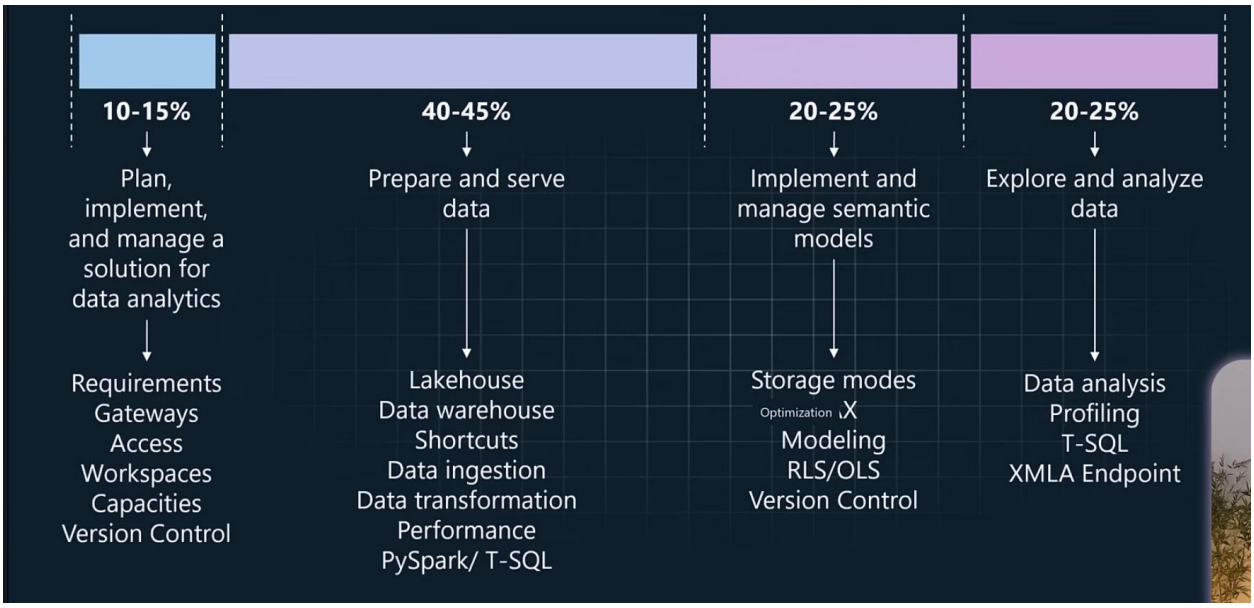
Fabric Analytics Concepts Notes

Fabric Analytics Concepts

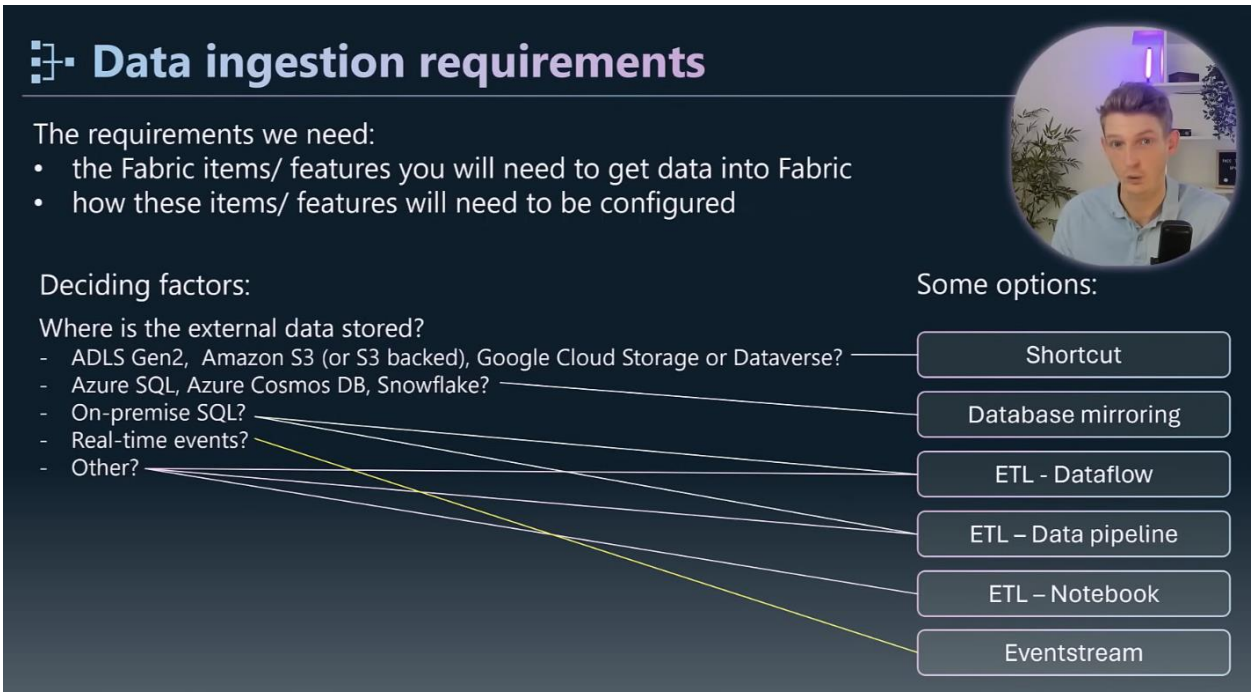
Fabric Analytics Concepts Notes

Ch1. Intro



S

Ch2. Plan Data Analytics Env



Fabric Analytics Concepts Notes

Data ingestion requirements

The requirements we need:

- the Fabric items/ features you will need to get data into Fabric
- how these items/ features will need to be configured

Deciding factors:

How is the data secured?

- On-premise SQL
- Azure virtual network / private endpoint

What is the volume of the data?

- Low (megabytes per day)
- Medium (gigabytes per day)
- High (many GB or terabytes per day)

Some features:

- On-premise data gateway
- VNet data gateway
- Fast copy
- Staging

Data gateways

Before we move on, let's explore data gateways in more detail

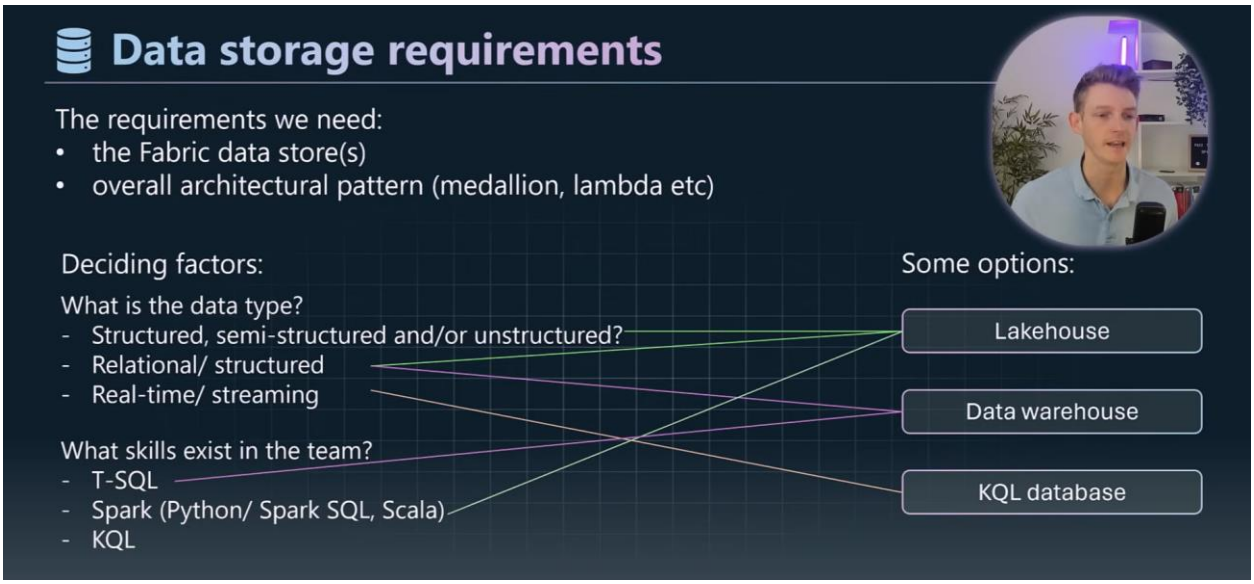
On-premise data gateway

1. Install the data gateway on the on-premise server (or update to Fabric version if already exists)
2. In Fabric, create a new On-Premise Data Gateway connection
3. Use the gateway in a Dataflow or Data Pipeline to get data into Fabric

Virtual network (VNet) data gateway

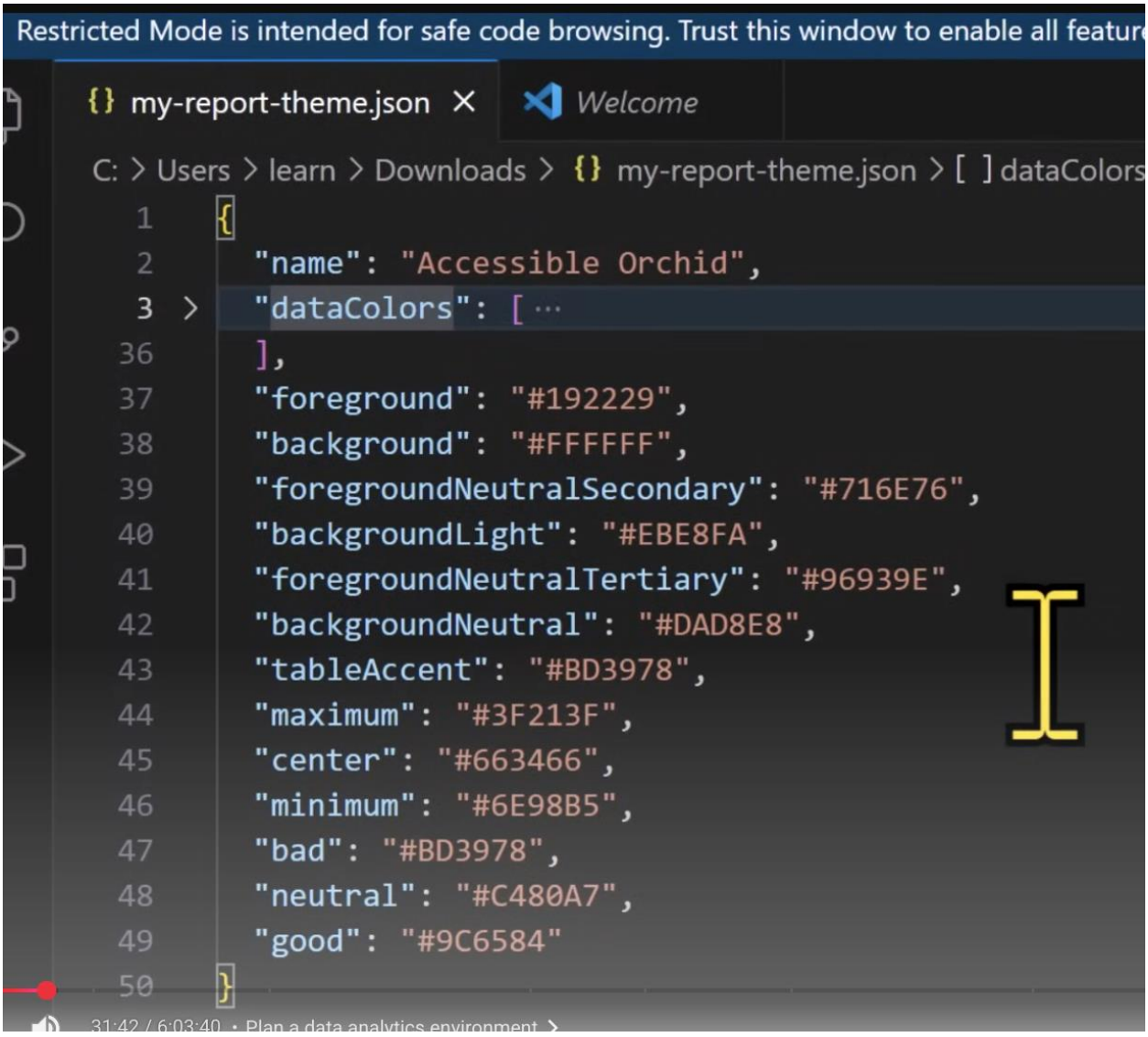
1. Set network configuration in Azure.
    1. Register Power Platform resource provider
    2. Create a private endpoint on your Azure object
    3. Create a subnet
2. In Fabric, create a new virtual network Data Gateway connection
3. Use the gateway in a Dataflow or Data Pipeline to get data into Fabric

Fabric Analytics Concepts Notes



PBI JSON Theme

Fabric Analytics Concepts Notes



Fabric Analytics Concepts Notes

Question
You are running an F2 capacity and you regularly experiencing throttling.

There are many long-running Spark jobs take on average 3 hours to complete. You need these to complete in under one hour, so you plan to increase the SKU of the capacity.

Where would go to go to make this change?

A) Go to Workspace Settings > Spark settings

B) Go to Admin Portal > Capacity Settings and click through to Azure to update your capacity

C) Go to the monitoring hub, and look at the run history.

D) Use the Capacity Metrics app

Question

Your data governance team would like to certify a Semantic Model to make it discoverable in the organisation.

Only the data governance team should be able to do this.

In what order should you complete the following tasks to certify a Semantic Model?

1) Create a security group for the data governance team
2) Enable the 'Make certified content discoverable' in Admin Portal > Tenant Settings > Discovery Settings
3) Make sure the 'Make certified content discoverable' setting applies only to the Data Governance security group.
4) Ask the data governance team to go to the Semantic Model Settings > Endorsement and Discovery, and click Certify
5) Ask a business user to visit the OneLake Data Hub to validate they can see the semantic model.

Question
You join a new company and are given a Power BI report theme (JSON file) to use for all new projects.

How do you apply this JSON file theme to the report you are developing?

A) In Power BI Desktop, go to View > Themes > Customize current theme

B) Go to Fabric Admin Portal > Custom branding and set the Default Report Theme

C) Use Tabular Editor 2 to update the theme.

D) In Power BI Desktop, go to View > Themes > Browse for themes

Fabric Analytics Concepts Notes

- **Azure Data Lake Storage (ADLS)**
    - A scalable, secure **cloud storage service** optimized for big data analytics.
    - Stores structured/unstructured data (e.g., files, logs, images) in a hierarchical namespace.
    - Acts as a *data lake* for raw data ingestion and batch/stream processing.
- **Microsoft Fabric**
    - An **end-to-end analytics platform** unifying data engineering, warehousing, science, and BI.
    - Integrates compute (Spark, SQL), storage (OneLake), and services (Data Factory, Power BI) into a single SaaS solution.
    - Built on **OneLake** (a unified data lake for all Fabric workloads).

Feature	ADLS	Microsoft Fabric
Storage	Hierarchical storage (Gen1/Gen2)	OneLake (ADLS Gen2-based, unified for all Fabric workloads)
Compute	Requires external services (e.g., Databricks, Synapse)	Built-in Spark, SQL, and real-time analytics
Data Integration	Needs Azure Data Factory/Synapse	Built-in Data Factory (Data Pipeline)
Governance	Manual setup (RBAC, ACLs)	Centralized governance (Purview integration)
Data Warehousing	Not natively supported	Built-in **Warehouse** (SQL-based)
BI & Visualization	Requires Power BI	Native Power BI integration
Pricing	Pay-as-you-go (storage + egress)	Unified capacity-based pricing

Fabric Analytics Concepts Notes

3. When to Use Which?

- **Choose ADLS if you need**:
  - A pure storage layer for raw data.
  - Flexibility to use external compute tools (e.g., Databricks, Synapse).
  - Long-term archival or low-cost storage.
- **Choose Fabric if you need**:
  - An all-in-one platform for analytics (storage + compute + BI).
  - Simplified collaboration (OneLake, shared datasets).
  - AI/ML integration (e.g., Synapse Data Science).

---

4. Integration

- **Fabric uses ADLS Gen2 under the hood** (via OneLake).
- You can **connect ADLS to Fabric** for existing data lakes.

Example Workflow:

1. **ADLS**: Ingest raw data.
2. **Fabric**: Transform (Spark), model (Warehouse), and visualize (Power BI).

Question
You have 1000 JSON files stored in Azure Data Lake Storage (ADLS) Gen2 that you want to bring into Fabric.

The ADLS Gen2 storage is secured using a virtual network.

Which of these actions would you need to perform FIRST?

A) In Fabric, go to 'Manage connections and gateways' and then click 'Create a virtual network data gateway'.

B) Create a shortcut to the ADLS Gen2 storage account.

C) **In Azure, register a new Resource Provider, create a Private Endpoint and subnet.**

D) Install an on-premise data gateway on an Azure Virtual Machine in the same virtual network.

E) Enable public access in the storage account network settings

To securely bring 1000 JSON files from **Azure Data Lake Storage (ADLS) Gen2 (secured via a virtual network)** into **Microsoft Fabric**, the **first step** is:

Fabric Analytics Concepts Notes

Correct Answer: C)

"In Azure, register a new Resource Provider, create a Private Endpoint and subnet."

Why?

1. **ADLS Gen2 is secured via a virtual network** (private, no public access).
2. To access it from Fabric, you need a **Private Endpoint** to establish a secure, private connection between Fabric and ADLS *without* exposing data to the public internet.
3. This requires:
    - Registering the **Microsoft.Synapse** Resource Provider (if not already done).
    - Creating a **Private Endpoint** in Azure to link Fabric to the ADLS virtual network.
    - Configuring a **subnet** for the endpoint.

Why Not the Others?

- **A)** Virtual Network Data Gateway is for **on-premises** data sources, not Azure-native services like ADLS.
- **B)** Shortcuts require connectivity *first* (which the Private Endpoint enables).
- **D)** On-premises gateways are unnecessary for Azure-to-Azure connections.
- **E)** Enabling public access violates security best practices (question states the storage is private).

Steps to Implement (After Option C):

1. Create the Private Endpoint in Azure.
2. In Fabric, create a **shortcut** to ADLS Gen2 (using the private connection).
3. Ingest JSON files via Spark or Data Pipeline.

This ensures secure, network-compliant data transfer.

Fabric Analytics Concepts Notes

Question
You have data stored in tables in Snowflake.

Which of the following CANNOT be used to bring the data into Fabric?

A) Use the Data Pipeline CopyData activity.

B) **Create a Shortcut to the Snowflake tables from your Lakehouse.**

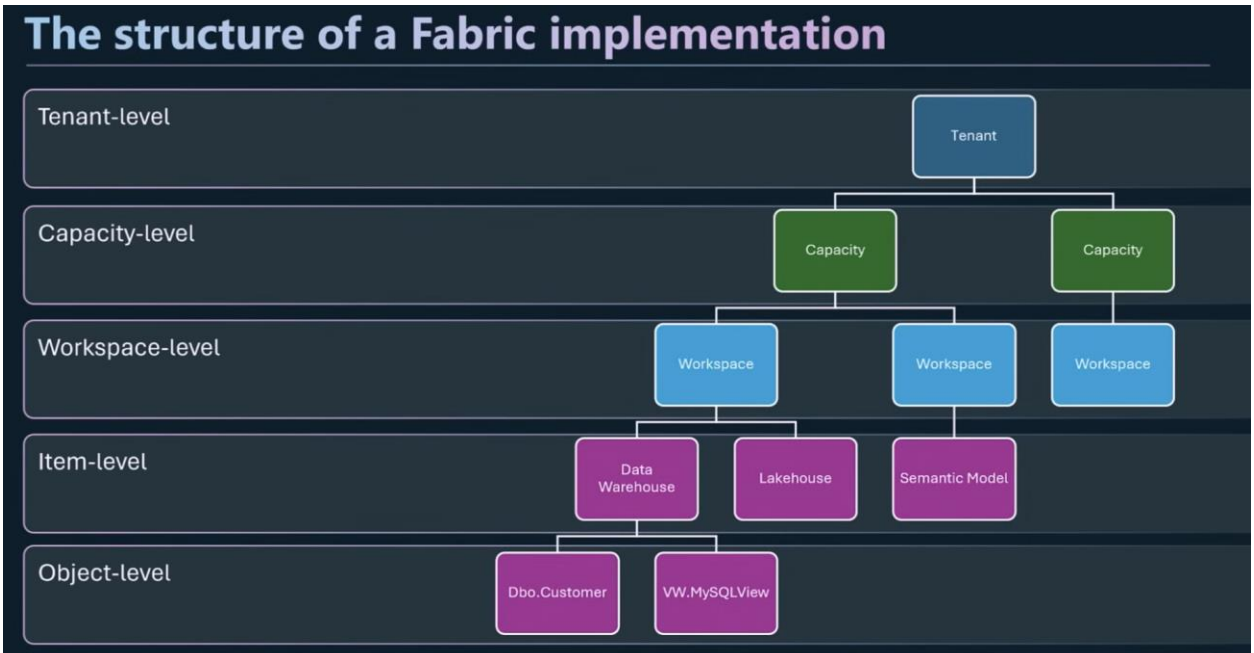
C) Use a Dataflow Gen2 with a Snowflake connection.

D) Use database mirroring to create a mirrored Snowflake database in Fabric.

In Microsoft Fabric, **shortcuts** are currently supported only for **OneLake data storage locations** like Azure Data Lake or other Lakehouses.
Snowflake is not a supported source for shortcuts — you cannot directly create a shortcut to a Snowflake table from a Lakehouse.

- **A) Use the Data Pipeline CopyData activity**
  ✅ This is a standard way to copy data from Snowflake into Fabric via data pipelines.
- **C) Use a Dataflow Gen2 with a Snowflake connection**
  ✅ Dataflows Gen2 support Snowflake connectors, enabling data ingestion and transformation.
- **D) Use database mirroring to create a mirrored Snowflake database in Fabric**
  ✅ Fabric supports **database mirroring** for Snowflake as one of its real-time data integration methods.

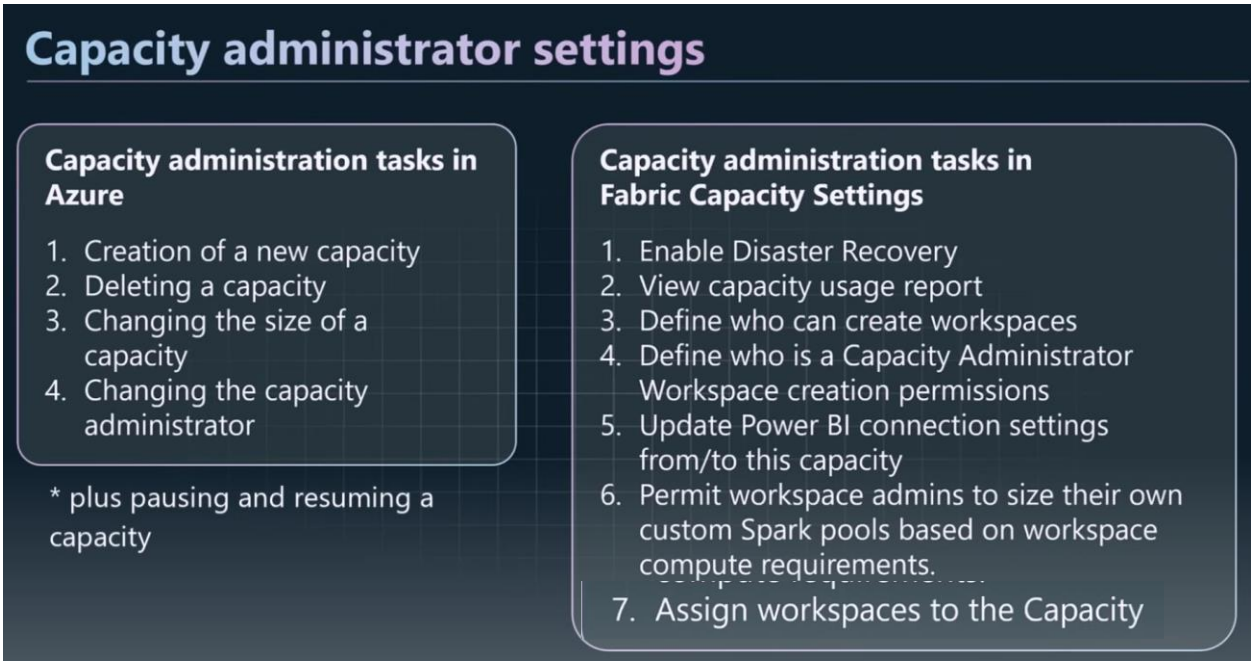
Ch3. Implement & Manage a Data Analytics Env



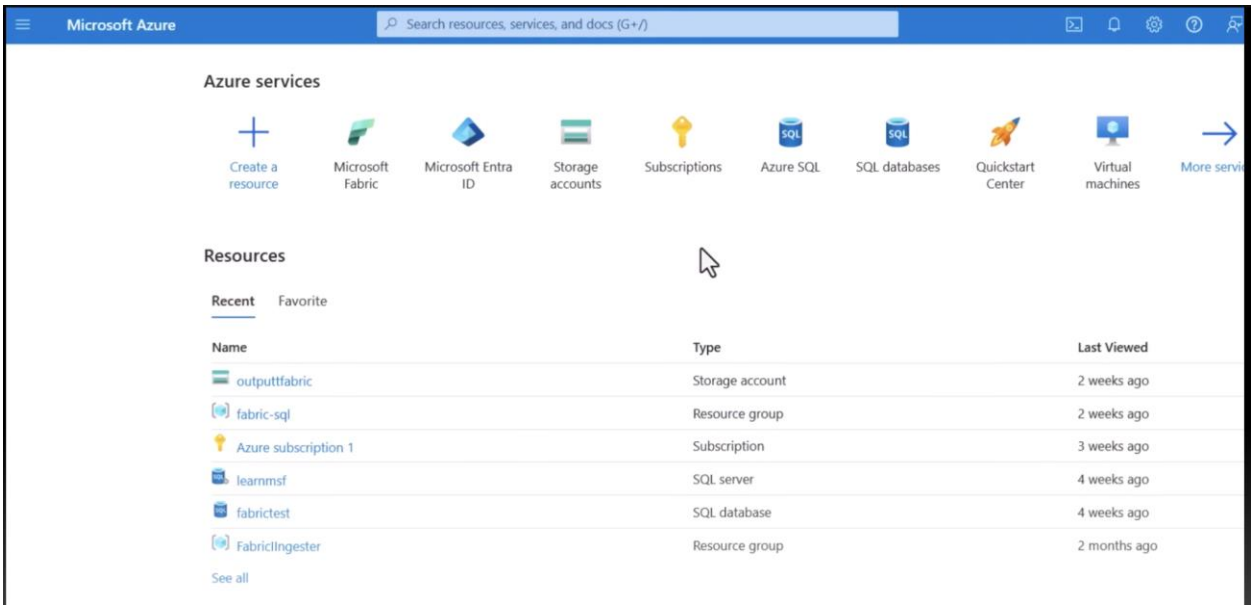
Fabric administration

Level	Administrator	Role required	Where admin happens	
Tenant-level Tenant	Tenant Admin	Entra ID role of Global administrator, Power Platform administrator or Fabric administrator	Fabric Admin Portal (Tenant Settings)	→ Last lesson
Capacity-level Capacity	Capacity Admin	Assigned in Azure when provisioning the capacity (must be person or Service Principal in Entra ID tenant) – can also be updated in Azure Portal	Azure Portal & Fabric Admin Portal (Capacity Settings)	→ This lesson
Workspace-level Workspace	Workspace Admin	Person or group with the Workspace Role of 'Admin'	Workspace Settings (in the workspace)	→ This lesson

