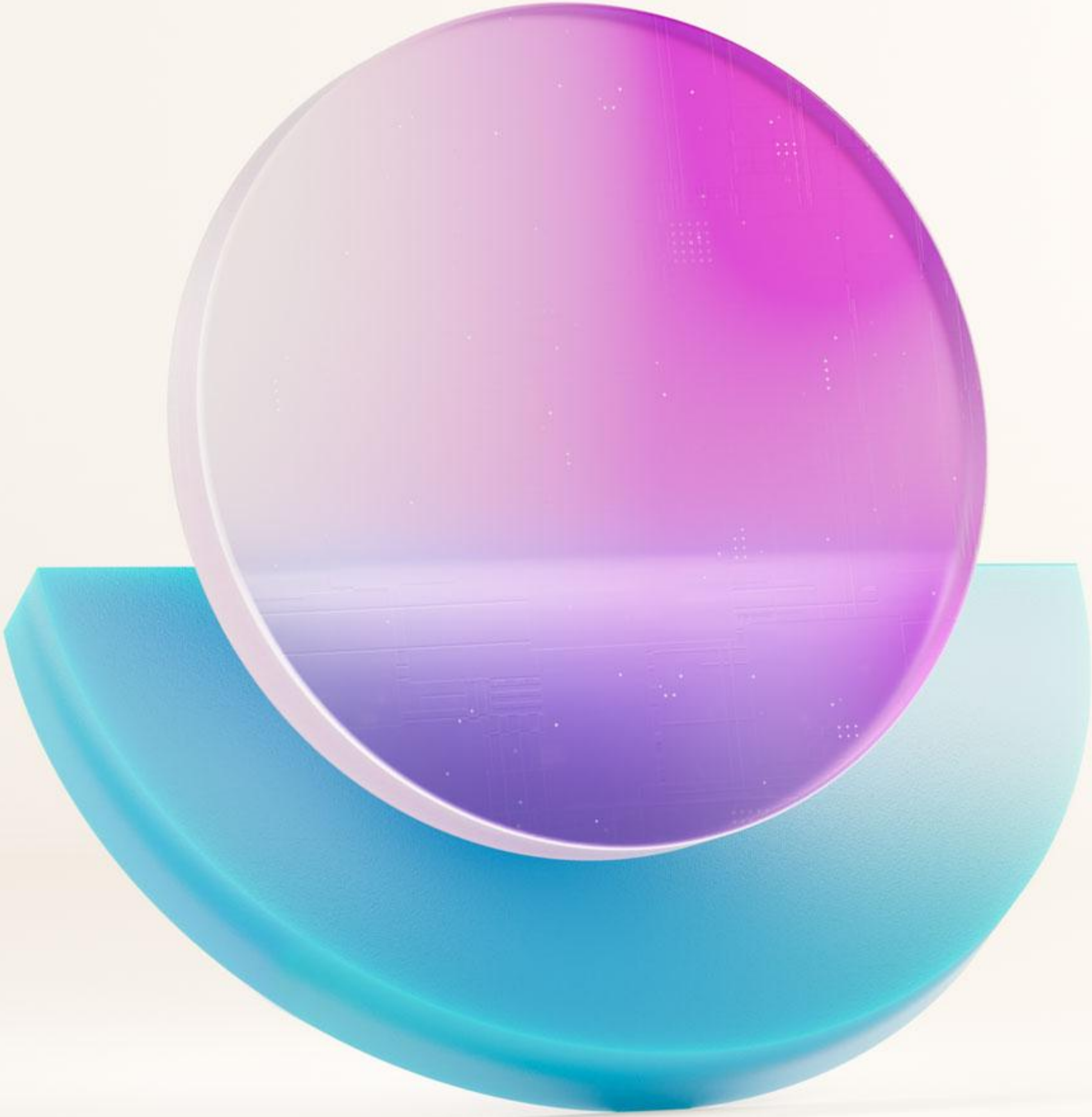
Users can try Microsoft Fabric paid features

Enabled for the entire organization

When users sign up for a Microsoft Fabric **trial**, they can try Fabric paid features for free for 60 days from the day they signed up. Learn More

Enabled

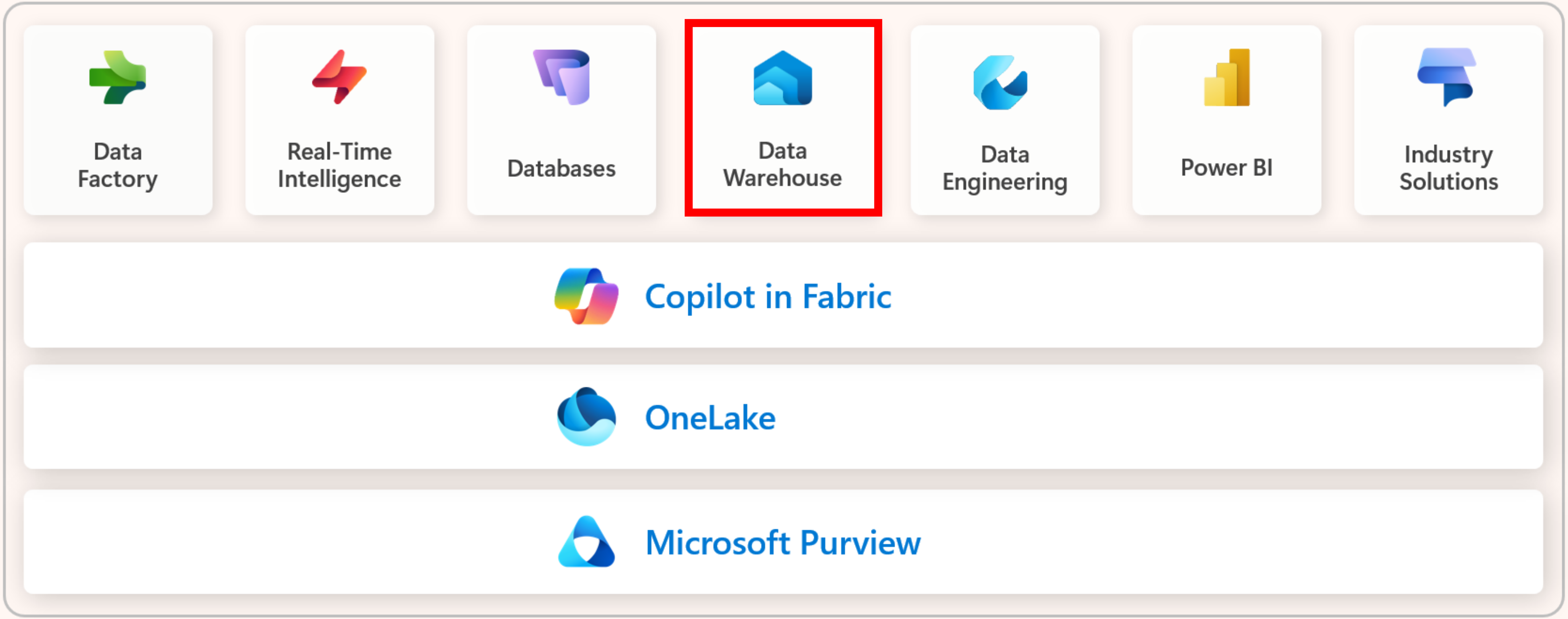
Microsoft Fabric
Community Conference



Introduction to Data Warehouse

Module 2

Microsoft Fabric



Data Factory

Real-Time Intelligence

Databases

Data Warehouse

Data Engineering

Power BI

Industry Solutions

Copilot in Fabric

OneLake

Microsoft Purview

What is a data warehouse?

A data warehouse is a central repository of information that can be analyzed to make more informed decisions. Data flows into a data warehouse from transactional systems, relational databases, and other sources, typically on a regular cadence. Business analysts, data engineers, data scientists, and decision makers access the data through business intelligence tools, SQL clients, and other analytics applications.

Data Warehouse key capabilities

Truly serverless data warehouse

Data warehouse redesigned from the ground up

Optimized for star schema designs

Familiar T-SQL code

Easy migration path

Use it with a lakehouse or standalone

Auto stats

Auto file maintenance

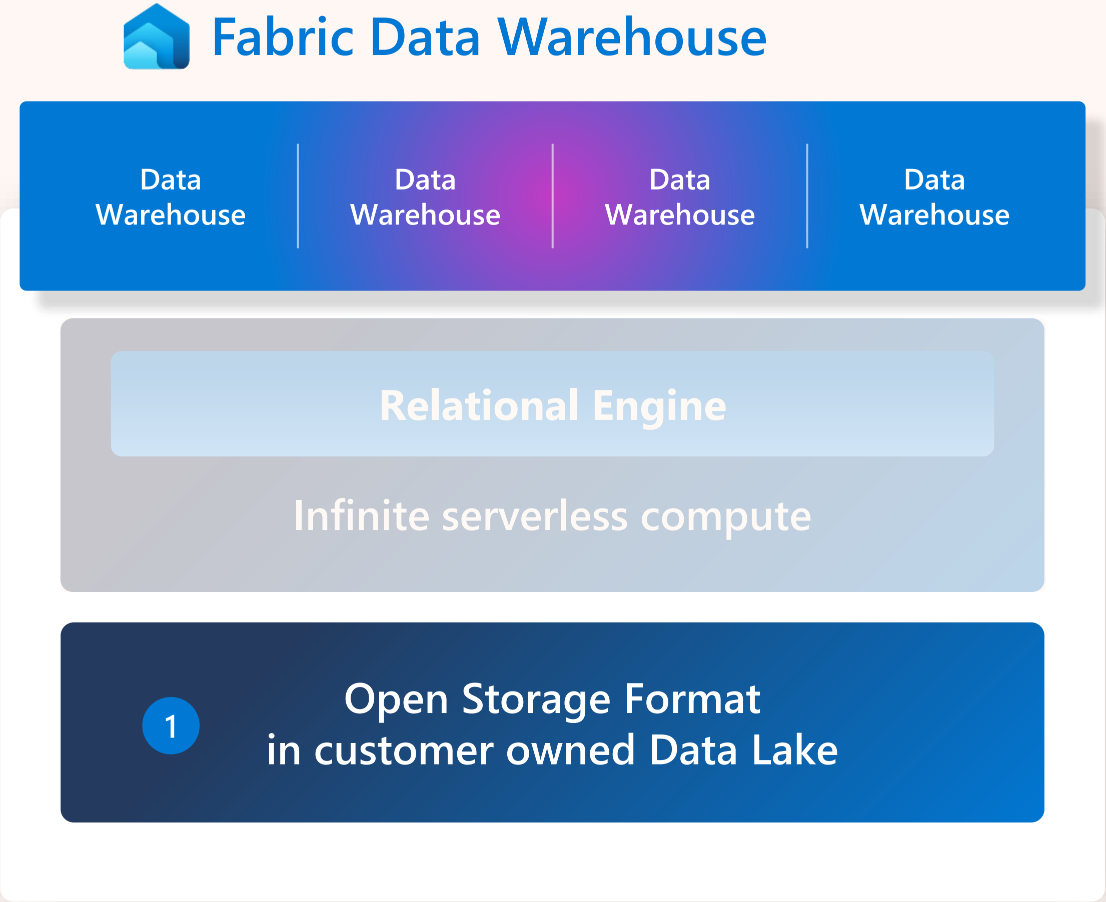
No indexes required

SQL Best Practices still apply

Rapid innovation (release cycle, new features, closing gaps)

Fabric Data Warehouse

Infinitely scalable and open

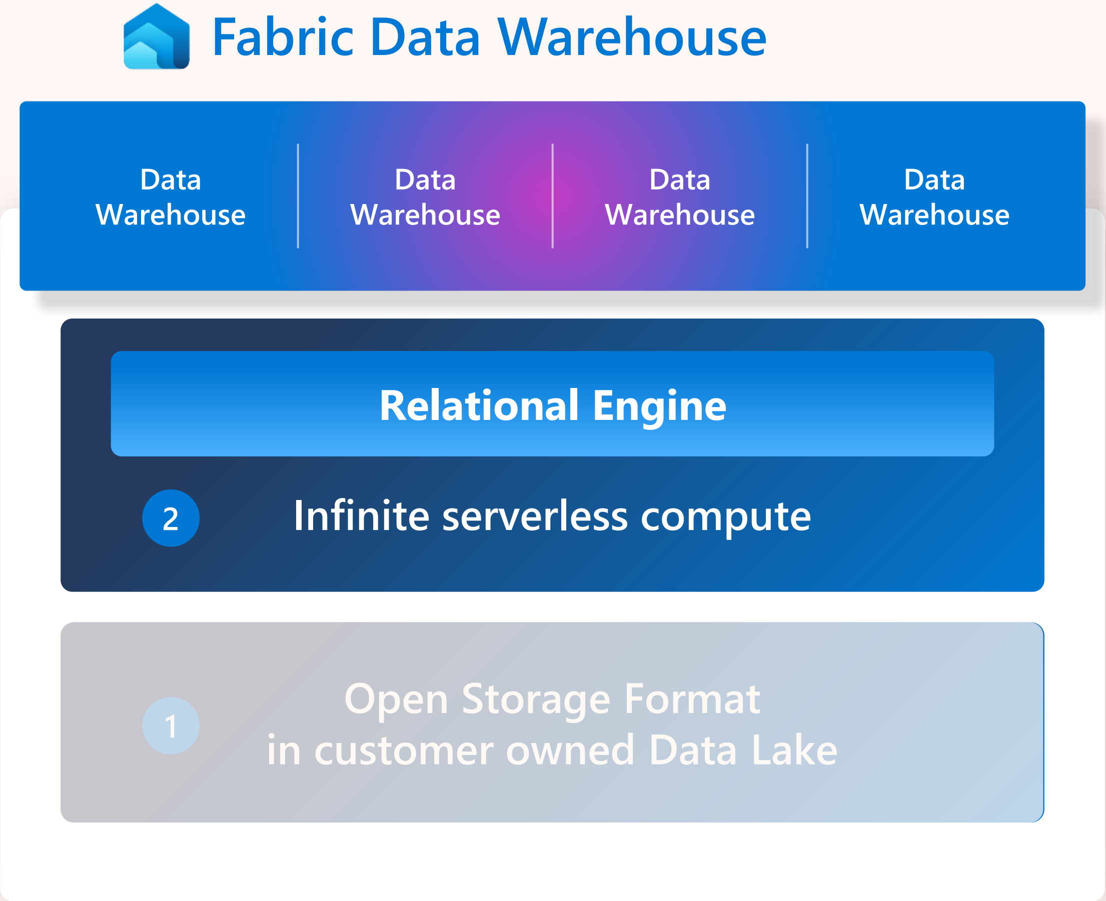


1 Open standard format in an open data lake replaces proprietary formats as the native storage

- 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 Data Warehouse

Infinitely scalable and open



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

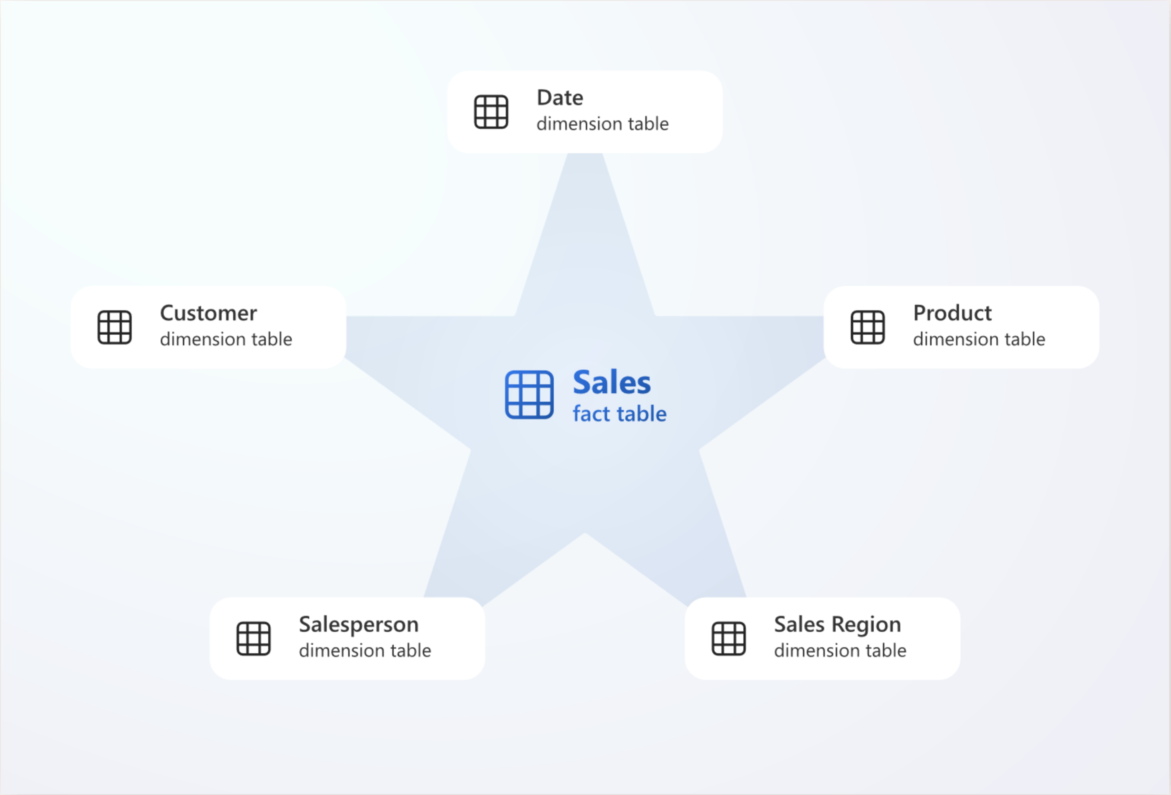
Data Modeling

👍 **Star schema** is the recommended design approach to take when creating a Fabric warehouse.

👎 An **OLTP** workload can run on Fabric warehouse but is it not recommended for optimal performance.

👍 The warehouse engine is optimized for **batch** writes and reads.

👎 Micro-batches can work well, but **trickle feeds** should be avoided where possible.



Date
dimension table

Customer
dimension table

Product
dimension table

Sales
fact table

Salesperson
dimension table

Sales Region
dimension table

Dimensional Modeling

Facts tell us…

What happened

When did it happen

Contain measures (counts, quantities, amounts, etc.)

Don’t typically change

Dimensions describe…

What happened in a “fact”

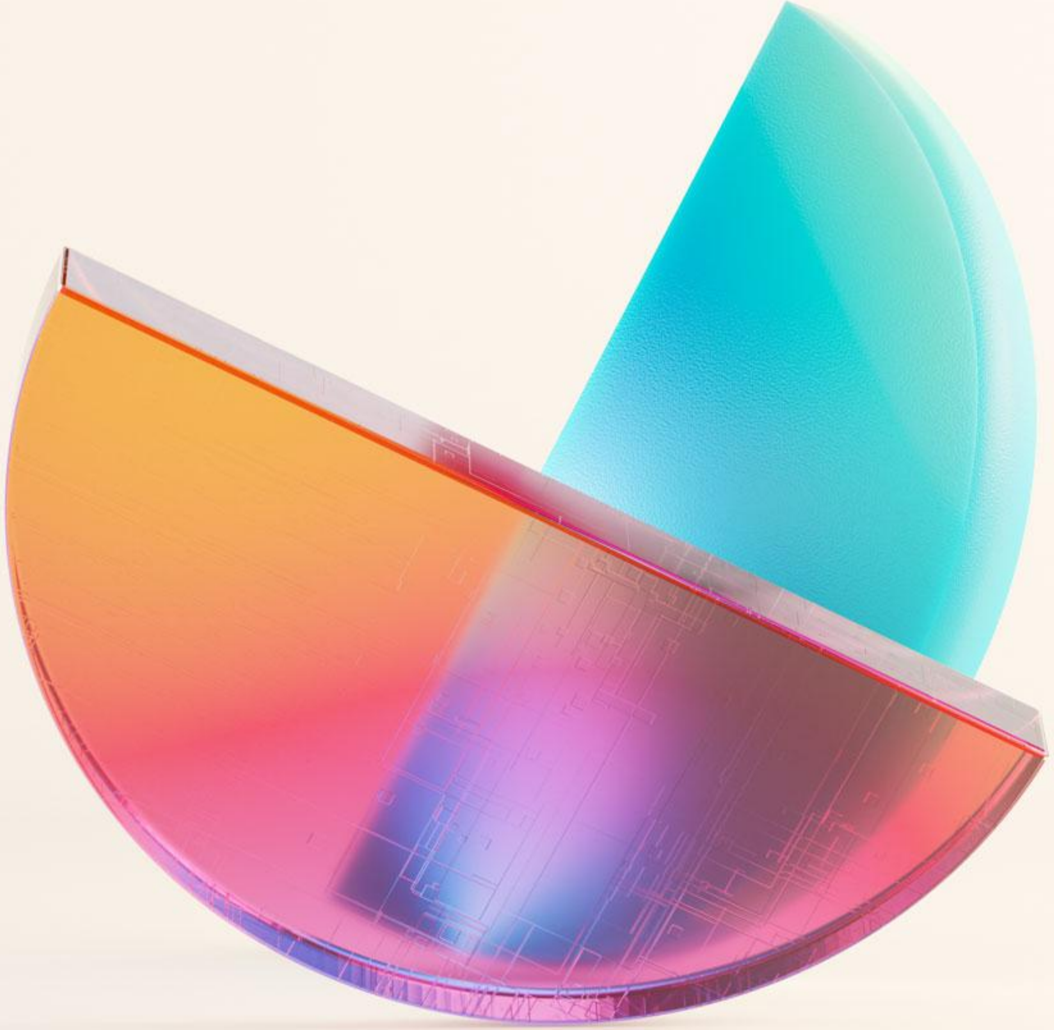
Ways to slide and dice an event

Attributes of the “member”

How attributes change over time

Microsoft

Microsoft Fabric
Community Conference



Creating Tables

Module 3

Creating Data Warehouse tables

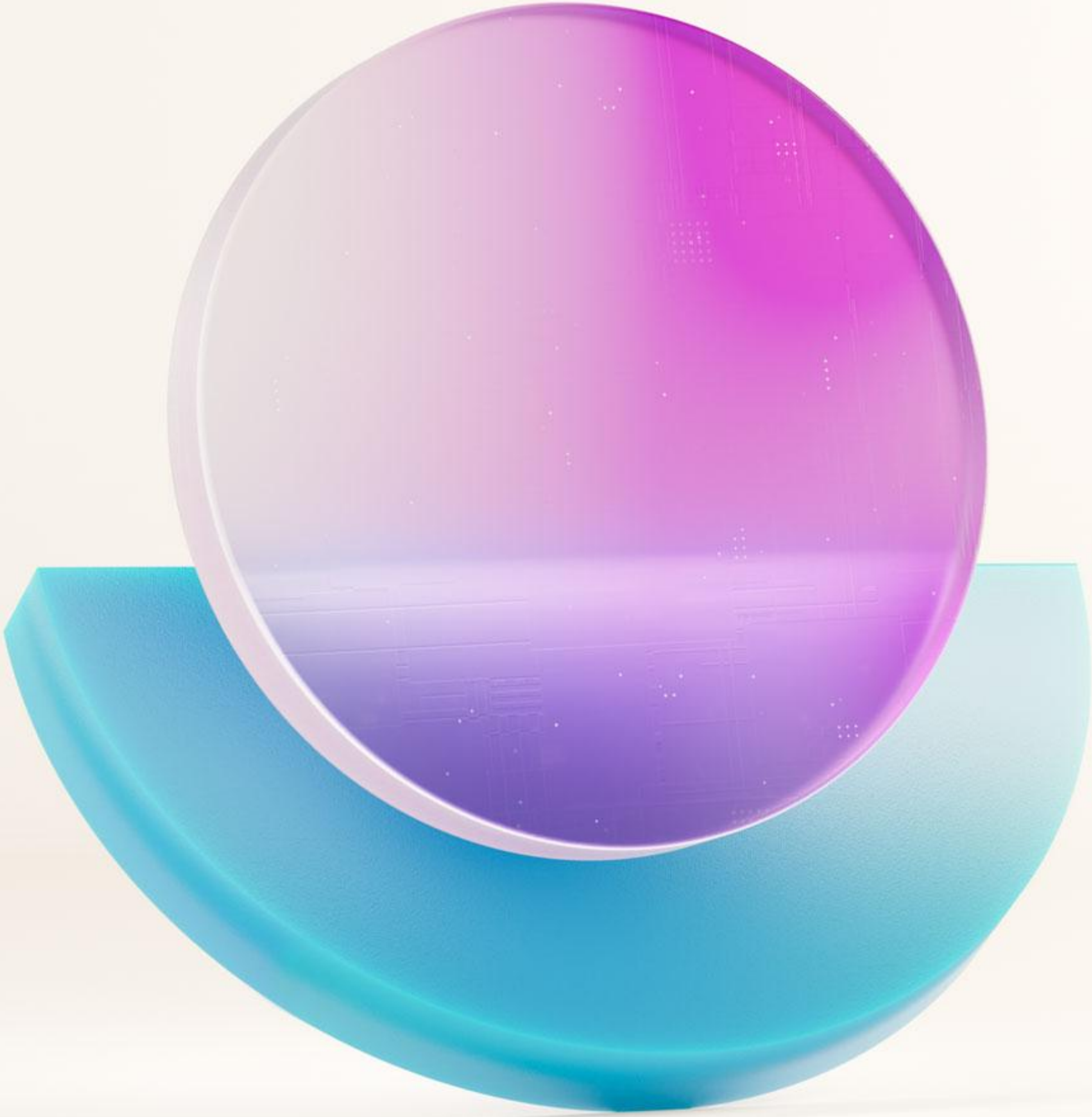
Basic syntax:

```
CREATE TABLE { database_name.schema_name.table_name | schema_name.table_name | table_name }
    (
      { column_name <data_type>  [ <column_options> ] } [ ,...n ]
    )
```

Example:

```
CREATE TABLE Bands (
    BandID INT NOT NULL,
    Name CHAR(45),
    FoundationDate DATE
)
```

Microsoft Fabric
Community Conference



Demo

Data type support

Exact numeric	Approximate numeric	Date and Time	Character strings	Binary strings	Other

bit
bigint
int
smallint
decimal(p,s)
numeric(p,s)
	float(n)
	real
		date
		datetime2(n)*
		time(n)*
			char(n)
			varchar(n)
			char(max)**
			varchar(max)**
				varbinary(n)	uniqueidentifier

* datetime2(6) and time(6) are the limits
** char/varchar(max) currently in preview

Optimizing data types for better performance

Best practices to maximize performance:

Use the appropriate data type

- If a column contains only integer values, use INT data types instead of DECIMAL/NUMERIC

Precision matters

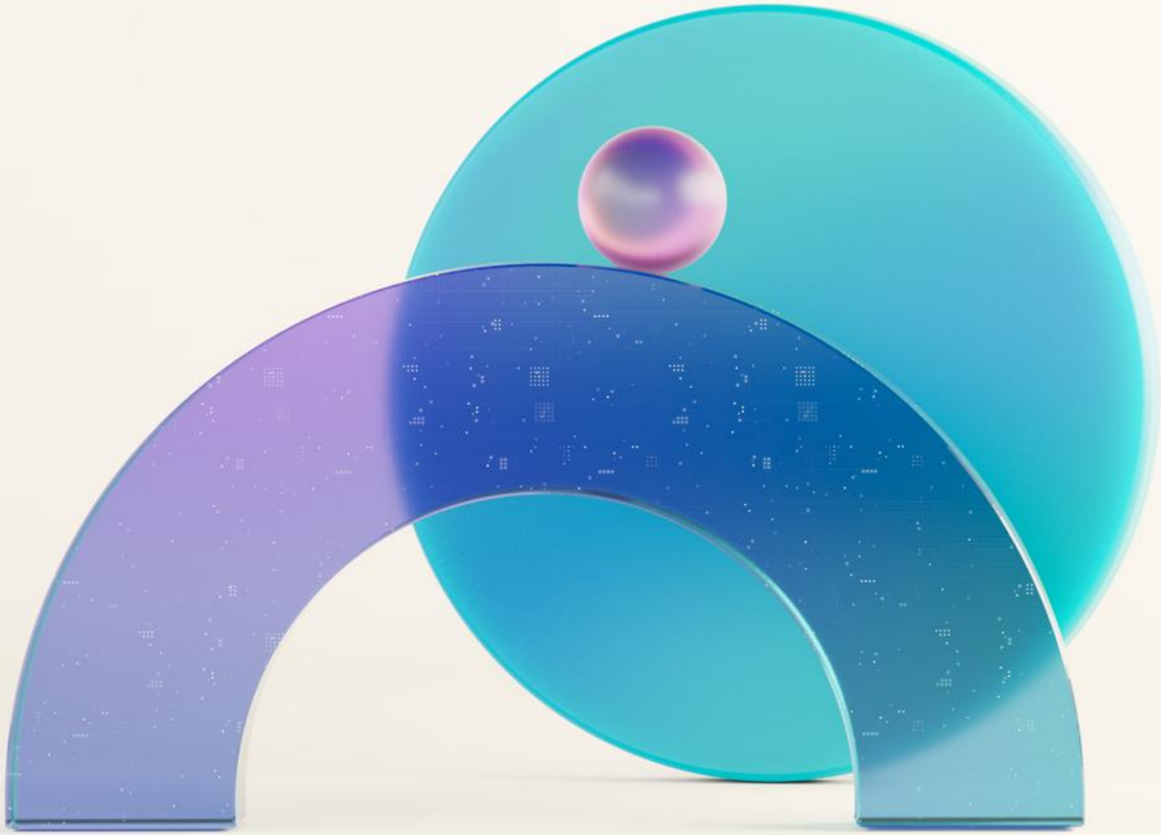
- Reduce DECIMAL/NUMERIC precision to the smallest one that can accommodate your data
- Don’t default to CHAR/VARCHAR(8000). Adjust the precision to your requirements (e.g.: VARCHAR(50))

Always use the smallest data type that supports your data

- Prefer INT over BIGINT when possible, or SMALLINT over INT
- Prefer DATE or TIME over DATETIME2 if you only need the date or time alone

Microsoft

Microsoft Fabric
Community Conference

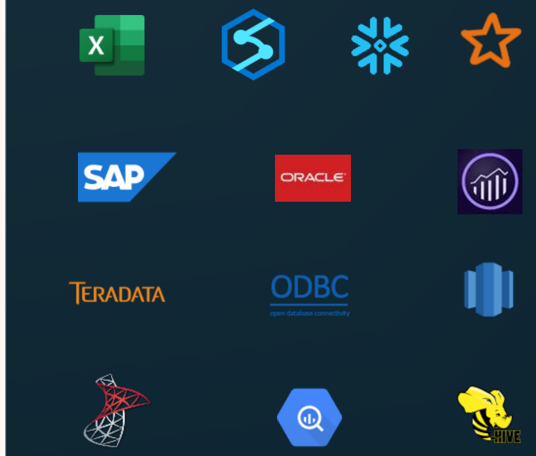


Data Ingestion

Module 4

Data orchestration and data prep with Pipelines

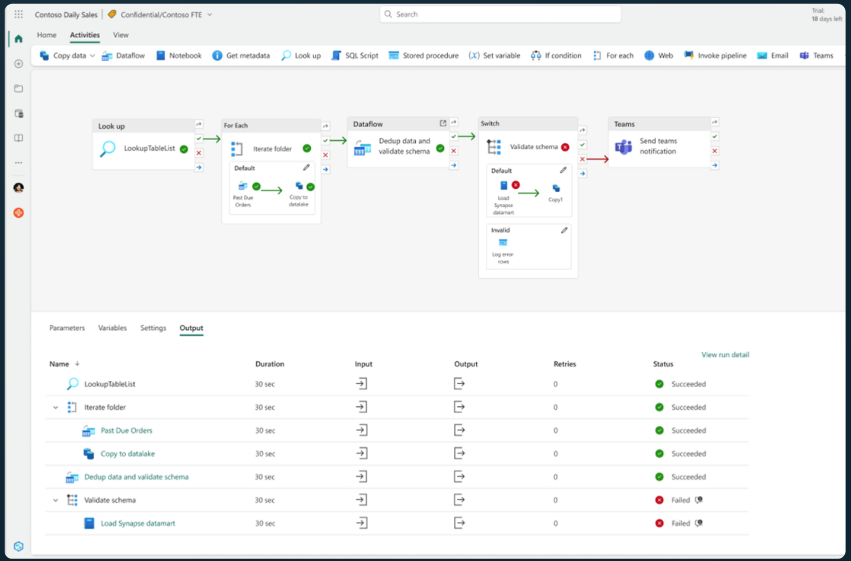
The worlds data



Through 170+ connectors

→

ETL, CDC, or ELT data prep needs



Name ↓	Duration	Input	Output	Retries	Status
LookupTableList	30 sec	→	→	0	● Succeeded
Iterate folder	30 sec	→	→	0	● Succeeded
Past Due Orders	30 sec	→	→	0	● Succeeded
Copy to datalake	30 sec	→	→	0	● Succeeded
Dedup data and validate schema	30 sec	→	→	0	● Succeeded
Validate schema	30 sec	→	→	0	● Failed
Load Synapse datamart	30 sec	→	→	0	● Failed

Rich library of activities and pro code data prep experiences

→

Warehouse

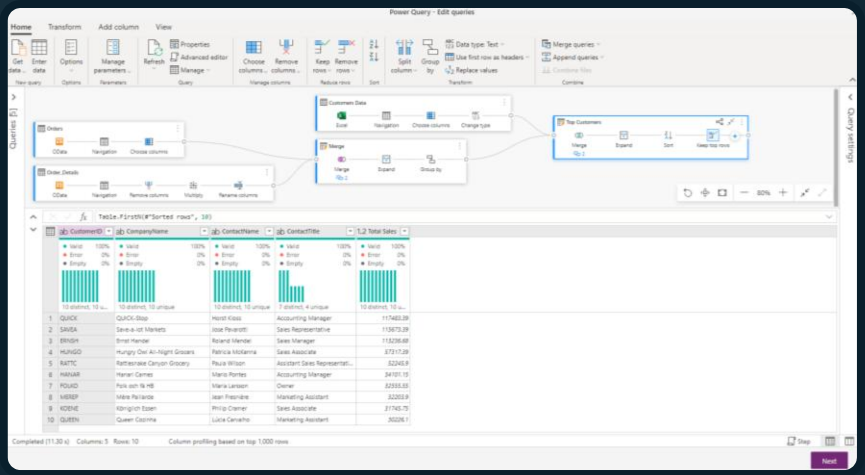
Ingest and transform data at scale

No code data prep with Dataflows gen 2

The worlds data

Through 170+ connectors

Familiar Power Query experiences



```
Table.FirstN(#"Sorted rows", 10)
```

	CustomerID	CompanyName	ContactName	ContactTitle	Total Sales
1	QUICK	QUICK-Stop	Horst Kloss	Accounting Manager	117483.39
2	SAVEA	Save-a-lot Markets	Jose Pavarotti	Sales Representative	115673.39
3	ERNSH	Ernst Handel	Roland Mendel	Sales Manager	113236.68
4	HUNGO	Hungry Owl All-Night Grocers	Patricia McKenna	Sales Associate	57317.39
5	RATTC	Rattlesnake Canyon Grocery	Paula Wilson	Assistant Sales Representat...	52245.9
6	HANAR	Hanari Carnes	Mario Pontes	Accounting Manager	34101.15
7	FOLKO	Folk och fä HB	Maria Larsson	Owner	32555.55
8	MEREP	Mère Paillarde	Jean Fresnière	Marketing Assistant	32203.9
9	KOENE	Königlich Essen	Philip Cramer	Sales Associate	31745.75
10	QUEEN	Queen Cozinha	Lúcia Carvalho	Marketing Assistant	30226.1

Completed (11.30 s) Columns: 5 Rows: 10 Column profiling based on top 1,000 rows

Through 400+ no code transformations

Warehouse

Ingest and transform data at scale

…and of course, you can use T-SQL too!