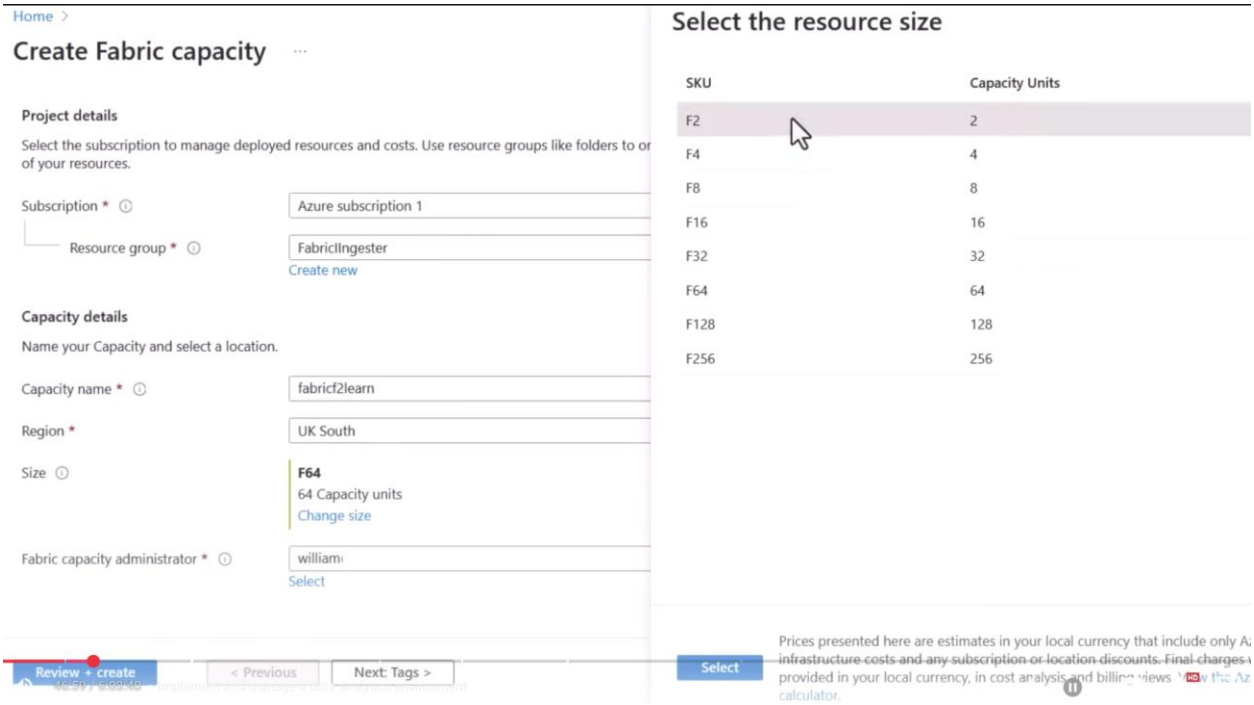
Fabric Analytics Concepts Notes



Capacity settings in Azure Portal



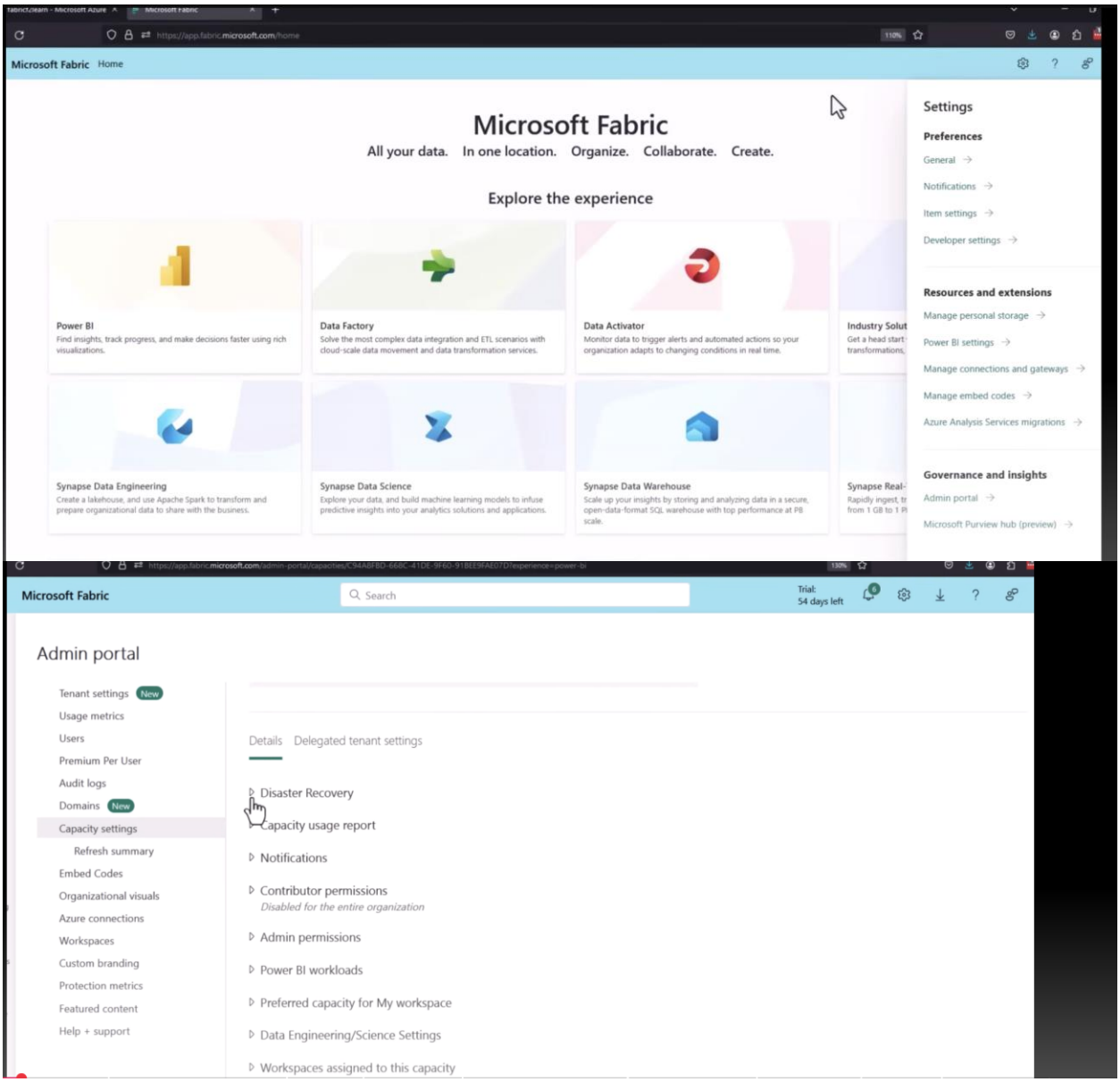
Fabric Analytics Concepts Notes



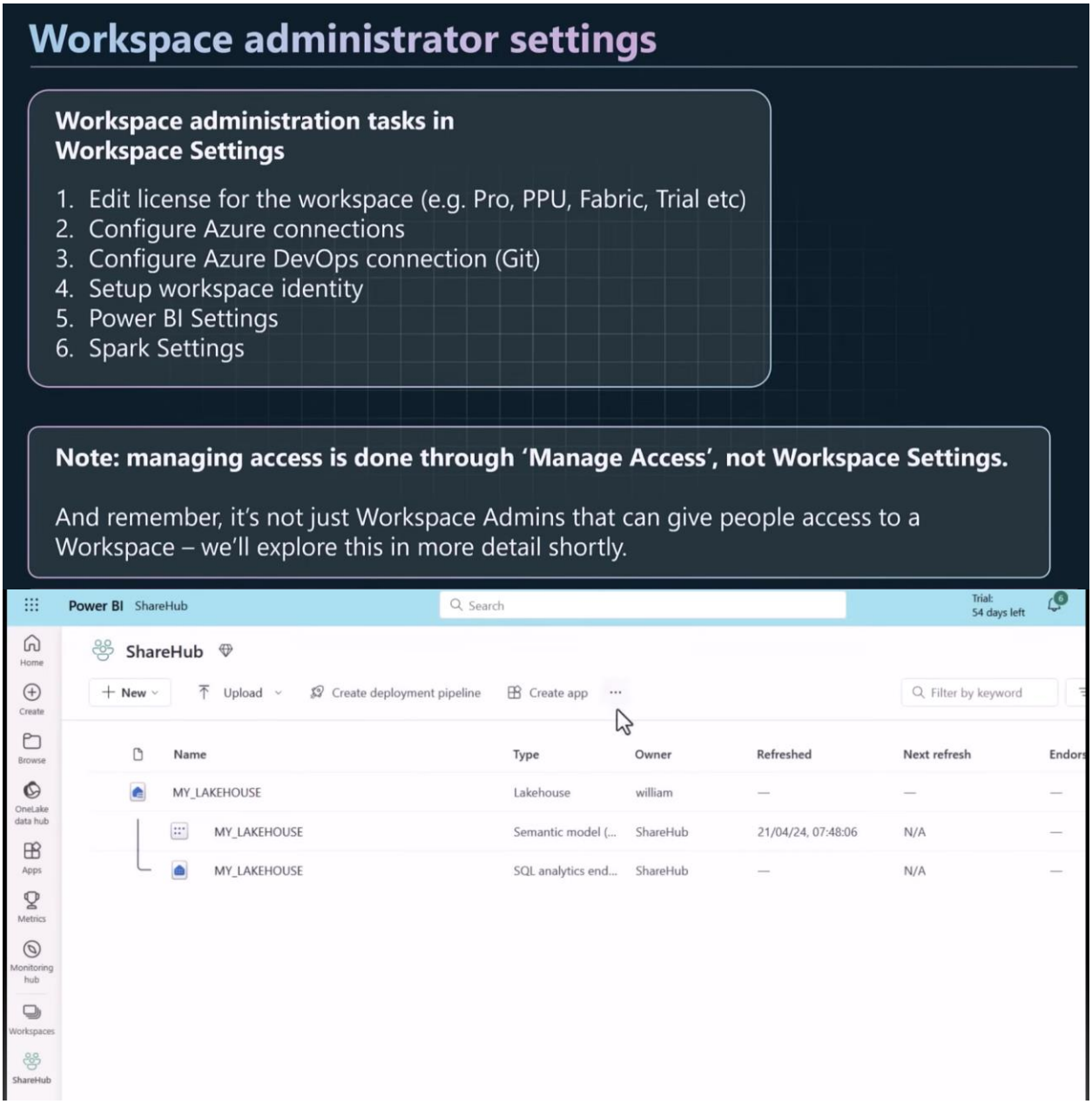
Costs for F2

Capacity settings in Fabric Portal

Fabric Analytics Concepts Notes



Fabric Analytics Concepts Notes

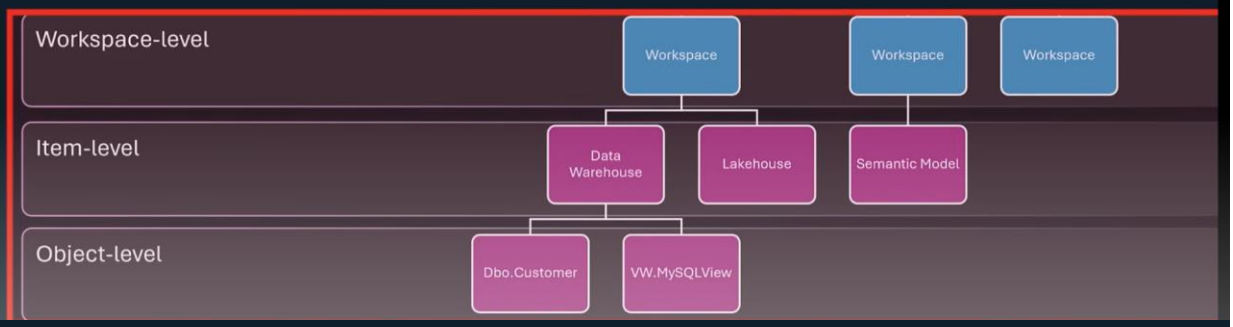


Fabric Analytics Concepts Notes

The structure of a Fabric implementation

Sharing things in Fabric can be done at one of these three levels.

(Object-level sharing is not assessed as part of the exam)



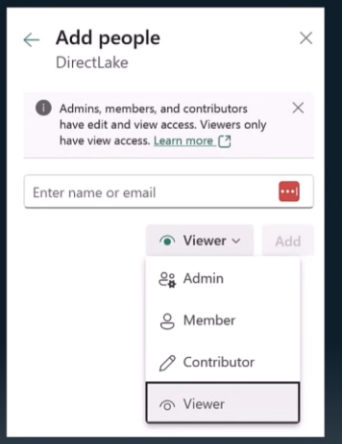
Workspace-level sharing

People or groups can be given workspace-level access.

When sharing, the person or group is assigned a workspace role:

- Admin
- Member
- Contributor
- Viewer

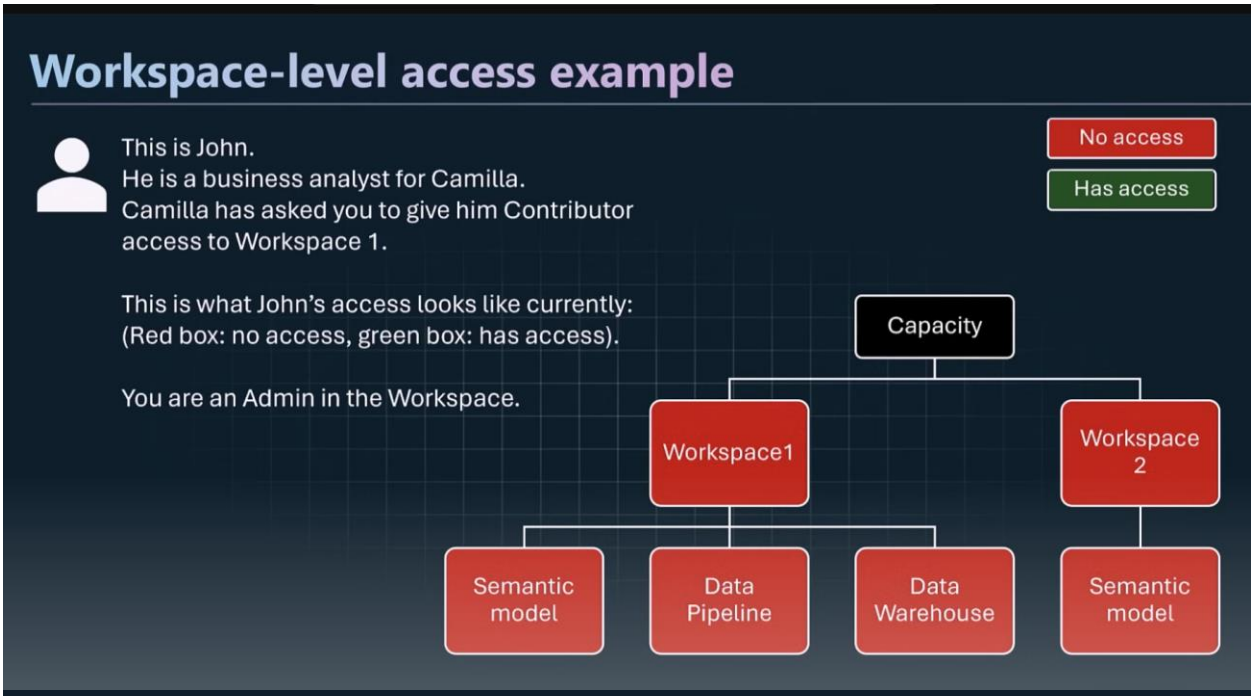
This role applies to **all items in the workspace**. For example a Viewer in the workspace will be able to View all items in the workspace.

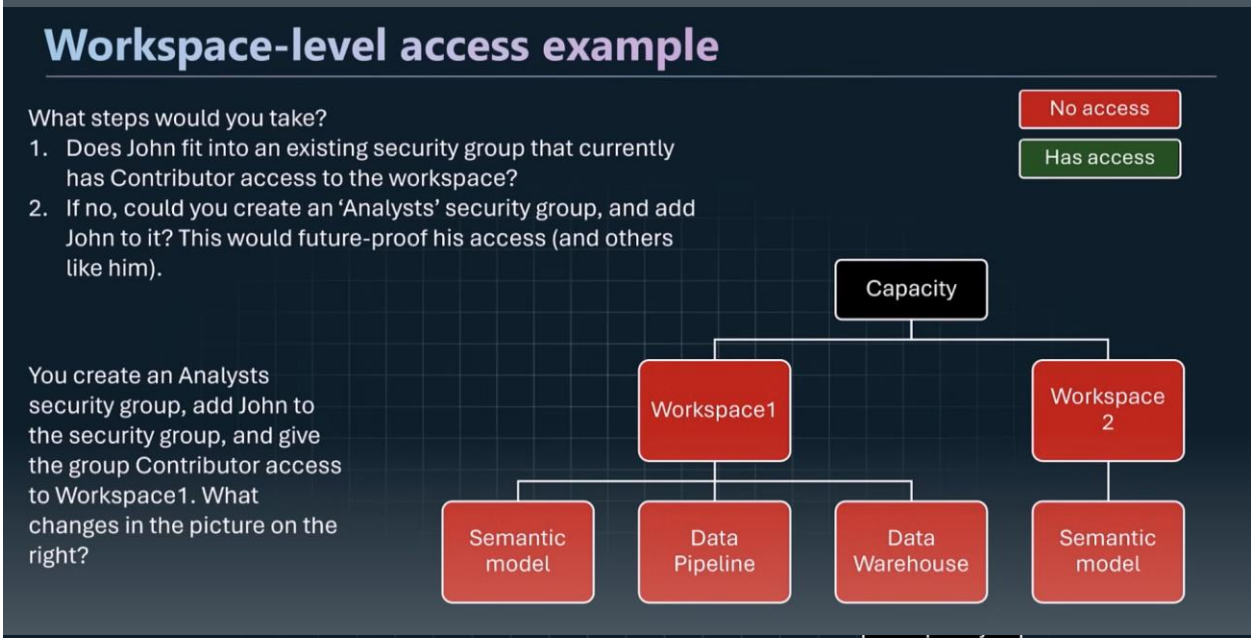


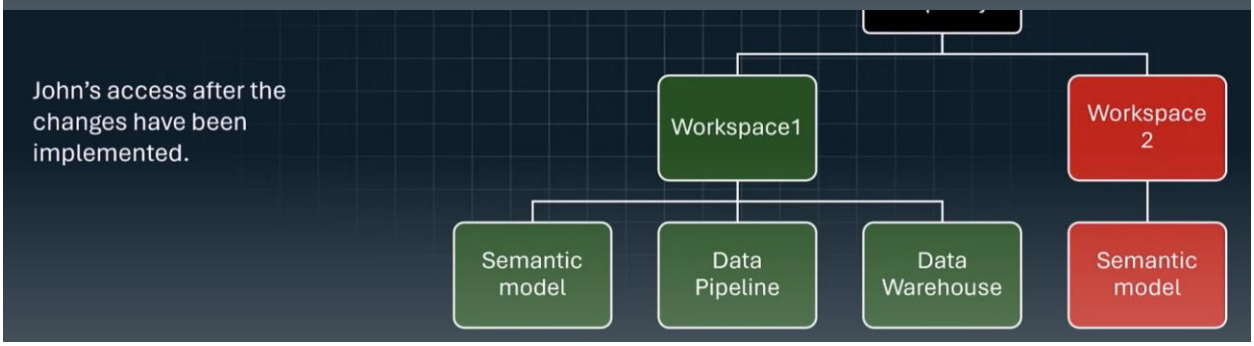
Fabric Analytics Concepts Notes

Capability	Admin	Member	Contributor	Viewer
Update and delete the workspace.	✅			
Add or remove people, including other admins.	✅			
Add members or others with lower permissions.	✅	✅		
Allow others to reshare items.[1]	✅	✅		
Create or modify database mirroring items.	✅	✅		
Create or modify warehouse items.	✅	✅		
Create or modify SQL database items.	✅	✅		
View and read content of data pipelines, notebooks, Spark job definitions, ML models and experiments, and eventstreams.	✅	✅	✅	✅
View and read content of KQL databases, KQL query-sets, and real-time dashboards.	✅	✅	✅	✅
Connect to SQL analytics endpoint of Lakehouse or the Warehouse	✅	✅	✅	✅
Read Lakehouse and Data warehouse data and shortcuts[2] with T-SQL through TDS endpoint (ReadData).	✅	✅	✅	✅
Read Lakehouse and Data warehouse data and shortcuts[2] through OneLake APIs and Spark (ReadAll).	✅	✅	✅	
Read Lakehouse data through Lakehouse explorer (ReadAll).	✅	✅	✅	
Subscribe to OneLake events.	✅	✅	✅	
Write or delete data pipelines, notebooks, Spark job definitions, ML models, and experiments, and eventstreams.	✅	✅	✅	
Write or delete Eventhouses[3], KQL Querysets, Real-Time Dashboards, and schema and data of KQL Databases, Lakehouses, data warehouses, and shortcuts.	✅	✅	✅	
Execute or cancel execution of notebooks, Spark job definitions, ML models, and experiments.	✅	✅	✅	
Execute or cancel execution of data pipelines.	✅	✅	✅	
View execution output of data pipelines, notebooks, ML models and experiments.	✅	✅	✅	✅
Schedule data refreshes via the on-premises gateway.[4]	✅	✅	✅	
Modify gateway connection settings.[4]	✅	✅	✅	

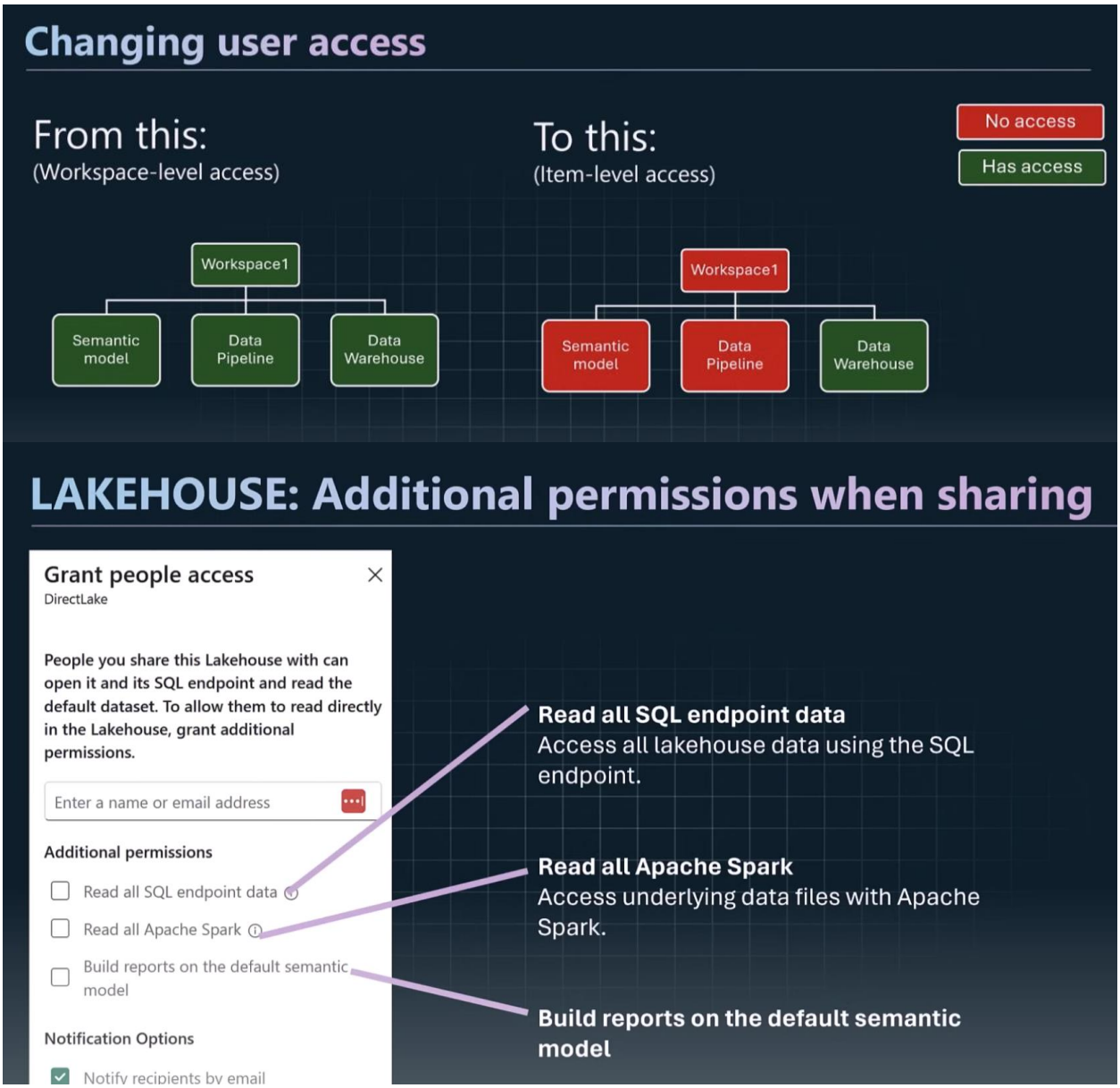
Fabric Analytics Concepts Notes



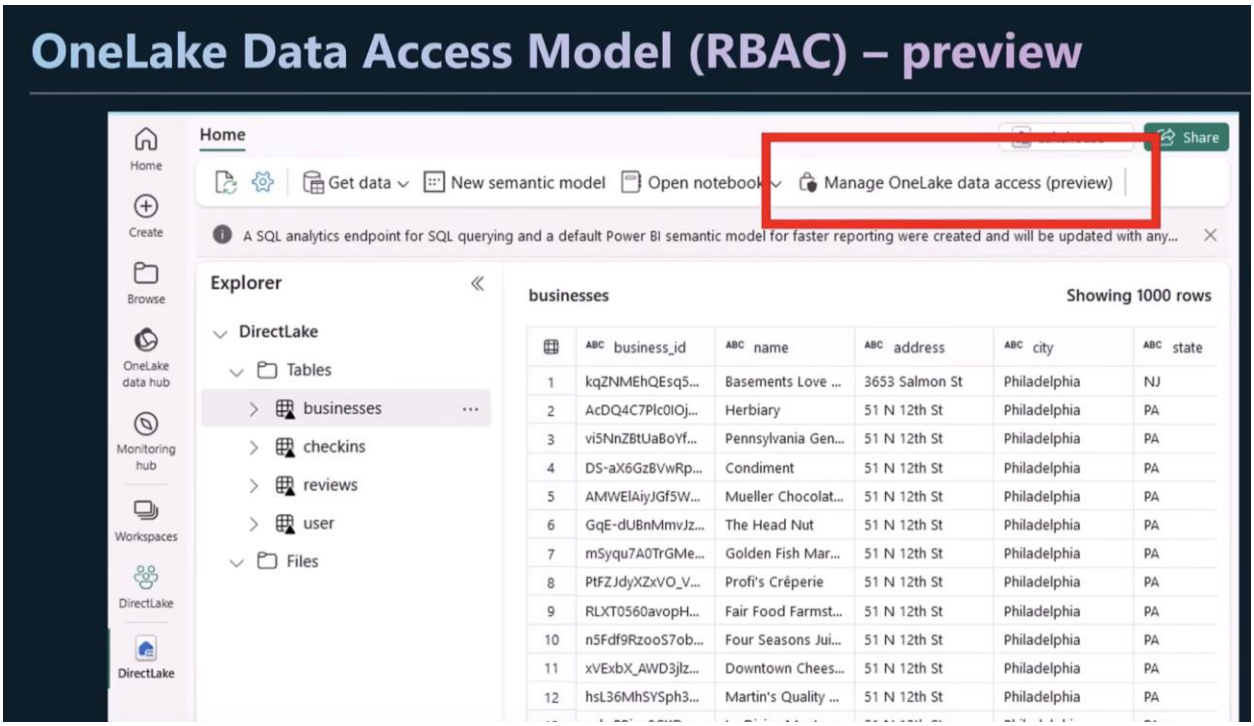




Fabric Analytics Concepts Notes



Fabric Analytics Concepts Notes



Question

Toby creates a new workspace with some Fabric items to be used by Data Analysts.

Toby creates a new security group called 'Data Analysts'. He includes himself as a member of this security group.

Toby gives the Data Analysts security group a **Viewer role** within the workspace.

What workspace role does Toby have?

A) Viewer

B) Member

C) Admin

D) Contributor

The Admin role supersedes and includes all permissions of the Viewer role.

Fabric Analytics Concepts Notes

Question

[..case study continued...]

Toby wants to delegate some of the management responsibilities in the workspace.

He wants to give this person the ability to share content within the workspace, invite new Contributors to the workspace, but not add new Admins to the workspace.

Which role should Toby give this person?

A) Admin

B) Member

C) Contributor

D) Viewer

Question

You have Admin role in a workspace.

Shiela is a data engineer in your team. Currently she has no access to the workspace.

Sheila needs to update a data transformation script in a PySpark notebook. The script gets data from a Lakehouse table, cleans it and then writes it to a table in the same Lakehouse.

You want to adhere to the principle of least privilege.

What actions should you take to enable this?

A) Give Sheila the Contributor role in the workspace.

B) Share the Lakehouse item with Read All Spark Data permission

C) Give Sheila the Admin role in the workspace

D) Share the Lakehouse item with Read All Spark Data permission and share the Notebook with Edit permission

Fabric Analytics Concepts Notes

Question

You have Admin role in a workspace.

You want to pre-install some useful Python packages to be used across all notebooks in your workspace.

How do you achieve this?

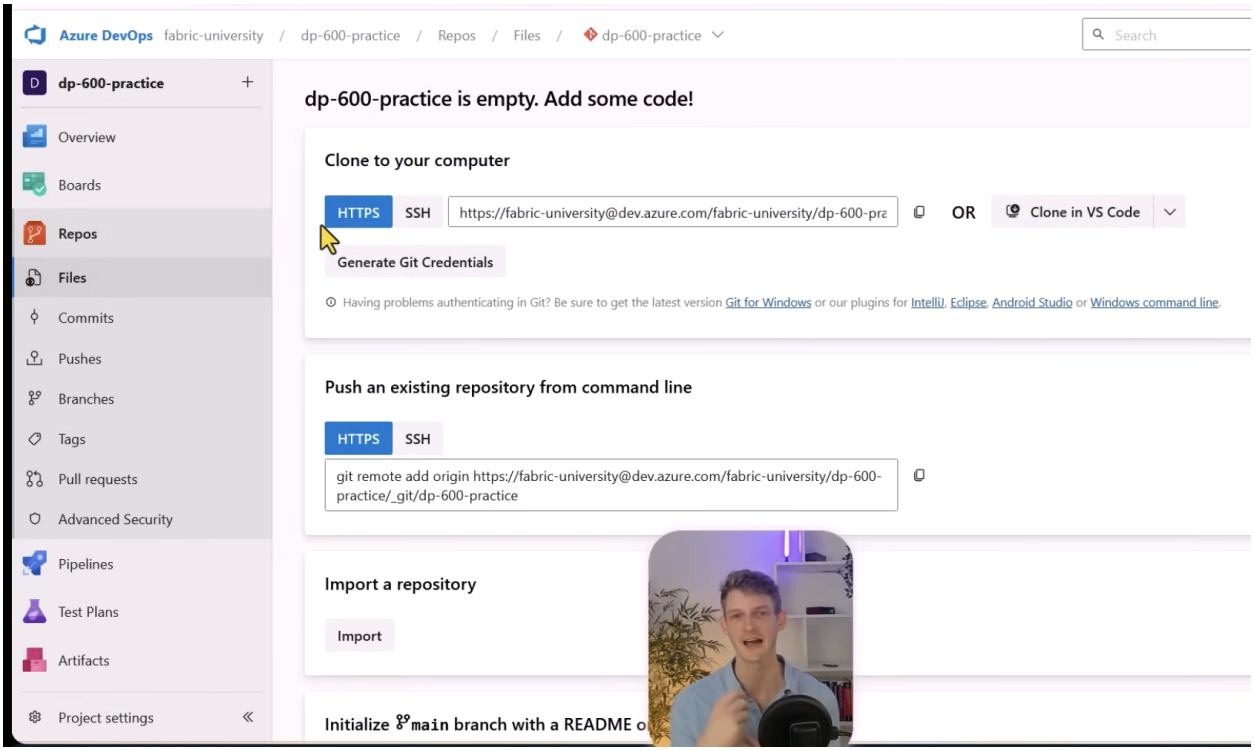
A) In the Fabric Admin Portal, go to Spark Settings and install the libraries

B) Go to Workspace Settings > Spark Settings > Library Management

C) **Create an Environment, install the packages in the Environment, then go to Workspace Settings > Spark Settings and Set the Default Environment**

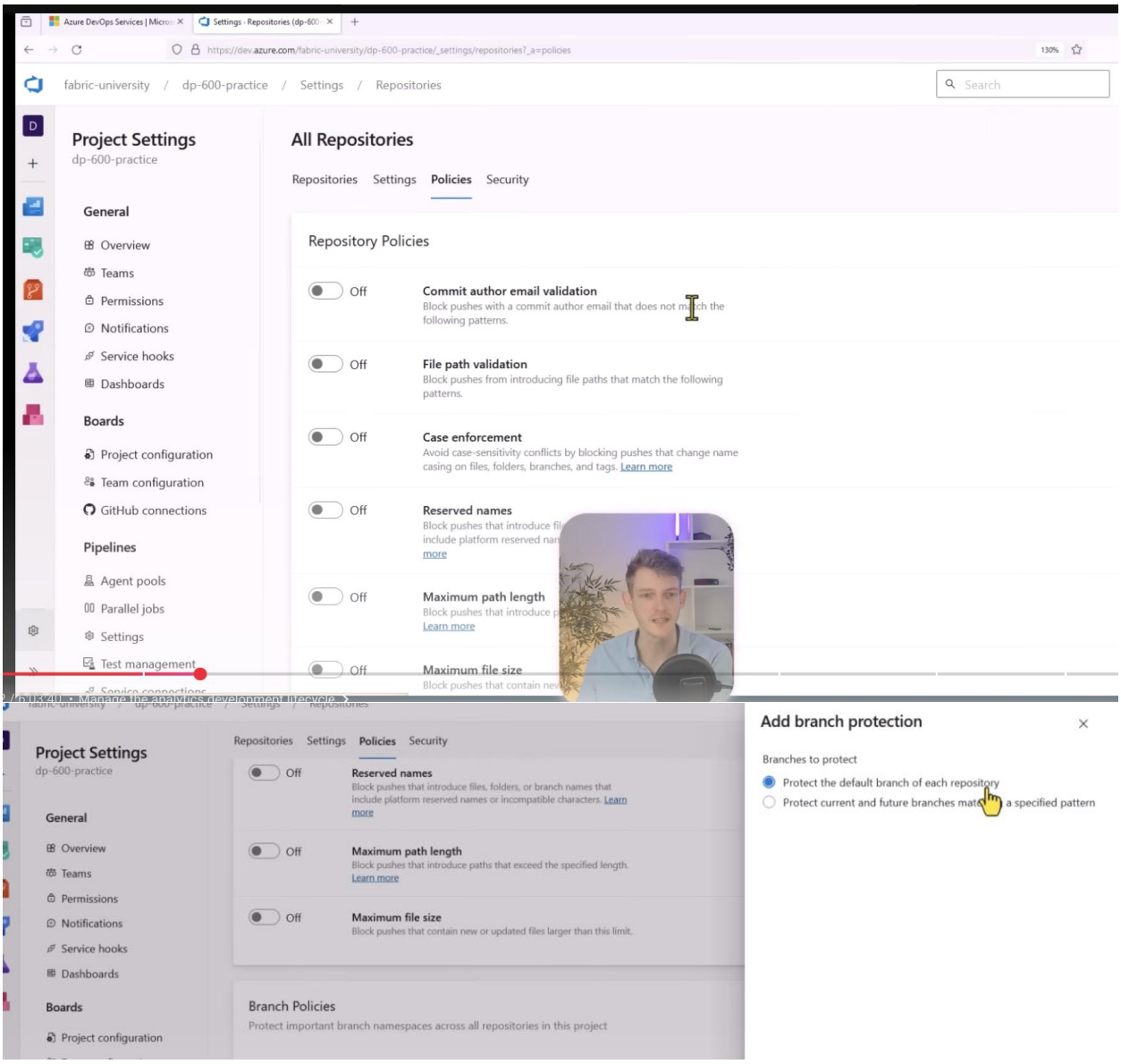
D) Go to Capacity Settings > Default Libraries

Ch4. Manage analytics dev lifecycle

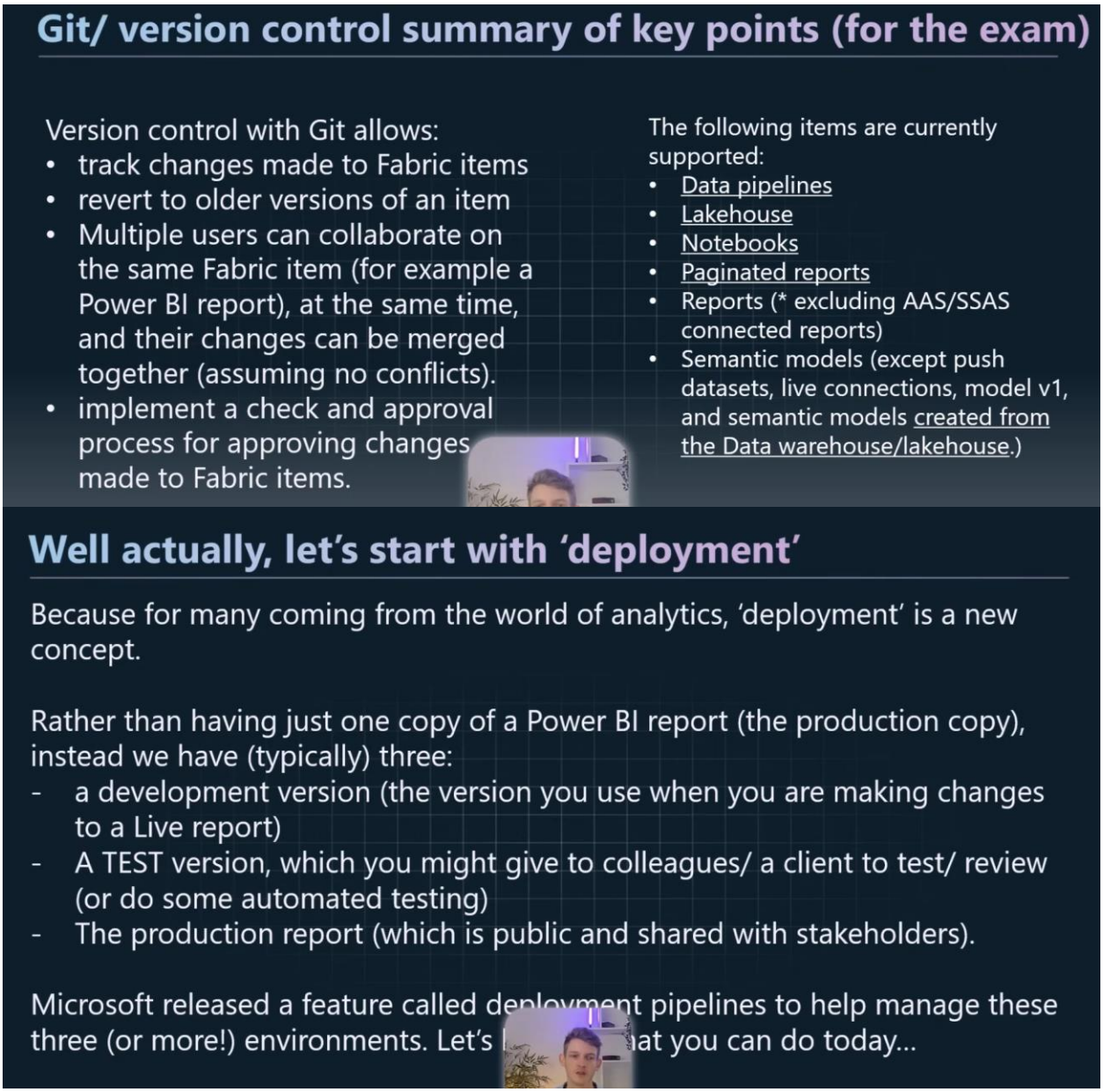


Restrict who can make changes to main branch of repo

Fabric Analytics Concepts Notes



Fabric Analytics Concepts Notes

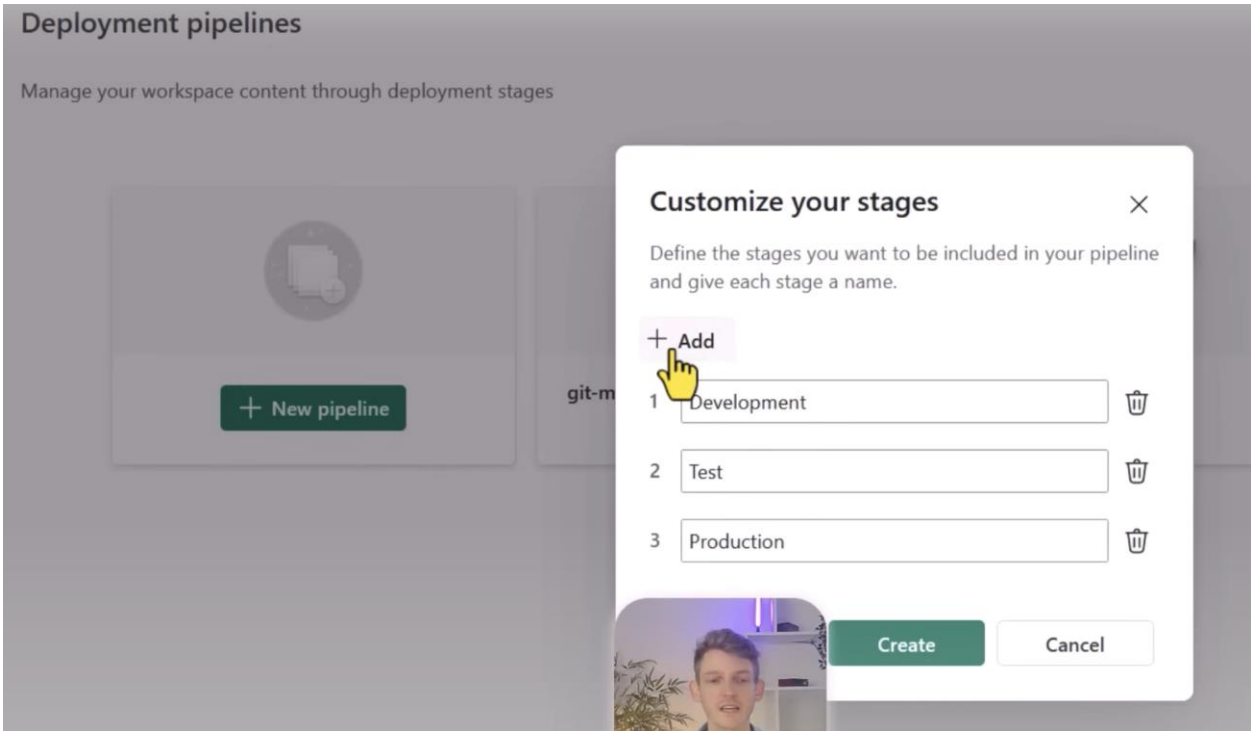


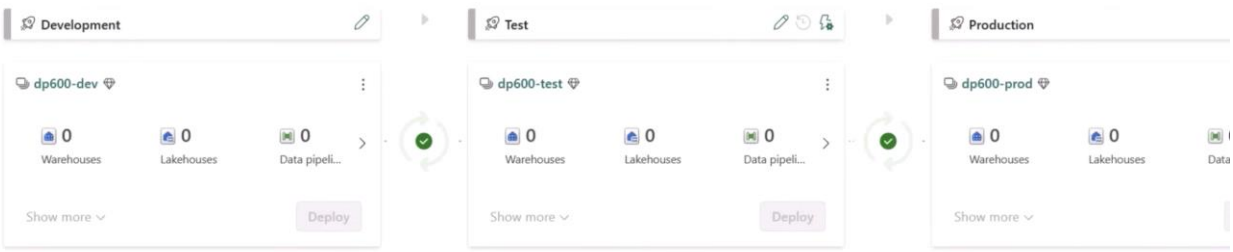
You worked in prod env in ergo & edeka projects

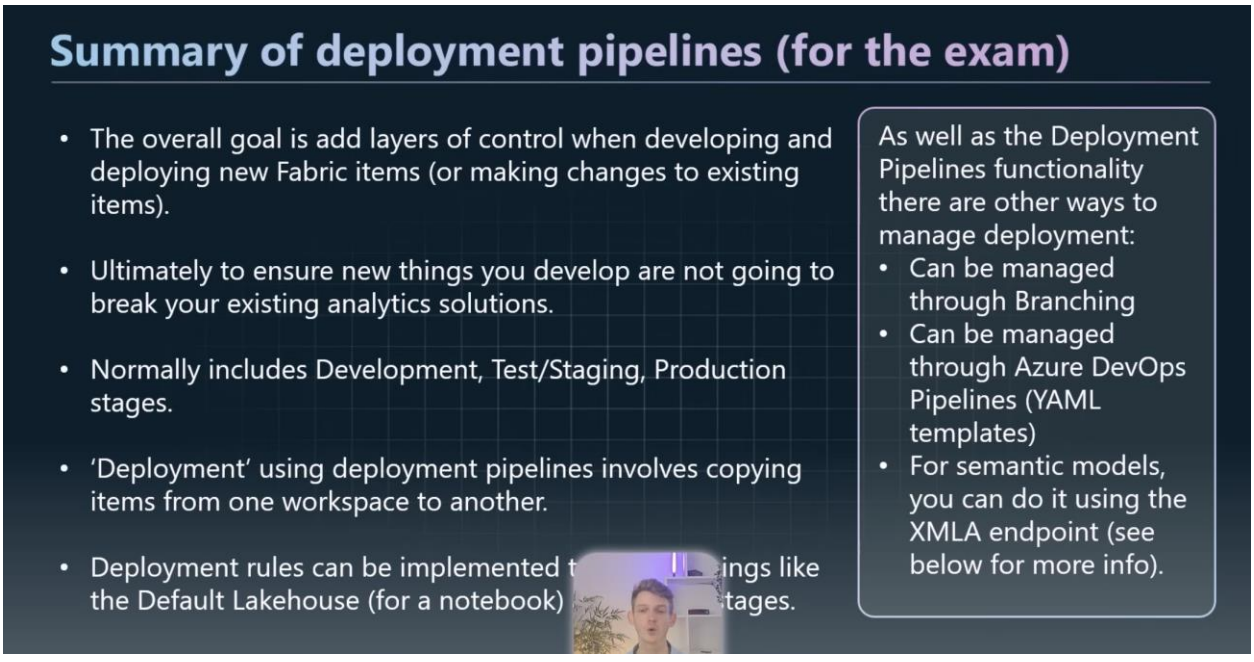
Setup deployment pipeline in MS Fabric

Testing: Integration test, unit tests, data validation tests

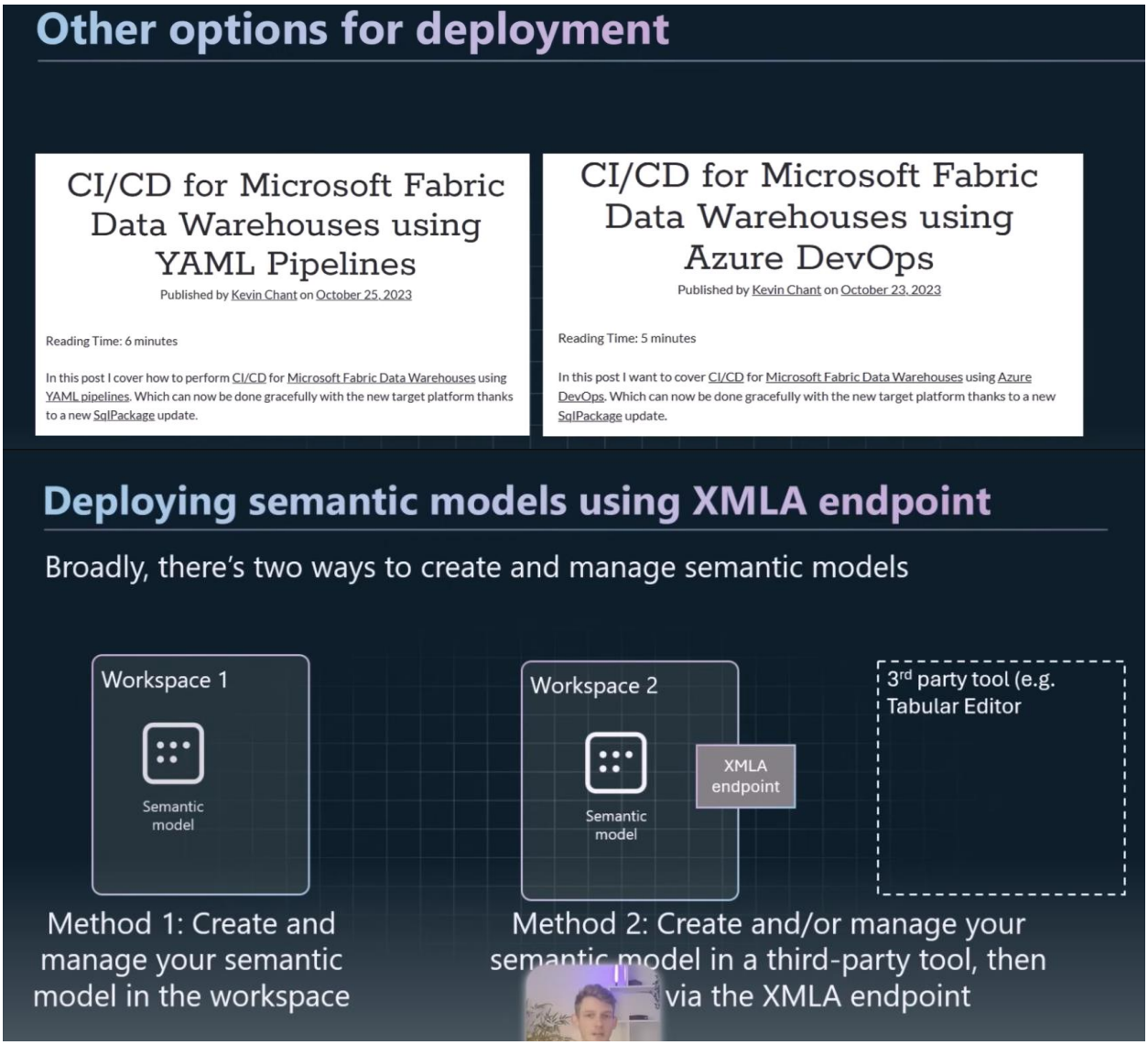
Fabric Analytics Concepts Notes







Fabric Analytics Concepts Notes



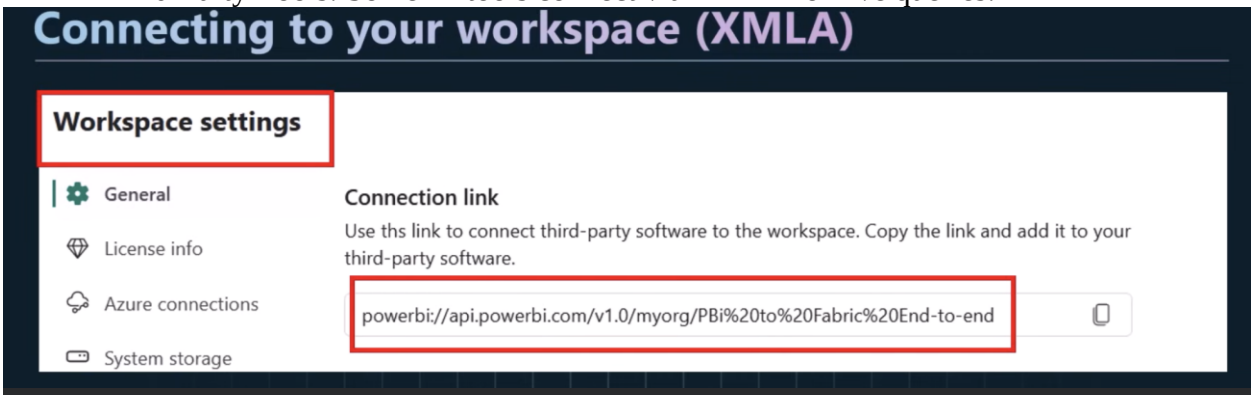
XLMA (XML for Analysis) Endpoint is a legacy SOAP-based protocol used for interacting with SQL Server Analysis Services (SSAS) and other OLAP (Online Analytical Processing) data sources. It allows client applications to execute queries (MDX, DAX, or XMLA commands) against multidimensional or tabular models.