SQL COPY INTO, transactions, and more

Support for:	COPY INTO
	CTAS
	INSERT … SELECT
	Common Table Expressions
	Stored Procedures
Transactions and locking:	Supports snapshot isolation
	INSERTS and SELECTS only require SCHEMA stability locks
	UPDATES and DELETES do not block tables

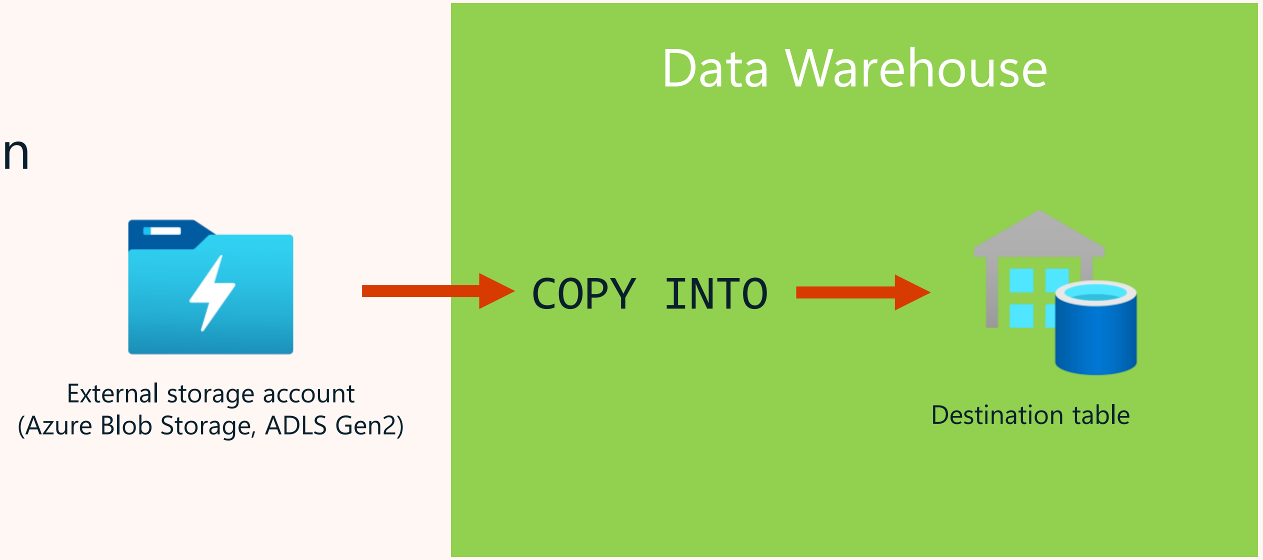
Simple, Familiar SQL Syntax

```
COPY INTO table_name.[(Column_list)]
FROM ‘<external_location>’ [,...n]
WITH
(
 [FILE_TYPE = {'CSV' | 'PARQUET' '} ]
 [,CREDENTIAL = (AZURE CREDENTIAL) ]
 [,COMPRESSION = { 'Gzip' | ’Snappy’}]
 [,FIELDQUOTE = ‘string_delimiter’]
 [,FIELDTERMINATOR = ‘field_terminator’]
 [,ROWTERMINATOR = ‘row_terminator’]
 [,FIRSTROW = first_row]
 [,ENCODING = {'UTF8'|'UTF16’}]
);

CREATE TABLE [dbo].[FactInternetSales_new]
AS
SELECT * FROM [dbo].[FactInternetSales];

INSERT INTO dbo.EmployeeSales
    SELECT 'SELECT', sp.BusinessEntityID, c.LastName, sp.SalesYTD
    FROM warehouseA.Sales.SalesPerson AS sp
    INNER JOIN warehouseB.Person.Person AS c
        ON sp.BusinessEntityID = c.BusinessEntityID
    WHERE sp.BusinessEntityID LIKE '2%'
    ORDER BY sp.BusinessEntityID, c.LastName;

WITH sales_data AS (
           SELECT customer_id, SUM(total_sales) AS total_sales
           FROM sales
           GROUP BY customer_id
)
SELECT customer_id, total_sales
FROM sales_data
WHERE total_sales > 1000;
```

COPY INTO

Enables flexible, high-throughput data ingestion from external Azure storage accounts

- Supports PARQUET and CSV formats
- No additional database objects required



Data Warehouse

External storage account
(Azure Blob Storage, ADLS Gen2)

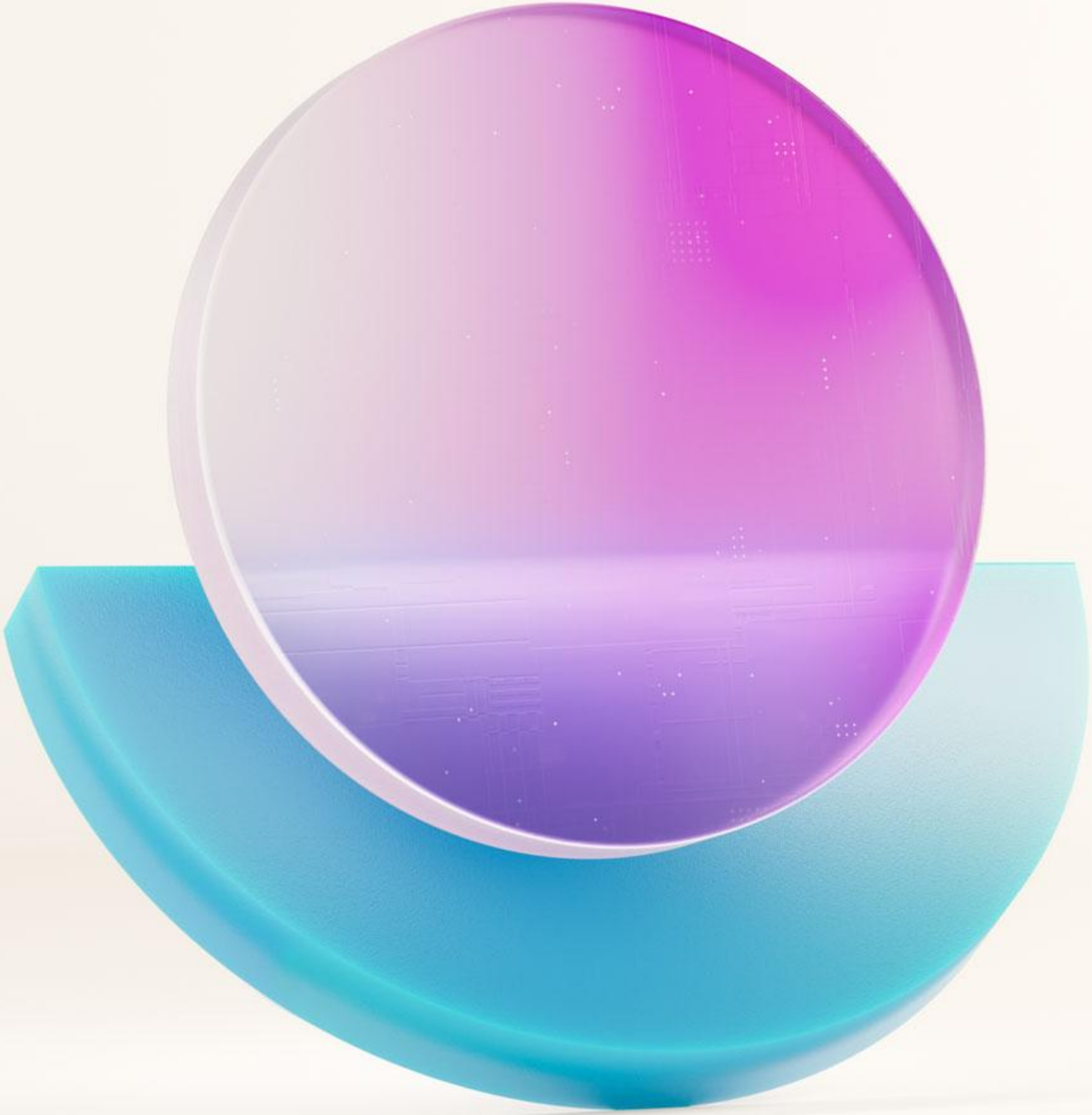
COPY INTO

Destination table

Example:

```
COPY INTO SalesOrderItems
FROM 'https://myaccount.blob.core.windows.net/mycontainer/*.csv'
WITH (
  FILE_TYPE = 'CSV’,
  FIRSTROW = 2
)
```

Microsoft Fabric
Community Conference



Demo

Ingesting data from tables on other warehouses

Cross-warehouse queries empower seamless integration of data from multiple warehouses, enabling comprehensive analysis, actionable insights, and efficient workflows for improved productivity.

- **Unified Data Analysis**: Combine datasets from different warehouses to gain a holistic view of business operations.
- **Enhanced Insights**: Identify trends and correlations across isolated datasets to drive informed decisions.
- **Boosted Productivity**: Streamline workflows by accessing and analyzing data from multiple sources in a single query.

Ingesting data from tables on other warehouses (cont’d)

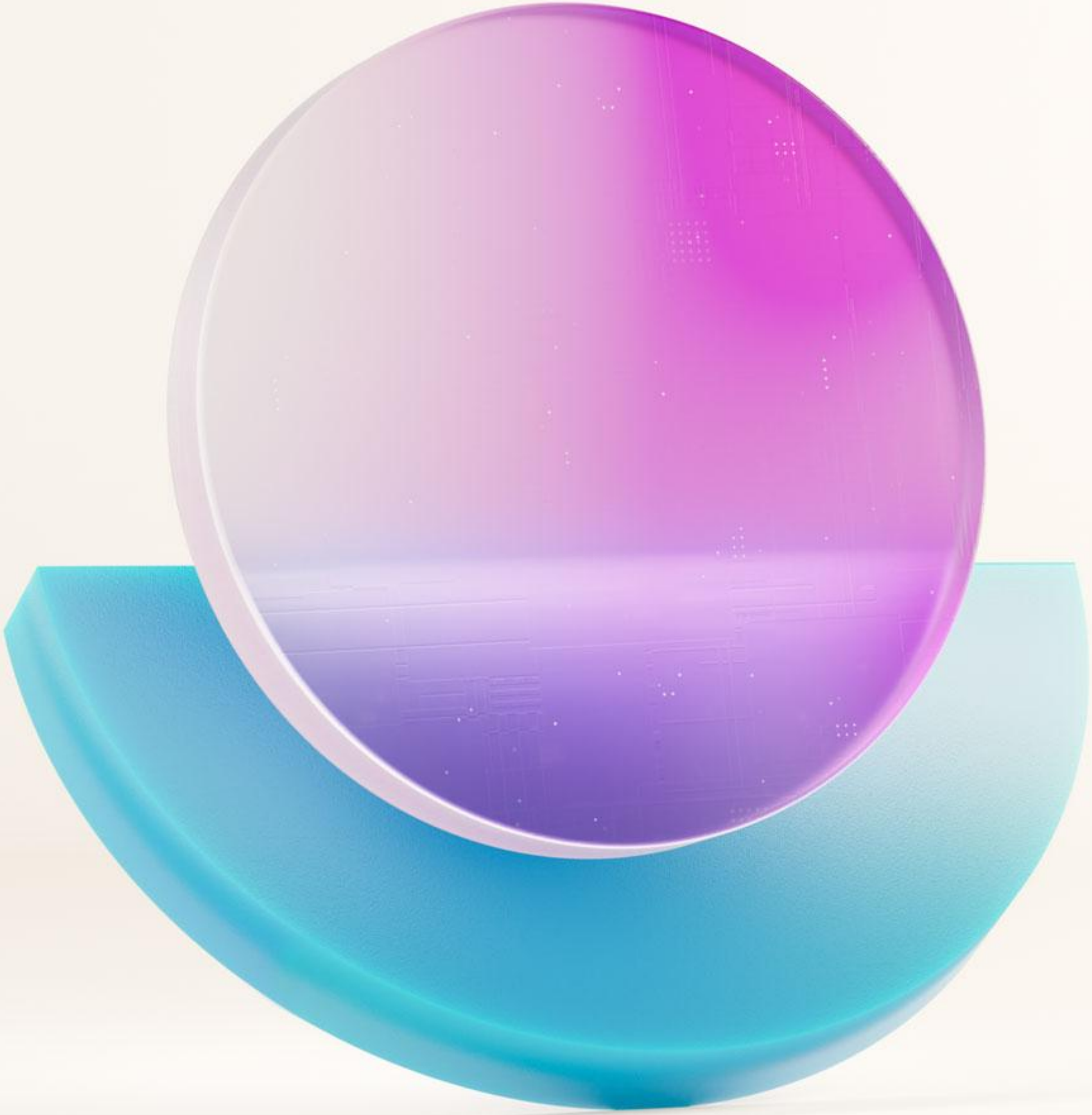
Cross-warehouse queries are allowed using three-part naming:

```
SELECT <column list>
FROM <WarehouseName>.<SchemaName>.<TableName>
```

Example using CREATE TABLE AS SELECT (CTAS):

```
CREATE TABLE WarehouseC.Analytics.CustomerSalesSummary AS
SELECT
    WHa.CustomerName,
    WHb.OrderDate,
    WHb.TotalOrderValue
FROM WarehouseA.Customers.CustomerData AS WHa
JOIN WarehouseB.Orders.SalesData AS WHb
ON WHa.CustomerID = WHb.CustomerID
WHERE WHb.TotalOrderValue > 100;
```

Other T-SQL patterns are also supported (e.g.: SELECT INTO, INSERT INTO… SELECT)



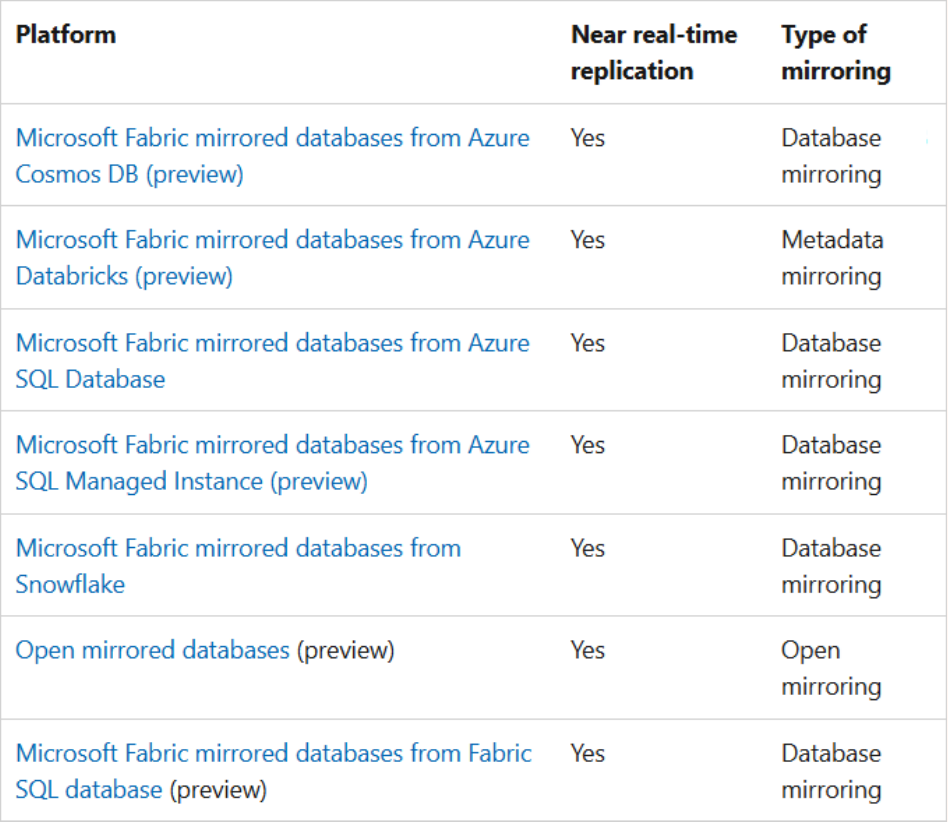
Demo

Mirroring

Mirroring in Fabric is a low-cost and low-latency solution to bring data from various systems together into a single analytics platform. You can continuously replicate your existing data estate directly into Fabric's OneLake from a variety of Azure databases and external data sources.

Enabling Mirroring is simple and intuitive, without having the need to create complex ETL pipelines, allocate other compute resources, and manage data movement.

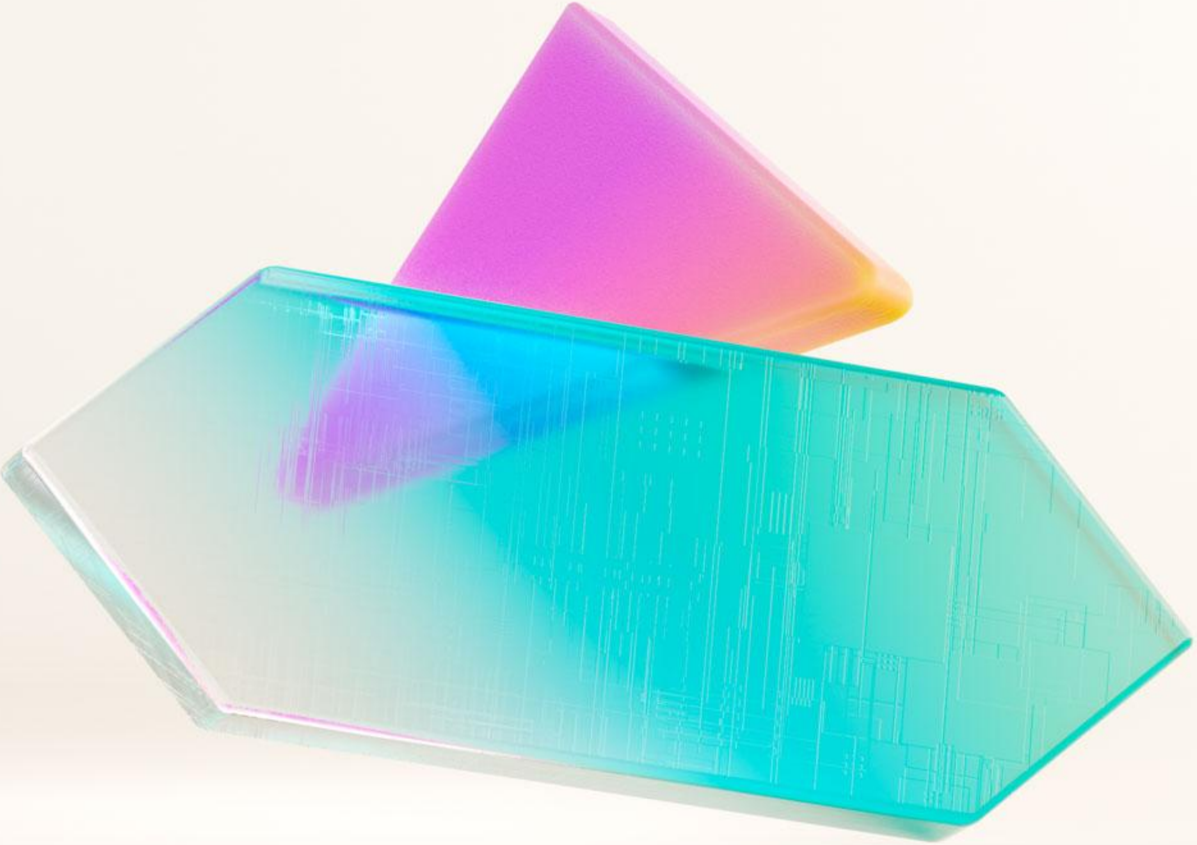
It is a fully managed service, so you don't have to worry about hosting, maintaining, or managing replication of the mirrored connection.