- Protocol: Uses XMLA over HTTP/HTTPS (SOAP-based).
- Supported SSAS Models: Multidimensional (OLAP) models & Tabular models (compatibility level 1200+)

When is XLMA Endpoint Used Today?

- Power BI Premium/PPU: Power BI uses XMLA endpoints for:
    - Advanced dataset management (via SSMS/Tabular Editor).
    - Automated deployments (using PowerShell or DevOps).
    - Use Cases: Executing MDX (Multidimensional Expressions) **or DAX (Data Analysis Expressions) queries**. Performing administrative tasks (e.g., processing cubes, deploying models). Used by tools like Excel, Power BI, SQL Server Management Studio (SSMS), and custom apps.
- Legacy SSAS Systems: Older on-premises SSAS deployments still rely on it.

- Third-Party Tools: Some BI tools connect via XMLA for live queries.



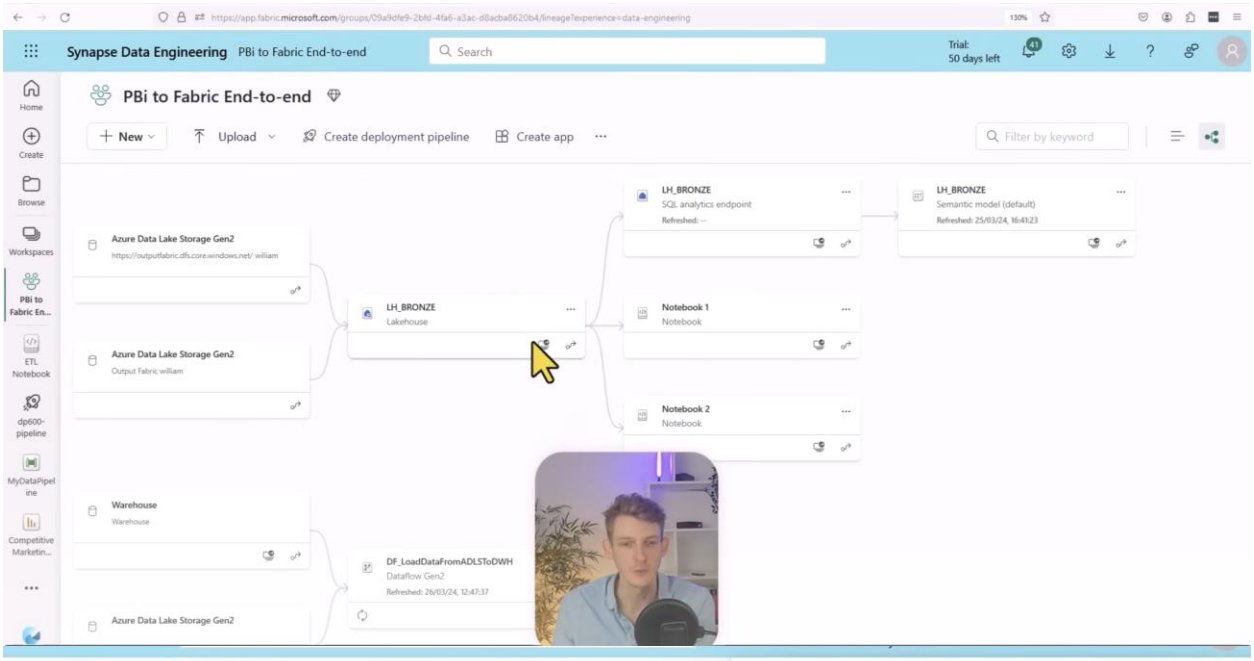
Comparison of Power BI File Formats

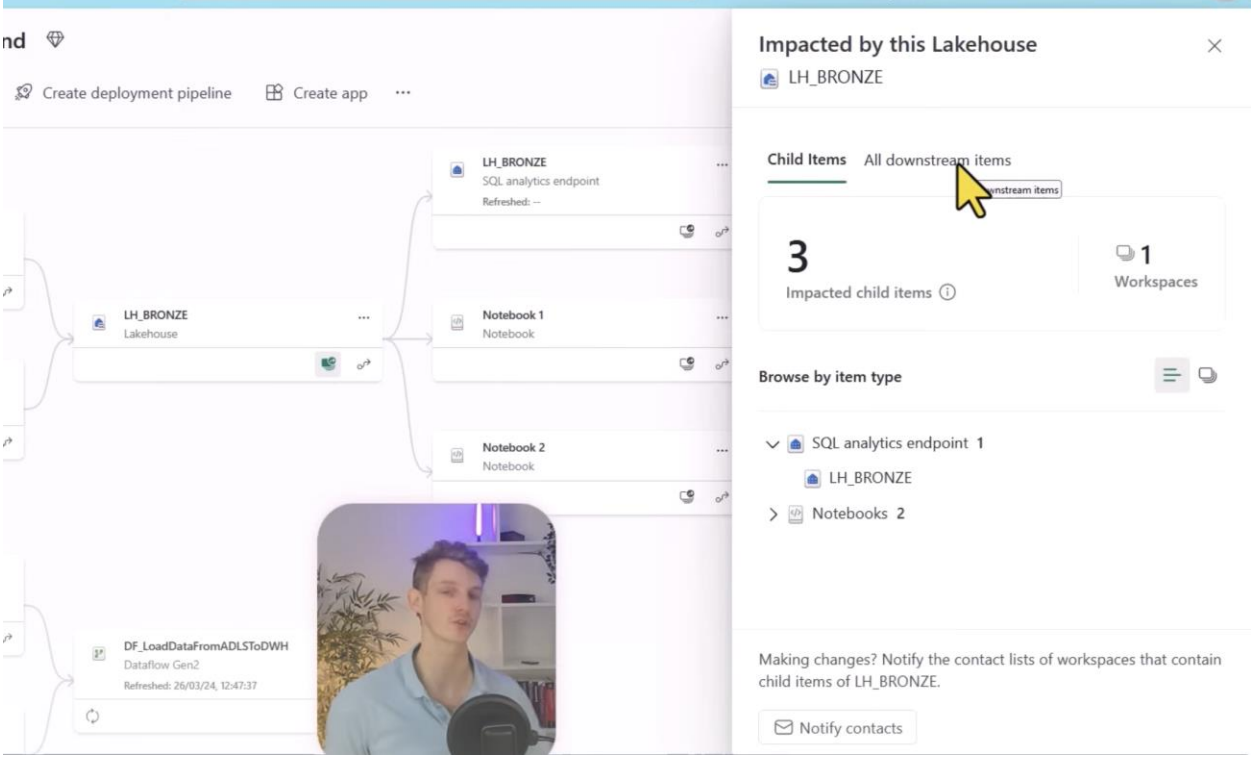
Feature	`.pbix`	`.pbip`	`.pbit`	`.pbids`
Full Name	Power BI Desktop File	Power BI Project File	Power BI Template File	Power BI Data Source File
Contains Data?	✅ Yes (embedded)	❌ No (Live/DQ only)	❌ No (template)	❌ No (connection only)
Git-Friendly?	❌ No (binary)	✅ Yes (folder-based)	❌ No	✅ Yes (lightweight)
Editable in PBI Desktop?	✅ Yes	✅ Yes	✅ Yes (prompts for data)	✅ Yes (opens as connection)
Used for Publishing?	✅ Yes	❌ No (must publish `.pbix`)	❌ No	❌ No
Primary Purpose	Standalone reports	Team projects (DevOps)	Reusable templates	Quick data source setup

- .pbix → Standard Power BI reports (single-user or small teams).
- .pbip → Team projects with Git/DevOps (future of Power BI development).
- .pbit → Distributing reusable templates (e.g., company-wide dashboards).
- .pbids = Just a connection (no report).

Fabric Analytics Concepts Notes

Impact Analysis Tool of Fabric (Lineage View)





Fabric Analytics Concepts Notes

Question

You are looking to improve the efficiency and consistency of your Power BI Development team.

You want each report created by the team to always consist of three pages: Intro, Context, Analysis.

The reports should always align to the company branding guidelines.

Which of the following would help you achieve this

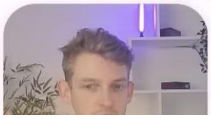
A) Create . pbip file

B) Create a .pbix file

C) Create a .pbit file

D) Create a .pbids file

E) Using a JSON custom report theme



Question

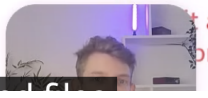
Which of the following most accurately describes Git?

A) To use Git, you must be using GitHub.

B) Git is a Microsoft product for tracking changes made to Fabric items.

C) Git is a open-source version control system that tracks changes in any set of (text-based) files.

allows us to add deployment rules to ric deployment pipelines.



Question

In a Azure DevOps repo, the MAIN branch is 'protected' (needs approval before any changes are merged into it).

The repo contains one PBIP. You have to update the Title in the report, merging these changes to the Main branch.

In which order should you carry out the following tasks to achieve this?

1. **Clone the repository to your local machine.**
2. **Checkout a new feature branch from the MAIN branch**
3. **Make the required changes to the report.**
4. **Commit and push the feature branch.**
5. **Open a pull request in Azure Repos.**
6. **Wait for approval, then merge into the main**

Fabric Analytics Concepts Notes

Question

You want to deploy a semantic model using the XMLA endpoint.

Where can you do to find the XMLA endpoint to set up a connection with a third-party tool?

A) **Go to the workspace settings for the workspace you want to deploy your model to**

B) Go to the Fabric Admin Portal > Capacity Settings

C) In your workspace, find your semantic model and click on settings to get the XMLA endpoint address

D) In the Azure Portal, in your Fabric Capacity, go to the XMLA endpoint connection string settings.

1. The XMLA endpoint is **workspace-specific** and tied to the underlying capacity.
    - Open the **Power BI Service** (app.powerbi.com) & Navigate to the **workspace** containing your semantic model.
    - Format: powerbi://api.powerbi.com/v1.0/[tenant-name]/[workspace-name]

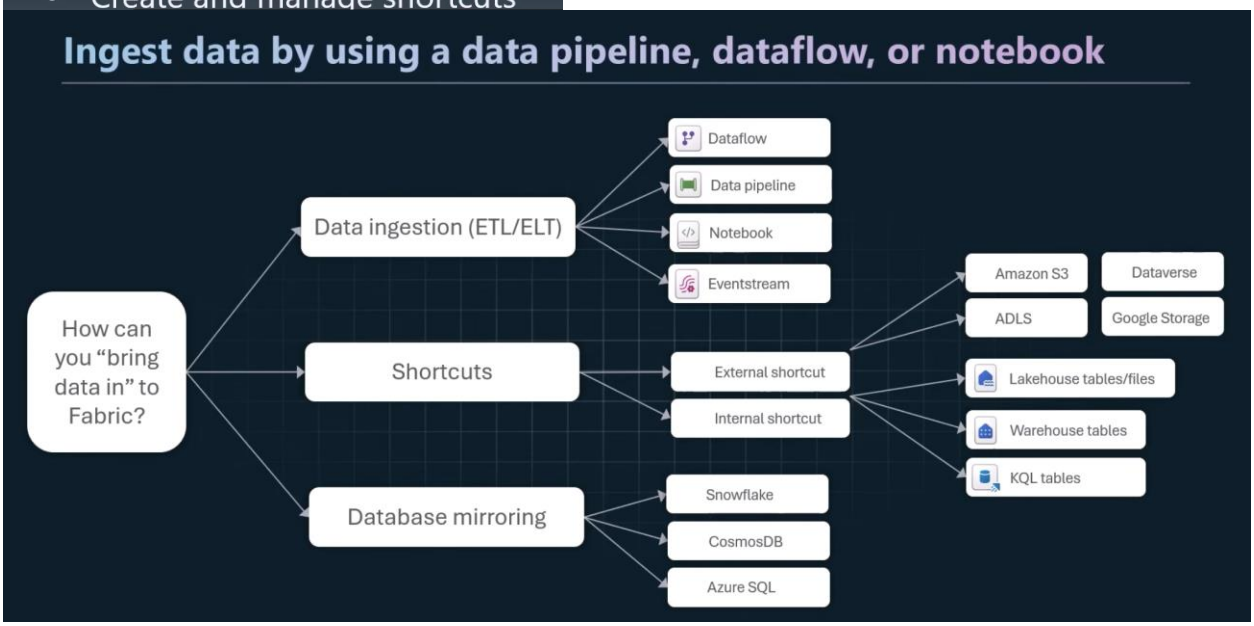
2. To deploy a semantic model via XMLA (e.g., using **SSMS, Tabular Editor, or DevOps**), you need the endpoint address from:

    - **Workspace Settings** > **Premium**/ **Fabric Capacity** tab > **XMLA Endpoint** field.
    - Requires Premium/Fabric Capacity: XMLA endpoints are only available for workspaces on Premium Per User (PPU), Premium, or Fabric SKUs.

    - Third-Party Tools: Use the endpoint to connect via: SQL Server Management Studio (SSMS) (or) Tabular Editor (or) PowerShell (for automation)

Ch5. Getting Data to Fabric

- Ingest data by using a data pipeline, dataflow, or notebook
- Copy data by using a data pipeline, dataflow, or notebook
- Choose an appropriate method for copying data from a Fabric data source to a lakehouse or warehouse
- Create and manage shortcuts



Fabric Analytics Concepts Notes

Fabric Analytics Concepts Notes

Ingest data with a notebook

General purpose coding notebook which can be used to bring data into Fabric, via connecting to APIs or by using client Python libraries

Pros (when you should consider using)	Cons (maybe when you shouldn't use)
• Extraction from APIs (using Python requests library, or similar!) • To use client libraries (e.g. Azure libraries, or the Hubspot client library in Python to access Hubspot data). • Good for code re-use (and can be parameterized) • For data validation and data quality testing of incoming data • The fastest in terms of data ingestion (and most efficient for CU spend - see here)	• When you don't have a Python capability in your organisation • When you want to write to a Data Warehouse (currently not possible)

Create and manage shortcuts

- A shortcut enables you to create a live link to data stored either in another part of Fabric (internal shortcut) or in external storage locations:
  - ADLS Gen2
  - Amazon S3
  - Other services that USE Amazon S3 for storage (like Cloudflare)
  - Google Cloud Storage
  - Dataverse
- Shortcuts can be setup for individual files, but more commonly to a folder. If you create a shortcut to a 'base' folder, it will monitor and sync all files in subfolders.
- Be careful of cross-region egress fees. If your Fabric capacity is in UK-South region (for example), but your ADLS storage account is in West-US, you will be charged for 'cross-region' egress fees ($0.01 per GB)
- OneLake shortcuts can now be created using the Fabric REST API

Fabric Analytics Concepts Notes

The following table shows the shortcut-related permissions for each workspace role. For more information, see Workspace roles.

Expand table

Capability	Admin	Member	Contributor	Viewer
Create a shortcut	Yes[1]	Yes[1]	Yes[1]	-
Read file/folder content of shortcut	Yes[2]	Yes[2]	Yes[2]	-
Write to shortcut target location	Yes[3]	Yes[3]	Yes[3]	-
Read data from shortcuts in table section of the lakehouse via TDS endpoint	Yes	Yes	Yes	Yes

[1] Users must have a role that provides write permission the shortcut location and at least read permission in the target location.

[2] Users must have a role that provides read permission both in the shortcut location and the target location.

[3] Users must have a role that provides write permission both in the shortcut location and the target location.

Feature	Lakehouse	Warehouse	Eventhouse
Data Types	All (structured/unstructured)	Structured only	Time-series/logs
Query Language	Spark SQL, T-SQL	T-SQL	KQL
Latency	Minutes-hours	Seconds-minutes	**Sub-second**
Best For	Raw data exploration	Business reports	Real-time streams

Fabric Analytics Concepts Notes

Some other decision criteria

- Need for 'real-time' data - if there is a requirement for near-real-time, you will prefer either database mirroring (for data in Azure SQL, Azure Cosmos DB, or Snowflake), or Shortcuts for Files and Folders.
- Skills in your team:
  - No/Low-code? Dataflow + data pipeline
  - SQL-based? Data pipeline + stored proc activity/ script activity
  - Python? Notebooks
- Cross-workspace limitations: data pipeline has quite a few limitations working 'cross-workspace'. This means that currently your data pipeline must be in the same workspace as your destination datastore (for data ingestion).
- Scalability / Size of your data / Cost / capacity usage - in general, the notebook is the most efficient method for data ingestion.

Question

You are trying to create a shortcut to a folder of CSV files in Azure Data Lake Storage Gen2.

Which of the following is a valid connection string you can connect to?

A) https://contosoadlscdm.database.windows.net

B) https://contosoadlscdm.blob.core.windows.net

C) https://contosoadlscdm.file.core.windows.net

D) https://contosoadlscdm.dfs.core.windows.net

Key Difference: `blob` vs. `dfs` Endpoints

Feature	`blob.core.windows.net`	`dfs.core.windows.net`
Purpose	Blob Storage (flat namespace)	ADLS Gen2 (hierarchical folders)
Shortcuts	❌ No	✅ Yes
Folder Operations	Limited	Optimized (e.g., `ls`, `mkdir`)

Fabric Analytics Concepts Notes

- **A)** `database.windows.net`:
  - Used for **Azure SQL Database**, not ADLS Gen2.
- **B)** `blob.core.windows.net`:
  - Blob Storage endpoint (works for files, but lacks **hierarchical namespace** features like folders).
- **C)** `file.core.windows.net`:
  - For **Azure Files** (SMB/NFS shares), not ADLS Gen2.

3. Key Differences: HDFS vs. ADLS Gen2

Feature	HDFS (On-prem)	ADLS Gen2 (Cloud)
Namespace	Flat (emulated folders)	True hierarchical folders
Scalability	Manual cluster scaling	Automatic (unlimited storage)
Replication	3x (default)	Geo-redundancy (GRS, ZRS)
Access	`hdfs://` paths	`abfss://` (Azure Blob FS)

Why This Matters for Your Question

- The `dfs` **endpoint** in ADLS Gen2 (Option D) is the **cloud equivalent of HDFS** for Hadoop-based tools.
- When you use `dfs.core.windows.net`, you're leveraging:
  - HDFS-like paths (e.g., `/myfolder/data.csv`).
  - Compatibility with **Spark, Fabric, and Hadoop ecosystems**.

File Sharing Protocols	***SMB** (Server Message Block)*	***NFS (**Network File System**)***
OS Support	Windows (native), Linux (Samba)	Linux/Unix (native), Windows (NFS client)
Security	Kerberos, AD integration/Entra id	IP-based, Kerberos (v4/v3)
Latency	Higher (chatty protocol)	Lower (lightweight)
Use Cases	Office docs, user home dirs	VMs, big data, media

Fabric Analytics Concepts Notes

Question

You are implementing the first stage in a medallion architecture.

Your goal is to retrieve data from a REST API (Get Request) and save the raw JSON response in the Files area of a BRONZE Lakehouse.

Which of the following methods can you use to achieve this?

(***THREE CORRECT ANSWERS***)

A) Data Pipeline (Copy Data Activity)

B) Dataflow (Web API connector)

C) Eventstream

D) Data Pipeline (Web Activity)

E) Fabric Notebook

Editor's Note: the Data Pipeline Web Activity can't perform the whole action on its own. It can perform the GET request and pass the output to a Copy Data activity to write the data into a Lakehouse Files area

Question

Your goal is to extract a CSV file in Azure Blob Storage write it to a Fabric Data Warehouse table.

Which of the following methods can you **NOT** use to achieve this?

A) Data Pipeline (Copy Data activity)

B) Dataflow with the Blob Storage connector

C) In a Fabric notebook, use the Azure Blob Storage client library for Python to get the file and write the Data Warehouse table

D) Use the COPY INTO statement (T-SQL) from within the Fabric Data Warehouse

Question

Workspace A contains Lakehouse A.

Workspace B contains Lakehouse B.

You want to create a shortcut in Lakehouse A (pointing to a table from Lakehouse B).

What is the minimum level of workspace permissions you need to achieve this?

A) Contributor in Workspace A and Viewer in Workspace B.

B) Contributor in Workspace A and Contributor in Workspace B.

C) Member in Workspace A and Contributor in Workspace B.

D) Viewer in Workspace A and Viewer in Workspace B

Question

One of your team is a superstar using M code for data extraction.

In which of following data extraction tools can you write M code?

A) Fabric notebook

B) In a T-SQL script in the Data Warehouse

C) Dataflow

D) Data pipeline (Mapping dataflow)

Fabric Analytics Concepts Notes

1. **A) Fabric Notebook**
   - **Primary Language**: Python, SQL, Scala, R.
   - **M Code Support?** ❌ No (M code is specific to Power Query, not notebooks).
2. **B) T-SQL Script in Data Warehouse**
   - **Primary Language**: T-SQL (Transact-SQL).
   - **M Code Support?** ❌ No (T-SQL is for querying relational data, not ETL transformations).
3. **C) Dataflow**
   - **Primary Language**: **M Code** (Power Query).
   - **M Code Support?** ✅ **Yes** (Dataflows are built on Power Query; M is the native language).
4. **D) Data Pipeline (Mapping Dataflow)**
   - **Primary Language**: Power Query (via UI) or Spark/SQL.
   - **M Code Support?** ✅ **Partial** (Mapping Dataflows use Power Query transformations, but M code is hidden behind the UI).

Ch6. SQL, DWH, Scheduling

View is read only (It can't update /insert just like function). To insert / update, use appropriate dml statement or stored procedure

```sql
CREATE VIEW  dbo.vw_Employee_GetJack AS
SELECT
    EmployeeId
    , Name
    , Age
from dbo.Employees
WHERE Name like 'Jack%';

select * from dbo.vw_Employee_GetJack
```

```sql
-- KEY POINTS FOR A VIEW
-- [x] Transformed data is not stored in a view, the result is calculated at query time.
-- [x] You can create a View from another View (but be careful, this can lead to poor performance, and difficult to
-- [x] you cannot specify parameters for a View (this is possible with functions and stored procedures)
-- [x] Reading a SQL View into a semantic model will fallback to DirectQuery (Direct Lake not possible).
```

Fabric Analytics Concepts Notes

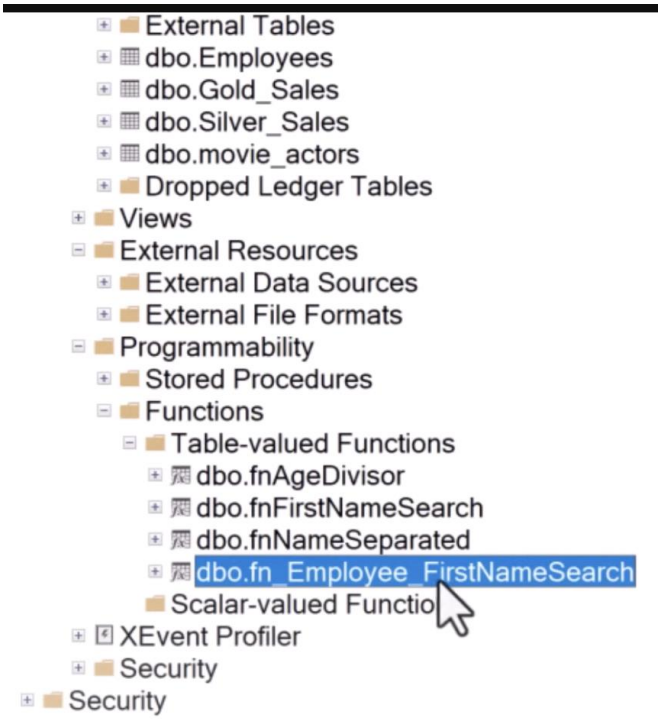
```sql
-- Create a Function that takes a first name as input
-- and Filters the Employees table
CREATE FUNCTION dbo.fn_Employee_FirstNameSearch (  @firstname varchar(20) = ''  )
RETURNS TABLE
AS
RETURN
(
    SELECT
        EmployeeId
        , Name
        , Age
    from dbo.Employees
    WHERE Name like @firstname + '%'
) ;
```

```sql
-- Test the function
select * from dbo.fn_Employee_FirstNameSearch('Jack')
```

Function can take only input parameter & not output parameter (in mysq, sql server. Postgres function can take both though)

```sql
-- Key Points for a FUNCTION
-- [x] Useful for packaging up logic, which can be parameterized
-- [x] Only SELECT statements, can never edit the underlying data (using UPDATE/ INSERT etc)
-- [x] Can be called from DDL statements, or from within Stored Procedures, or a View
-- [x] Can't be orchestrated (directly) from a data pipeline (although can embed in a stored proc).
-- [x] Allows one or many input parameters
-- [x] Output type should also be a Table
```

Fabric Analytics Concepts Notes



- **DDL:**
  - Creating tables (CREATE TABLE).
  - Adding columns (ALTER TABLE).
  - Dropping databases (DROP DATABASE).
- **DML:**
  - Adding records (INSERT).
  - Updating salaries (UPDATE).
  - Querying data (SELECT).

1. DDL (Data Definition Language) Statements

DDL commands define/modify database structures.
CREATE TABLE

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    HireDate DATE DEFAULT GETDATE(),
    Salary DECIMAL(10, 2) CHECK (Salary > 0)
);
```

ALTER TABLE

```
-- Add a column
ALTER TABLE Employees ADD DepartmentID INT;
```

-- Add a foreign key
ALTER TABLE Employees
ADD CONSTRAINT FK_Department
FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID);
CREATE INDEX
CREATE INDEX IX_Employees_LastName ON Employees(LastName);
DROP TABLE
DROP TABLE Employees; -- Deletes the table and its data

2. Stored Procedures
Stored procedures encapsulate reusable SQL logic.

```
drop procedure if exists dbo.sp_Employee_GetByFirstName

-- stored procedure with input parameter
CREATE PROCEDURE dbo.sp_Employee_GetByFirstName @firstname varchar(20)
as
select * from dbo.Employees
where Name like @firstname + '%'

exec dbo.sp_Employee_GetByFirstName 'Jack'
```

Basic Stored Procedure
CREATE PROCEDURE GetEmployeeByID @EmployeeID INT
AS
BEGIN
  SELECT * FROM Employees WHERE EmployeeID = @EmployeeID;
END;
Execution:
EXEC GetEmployeeByID @EmployeeID = 101;
Procedure with Output Parameter
CREATE PROCEDURE GetEmployeeCount
  @DepartmentID INT,
  @Count INT OUTPUT
AS
BEGIN
  SELECT @Count = COUNT(*)
  FROM Employees
  WHERE DepartmentID = @DepartmentID;
END;
Execution:
DECLARE @Result INT;
EXEC GetEmployeeCount @DepartmentID = 5, @Count = @Result OUTPUT;
PRINT @Result; -- Prints the count

Fabric Analytics Concepts Notes

Transaction Handling

```
CREATE PROCEDURE UpdateSalary
   @EmployeeID INT,
   @NewSalary DECIMAL(10, 2)
AS
BEGIN
   BEGIN TRY
      BEGIN TRANSACTION;
         UPDATE Employees
         SET Salary = @NewSalary
         WHERE EmployeeID = @EmployeeID;
      COMMIT TRANSACTION;
   END TRY
   BEGIN CATCH
      ROLLBACK TRANSACTION;
      THROW;  -- Re-throws the error
   END CATCH
END;
```

3. Dynamic SQL in Stored Procedures

```
CREATE PROCEDURE SearchEmployees
   @ColumnName NVARCHAR(50),
   @SearchValue NVARCHAR(100)
AS
BEGIN
   DECLARE @SQL NVARCHAR(MAX);
   SET @SQL = N'SELECT * FROM Employees WHERE ' +
         QUOTENAME(@ColumnName) + ' = @Value';
   EXEC sp_executesql @SQL, N'@Value NVARCHAR(100)', @Value = @SearchValue;
END;
```

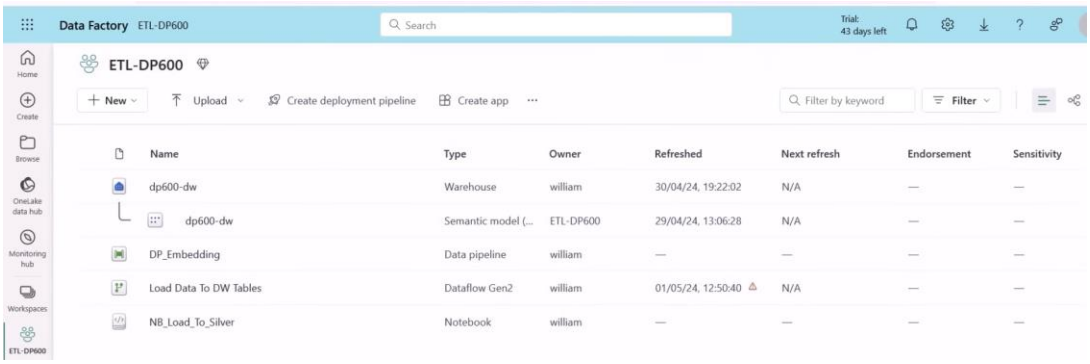
Execution:

```
EXEC SearchEmployees @ColumnName = 'LastName', @SearchValue = 'Smith';
```

```
-- KEY POINTS
-- [x] Ability to define input AND OUTPUT parameters
-- [x] Are called using EXEC (not as part of a select statement)
-- [x] Can call other stored procedures
-- [x] Can also give access to people to the Stored Procedure
--       (and not the underlying dataset)
-- [x] Can be embedded in a data pipeline (with parameters passed)
-- [x] Ability to do INSERTS, UPDATES, DELETES
```

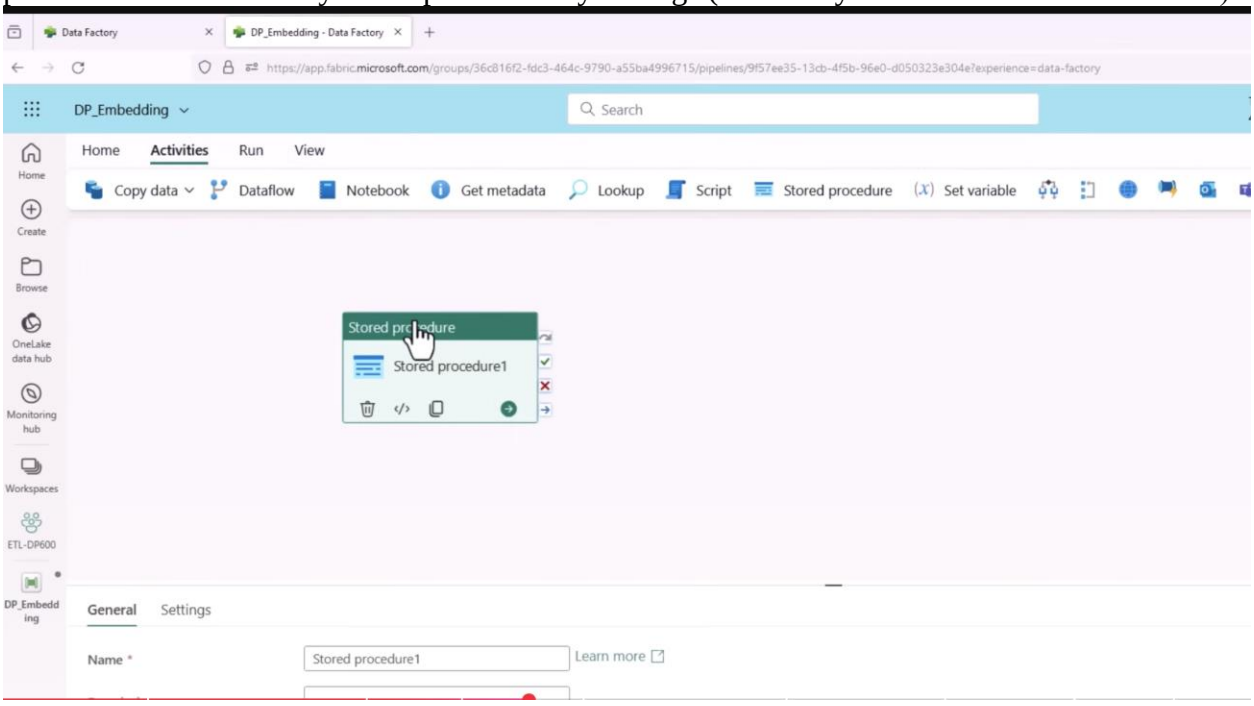
Fabric Analytics Concepts Notes

Azure DataFactory

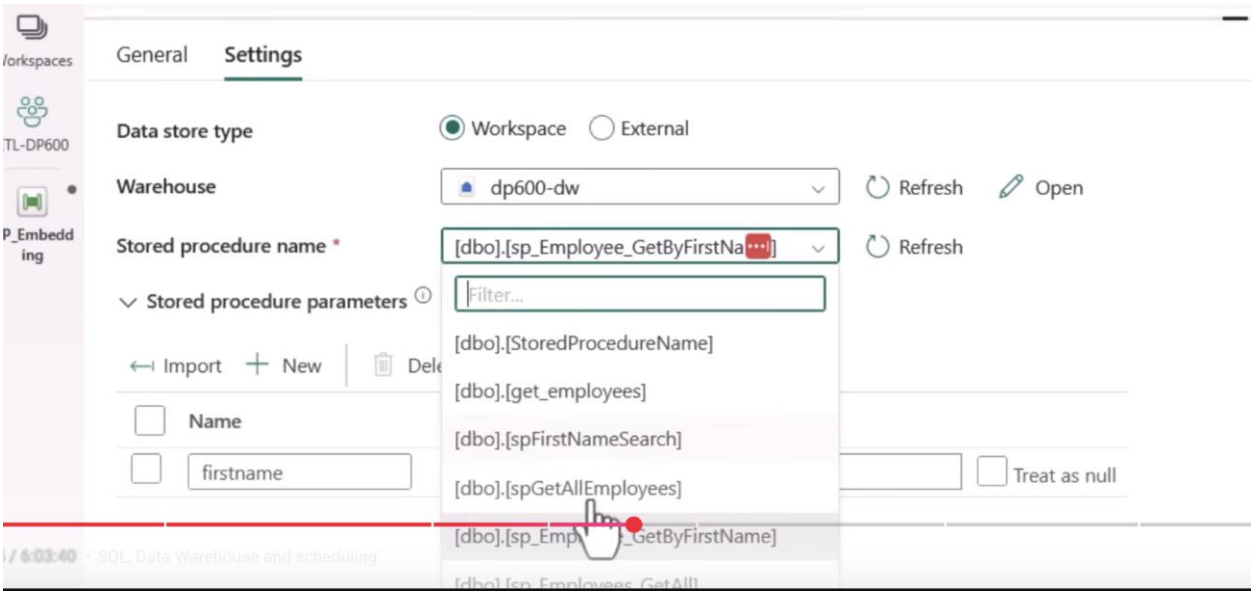


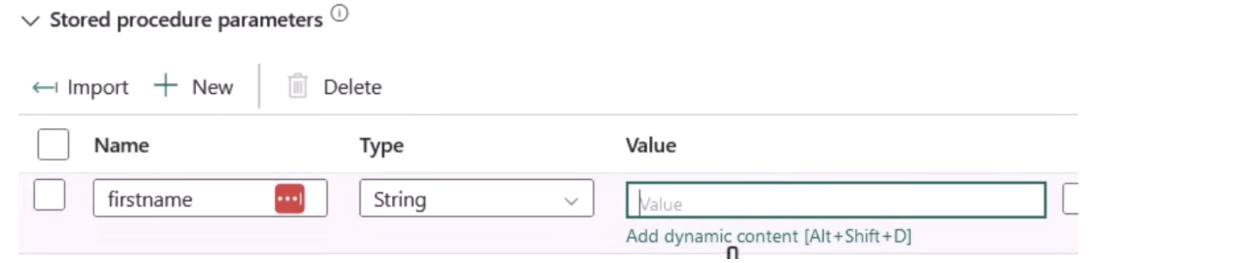
Data Pipeline

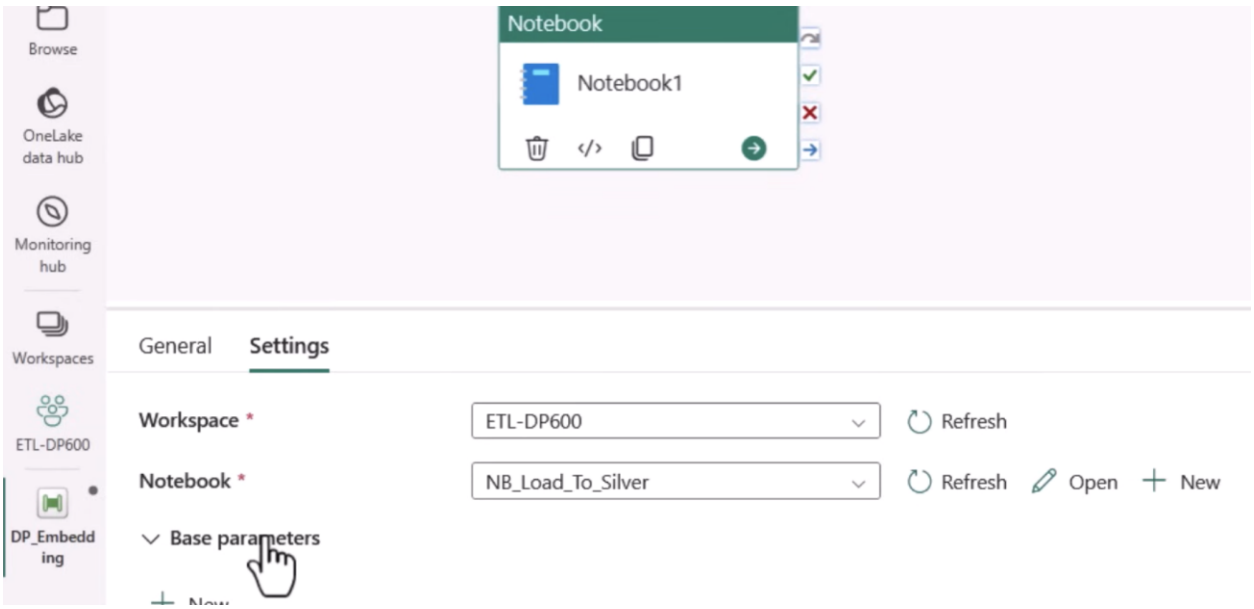
In pipeline, u can define what to do when a dataflow fails, and link to notebook & store procedures and have retry of scripts and many settings (which any one individual item can't do)



Fabric Analytics Concepts Notes

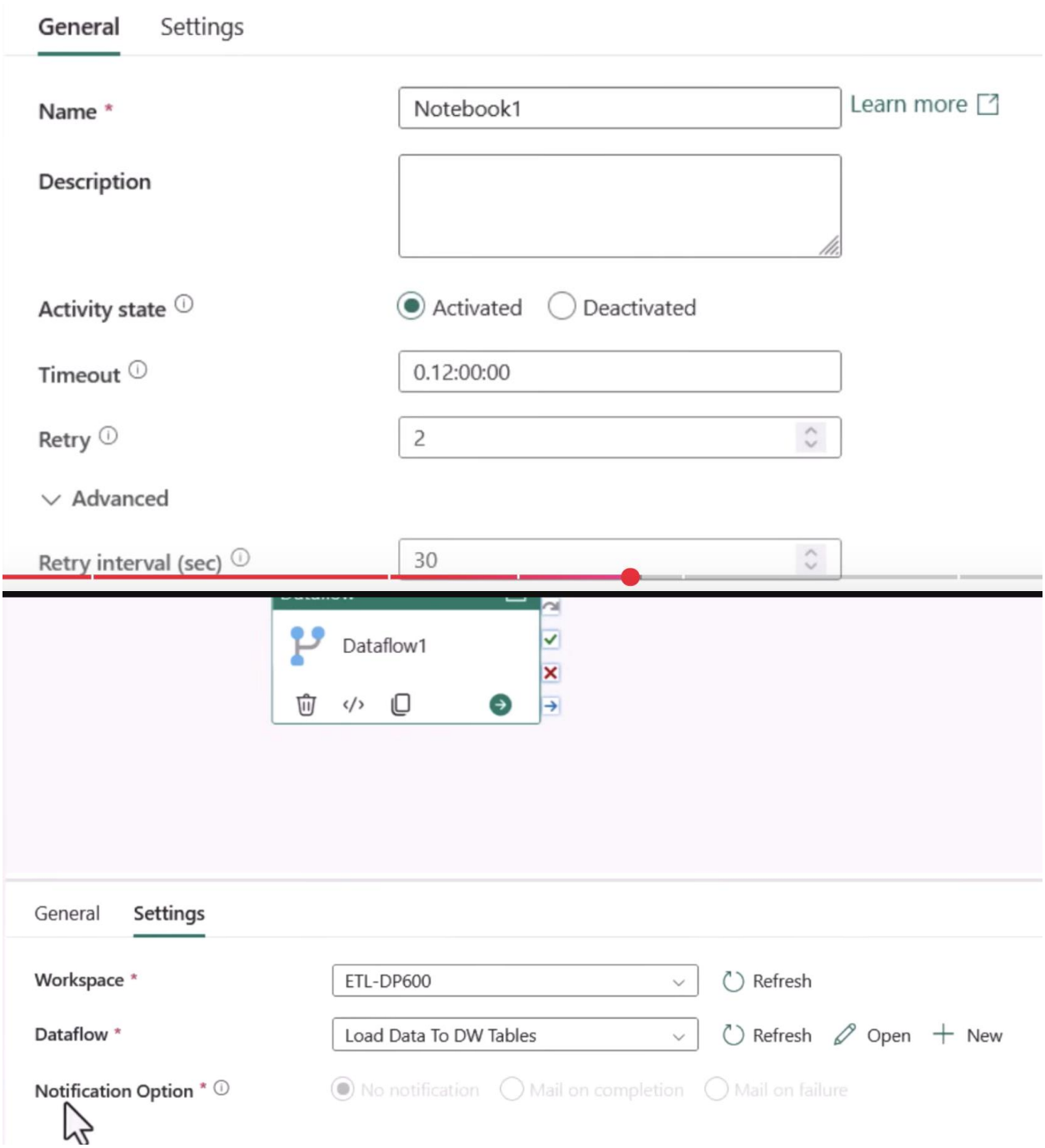






If spark cluster in the notebook code is busy, below retry option retries the code 2 times after retry interval of 30sec

Fabric Analytics Concepts Notes



Fabric Analytics Concepts Notes

The maximum number of scheduled refreshes allowed per day with a Dataflow Gen2 is...

A) 12 refreshes per day

B) 24 refreshes per day

C) 48 refreshes per day

D) 96 refreshes per day

E) Unlimited refreshes per day

Which of the following can you use to update a row in a Data Warehouse table?

A) Stored Procedure

B) View

C) Create Table

D) Function

Which of the following statements is **FALSE**, when talking about operations in a Fabric Data Warehouse:

A) You can call a function from a stored procedure

B) You can call a function from a view

C) You can query a view and another view

D) You can call a stored procedure from a function

You are orchestrating many Spark notebooks to perform data transformation activities at the same time.

Sometimes the notebook execution is failing because your cluster is busy.

What modification can you make to the data pipeline notebook activity to give the pipeline more chances to run successfully?

A) Deactivate and reactivate the activity

B) Set the number of retries to 2 or 3

C) Change the parameters you are passing into the notebook

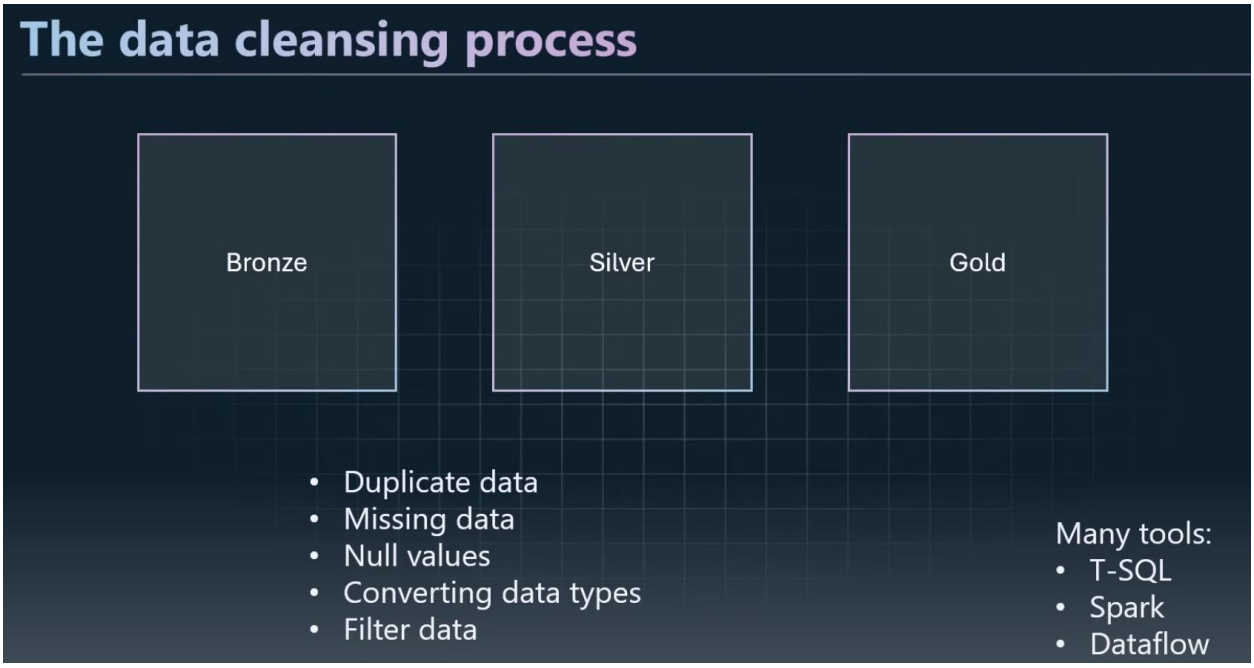
D) Add a Fail activity to handle the execution failure

Ch7. Transform Data

Data cleansing:
- Implement a data cleansing process
- Identify and resolve duplicate data, missing data, or null values
- Convert data types by using Dataflows or PySpark
- Filter data

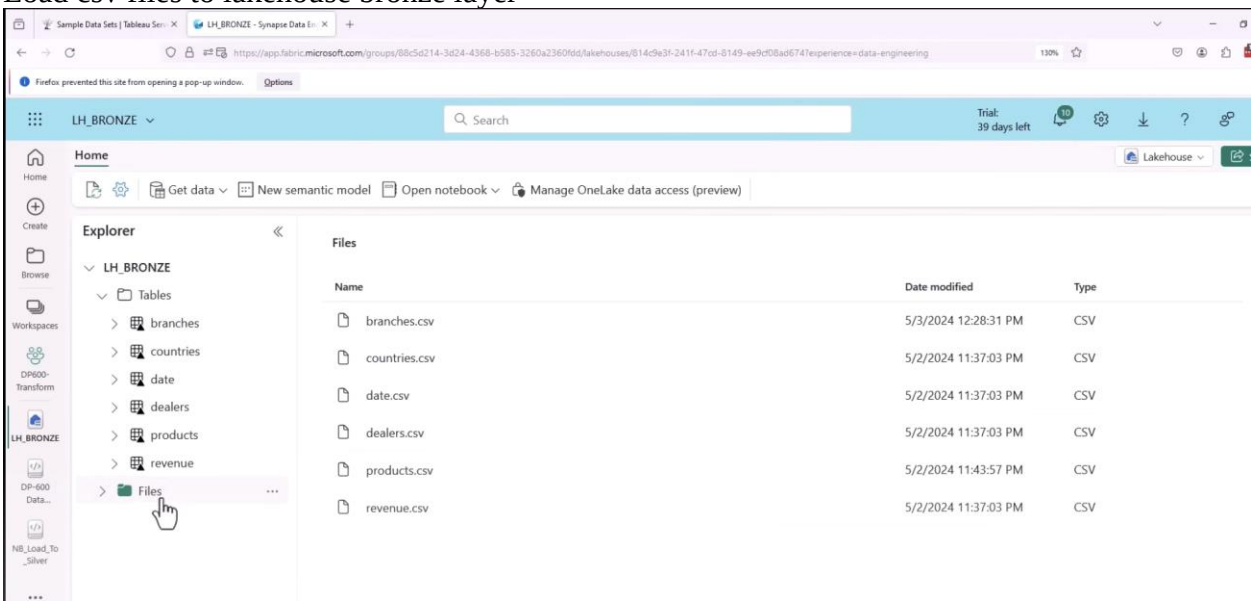
Data enrichment
- Merge or join data
- Enrich data by adding new columns or tables

Data modelling
- Implement a star schema for a lakehouse or warehouse, including Type 1 and Type 2 slowly changing dimensions
- Implement bridge tables for a lakehouse or a warehouse
- Denormalize data
- Aggregate or de-aggregate data

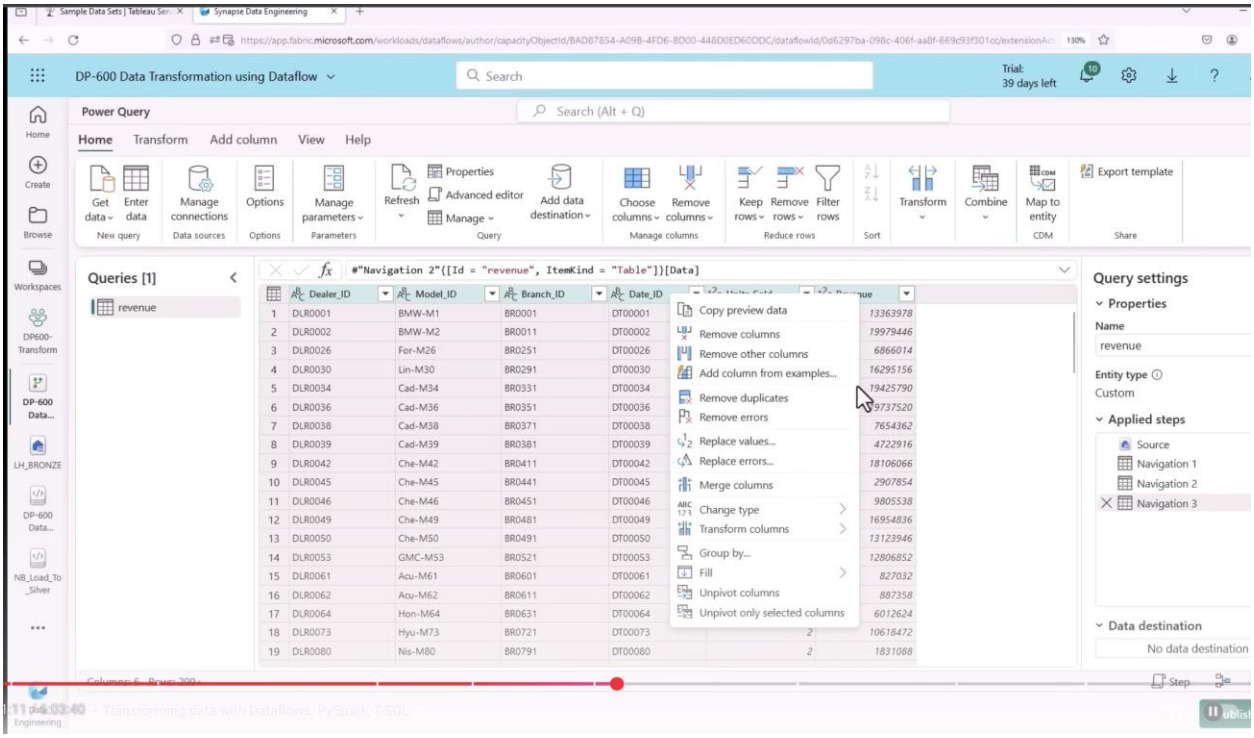


Fabric Analytics Concepts Notes

Load csv files to lakehouse bronze layer



Transformations in PowerQuery



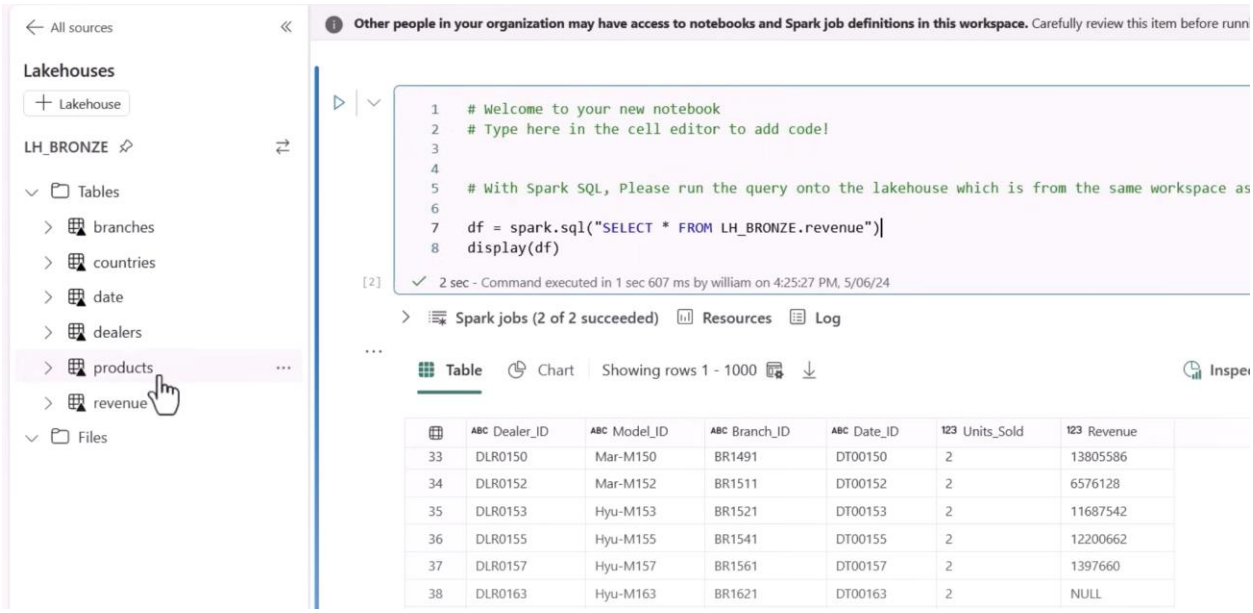
Transformations in SQL Endpoint

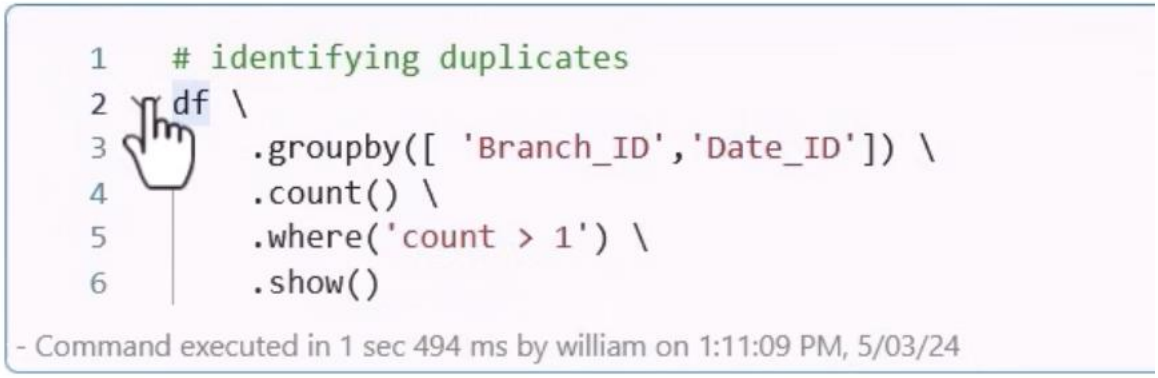
```sql
-- identifying/ removing nulls / filtering with WHERE

with removed_nulls as (
SELECT [Dealer_ID]
    ,[Model_ID]
    ,[Branch_ID]
    ,[Date_ID]
    ,[Units_Sold]
    ,[Revenue]
FROM [LH_BRONZE].[dbo].[revenue]
WHERE Revenue IS NOT NULL
)

select count(*) from removed_nulls
select count(*) from [LH_BRONZE].[dbo].[revenue]
```