Platform	Near real-time replication	Type of mirroring
Microsoft Fabric mirrored databases from Azure Cosmos DB (preview)	Yes	Database mirroring
Microsoft Fabric mirrored databases from Azure Databricks (preview)	Yes	Metadata mirroring
Microsoft Fabric mirrored databases from Azure SQL Database	Yes	Database mirroring
Microsoft Fabric mirrored databases from Azure SQL Managed Instance (preview)	Yes	Database mirroring
Microsoft Fabric mirrored databases from Snowflake	Yes	Database mirroring
Open mirrored databases (preview)	Yes	Open mirroring
Microsoft Fabric mirrored databases from Fabric SQL database (preview)	Yes	Database mirroring

Availability of Mirroring by platform

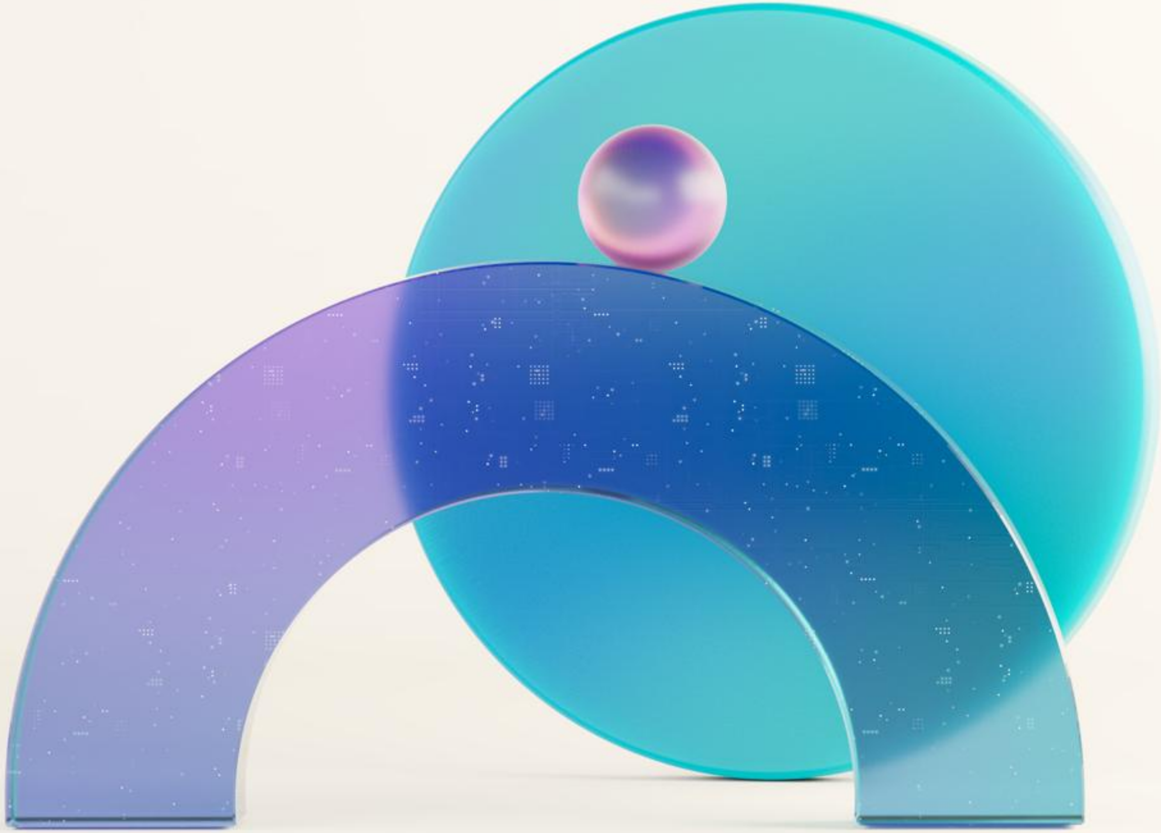
Microsoft Fabric
COMMUNITY CONFERENCE



10-minute break

Microsoft

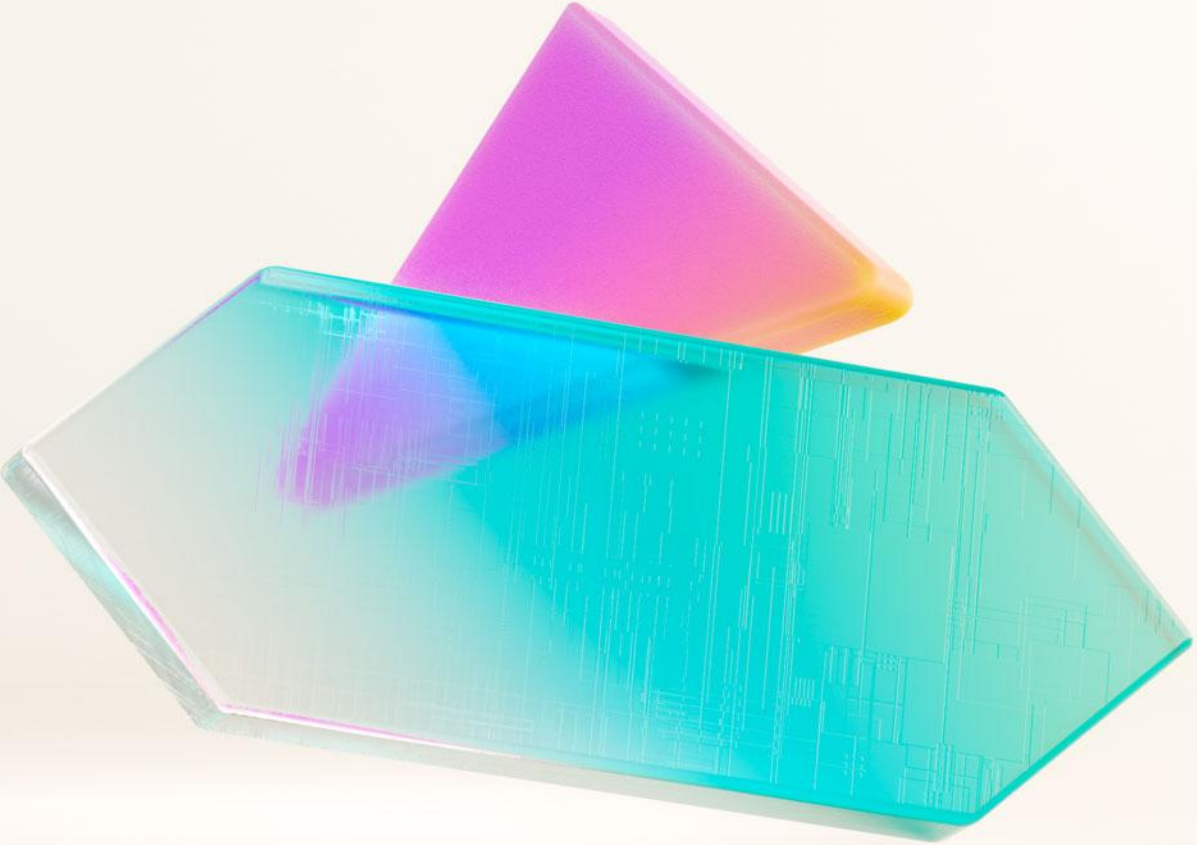
Microsoft Fabric
Community Conference



Lunch

We will start at
1:00 PM

Microsoft Fabric
COMMUNITY CONFERENCE



Data Transformation and T-SQL

Module 5

How do you perform data transformation?

At Load Time

Extract Transform Load

Dataflows Gen2

Spark

SQL Server Integration Services (SSIS)

After Loading

Extract Load Transform

T-SQL Scripts (at query time)

Stored Procedures

Functions

Programmability

Stored procedures

Encapsulate sets of logic written in T-SQL.

Can be called through T-SQL or through Data Factory pipeline activities.

Run any T-SQL statement in the supported surface area.

Functions

Support for table-valued functions

Scalar functions coming soon

Repeatable logic that can be used independently or referenced in other T-SQL statements

Common Data Warehouse Operations

Surrogate Keys

Synthetic key only used in the data warehouse

Identity support on the roadmap

```
DECLARE @MaxKey
SELECT MAX(Key) FROM Table

…@MaxKey + RowNumber…
```

Type 1 and 2 Attributes

MERGE coming soon!

Update

Type 1 attributes
End date type 2 records

Insert

Insert new records
Insert new type 2 versions

Late Arriving Members

Create "unknown" members in dimension

Best practice: use -1 or 0 for the unknown member

Time Travel

Historical analysis of data up to 30 days in the past.

Granularity of the time travel is at the statement level. Cannot vary times for each table within a query.

Audit changes for compliance purposes.

Only returns the latest version of the table schema.

Dropping a table removes the history.

T-SQL Syntax

```
SELECT
        [Columns]
FROM [schema].[Table or View]
[JOIN Table or View]
OPTION (FOR TIMESTAMP AS OF ‘YYYY-MM-DDTHH:MM:SS.ss’);
```

Cloning tables

“Copy” data with zero duplication of data.

Create historical reports reflecting data as it was at a specific point in time.

Clones become a new branch of the table allowing for full, independent DDL and DML without impact to the original table.

Security is inherited by the clone.

T-SQL Syntax

```
CREATE TABLE { database_name.schema_name.table_name | schema_name.table_name | table_name }
AS CLONE OF { database_name.schema_name.table_name | schema_name.table_name | table_name }
[AT {point_in_time}]
```

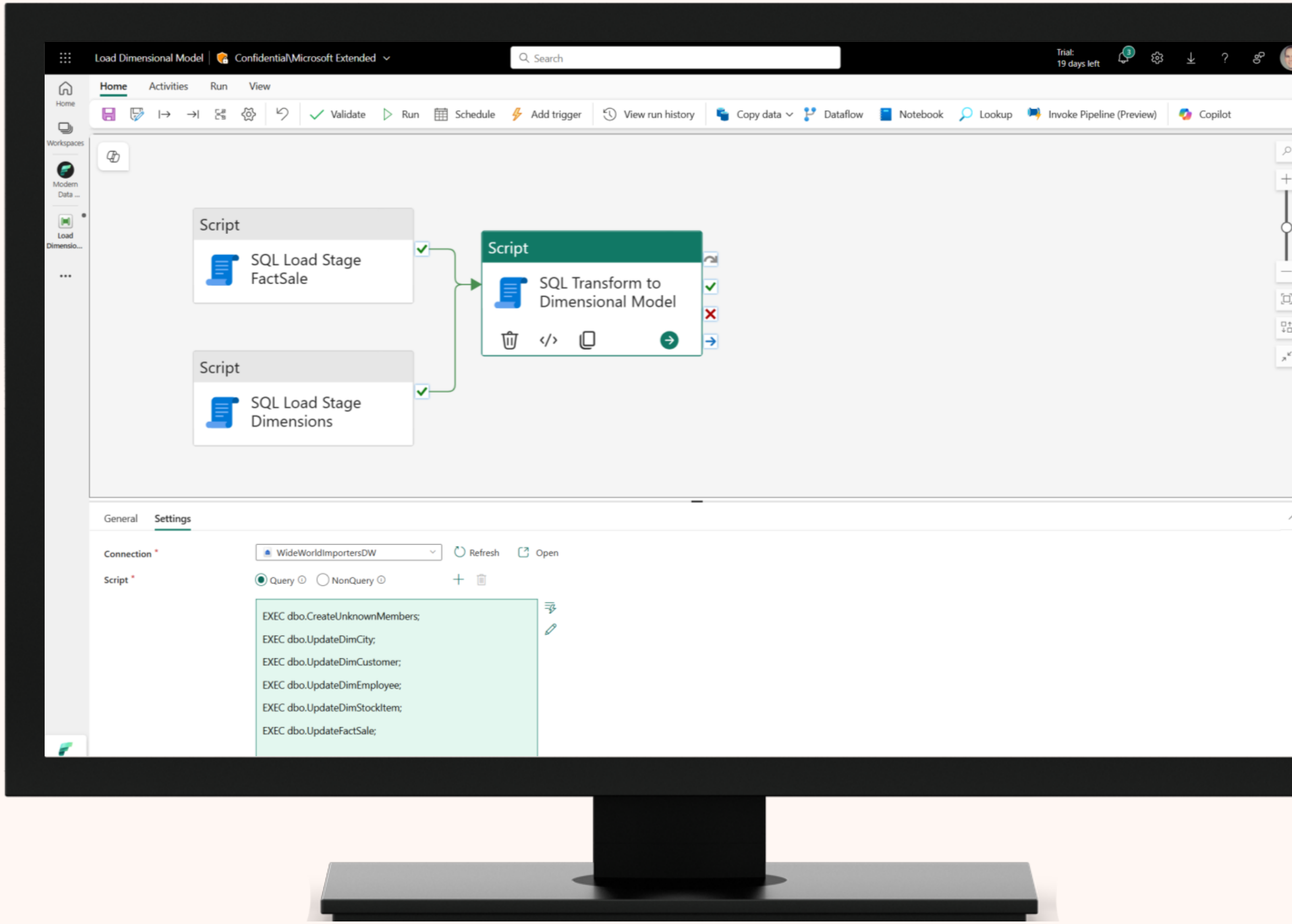
Orchestrating ETL Operations with Pipelines

Data Factory pipelines move data or dispatch operations to other compute engines.

Ensure proper order of operations.

Facilitate scheduling ETL operations.

Can be simple “run this code” or complex metadata driven frameworks.



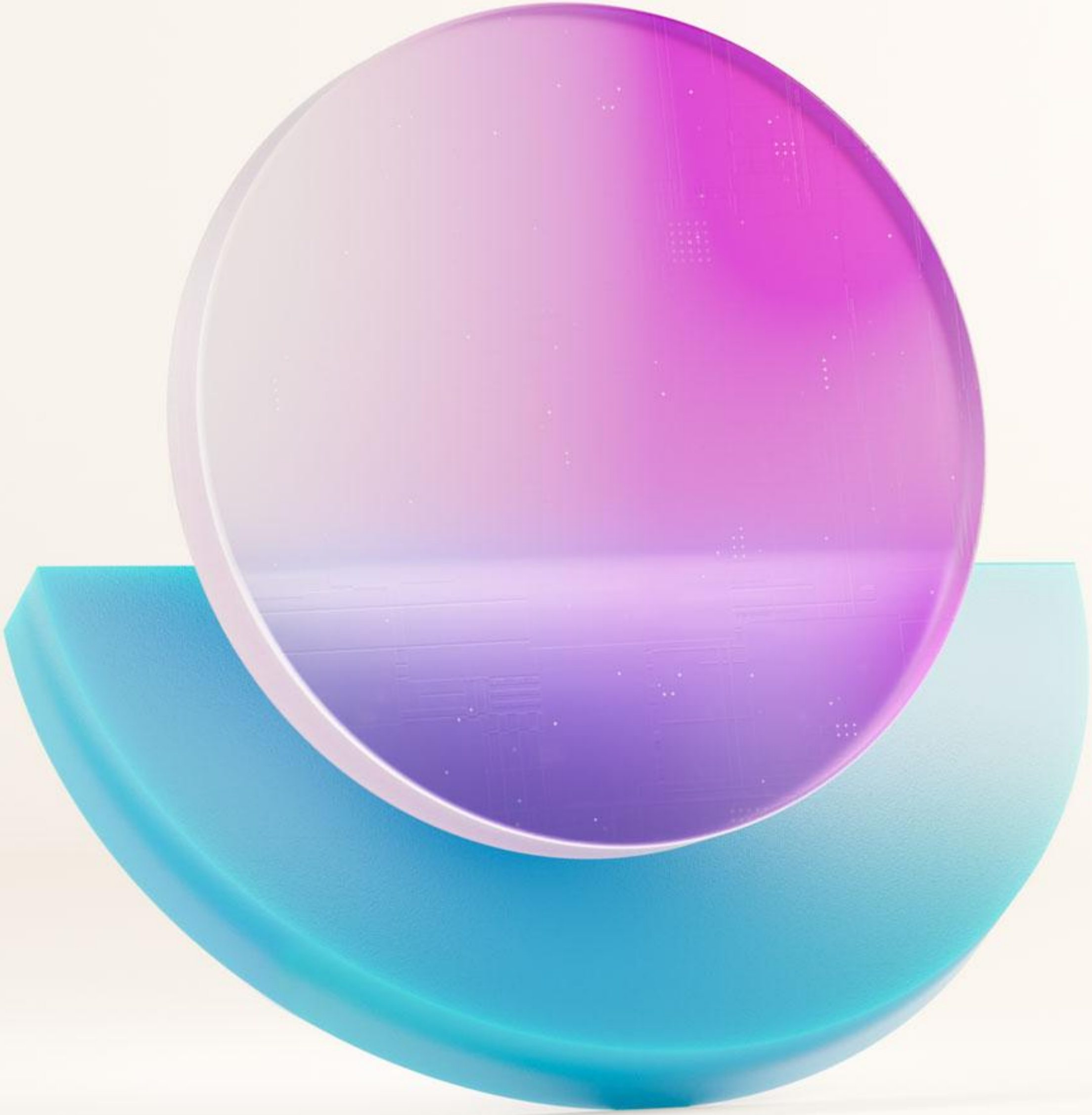
General | Settings

Connection * WideWorldImportersDW ⌄ Refresh Open

Script * ● Query ○ NonQuery

```
EXEC dbo.CreateUnknownMembers;
EXEC dbo.UpdateDimCity;
EXEC dbo.UpdateDimCustomer;
EXEC dbo.UpdateDimEmployee;
EXEC dbo.UpdateDimStockItem;
EXEC dbo.UpdateFactSale;
```

Microsoft Fabric
Community Conference



Monitoring Warehouse Activity

Module 6

Dynamic Management Views

sys.dm_exec_connections

Information about each connection established to the engine.

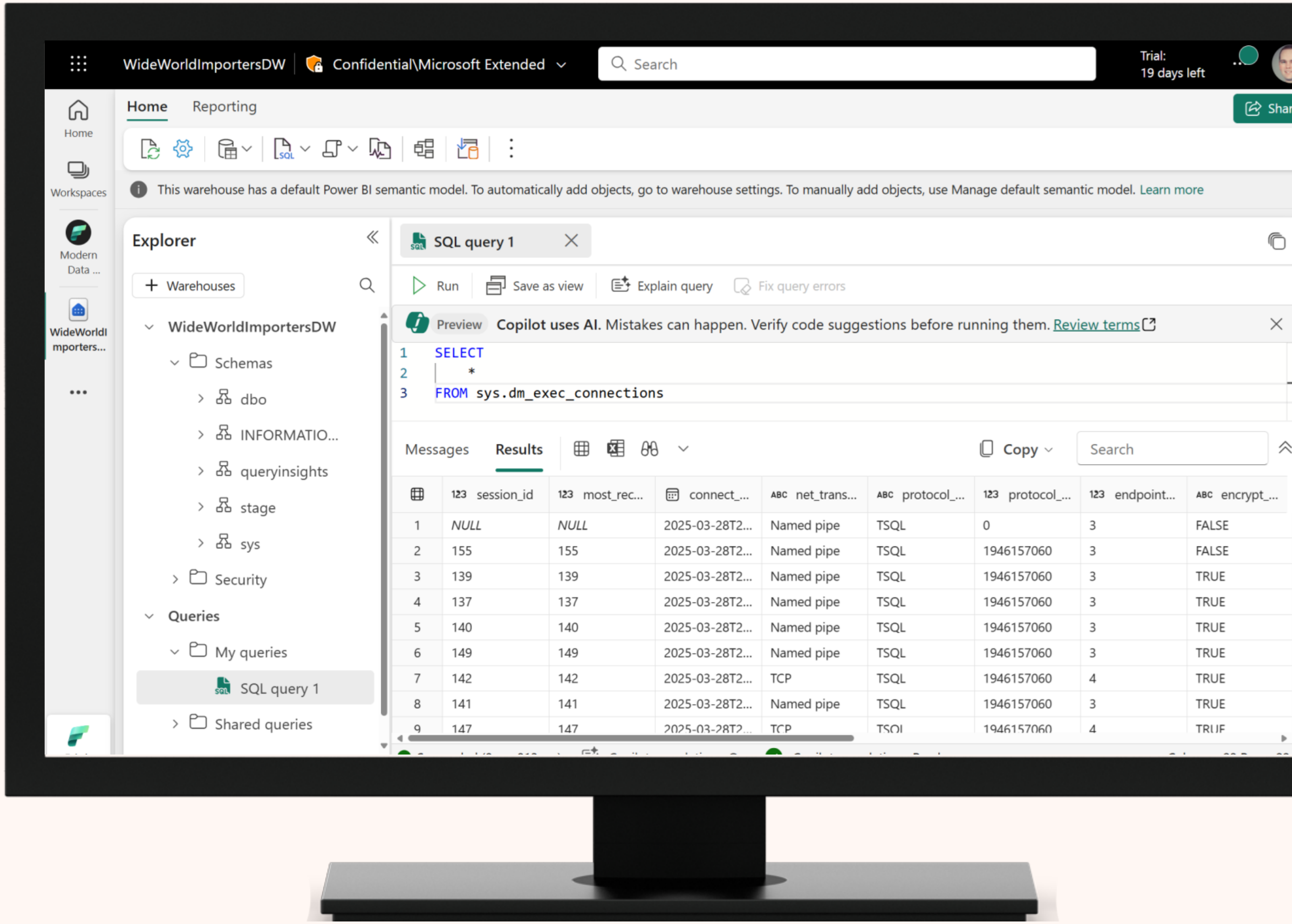
sys.dm_exec_sessions

Information about each session.

sys.dm_exec_requests

Information about each request made by users and generated by the system.