```sql
-- type conversion / adding new columns
SELECT [Dealer_ID]
    , [Revenue]
    , CAST([Revenue] as FLOAT) as RevFloat
    , [Revenue] / 2 as HalfRevenue
FROM [LH_BRONZE].[dbo].[revenue]
```

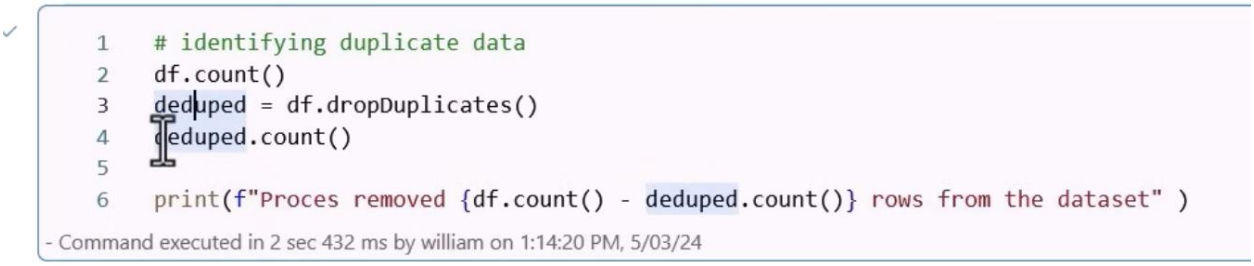
Fabric Analytics Concepts Notes

Transformations in PySpark





dropDuplicates()



Fabric Analytics Concepts Notes

Missing data/ nulls values

Identifying missing values in a column

```
1   # option 1: using isNull()
2   nulls = df.filter(df.Revenue.isNull())
3   display(nulls)
4
5   # option 2, using .where and col
6   from pyspark.sql.functions import col
7   nulls2 = df.where(col("Revenue").isNull())
8   display(nulls2)
9
```

Dropping nulls values using dropna()

See here for doumentation

Params explained:

- **how** - can be 'any' or 'all'. If 'any', drop a row if it contains any nulls. If 'all', drop a row only if all its values are null.
- **thresh** - drop rows that have less than thresh non-null values. This overwrites the how parameter.

```
1   no_nas = df.dropna(subset=['Revenue'])
2   print(f"Proces removed {df.count() - no_nas.count()} rows from the dataset" )
```

✓ 1 sec - Command executed in 796 ms by william on 4:32:36 PM, 5/06/24

> Spark jobs (5 of 5 succeeded) Resources Log

```
Proces removed 6 rows from the dataset
```

Fabric Analytics Concepts Notes

Type conversion + adding new columns

```
# take a look at the schema (including data types)
df.printSchema()

# using cast, create a new column
type_conv = df.withColumn('UnitsSoldConverted', df.Units_Sold.cast("string"))
type_conv.printSchema()
display(type_conv)
```

✓ 1 sec - Command executed in 849 ms by william on 4:34:35 PM, 5/06/24

Spark jobs (2 of 2 succeeded) Resources Log

```
root
 |-- Dealer_ID: string (nullable = true)
 |-- Model_ID: string (nullable = true)
 |-- Branch_ID: string (nullable = true)
 |-- Date_ID: string (nullable = true)
 |-- Units_Sold: integer (nullable = true)
 |-- Revenue: integer (nullable = true)

root
 |-- Dealer_ID: string (nullable = true)
 |-- Model_ID: string (nullable = true)
 |-- Branch_ID: string (nullable = true)
 |-- Date_ID: string (nullable = true)
 |-- Units_Sold: integer (nullable = true)
```

Filter data

```
from pyspark.sql.functions import col

# df.filter()
filtered_df = df.where(col("Revenue") > 10000000)
print(f"Proces removed {df.count() - filtered_df.count()} rows from the dataset" )
```

- Command executed in 1 sec 457 ms by william on 1:56:35 PM, 5/03/24

Part 2) Data enrichment

Adding new columns (withColumn or withColumns)

```
# using withColumn to create new columns.
enriched_df = df.withColumn('halfRevenue', df.Revenue/2)
display(enriched_df)
```

- Command executed in 1 sec 502 ms by william on 3:01:10 PM, 5/03/24

Joining and merging

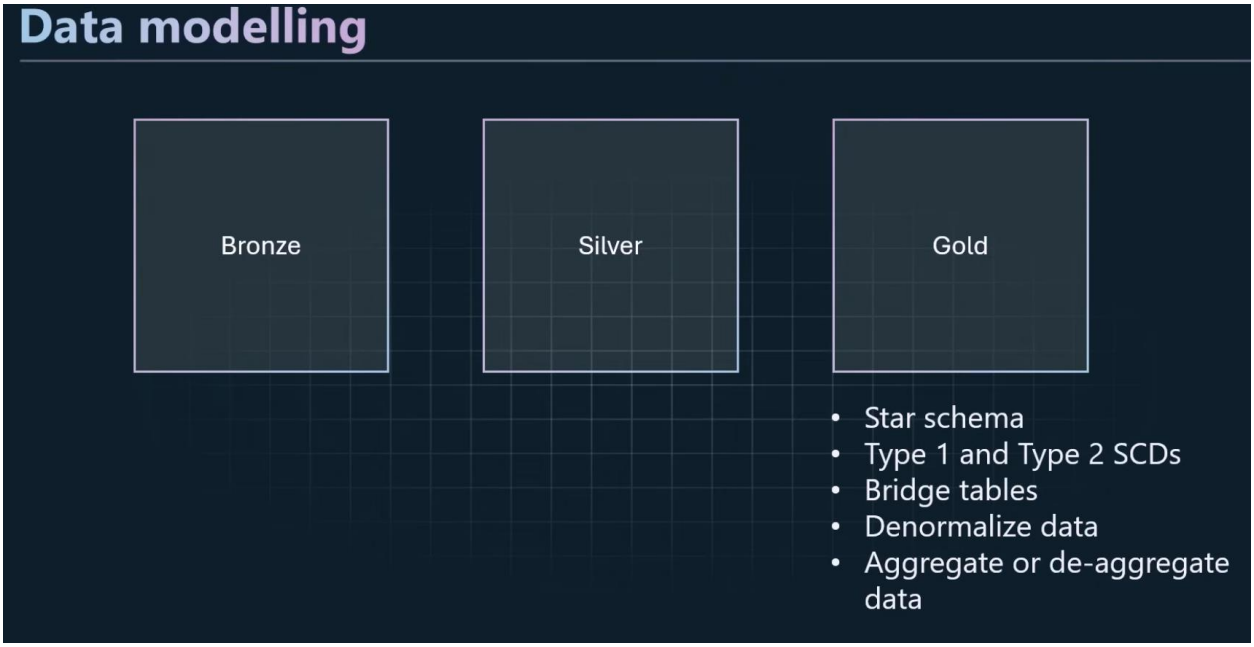
```
dealers_df = spark.sql("SELECT * FROM LH_BRONZE.dealers")
display(dealers_df)

countries_df = spark.sql("SELECT * FROM LH_BRONZE.countries")
display(countries_df)


```

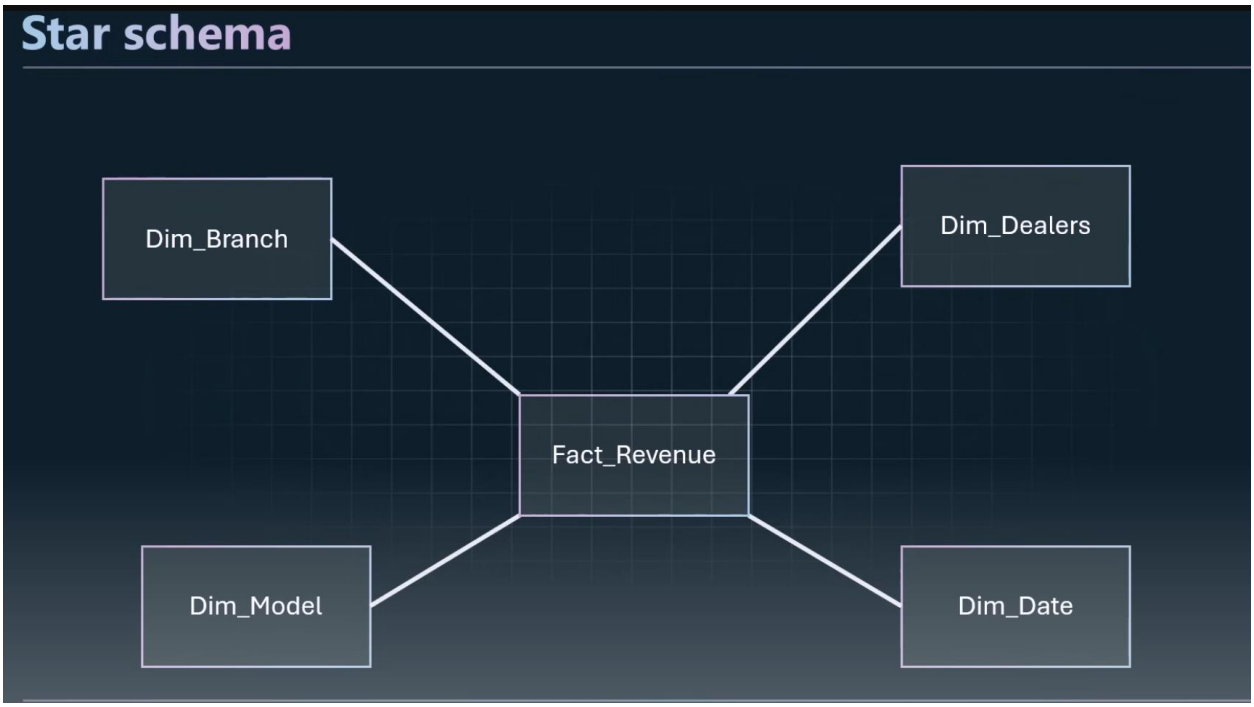
- Command executed in 4 sec 949 ms by william on 11:18:06 AM, 5/06/24

```
joined_df = (
    dealers_df
        .join(countries_df, dealers_df.Country_ID == countries_df.Country_ID)
        .select(dealers_df.Dealer_ID, dealers_df.Country_ID, countries_df.Country_Name)
    )
display(joined_df)
```

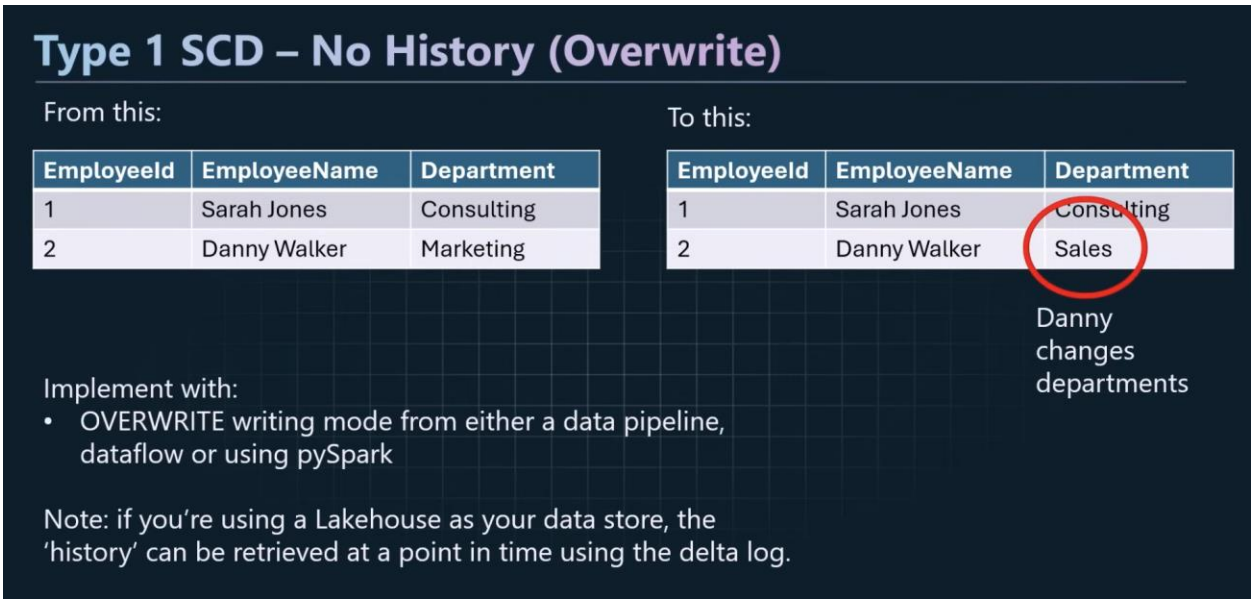
Fabric Analytics Concepts Notes

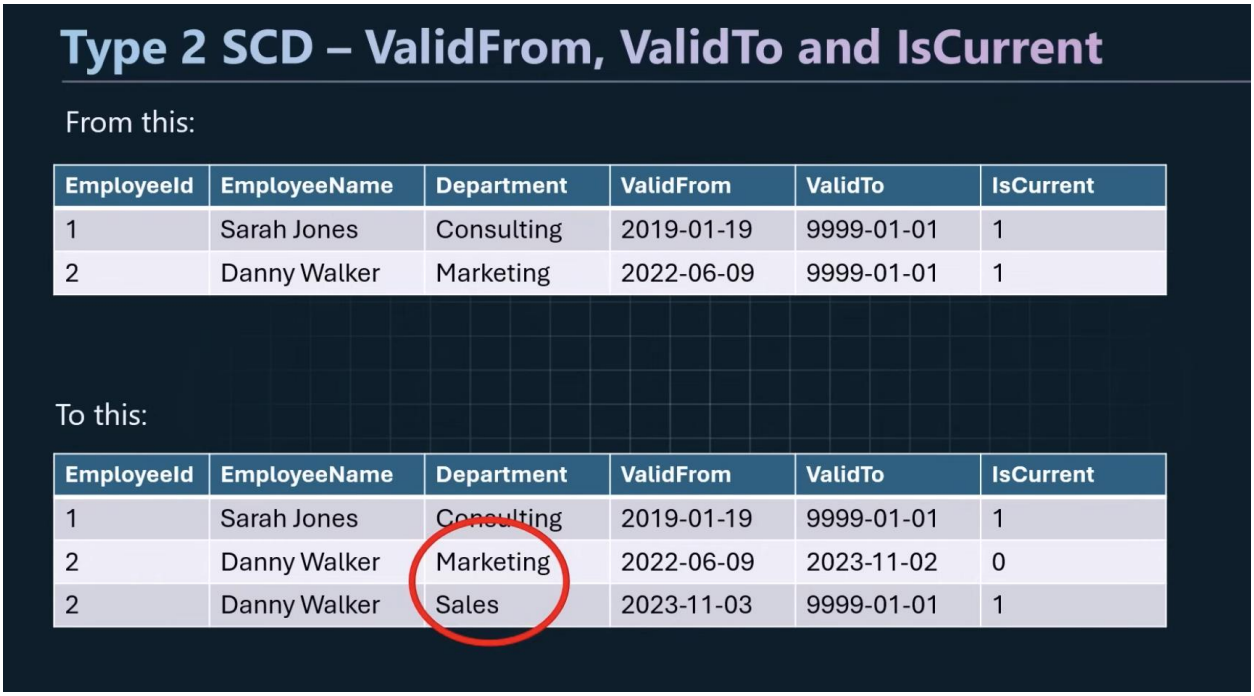
Data Modeling

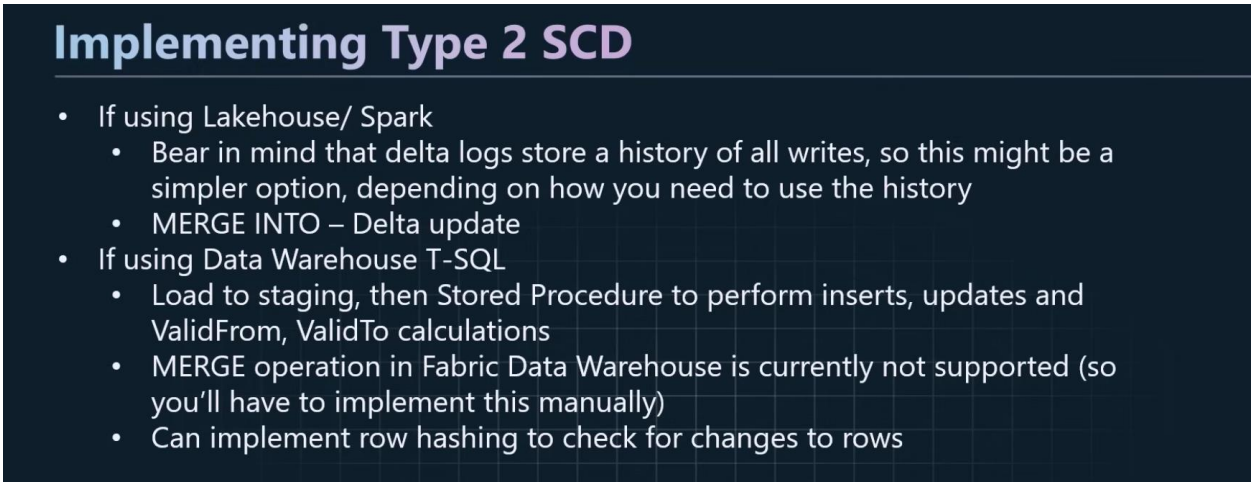




SCD


Type 1 SCD – No History (Overwrite)
From this:
EmployeeId | EmployeeName | Department
1 | Sarah Jones | Consulting
2 | Danny Walker | Marketing
To this:
EmployeeId | EmployeeName | Department
1 | Sarah Jones | Consulting
2 | Danny Walker | Sales
Danny changes departments
Implement with:
• OVERWRITE writing mode from either a data pipeline, dataflow or using pySpark
Note: if you're using a Lakehouse as your data store, the 'history' can be retrieved at a point in time using the delta log.


Type 2 SCD – ValidFrom, ValidTo and IsCurrent
From this:
EmployeeId | EmployeeName | Department | ValidFrom | ValidTo | IsCurrent
1 | Sarah Jones | Consulting | 2019-01-19 | 9999-01-01 | 1
2 | Danny Walker | Marketing | 2022-06-09 | 9999-01-01 | 1
To this:
EmployeeId | EmployeeName | Department | ValidFrom | ValidTo | IsCurrent
1 | Sarah Jones | Consulting | 2019-01-19 | 9999-01-01 | 1
2 | Danny Walker | Marketing | 2022-06-09 | 2023-11-02 | 0
2 | Danny Walker | Sales | 2023-11-03 | 9999-01-01 | 1



SCD Type 2 vs. Type 3: Key Differences

Feature	Type 2 SCD	Type 3 SCD
History Tracking	Full history (new row for each change).	Limited history (only current + previous value).
Storage Impact	High (adds rows over time).	Low (adds columns, fixed row count).
Implementation	Uses `ValidFrom` / `ValidTo` dates + `IsCurrent` flag.	Adds `PreviousValue` columns (e.g., `PreviousDept`).
Query Complexity	Requires time-based joins for historical data.	Simpler (current + previous value in one row).
Best For	Auditing, compliance, full historical analysis.	Short-term trend analysis (e.g., "last change").

Example: Employee Department Changes

SCD Type 2 (Full History)

EmployeeID	Name	Department	ValidFrom	ValidTo	IsCurrent
101	Alice	Sales	2023-01-01	2023-05-31	False
101	Alice	Marketing	2023-06-01	NULL	True

SCD Type 3 (Limited History)

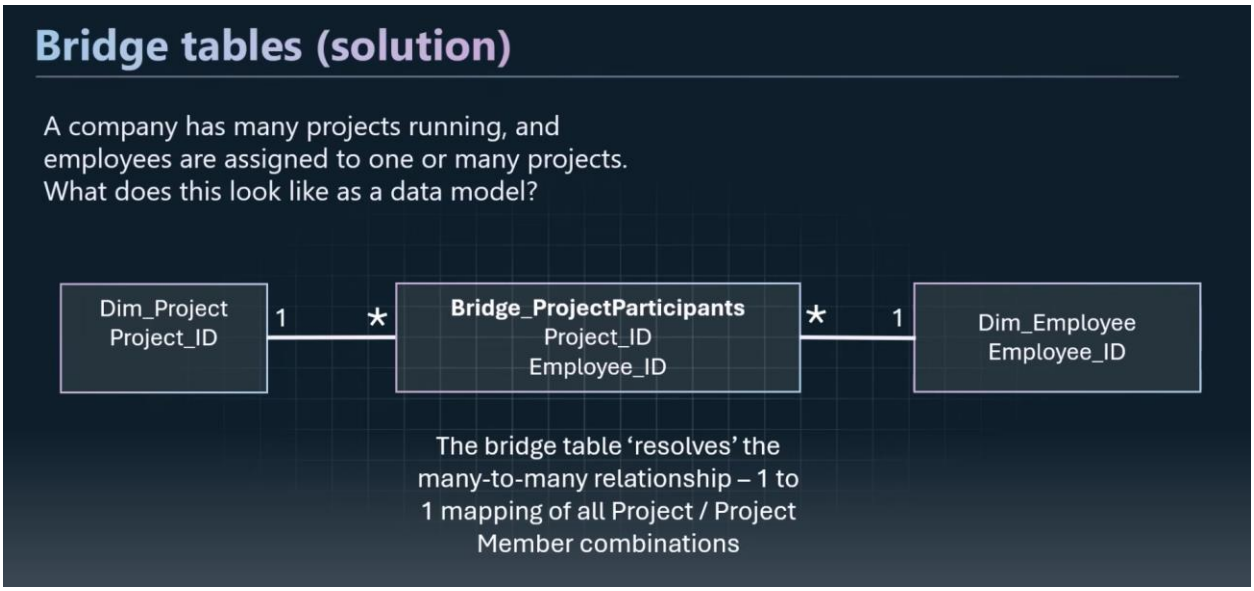
EmployeeID	Name	CurrentDept	PreviousDept	LastChangeDate
101	Alice	Marketing	Sales	2023-06-01

Bridge tables (problem)

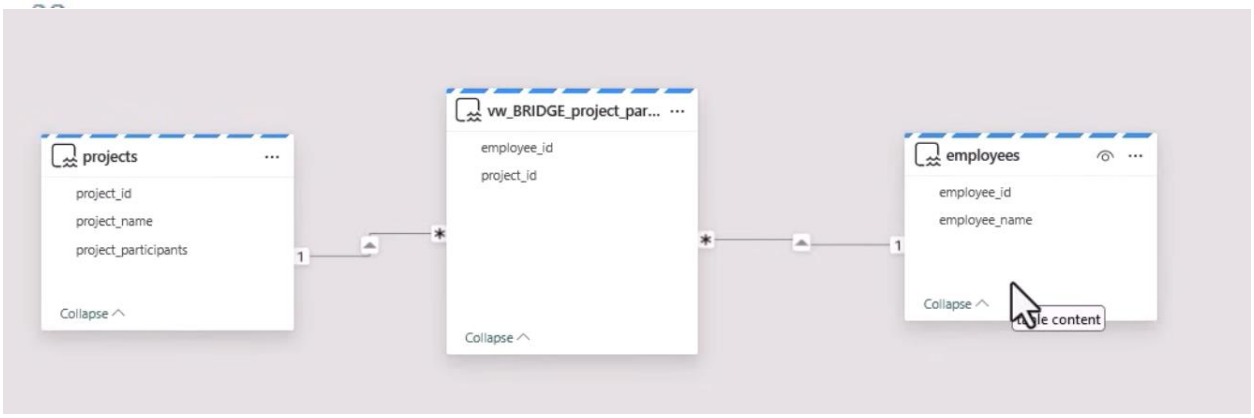
A company has many projects running, and employees are assigned to one or many projects. What does this look like as a data model?

A many-to-many relationship exists between Dim_Projects and Dim_Employee

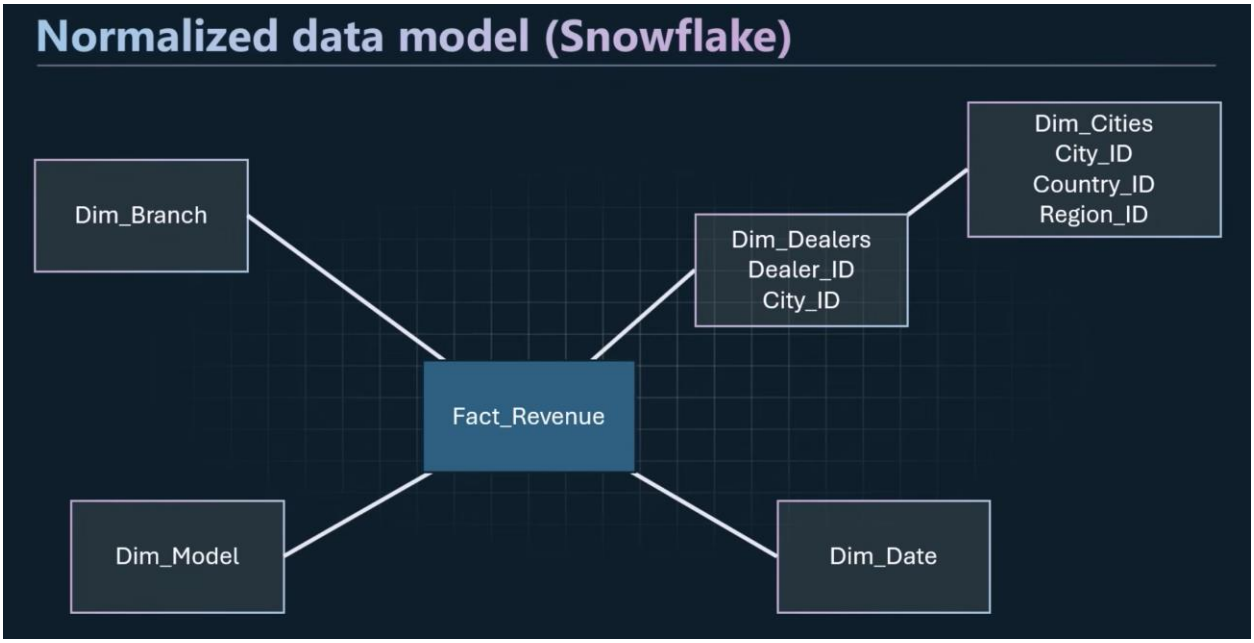
Fabric Analytics Concepts Notes



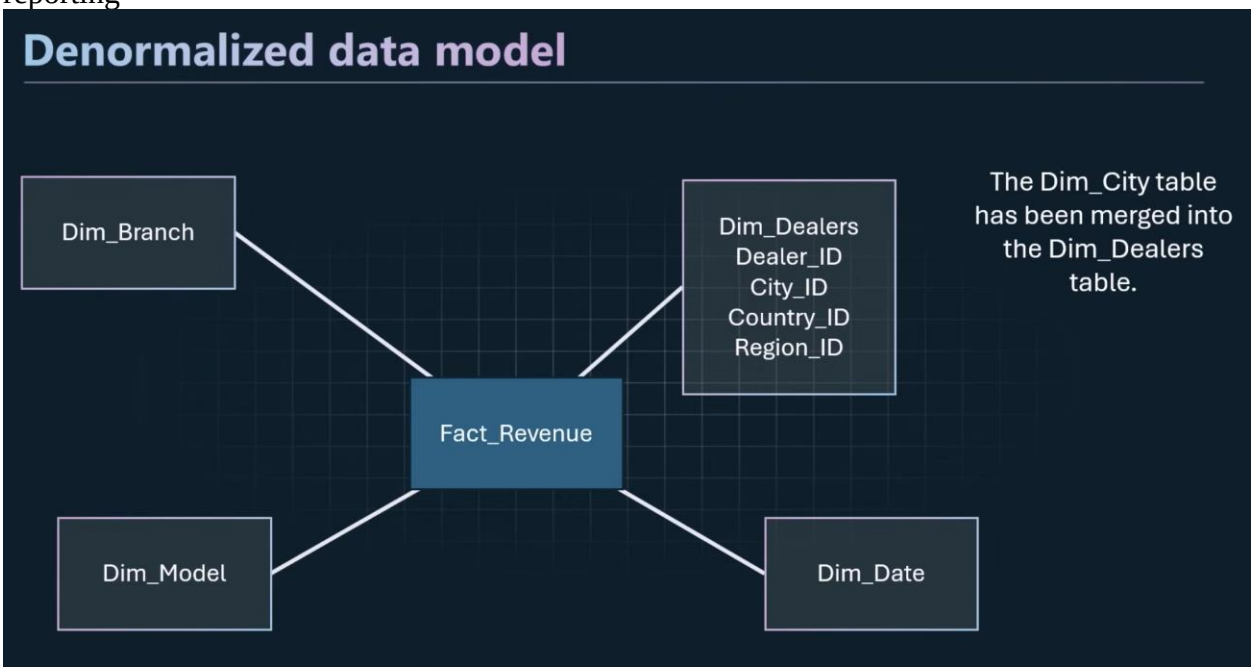