Member, Contributor, or View permissions.



This warehouse has a default Power BI semantic model. To automatically add objects, go to warehouse settings. To manually add objects, use Manage default semantic model. Learn more

Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. Review terms

```sql
SELECT
    *
FROM sys.dm_exec_connections
```

	session_id	most_rec...	connect_...	net_trans...	protocol_...	protocol_...	endpoint...	encrypt_...
1	NULL	NULL	2025-03-28T2...	Named pipe	TSQL	0	3	FALSE
2	155	155	2025-03-28T2...	Named pipe	TSQL	1946157060	3	FALSE
3	139	139	2025-03-28T2...	Named pipe	TSQL	1946157060	3	TRUE
4	137	137	2025-03-28T2...	Named pipe	TSQL	1946157060	3	TRUE
5	140	140	2025-03-28T2...	Named pipe	TSQL	1946157060	3	TRUE
6	149	149	2025-03-28T2...	Named pipe	TSQL	1946157060	3	TRUE
7	142	142	2025-03-28T2...	TCP	TSQL	1946157060	4	TRUE
8	141	141	2025-03-28T2...	Named pipe	TSQL	1946157060	3	TRUE
9	147	147	2025-03-28T2...	TCP	TSQL	1946157060	4	TRUE

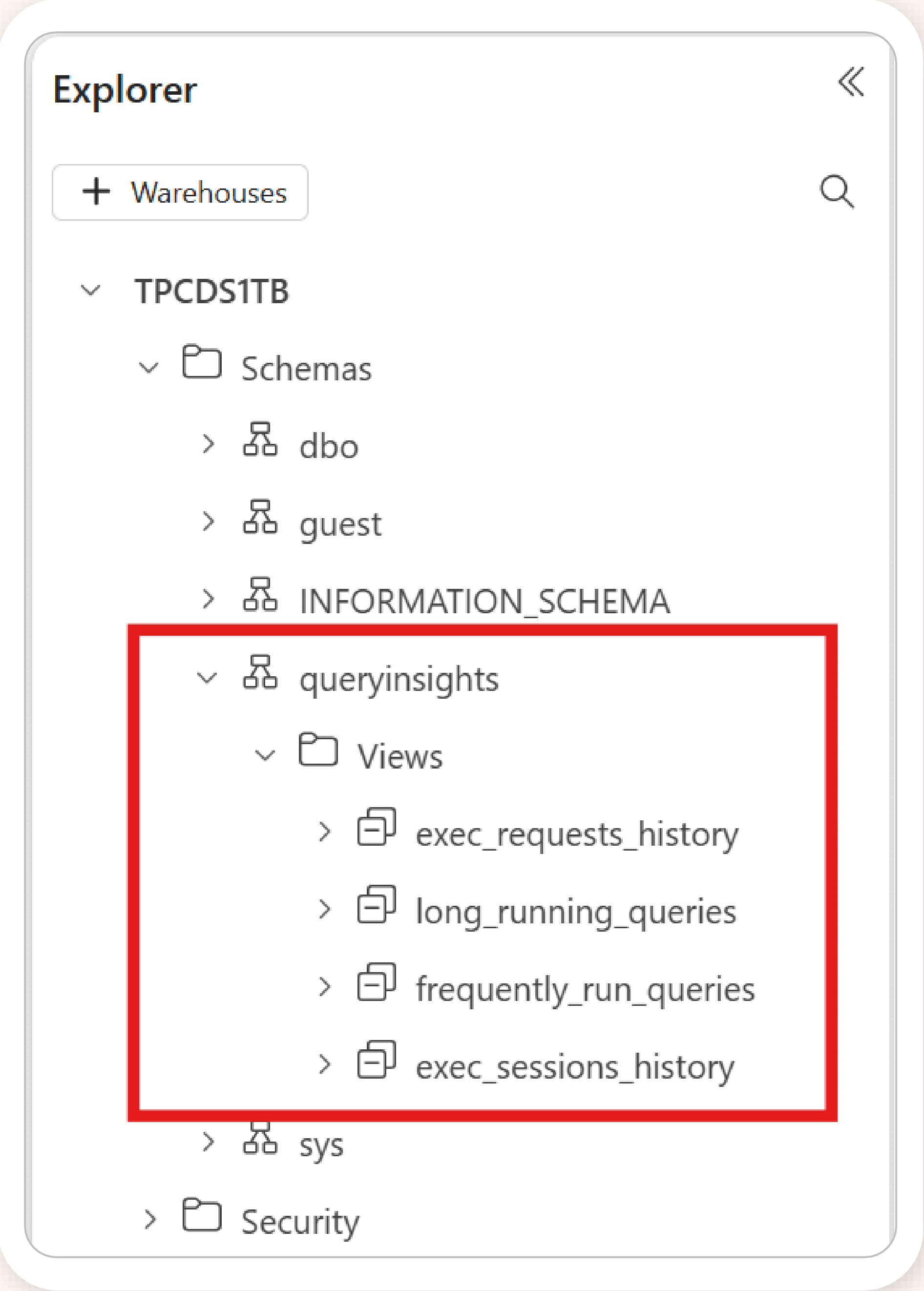
Query Insights

Each query reflected in queryinsights.exec_requests_history

Populated on a delay of 5-15 minutes.

Useful information about execution:

- Data scanned
- CPU time
- Historical runs
- Long running queries
- Highest cost queries



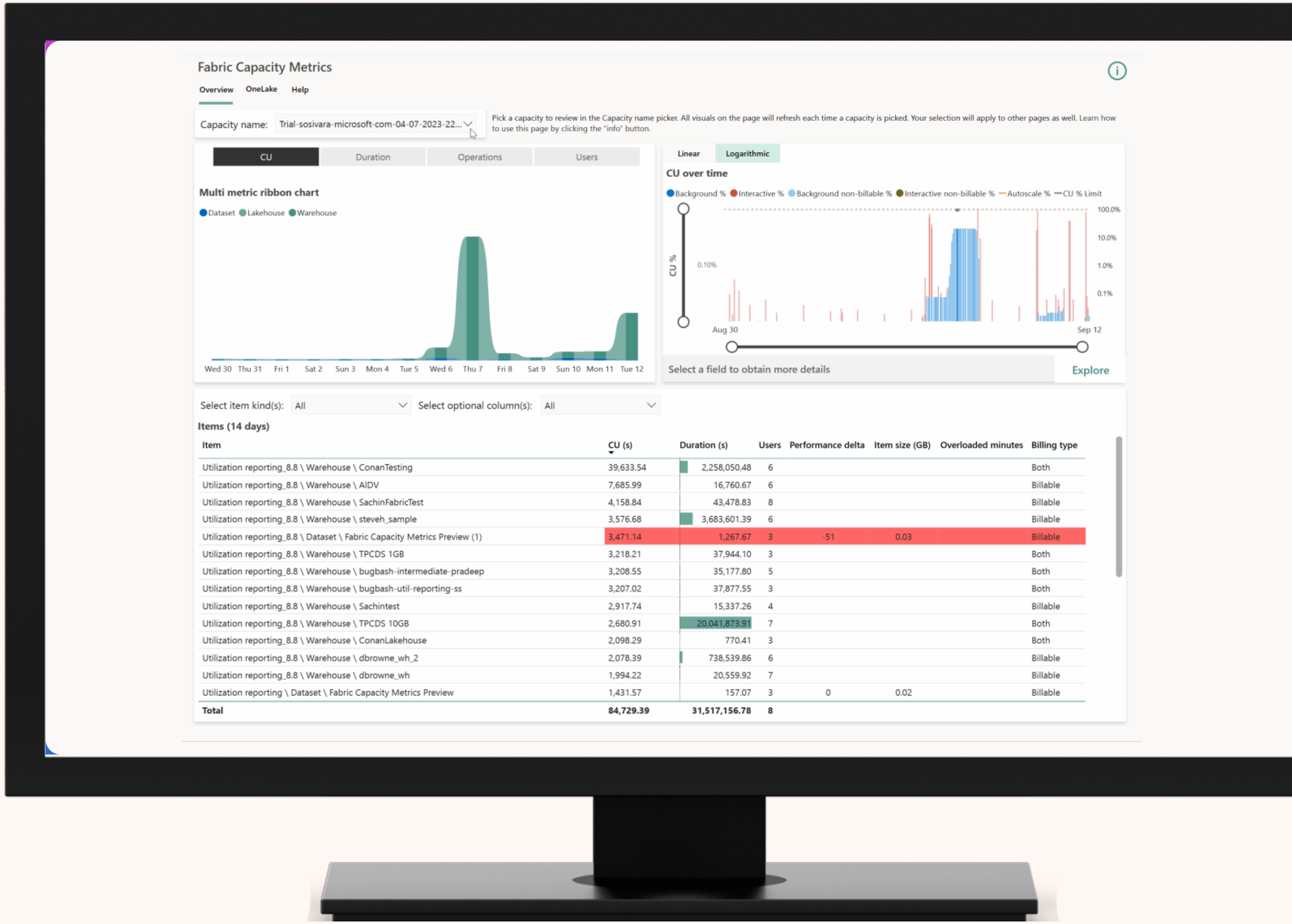
Capacity Metrics

Visibility into capacity usage for all Fabric workloads, including data warehouse, in one place.

New usage is reflected in the report approximately every 10-15 minutes.

Use statement id to find individual T-SQL query.

The **Total CU (s)** field represents the total CU cost of the query.



Capacity name: Trial-sosivara-microsoft-com-04-07-2023-22...

Pick a capacity to review in the Capacity name picker. All visuals on the page will refresh each time a capacity is picked. Your selection will apply to other pages as well. Learn how to use this page by clicking the "info" button.

Select item kind(s): All

Select optional column(s): All

Items (14 days)

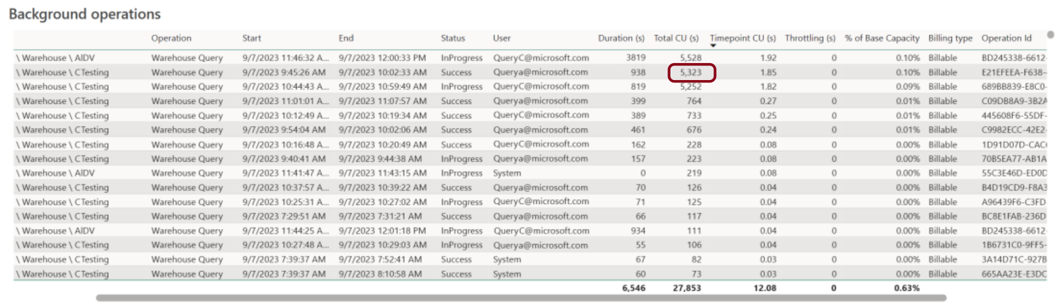
Item	CU (s)	Duration (s)	Users	Performance delta	Item size (GB)	Overloaded minutes	Billing type
Utilization reporting_8.8 \ Warehouse \ ConanTesting	39,633.54	2,258,050.48	6				Both
Utilization reporting_8.8 \ Warehouse \ AIDV	7,685.99	16,760.67	6				Billable
Utilization reporting_8.8 \ Warehouse \ SachinFabricTest	4,158.84	43,478.83	8				Billable
Utilization reporting_8.8 \ Warehouse \ steveh_sample	3,576.68	3,683,601.39	6				Billable
Utilization reporting_8.8 \ Dataset \ Fabric Capacity Metrics Preview (1)	3,471.14	1,267.67	3	-51	0.03		Billable
Utilization reporting_8.8 \ Warehouse \ TPCDS 1GB	3,218.21	37,944.10	3				Both
Utilization reporting_8.8 \ Warehouse \ bugbash-intermediate-pradeep	3,208.55	35,177.80	5				Both
Utilization reporting_8.8 \ Warehouse \ bugbash-util-reporting-ss	3,207.02	37,877.55	3				Both
Utilization reporting_8.8 \ Warehouse \ Sachintest	2,917.74	15,337.26	4				Billable
Utilization reporting_8.8 \ Warehouse \ TPCDS 10GB	2,680.91	20,041,873.91	7				Both
Utilization reporting_8.8 \ Warehouse \ ConanLakehouse	2,098.29	770.41	3				Both
Utilization reporting_8.8 \ Warehouse \ dbrowne_wh_2	2,078.39	738,539.86	6				Billable
Utilization reporting_8.8 \ Warehouse \ dbrowne_wh	1,994.22	20,559.92	7				Billable
Utilization reporting \ Dataset \ Fabric Capacity Metrics Preview	1,431.57	157.07	3	0	0.02		Billable
Total	**84,729.39**	**31,517,156.78**	**8**				

Identifying a query’s cost

Optionally, with the total CU (s) of the query, compute the monetary cost of the query

Using the hourly rate from the Fabric pricing page, calculate the price per CU second for your capacity’s region

Multiply the Total CU (s) value by the CU second cost to get the total query cost



Background operations

	Operation	Start	End	Status	User	Duration (s)	Total CU (s)	Timepoint CU (s)	Throttling (s)	% of Base Capacity	Billing type	Operation Id
\ Warehouse \ AIDV	Warehouse Query	9/7/2023 11:46:32 A...	9/7/2023 12:00:33 PM	InProgress	QueryC@microsoft.com	3819	5,528	1.92	0	0.10%	Billable	BD245338-6612-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 9:45:26 AM	9/7/2023 10:02:33 AM	Success	Querya@microsoft.com	938	5,323	1.85	0	0.10%	Billable	E21EFEEA-F638-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 10:44:43 A...	9/7/2023 10:59:49 AM	InProgress	QueryC@microsoft.com	819	5,252	1.82	0	0.09%	Billable	689BB839-E8C0-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 11:01:01 A...	9/7/2023 11:07:57 AM	Success	Querya@microsoft.com	399	764	0.27	0	0.01%	Billable	C09DB8A9-3B2A
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 10:12:49 A...	9/7/2023 10:19:34 AM	Success	QueryC@microsoft.com	389	733	0.25	0	0.01%	Billable	445608F6-55DF-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 9:54:04 AM	9/7/2023 10:02:06 AM	Success	Querya@microsoft.com	461	676	0.24	0	0.01%	Billable	C9982ECC-42E2-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 10:16:48 A...	9/7/2023 10:20:49 AM	Success	QueryC@microsoft.com	162	228	0.08	0	0.00%	Billable	1D91D07D-CAC
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 9:40:41 AM	9/7/2023 9:44:38 AM	InProgress	Querya@microsoft.com	157	223	0.08	0	0.00%	Billable	70B5EA77-AB1A
\ Warehouse \ AIDV	Warehouse Query	9/7/2023 11:41:47 A...	9/7/2023 11:43:15 AM	InProgress	System	0	219	0.08	0	0.00%	Billable	55C3E46D-ED0D
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 10:37:57 A...	9/7/2023 10:39:22 AM	Success	Querya@microsoft.com	70	126	0.04	0	0.00%	Billable	B4D19CD9-F8A3
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 10:25:31 A...	9/7/2023 10:27:02 AM	InProgress	QueryC@microsoft.com	71	125	0.04	0	0.00%	Billable	A96439F6-C3FD-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 7:29:51 AM	9/7/2023 7:31:21 AM	Success	Querya@microsoft.com	66	117	0.04	0	0.00%	Billable	BC8E1FAB-236D
\ Warehouse \ AIDV	Warehouse Query	9/7/2023 11:44:25 A...	9/7/2023 12:01:18 PM	InProgress	QueryC@microsoft.com	934	111	0.04	0	0.00%	Billable	BD245338-6612-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 10:27:48 A...	9/7/2023 10:29:03 AM	InProgress	Querya@microsoft.com	55	106	0.04	0	0.00%	Billable	1B6731C0-9FF5-
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 7:39:37 AM	9/7/2023 7:52:41 AM	Success	System	67	82	0.03	0	0.00%	Billable	3A14D71C-927B
\ Warehouse \ CTesting	Warehouse Query	9/7/2023 7:39:37 AM	9/7/2023 8:10:58 AM	Success	System	60	73	0.03	0	0.00%	Billable	665AA23E-E3DC
						6,546	**27,853**	**12.08**	**0**	**0.63%**		

Cost Per Capacity Unit Second Calculation

- F2 Hourly Rate / 3,600 seconds per hour / 2 CU in an F2 = $ per CU second
- F2 Hourly Rate in East US 2 = $0.36
- $0.36 / 3600 / 2 = $0.00005 per CU second

Total Query Cost Calculation

- $ Per CU second * Total CU (S) for the query = Total Query Cost
- $0.00005 * 5323 = $0.26615

Monitoring Warehouse Activity

What is happening right now?

DMVs

Fabric Data Warehouse Query Monitor

What happened in the past?

Query Insights

Fabric Data Warehouse Query Monitor

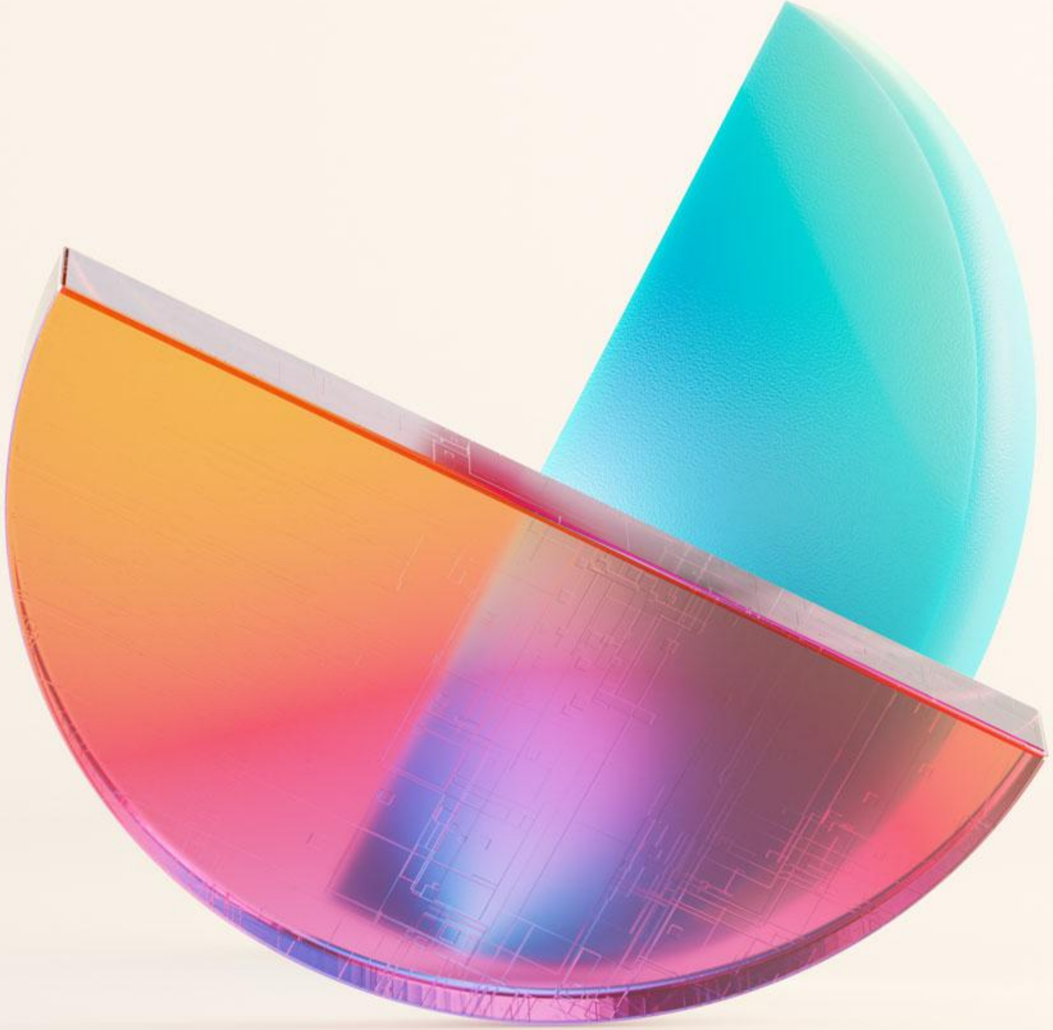
What happened on my capacity?

Capacity Metrics

Query Insights*

Microsoft

Microsoft Fabric
Community Conference

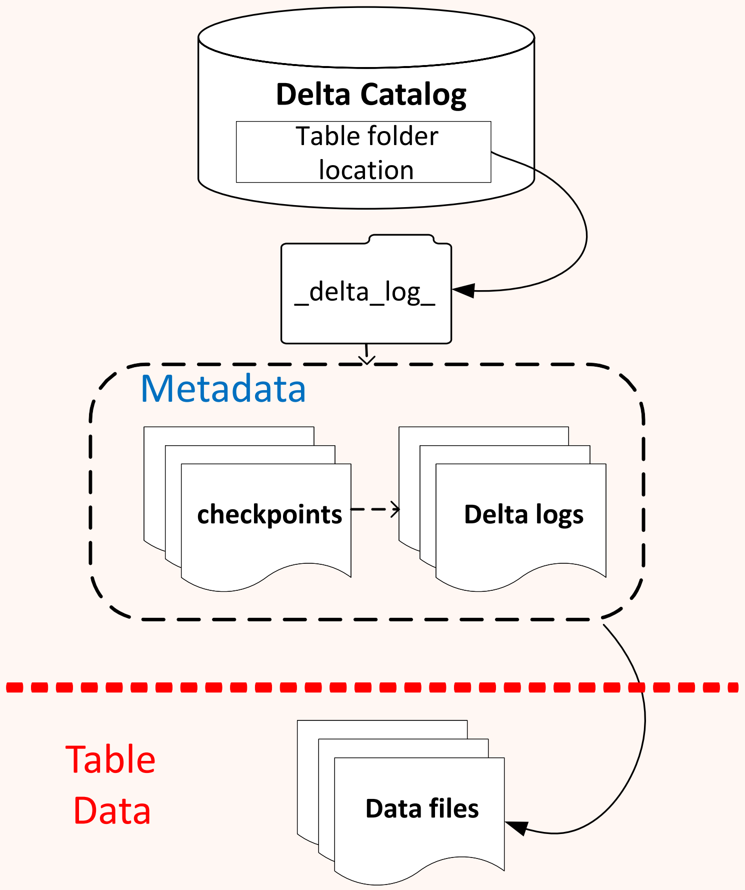


Statistics and V-Order

Module 7

Storage engine design principles

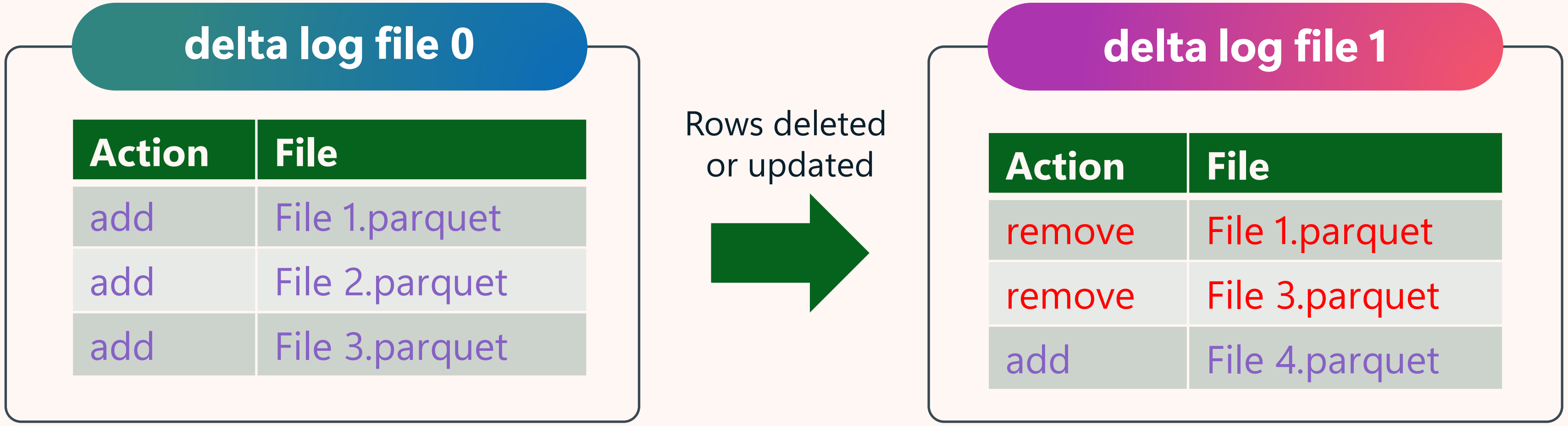
- Cloud first. Separation of State and Compute
- Columnar, immutable
- Open storage format
- Support for lineage-based features
- Snapshot Isolation semantics with multi-table transaction support
- No cross-component state sharing



Data Compaction

As data changes, the warehouse engine maintains storage optimal

No user action is required



delta log file 0

Action	File
add	File 1.parquet
add	File 2.parquet
add	File 3.parquet

Rows deleted or updated

delta log file 1

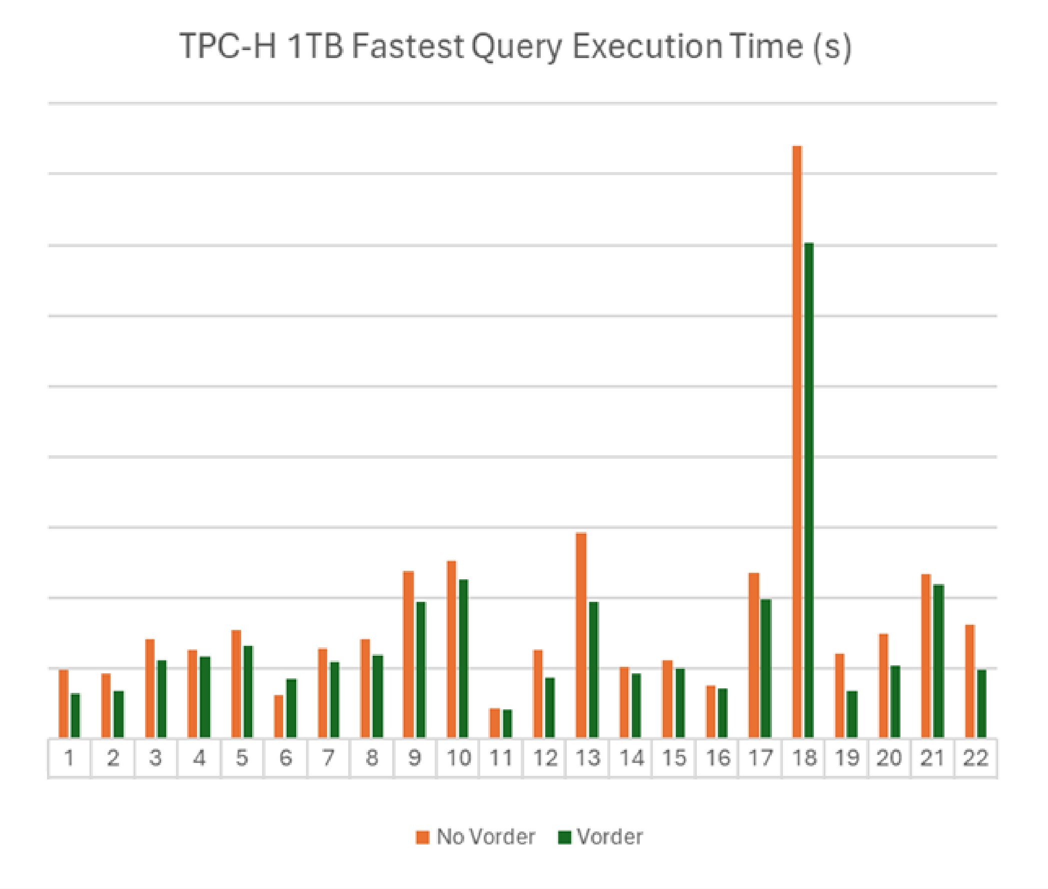
Action	File
remove	File 1.parquet
remove	File 3.parquet
add	File 4.parquet

V-Order

V-Order is a write time optimization to the parquet file format that enables lightning-fast reads under Microsoft Fabric compute engines, such as Power BI, SQL, Spark and others. It applies special sorting and compression to Parquet.

Warehouse queries benefit from faster read times with v-order, still ensuring files are 100% compliant to Parquet's open-source specification.

All warehouse data is written with v-order optimization during data ingestion.



Query performance: no V-Order vs V-Order

V-Order (cont'd)

IMPORTANT: Disabling V-Order can only be done at the warehouse level, and it is irreversible. Once disabled, it cannot be enabled again. Users must consider all the performance impact of disabling V-Order before deciding to do so. WE DO NOT RECOMMEND DISABLING V-ORDER WITHOUT A THOROUGH INVESTIGATION OF THE PERFORMANCE IMPACT.

Disabling V-Order can make data ingestion faster, but it has the following impact:

- V-Order optimizes Parquet files for faster reads by applying sorting, row group distribution, dictionary encoding, and compression. Disabling it can lead to slower query performance, especially for analytics scenarios.
- Direct Lake mode in Power BI requires V-Order. If you disable V-Order, Power BI will fall back to direct query mode when querying tables in your V-Order-disabled warehouse.
- Only newly ingested data will not use V-Order; all existing files in your warehouse will remain with V-Order until rows are re-written (through ingestion or compaction).

V-Order (cont'd)

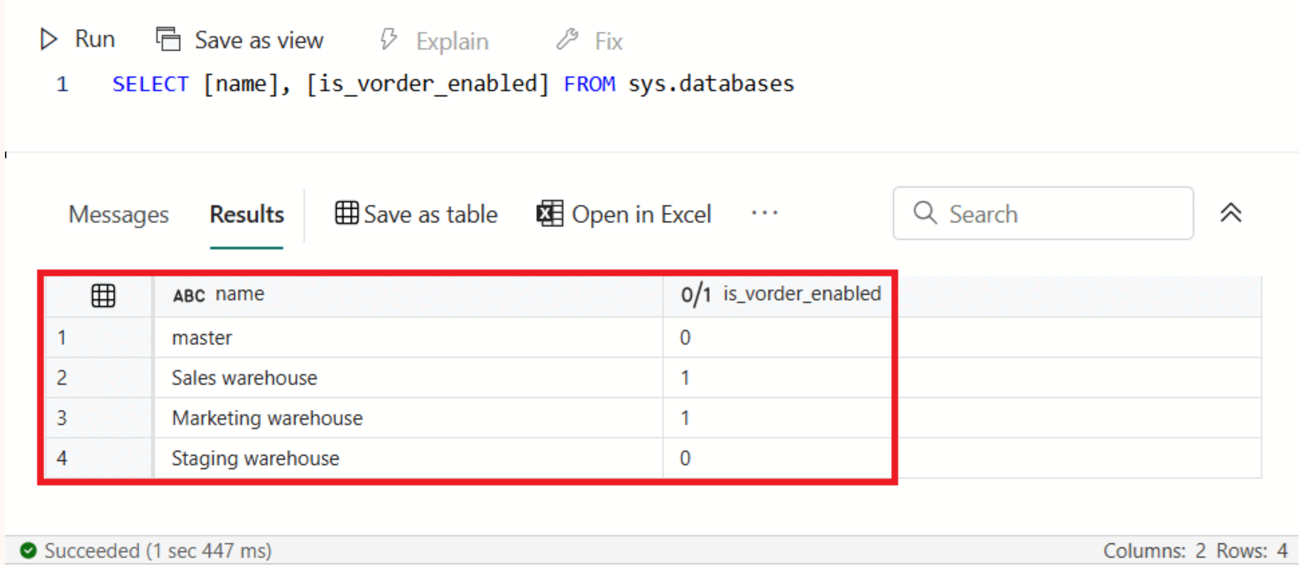
In special cases (e.g.: write intensive warehouses where reads are limited; staging warehouses; Power BI Direct Lake mode not needed), V-ORDER can be disabled.

To disable V-Order in the current Warehouse:

```
ALTER DATABASE CURRENT SET VORDER = OFF
```

To check the current configuration of V-ORDER on your warehouse:

```
SELECT [name], [is_vorder_enabled]
FROM sys.databases
```

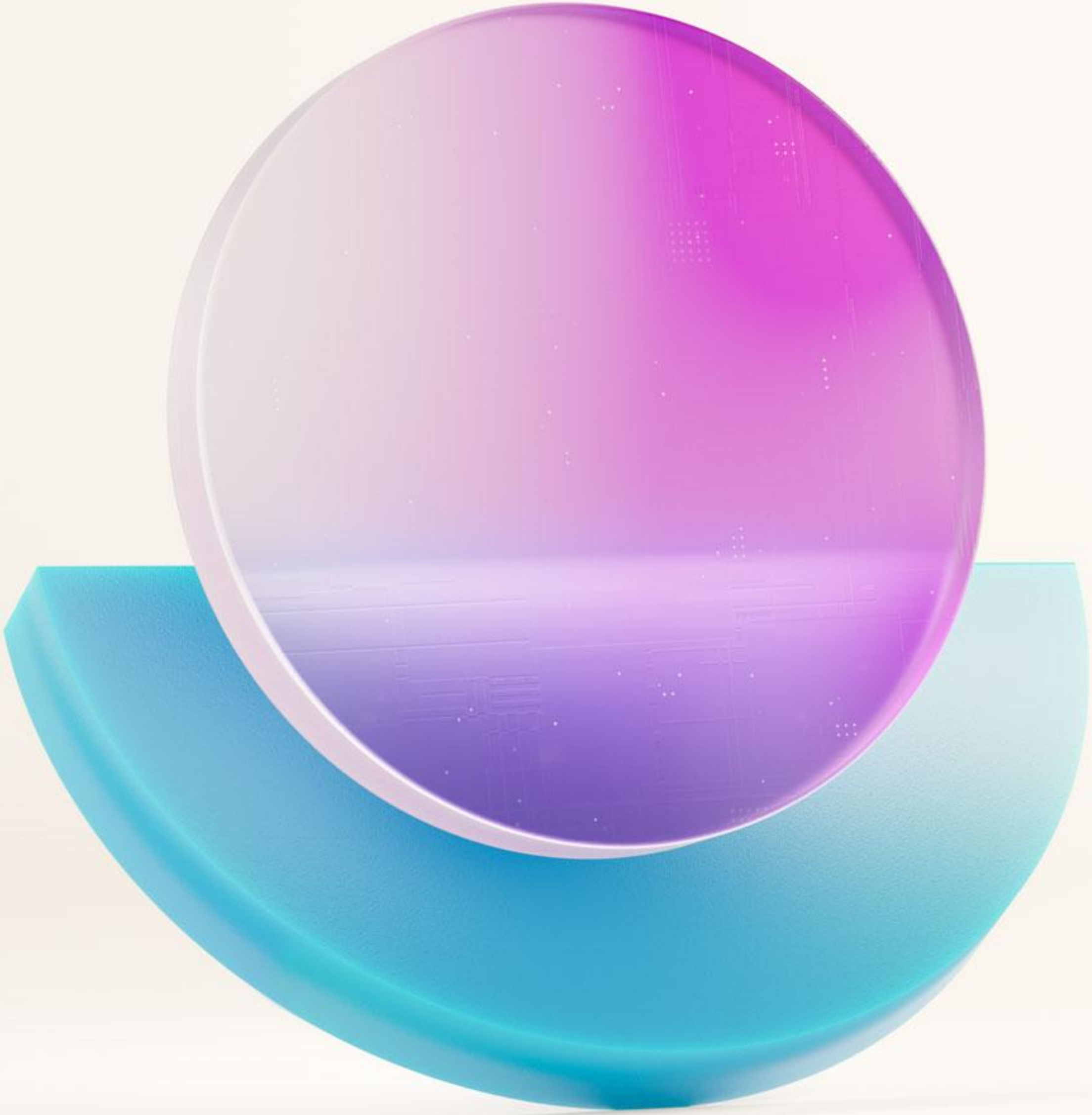


```
SELECT [name], [is_vorder_enabled] FROM sys.databases
```

	ABC name	0/1 is_vorder_enabled
1	master	0
2	Sales warehouse	1
3	Marketing warehouse	1
4	Staging warehouse	0

Succeeded (1 sec 447 ms)

Columns: 2 Rows: 4



Optimizing queries

Managing Statistics

Statistics are objects that contain relevant information about your data. They enable the query optimizer to estimate a query plan with the least system overhead (I/O, CPU) possible, resulting in faster query execution.