```
21
22
23  create view dbo.vw_BRIDGE_project_participants as
24  SELECT project_id, CAST(value as INT) as employee_id
25  FROM dbo.projects
26      CROSS APPLY STRING_SPLIT(project_participants, ',');
27
```



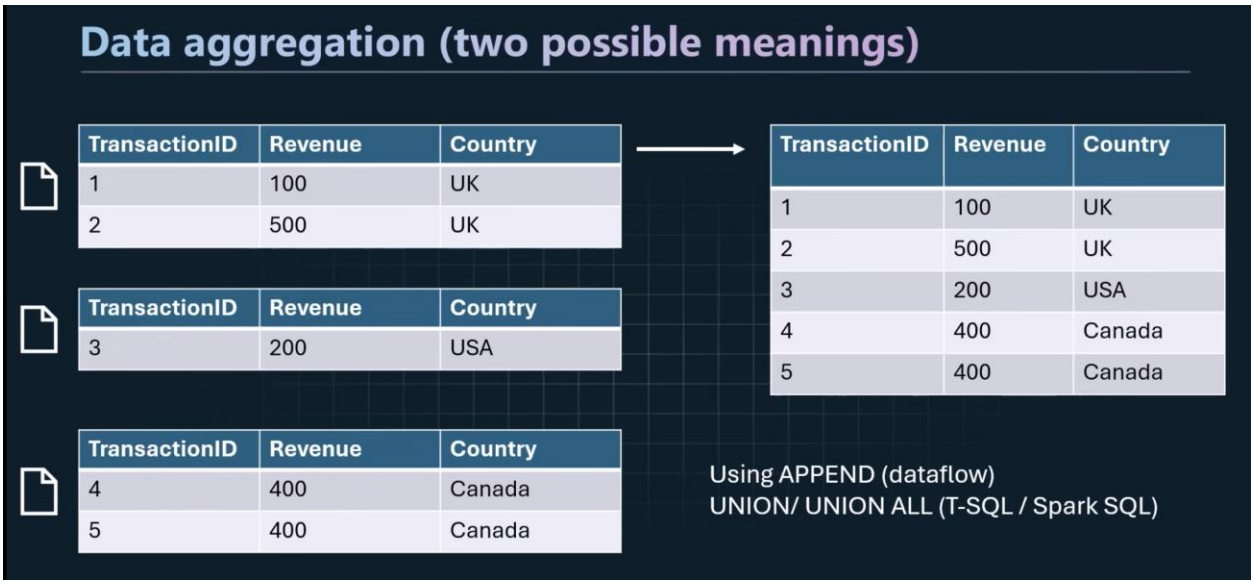
Fabric Analytics Concepts Notes



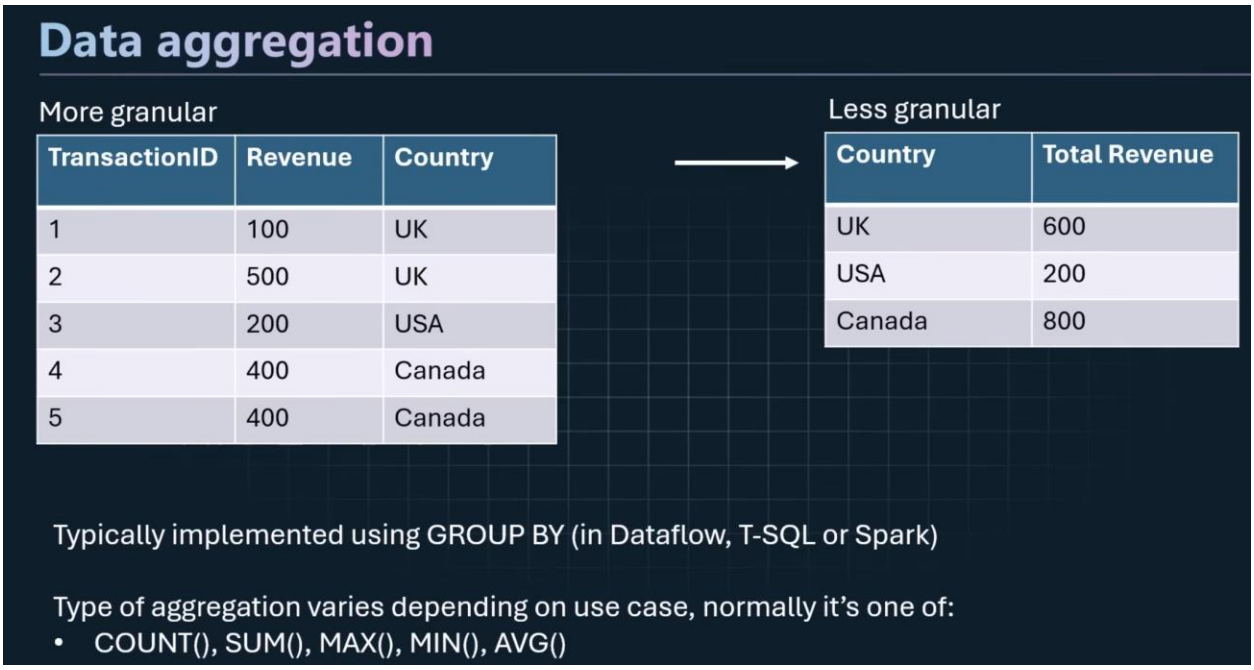
Denormalized from snowflaske to star schema for efficiency (although redundant) in powerbi reporting



Fabric Analytics Concepts Notes



Union – appends all datasets (no chk for duplicates)
Union – removes any duplicate rows in result dataset



Question

When using df.join() in a pySpark notebook, the default join type

A) FULL OUTER JOIN

B) INNER JOIN

C) LEFT JOIN

D) RIGHT JOIN

E) ANTI JOIN

JOIN Types

DataFrames

Left DF (Customers)	Right DF (Orders)
`ID: A, Name: Alice`	`ID: B, Order: 01`
`ID: B, Name: Bob`	`ID: C, Order: 02`
`ID: C, Name: Carol`	`ID: D, Order: 03`

LEFT JOIN Result

```python
customers.join(orders, "ID", "left").show()
```

Copy

ID	Name	Order	
A	Alice	NULL	← "NULL for A" (no matching order)
B	Bob	01	
C	Carol	02	

```
Left DF: [A, B, C]    Right DF: [B, C, D]