Automatic statistics - RECOMMENDED

- Engine automatically creates and maintains statistics at query time
- Because these automatic operations are done synchronously, expect the query duration to include this time if the needed statistics do not yet exist, or significant data changes have happened since the last statistics refresh

User-defined statistics

- The user manually uses data definition language (DDL) syntax to create, update, and drop statistics as needed
- Consider updating column-level statistics regularly after data updates that significantly change the row count or distribution of the data

Manually managing statistics

If creating statistics manually, consider focusing on columns heavily used in your query workload (specifically in GROUP BYs, ORDER BYs, filters, and JOINs). Examples:

Create statistics for the CustomerKey column of a table:

```
CREATE STATISTICS DimCustomer_CustomerKey_FullScan
 ON dbo.DimCustomer (CustomerKey) WITH FULLSCAN;
```

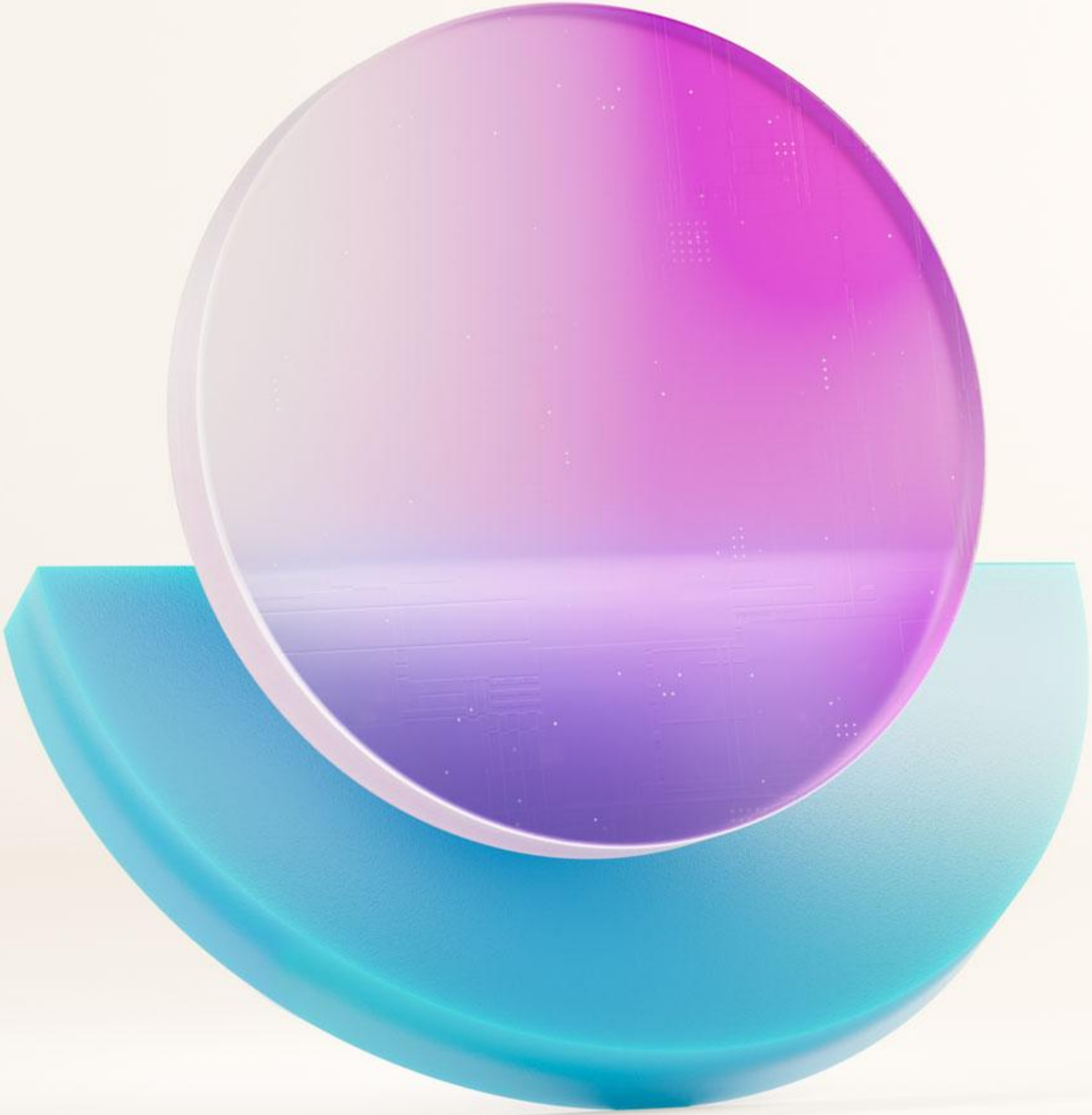
Update statistics for the statistics object previously created (perhaps after a large data update):

```
UPDATE STATISTICS dbo.DimCustomer (DimCustomer_CustomerKey_FullScan) WITH FULLSCAN;
```

To show information about the histogram of the statistics object:

```
DBCC SHOW_STATISTICS ("dbo.DimCustomer", "DimCustomer_CustomerKey_FullScan") WITH HISTOGRAM;
```

Microsoft Fabric
Community Conference

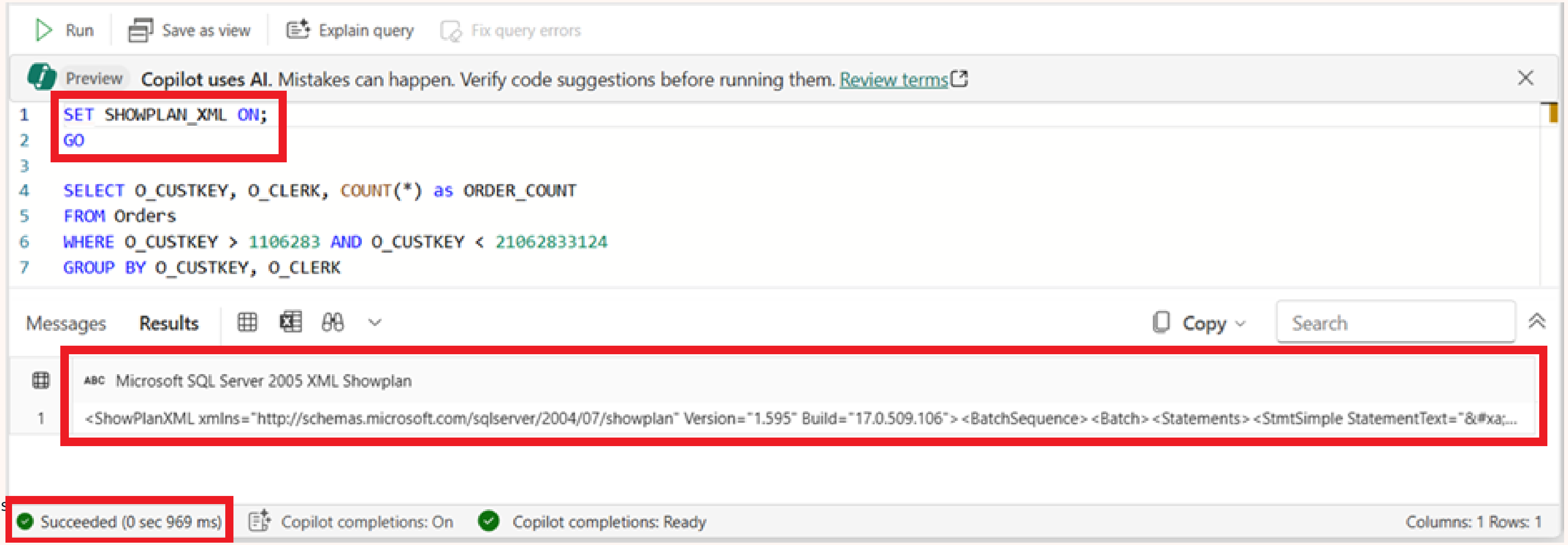


Demo

Obtaining the query plan (preview)

Now in preview: `SET SHOWPLAN_XML { ON | OFF }`

- Using SET SHOWPLAN_XML helps you analyze and optimize complex queries without executing them, ensuring you can make informed decisions about query tuning.
- Fabric Data Warehouse returns detailed information about with the query plan for query execution.



Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. Review terms

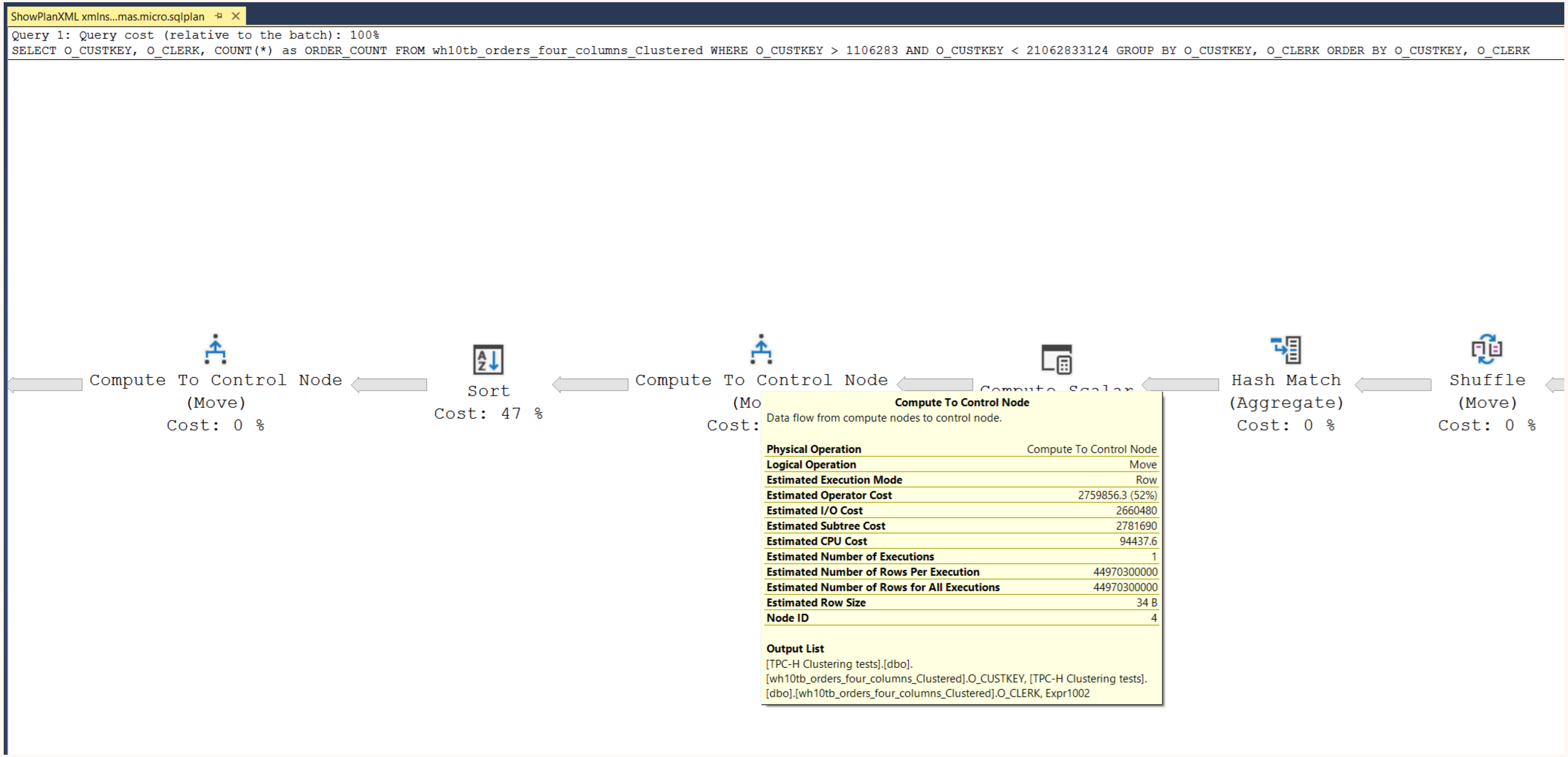
```sql
SET SHOWPLAN_XML ON;
GO

SELECT O_CUSTKEY, O_CLERK, COUNT(*) as ORDER_COUNT
FROM Orders
WHERE O_CUSTKEY > 1106283 AND O_CUSTKEY < 21062833124
GROUP BY O_CUSTKEY, O_CLERK
```

	Microsoft SQL Server 2005 XML Showplan
1	<ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan" Version="1.595" Build="17.0.509.106"><BatchSequence><Batch><Statements><StmtSimple StatementText="&#xa;...

Succeeded (0 sec 969 ms) Copilot completions: On Copilot completions: Ready Columns: 1 Rows: 1

Obtaining the query plan (preview) (cont'd)

Query plans can be visualized in SQL Server Management Studio



```
Query 1: Query cost (relative to the batch): 100%
SELECT O_CUSTKEY, O_CLERK, COUNT(*) as ORDER_COUNT FROM wh10tb_orders_four_columns_Clustered WHERE O_CUSTKEY > 1106283 AND O_CUSTKEY < 21062833124 GROUP BY O_CUSTKEY, O_CLERK ORDER BY O_CUSTKEY, O_CLERK
```

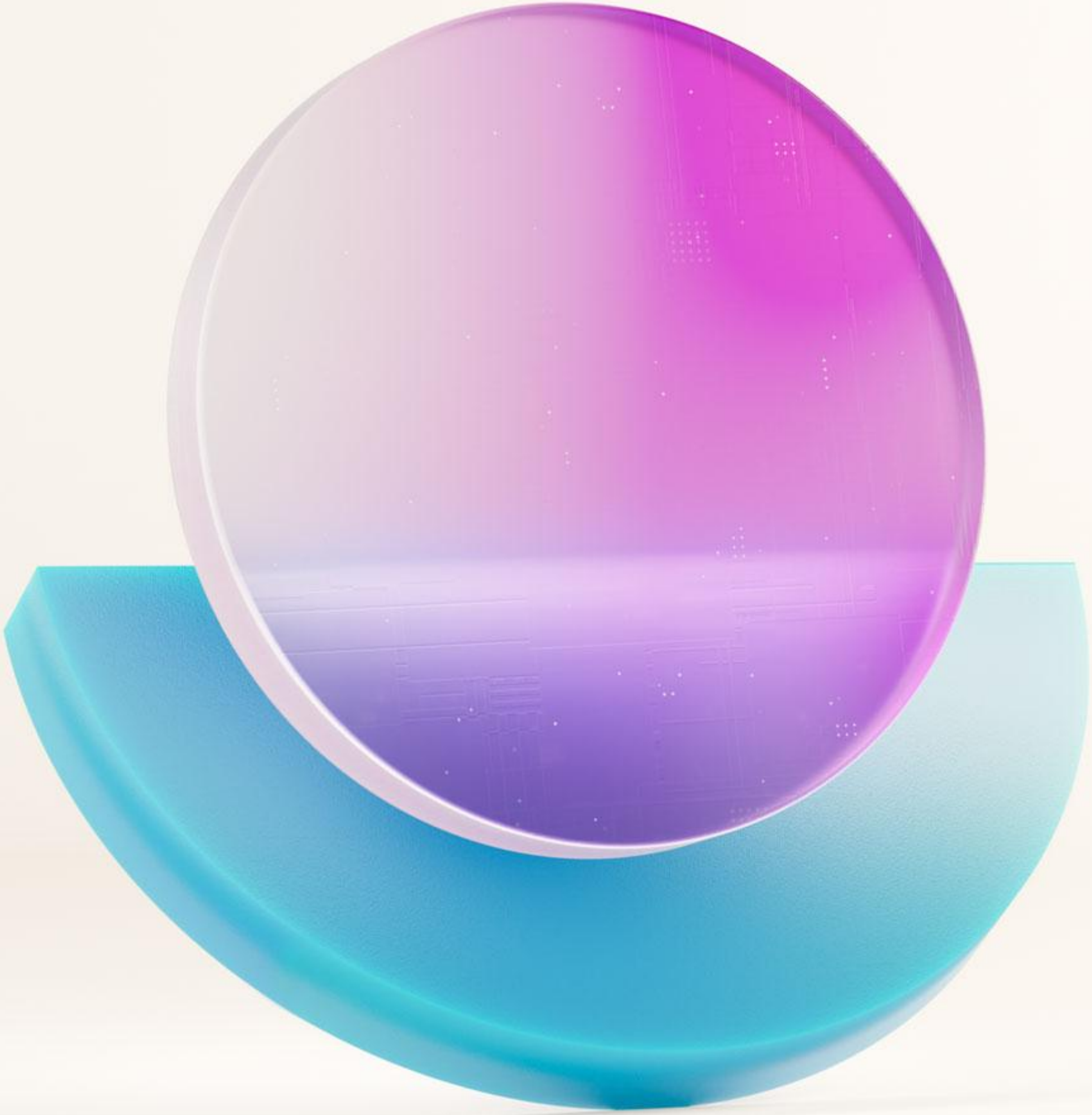
Compute To Control Node

Data flow from compute nodes to control node.

Physical Operation	Compute To Control Node
Logical Operation	Move
Estimated Execution Mode	Row
Estimated Operator Cost	2759856.3 (52%)
Estimated I/O Cost	2660480
Estimated Subtree Cost	2781690
Estimated CPU Cost	94437.6
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	44970300000
Estimated Number of Rows for All Executions	44970300000
Estimated Row Size	34 B
Node ID	4

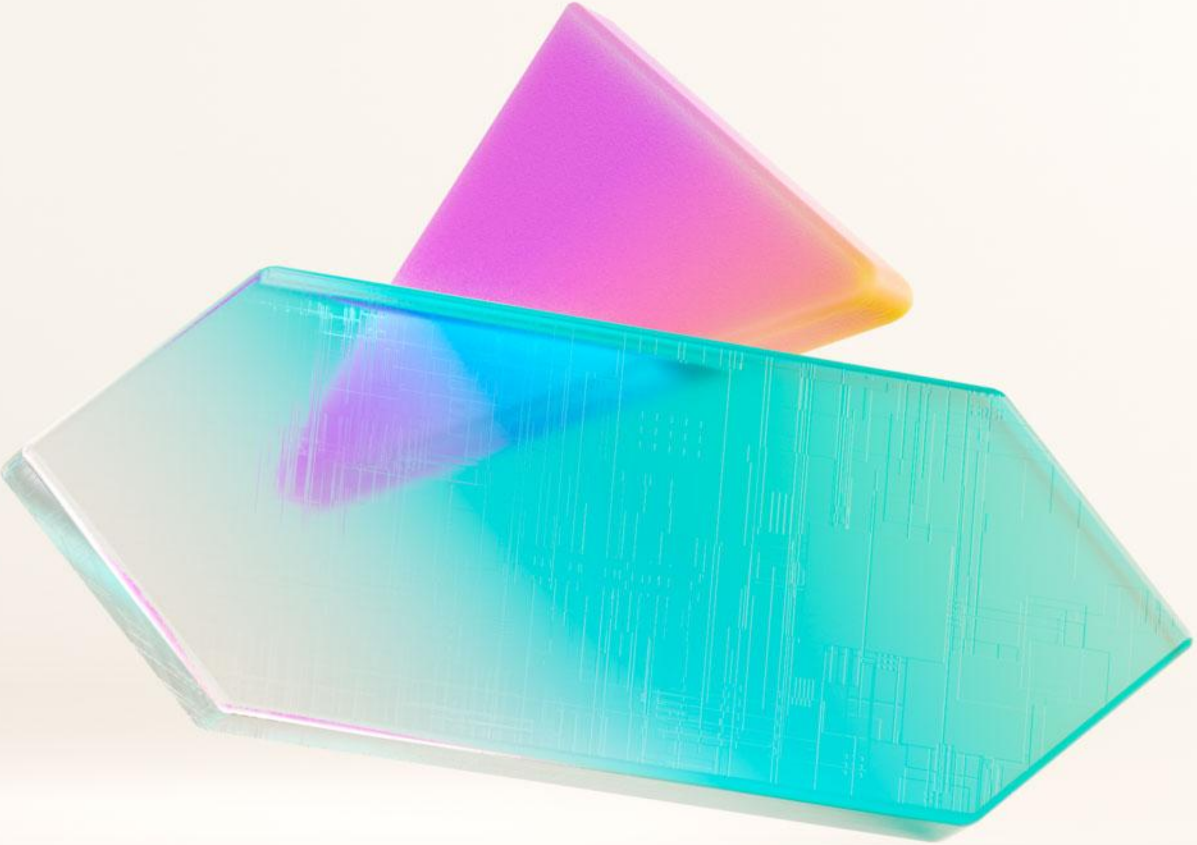
Output List

[TPC-H Clustering tests].[dbo].[wh10tb_orders_four_columns_Clustered].O_CUSTKEY, [TPC-H Clustering tests].[dbo].[wh10tb_orders_four_columns_Clustered].O_CLERK, Expr1002



Demo

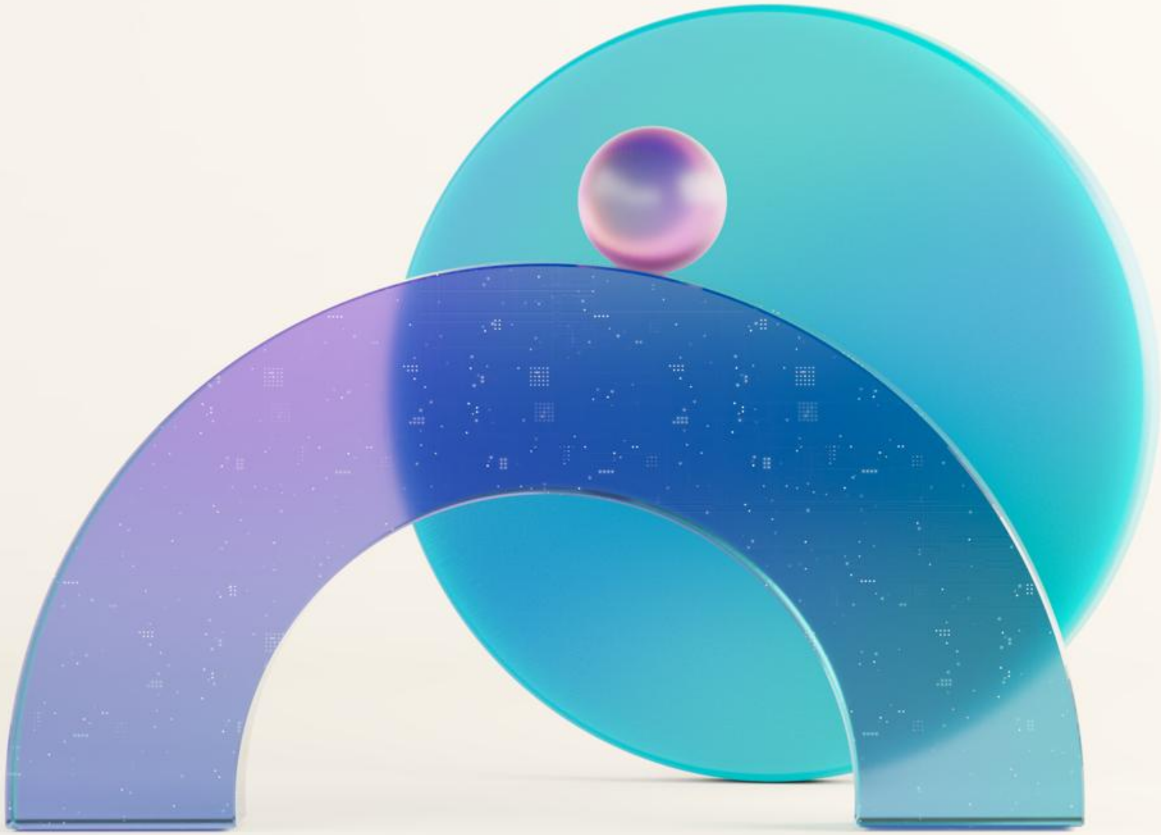
Microsoft Fabric
COMMUNITY CONFERENCE



10-minute break

Microsoft

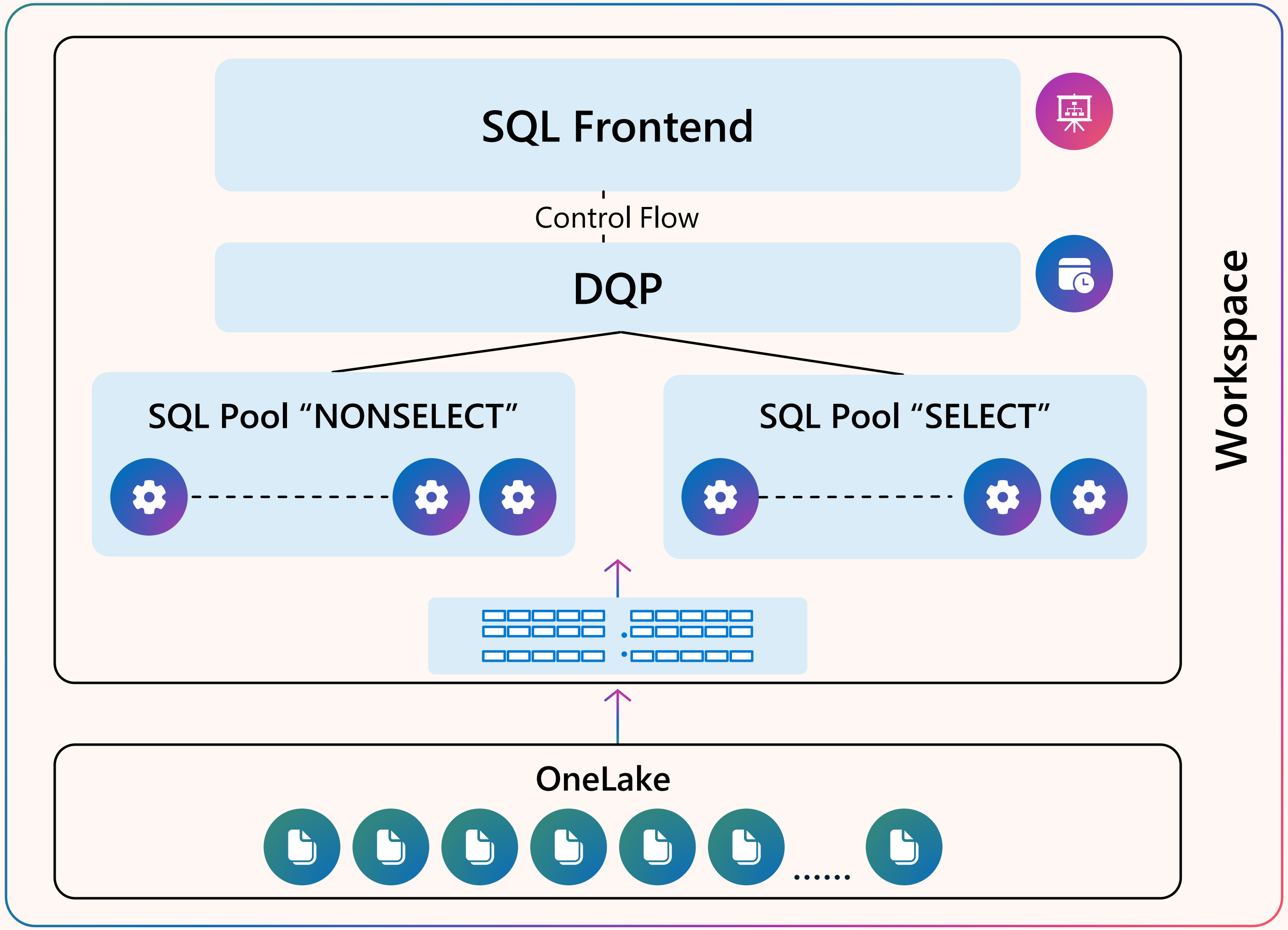
Microsoft Fabric
Community Conference



Workload Management

Module 8

Intelligent Workload Management



Physical isolation exists between NONSELECT (ETL workloads) and SELECT (analytics/reporting workloads)

Maximum burstable capacity is split evenly between these two resource pools.

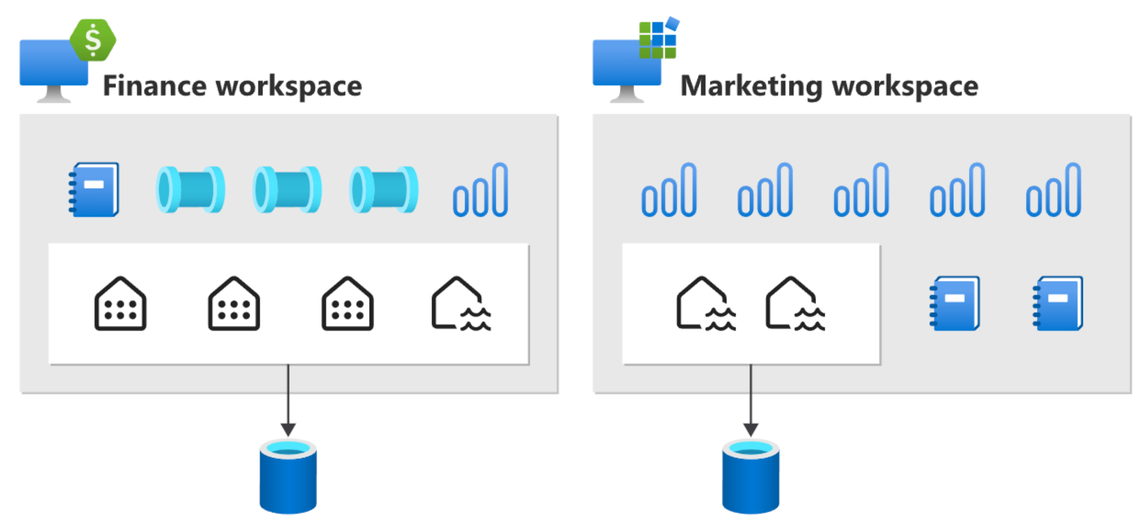
Burstable Compute is available when needed, up to a SKU specified maximum

Workspace isolation for different workloads

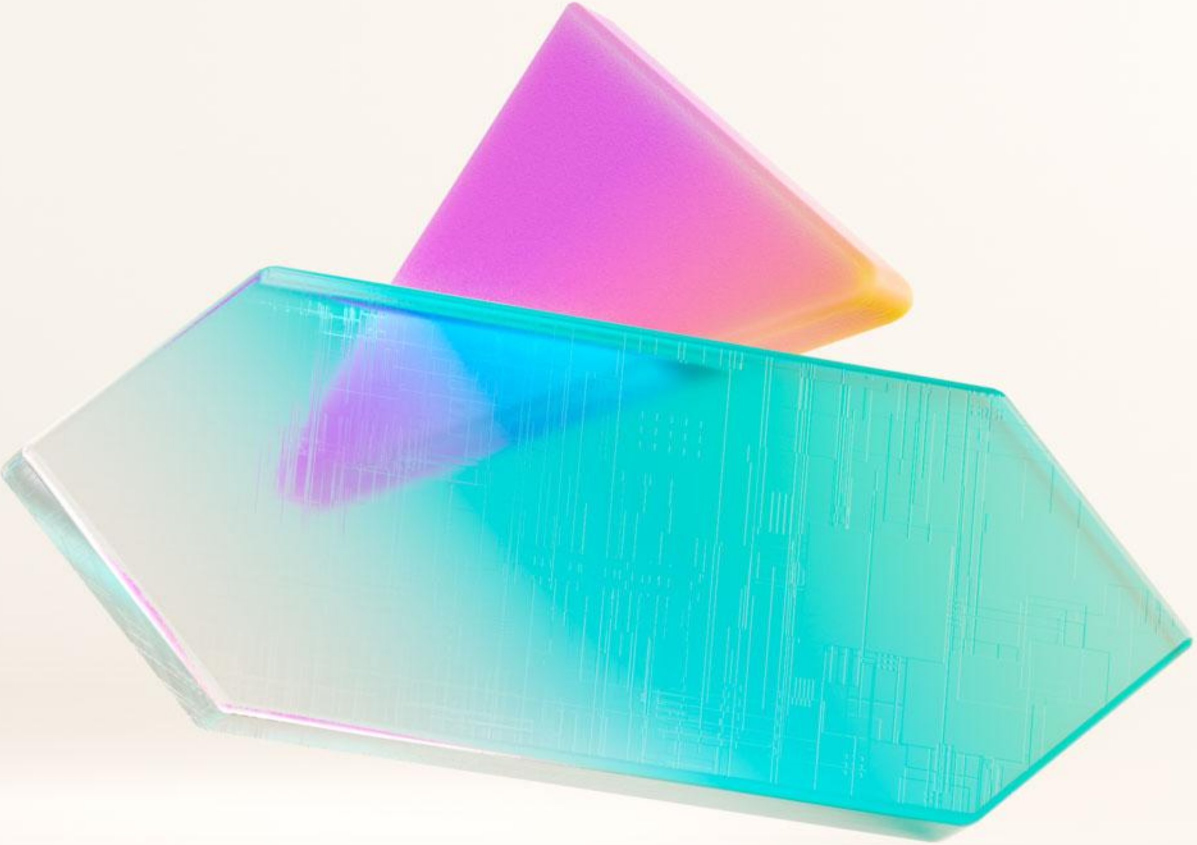
Microsoft Fabric workspaces provides a natural isolation boundary of the distributed compute system. Workloads can take advantage of this boundary to manage both cost and performance.

OneLake shortcuts can be used to create read-only replicas of tables in other workspaces to distribute load across multiple SQL engines, creating an isolation boundary.

This can effectively increase the maximum number of sessions performing read-only queries.



Microsoft Fabric
COMMUNITY CONFERENCE



Security

Module 9

Permission model

Microsoft Fabric permissions and granular SQL permissions work together to govern Warehouse access and the user permissions once connected

Workspace roles provide Microsoft Fabric permissions for all warehouses within a workspace

Fabric capability	Admin	Member	Contributor	Viewer
Delete the workspace	✅	❌	❌	❌
Add admins	✅	❌	❌	❌
Add members	✅	✅	❌	❌
Write data	✅	✅	✅	❌
Create items	✅	✅	✅	❌
Read data	✅	✅	✅	✅

Learn more about workspace roles on https://learn.microsoft.com/en-us/fabric/fundamentals/roles-workspaces

Workspace roles in Fabric Data Warehouse

Assigning users to the various workspace roles uses the following SQL permissions

Workspace role	Description
Admin	Grants the user **CONTROL** access for each Warehouse and SQL analytics endpoint within the workspace, providing them with full read/write permissions and the ability to manage granular user SQL permissions. Allows the user to see workspace-scoped session, monitor connections and requests in DMVs via TSQL, and KILL sessions.
Member	Grants the user **CONTROL** access for each Warehouse and SQL analytics endpoint within the workspace, providing them with full read/write permissions and the ability to manage granular user SQL permissions. Share all content within the warehouse.
Contributor	Grants the user **CONTROL** access for each Warehouse and SQL analytics endpoint within the workspace, providing them with full read/write permissions and the ability to manage granular user SQL permissions.
Viewer	Grants the user **CONNECT** and ReadData permissions for each Warehouse and SQL analytics endpoint within the workspace. Viewers have SQL permissions to read data from tables/views using T-SQL.

Granular security on SQL objects

- Object-level-security can be managed using GRANT, REVOKE, and DENY T-SQL syntax
- Users can be assigned to SQL roles, both custom and built-in database roles

To grant users access to database objects:

```
GRANT EXECUTE ON OBJECT::dbo.UpdateDimCity TO [roger@contoso.com]
```

Seamlessly, to deny access:

```
DENY EXECUTE ON OBJECT::dbo.UpdateDimCity TO [roger@contoso.com]
```

Row-level security

- Enables you to use group membership or execution context to control access to rows in a database table.
- Examples:
  - Ensure that workers access only those data rows that are pertinent to their department
  - Allow sales managers access to a global sales table, but only to rows pertinent to their region

SaleID	CustomerID	TotalAmmount	Date	Region
1001	3345	121.00	3/29/2025	Southeast
1002	2561	629.00	3/29/2025	Southeast

Row-level security

```
-- Creating a function for the SalesRep evaluation
CREATE FUNCTION Security.tvf_securitypredicate(@SalesRep AS nvarchar(50))
    RETURNS TABLE
WITH SCHEMABINDING
AS
    RETURN SELECT 1 AS tvf_securitypredicate_result
WHERE @SalesRep = USER_NAME() OR USER_NAME() = 'manager@contoso.com';

-- Using the function to create a Security Policy
CREATE SECURITY POLICY SalesFilter
ADD FILTER PREDICATE Security.tvf_securitypredicate(SalesRep)
ON sales.Orders
WITH (STATE = ON);
```

Column-level security

- Enables you to use group membership or execution context to control access to columns in a database table.
- Examples:
  - Ensure that workers can’t see sensitive information about a customer (e.g.: SSN, credit card number)
  - Allow HR personnel to see employee names, but hide salaries

🔒

First name	Last name	Job role	Manager ID	Salary
Dany	Bélisle	Data Engineer	627819	
Camelia	Banica	Data Analyst	726541	
Emil	Moldovan	SQL Developer	726541	
Ioana	Costache	Data guru	019275	

Column-level security

```
GRANT SELECT ON Employees(FirstName, LastName, JobRole, ManagerID) TO
[charlie@contoso.com]
```

Attempts to read the Salary column will fail:

```
SELECT * FROM Employees
```

```
Msg 230, Level 14, State 1, Line 12
The SELECT permission was denied on the column 'Salary' of the object
‘Employees', database ‘HumanResources', schema 'dbo'.
```

Dynamic data masking

- Limits sensitive data exposure by masking it to nonprivileged users. It can be used to greatly simplify the design and coding of security in your application
- The data isn't changed, so it can be used with existing applications since masking rules are applied to query results
- Minimal (if any) effect in the application layer

First name	Last name	E-mail	Phone	Credit Card Number
Dany	Bélisle	dXXXXX@XXXXX.com	(425) ###-1254	XXXX-XXXX-XXXX-1234
Camelia	Banica	cXXXXX@XXXXX.com	(425) ###-7284	XXXX-XXXX-XXXX-4321
Emil	Moldovan	eXXXXX@XXXXX.com	(425) ###-8271	XXXX-XXXX-XXXX-0123
Ioana	Costache	IXXXXX@XXXXX.com	(425) ###-0021	XXXX-XXXX-XXXX-3210

Dynamic data masking

```
CREATE TABLE dbo.EmployeeData (
    EmployeeID INT
    ,FirstName VARCHAR(50) MASKED WITH (FUNCTION = 'partial(1,"-",2)') NULL
    ,LastName VARCHAR(50) MASKED WITH (FUNCTION = 'default()') NULL
    ,SSN CHAR(11) MASKED WITH (FUNCTION = 'partial(0,"XXX-XX-",4)') NULL
    ,email VARCHAR(256) NULL);
```

In this example:

- The **FirstName** column shows only the first and last two characters of the string, with **-** in the middle.
- The **LastName** column shows **XXXX**.
- The **SSN** column shows **XXX-XX-** followed by the last four characters of the string.