INNER:    [B, C]
LEFT:     [A, B, C] + NULL for A
RIGHT:    [B, C, D] + NULL for D
FULL:     [A, B, C, D] + NULLs for A/D
ANTI:     [A]
SEMI:     [B, C] (no duplicates)
```

Question

You are looking to migrate a data transformation workload that currently is done using a Dataflow Gen2. You need to convert the same transformation into a T-SQL script.

The Dataflow appends two datasets together and removes any duplicate rows.

Which T-SQL command can you use to implement this transformation?

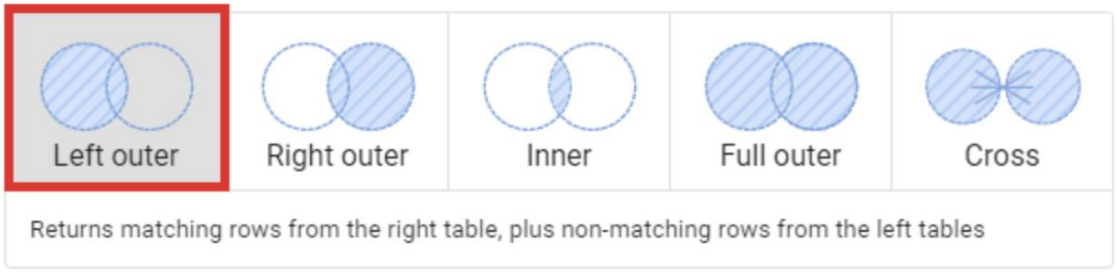
A) LEFT JOIN

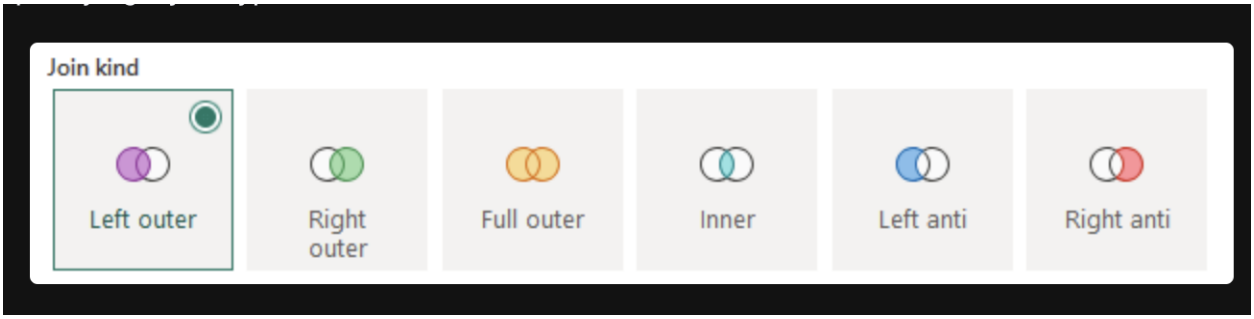
B) UNION ALL

C) CONCAT

D) APPEND

E) UNION





Question

You have a Spark Dataframe (df). Your goal is to remove rows that contain a null value in the TransactionDate column.

Which of the following will help you achieve this?

A) df.dropDuplicates()

B) df.dropna(how='all')

C) df.filter(df.TransactionDate.isNull())

D) df.dropna(how='any')

E) df.dropna(subset=['TransactionDate'])

Why the Other Options Are Wrong

Option	Problem	What It Actually Does
A) `df.dropDuplicates()`	❌ Doesn't handle nulls.	Removes duplicate rows (ignores nulls).
B) `df.dropna(how='all')`	❌ Too broad.	Only drops rows where **all** columns are null (your goal is to target one column).
C) `df.filter(df.TransactionDate.isNull())`	❌ Opposite effect.	**Keeps** rows with nulls (instead of removing them). Use `isNotNull()` to fix.
D) `df.dropna(how='any')`	❌ Overkill.	Drops rows if **any** column has a null (not just `TransactionDate`).

Question

You inherit a data project and are inspecting the tables in the data warehouse.

One of the tables is a dimension table (Dim_Contacts), with the following columns:
- Contact_ID
- Contact_Name
- Contact_Address
- Effective_Date
- Effective_Until

Make an assumption about the type of data modelling that is being implemented in this dimension table

A) Type 0 SCD

B) Type 1 SCD

C) Type 2 SCD

D) Type 3 SCD

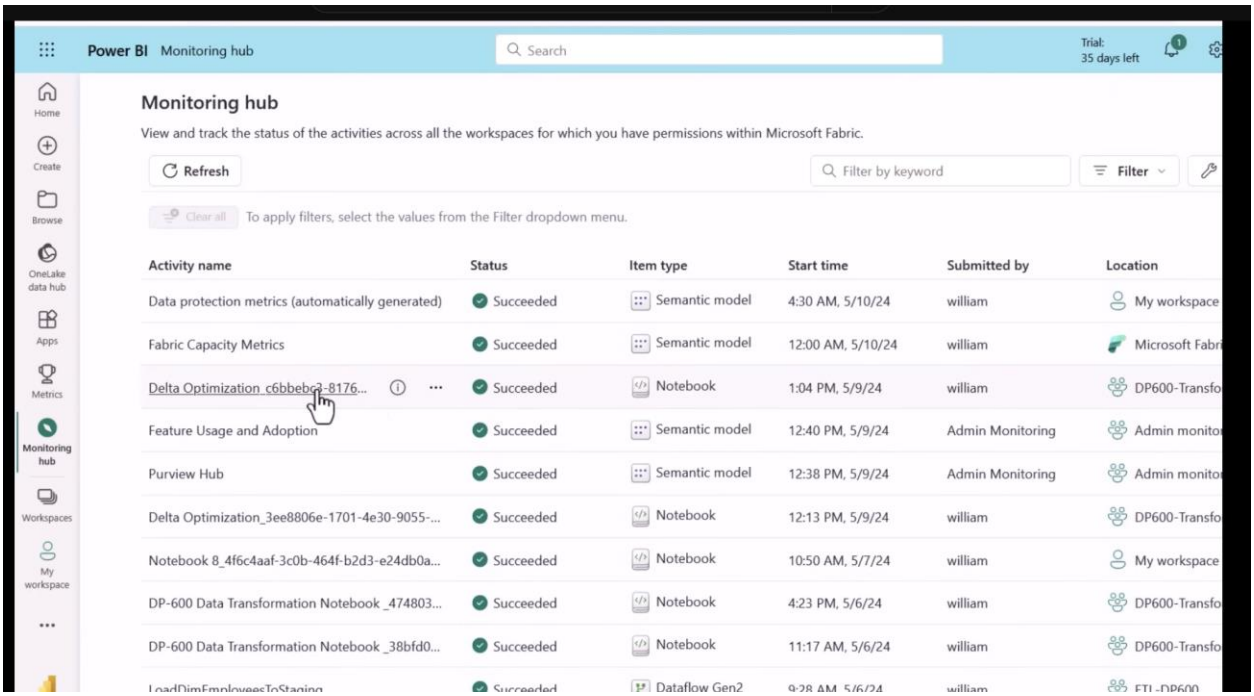
Ch8. Optimizing Performance

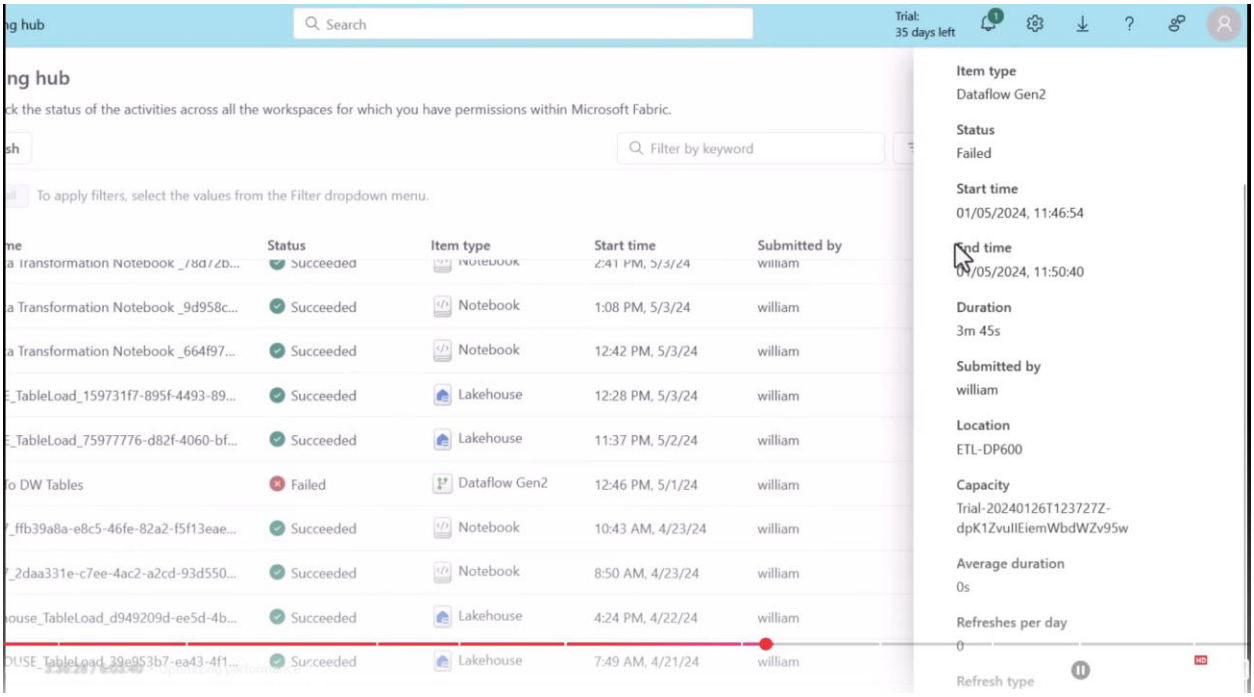
Framing performance optimization

	Identify performance issues	Possible methods to resolve issues
Dataflow	Refresh History Monitoring Hub Capacity Metrics App	Refactoring Staging Fast copy
SQL Data Warehouse	Query Insights DMVs Capacity Metrics App Statistics & Query Plan (in future)	Refactoring/ Query optimization
Notebook (Spark)	Spark History Server Monitoring Hub Capacity Metrics App	Refactoring/ Query optimization
Delta files	DESCRIBE	V-Order Optimization File partitioning VACUUM & OPTIMIZE

See logs

Fabric Analytics Concepts Notes



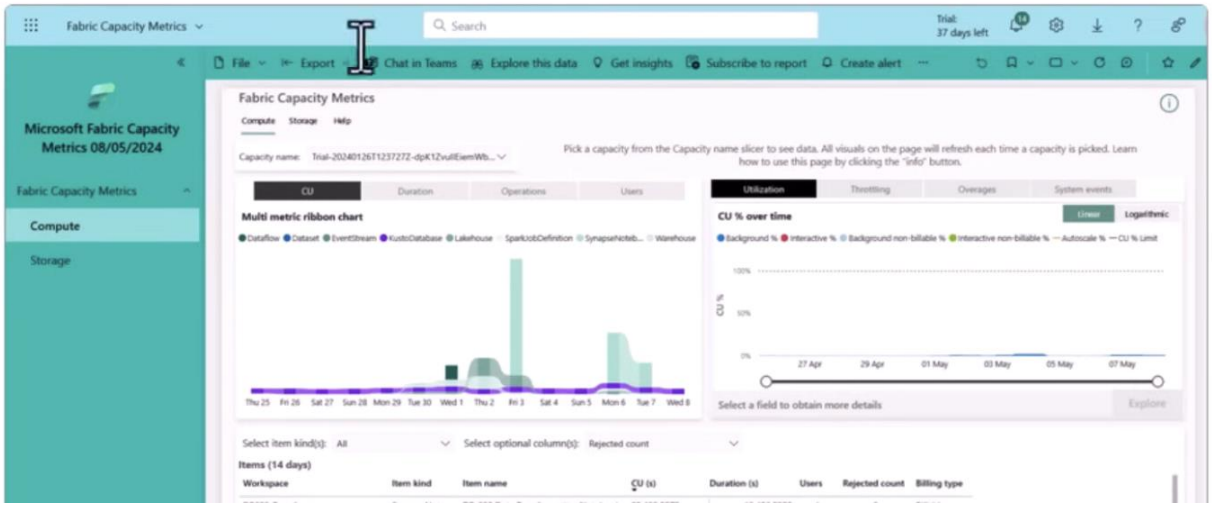


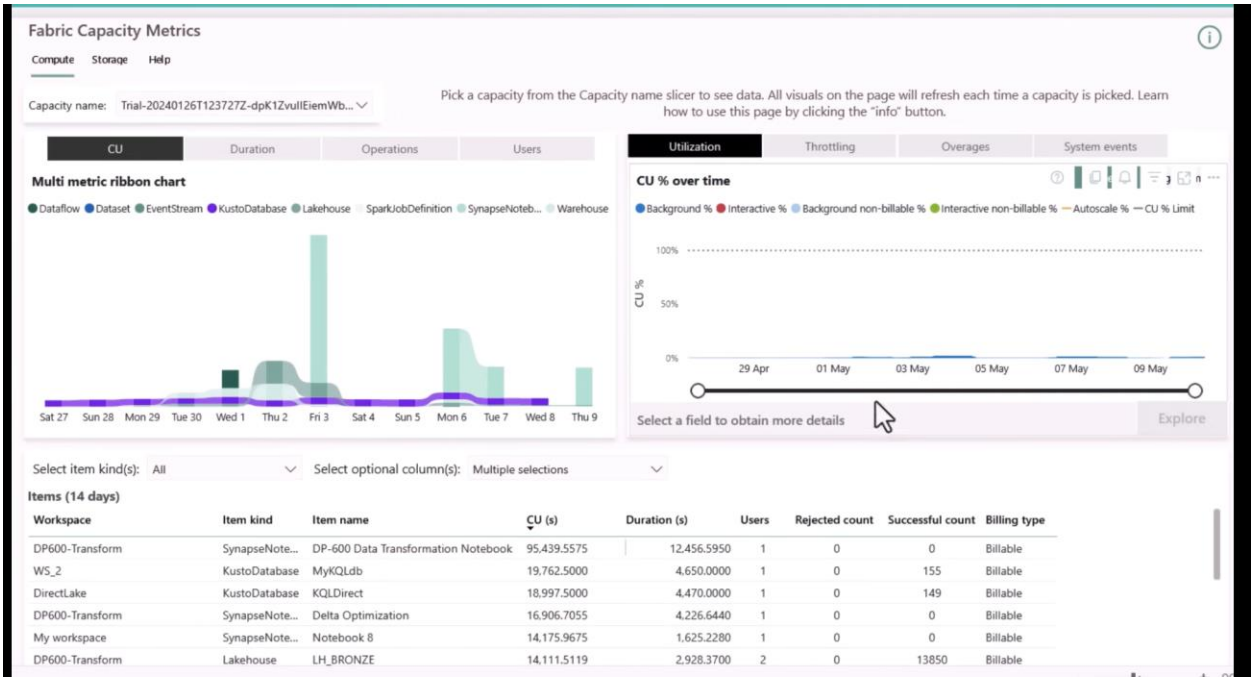
Fabric Analytics Concepts Notes

Capacity Metrics App

The Capacity Metrics App (install instructions), is a Power BI App that you can download (for free) and install onto your Fabric tenant.

You give it your Capacity ID (you can find this in the Capacity Settings part of the Admin Portal), and it retrieves capacity usage statistics like so:





Dataflows + performance

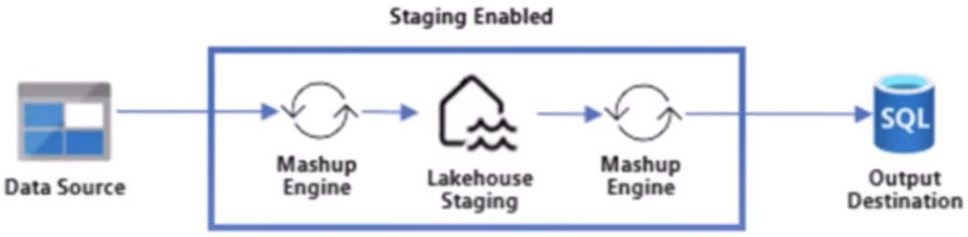
Performance monitoring (Dataflows)

- High-level summary of a dataflow run is available in the **Monitoring Hub**.
- Lower-level data and understanding of what is happening in a particular dataflow is available in the Refresh History of a particular dataflow.
    - You can inspect error messages,
    - Get a breakdown of the different sub-activities being performed by the Dataflow.

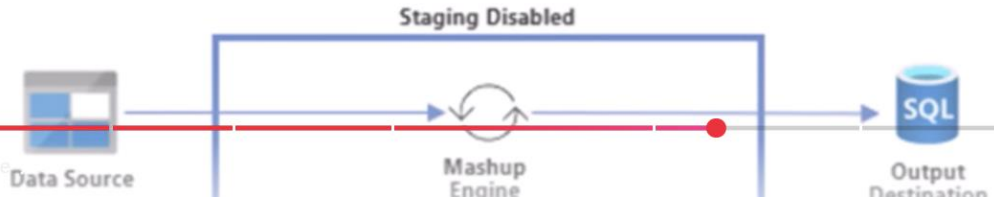
Performance optimization/ features:

Staging - what is staging? By default, transformations in a dataflow are carried out by the PowerQuery engine (also known as the Mashup engine), this is relatively slow, compared to the Spark engine. So staging involves using the Mashup engine to first get the data from source, then it writes the data out to a Staging Lakehouse, performs the transformation using the Spark engnie (rather than the Mashup engine), then the Mashup engine reads the result and then writes to data to destination.

Behind the scenes, controlling the staging setting of a query will change orchestration from its default setting:



To orchestration that extracts, transforms, and loads data from source to destination in memory:



The added overhead of writing the data out to a Lakehouse (and then back) adds considerable overhead, so Staging is only ever more efficient when you've got large datasets, or lots of transformation steps (or both).

Fast Copy (preview) - a new feature to speed up getting new data from sources (but not on-premise). I believe it uses the same technology as the Data Pipeline CopyData activity. Still a preview feature, and relatively recent, so might not be on the exam yet.

More resources:

- Spotlight blog on Dataflow (including performance optimization)
- Best practices for complex dataflows

SQL Data Warehouse + Performance

So you've got a really long-running SQL query in Data Warehouse. What to do about it?? There's a few things we can do:

1) Monitor using the Capacity Metrics App (see here)

The Capacity Metrics App can be used to see how much CU(s) has been used for different read/write queries you've executed in the Data Warehouse.

2) Dynamic Management Views (DMVs)

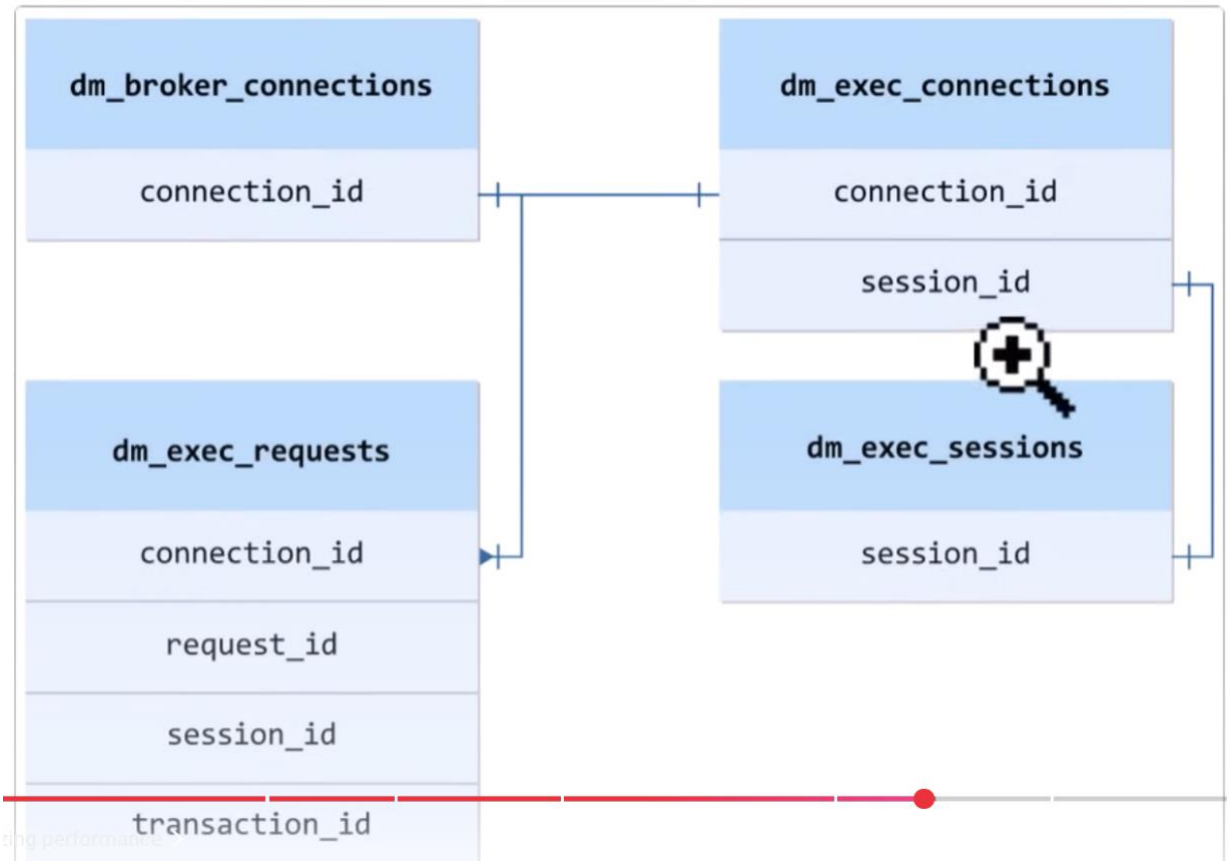
For the current version, there are three dynamic management views (DMVs) provided for you to receive live SQL query lifecycle insights.

- sys.dm_exec_connections
    - Returns information about each connection established between the warehouse and the engine.
- sys.dm_exec_sessions
    - Returns information about each session authenticated between the item and engine.

Fabric Analytics Concepts Notes

- sys.dm_exec_requests
  - Returns information about each active request in a session.

How these queries are related:



These three DMVs provide detailed insight on the following scenarios:

- Who is the user running the session?
- When was the session started by the user?
- What's the ID of the connection to the data Warehouse and the session that is running the request?
- How many queries are actively running?
- Which queries are long running?

DMV are low level logs

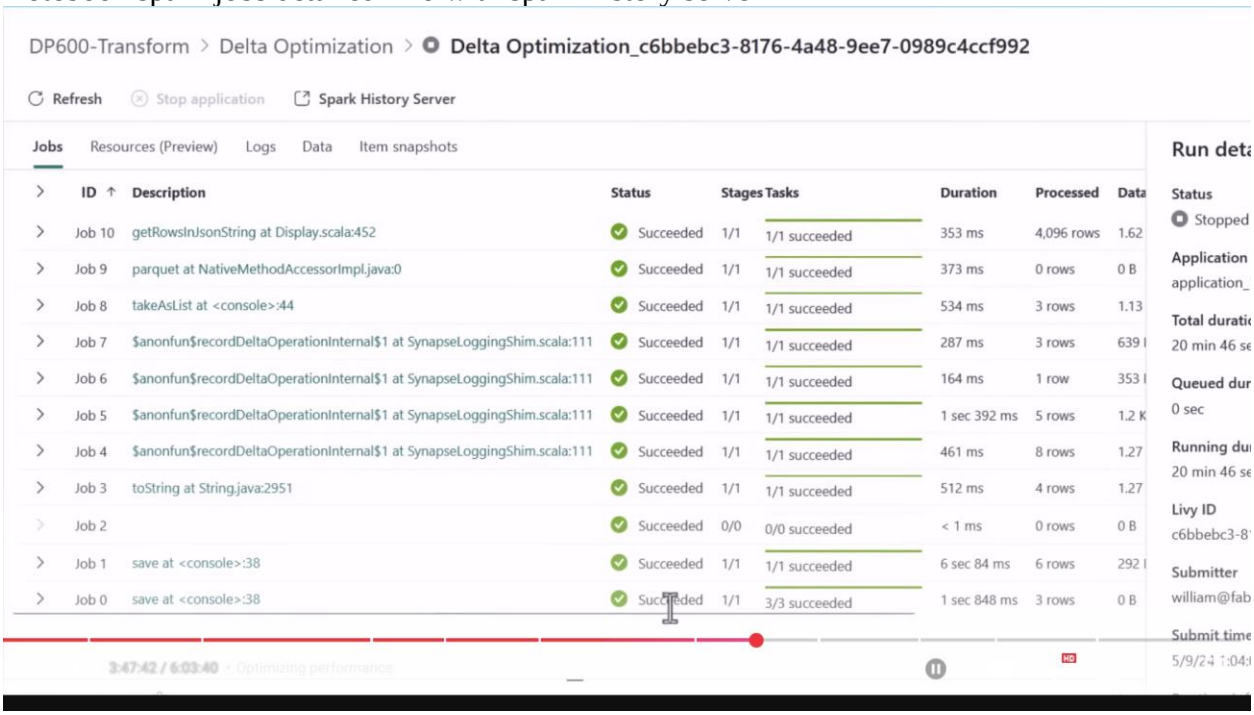
Fabric Analytics Concepts Notes

Queryinsights is a schema exposed in every Data Warehouse, it consists of four main views, all of which are useful for performance monitoring:

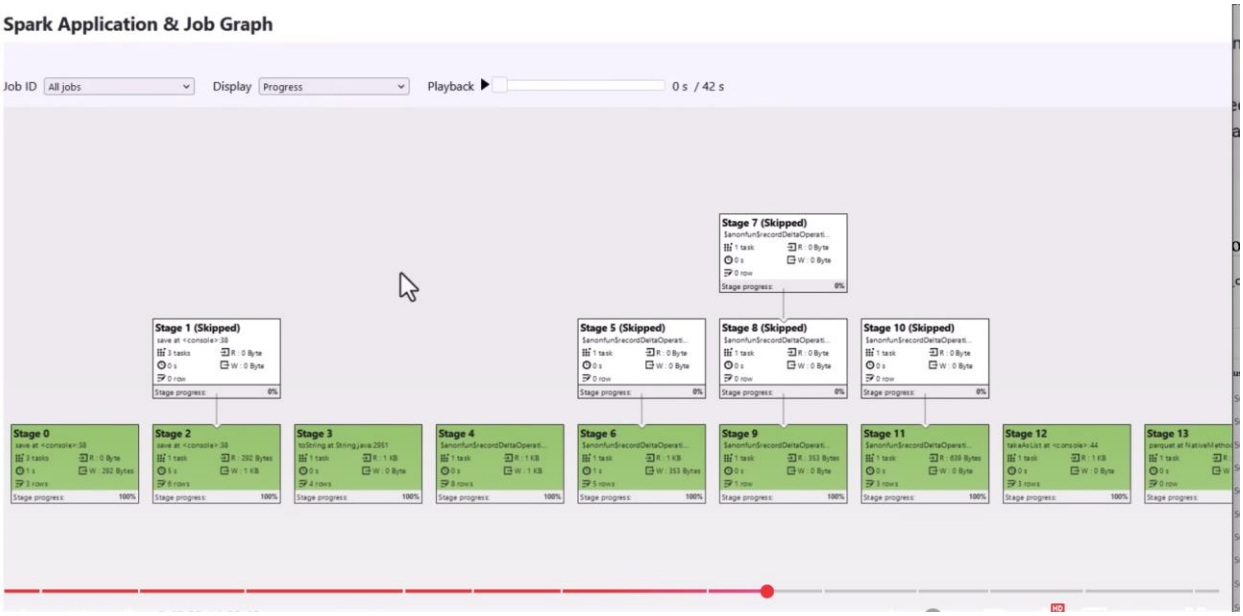
- exec_requests_history: Returns information about each completed SQL request/query.
- frequently_run_queries: Returns information about frequently run queries.
- long_running_queries: Returns the information about queries by query execution time.

> Note: it's not currently possible to view the Execution/ Query Plan for SQL query in the Fabric Data Warehouse - although there are plans to support/ expose this in the future.

Notebook spark jobs detailed info with spark history server



Fabric Analytics Concepts Notes



Delta Table Optimization

Fabric is built on delta tables, so this is a pretty important topic. The delta file format is great, but it can lead to poor performance and bloated storage sizes, if not managed correctly.

This is a topic that can run very deep, but we'll just cover some of the basics for the exam.

Download and run the Spark notebook below

In this notebook, we will first look at file partitioning, then we will look at other methods for optimizing delta table (and Spark) performance.

This can improve:

- read/ retrieval performance (no need to scan through millions of rows to find the record(s) you need)
- transformation performance (partitions can be transformed in parallel)

A client you're working with wants to reduce the SKU of their Fabric Capacity from an F16 to an F8 to save money.

They want to find the most resource-intensive workloads and optimize them to use less CU(s).

Where should they look to find this information?

A) Monitoring Hub

B) Capacity Metrics App

C) Query Insights

D) Spark History Server

E) OneLake Hub

Fabric Analytics Concepts Notes

Correct Answer: B) Capacity Metrics App

Why?

- The **Capacity Metrics App** provides detailed insights into **Capacity Unit (CU) consumption** across all workloads (Power BI, Data Factory, Spark, etc.).
- Key metrics:
  - **CU usage per item/workspace** (identify high-cost queries).
  - **Peak utilization times** (optimize scheduling).
  - **Resource-heavy operations** (e.g., long-running Spark jobs).

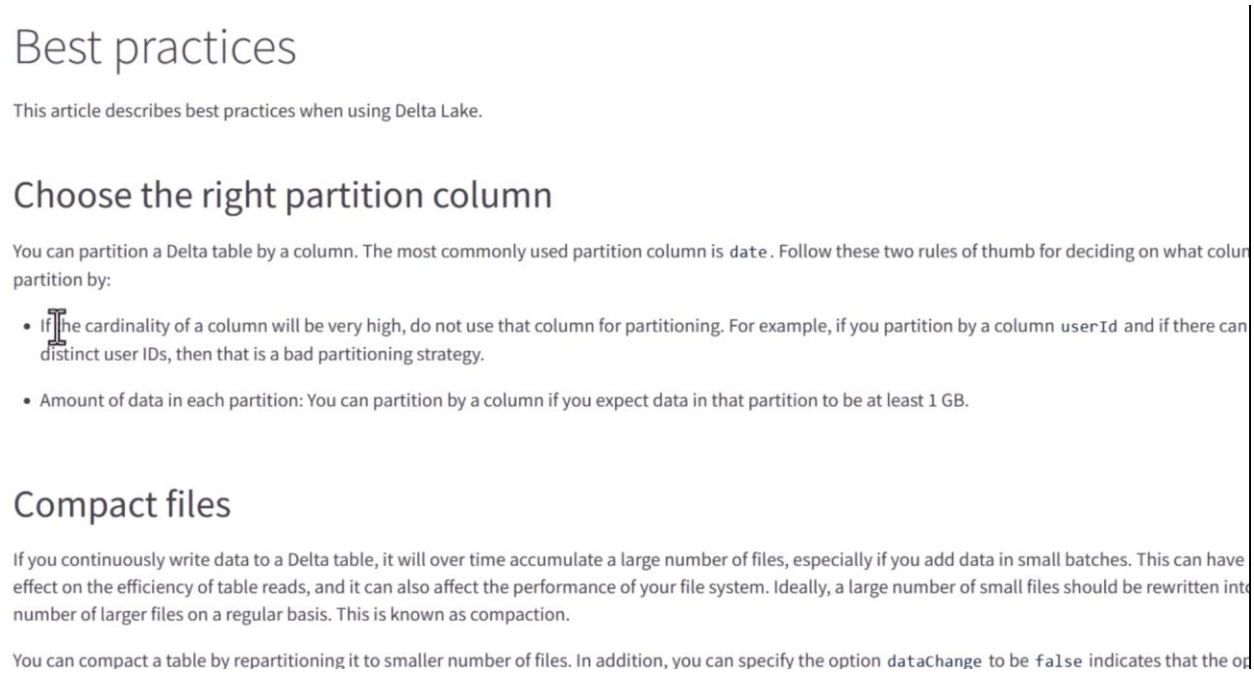
How to Access:

1. Go to your Fabric **admin portal**.
2. Navigate to **Capacity Settings → Metrics App**.

Why Not the Others?

Option	Why It's Not the Best Fit
A) Monitoring Hub	Shows activity logs (who did what) but **not CU consumption**.
C) Query Insights	Optimizes **individual queries** (Power BI), not overall capacity planning.
D) Spark History Server	Limited to **Spark job logs** (ignores other workloads like Dataflows).
E) OneLake Hub	Manages **data storage**, not compute resources.

Fabric Analytics Concepts Notes



Best practices

This article describes best practices when using Delta Lake.

Choose the right partition column

You can partition a Delta table by a column. The most commonly used partition column is `date`. Follow these two rules of thumb for deciding on what colu
partition by:

- If the cardinality of a column will be very high, do not use that column for partitioning. For example, if you partition by a column `userId` and if there can
distinct user IDs, then that is a bad partitioning strategy.
- Amount of data in each partition: You can partition by a column if you expect data in that partition to be at least 1 GB.

Compact files

If you continuously write data to a Delta table, it will over time accumulate a large number of files, especially if you add data in small batches. This can have
effect on the efficiency of table reads, and it can also affect the performance of your file system. Ideally, a large number of small files should be rewritten int
number of larger files on a regular basis. This is known as compaction.

You can compact a table by repartitioning it to smaller number of files. In addition, you can specify the option `dataChange` to be `false` indicates that the o

High cardinality refers to a column or field in a database that contains a large number of unique values (close to the total number of rows). Ex: Employee ID,

Write data to delta (three ways)

```
# write WITHOUT Partitions, first
transformed_df.write.mode("overwrite").format("delta").save("Tables/flights_not_partitioned")

# then write WITH partitions
transformed_df.write.mode("overwrite").format("delta").partitionBy("year", "month").save("Tables/flights_partitioned")

# then write WITH (more) partitions
transformed_df.write.mode("overwrite").format("delta").partitionBy("year", "month", "day").save("Tables/flights_partitioned_daily")
```

Fabric Analytics Concepts Notes

Inspecting our delta tables (and parquet files)

```
1   %%sql
2   describe detail flights_partitioned
3
```

✳ 4 sec - Running

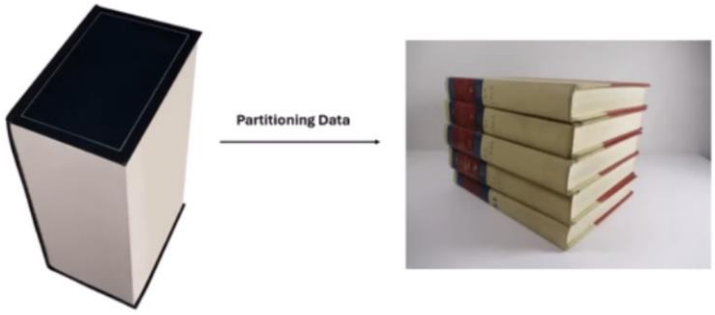
```
1   %%sql
2   describe detail flights_not_partitioned
```

- Command executed in 1 sec 459 ms by william on 12:40:25 PM, 5/09/24

```
1   %%sql
2   describe detail flights_partitioned_daily
```

If partitioned column has data that is less than 1gb, u come across small file pb. Partitioning is balancing act.

- As we know, Spark is a distributed processing engine, meaning it is capable of processing multiple sets of data 'in-parallel'.
- To make this possible, it helps to 'partition' the data. Rather than passing in one massive file to the Spark engine, multiple partitions allow the process/ transform the data in parallel, greatly improving performance.
- It is a balancing act though, because if you go too far, then you will suffer from the 'small file problem'.



Fabric Analytics Concepts Notes

V-Order optimization

What is V-Order?

V-Order (not to be confused with Z-order) is a Microsoft proprietary algorithm. It works by providing some special-sauce sorting/ compaction. compression files to reduce file size, and optimize read performance across all engines within Fabric. The outputted Parquet file is fully parquet-compliant.

Checking if V-Order is enabled

In a notebook you can check is V-Order is enabled (it should be ENABLED by default):

```
1  # Check if v-order optimization is Enabled
2  spark.conf.get('spark.sql.parquet.vorder.enabled')
```

- Command executed in 277 ms by william on 12:35:05 PM, 5/09/24 PySp

Disabling/ re-enabling it:

```
1  # disable v-order
2  spark.conf.set('spark.sql.parquet.vorder.enabled', 'false')
3
4  # enable v-order
5  spark.conf.set('spark.sql.parquet.vorder.enabled', 'true')
```

Delta table maintenance & optimization techniques

The following section describes a few delta table maintenance and optimization techniques:

OPTIMIZE & VACUUM (Delta)

- `OPTIMIZE` is a delta lake method that performs bin-compaction (read more here). This means is can tidy up lots of small files. It's also idempotent.
- `VACUUM` (read more here) involves the removal of files no longer referenced by a Delta table.

Coalesce & Repartition (Spark)

- `coalesce()` (read more here) is a Spark method for reducing the amount of partitions in a delta table. Say you have 100 partitions, you can 'coalesce' th into 10 partitions. Importantly, this is quite an efficient operation, because it doesn't require a shuffle of your data.
- `repartition()` (read more here) involves the breaking of existing partitions to create new partitions, these can either be more or less than the original partitions. Repartitioning is an expensive operation because it involves shuffling (unlike `coalesce`)

Fabric Analytics Concepts Notes

Question

You notice one of your Dataflow Gen2 runs failed to refresh last night.

Where would you go to find out why a particular Dataflow might have failed a particular run?

A) Capacity Metrics App

B) Monitoring Hub

C) Power Query Error Hub

D) Dataflow Refresh History

E) Data Pipeline Run History

Question

When talking about delta table optimization, which of the following operations removes old files no longer referenced by a Delta table log?

A) V-order optimization

B) Z-ordering

C) VACUUM

D) OPTIMIZE

E) Bin compaction

Z-Order Optimization & Bin Compaction

These are advanced data organization techniques used in **distributed computing** and **big data systems** (like Apache Spark, Delta Lake, or Iceberg) to improve query performance, reduce I/O, and optimize storage.

---

1. Z-Order Optimization (Morton Order)

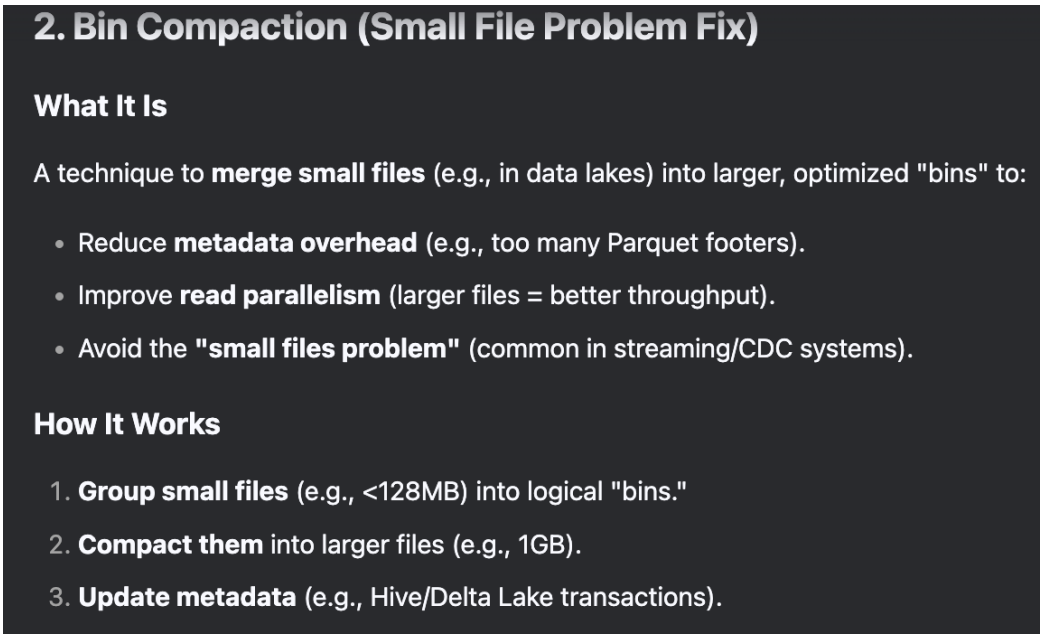
What It Is

Z-Order is a **multi-dimensional clustering** method that maps data with multiple columns into a single **Z-curve** (space-filling curve), keeping similar values close on disk.

Why Use It?

- Improves **query performance** for multi-column filters (e.g., `WHERE date = X AND region = Y`).
- Reduces **data skipping** (fewer files scanned).
- Works well in **columnar formats** (Parquet, ORC).

Feature	Z-Order Optimization	Bin Compaction
Purpose	Optimize query performance	Reduce small-file overhead
Method	Multi-column clustering	Merging small files
Impact	Faster reads (data skipping)	Faster metadata ops & scans
Used in	Delta Lake, Iceberg, databases	Data lakes, streaming systems



When to Use Both?

- First compact small files (Bin Compaction).
- Then apply Z-Order for query optimization.

V-Order (short for "Vertex Order") is a **write-time optimization** technique developed by **Databricks** to enhance the performance of **Delta Lake and Parquet files** in Apache Spark. It reorganizes data on disk to improve **compression, query speed, and cost efficiency** in cloud storage (e.g., S3, ADLS).

Key Features of V-Order

1. Automatic Sorting & Clustering

- Unlike **Z-Order** (which requires explicit `ZORDER BY`), **V-Order automatically sorts data** during writes.
- It groups similar values together, improving **dictionary encoding** and **columnar compression**.

2. Better Compression

- Achieves **smaller file sizes** (up to **2–4x better compression** vs. unsorted data).
- Reduces **storage costs** and **I/O operations** in cloud environments.

3. Faster Queries

- Enables **data skipping** (fewer files scanned).
- Works well with **Delta Lake's metadata** for optimized reads.

4. No Extra Maintenance

- Unlike **Z-Order**, which requires manual optimization (`OPTIMIZE ... ZORDER BY`), **V-Order wo**
  at write time without additional commands.

Question

Which of the following statements about V-Order optimization is FALSE?

A) V-Order optimization is enabled by default in the Fabric Spark Runtime.

B) V-Order can be enabled during table creation using TBLPROPERTIES.

C) A table can be both V-ordered and Z-ordered.

D) V-Order improves the read performance for parquet files.

E) V-Order speeds up the write time of a Parquet file.

Vorder works at write time (hence increase write time than decrease / speed up the write time). Vorder takes longer time to write files but it massively improves the read performance.

Question

You want to analyse long-running queries in a Fabric Data Warehouse.

What is the minimum workspace role you need to run the following query:

```
SELECT * FROM [queryinsights].[long_running_queries]
```

A) Admin

B) Member

C) Contributor

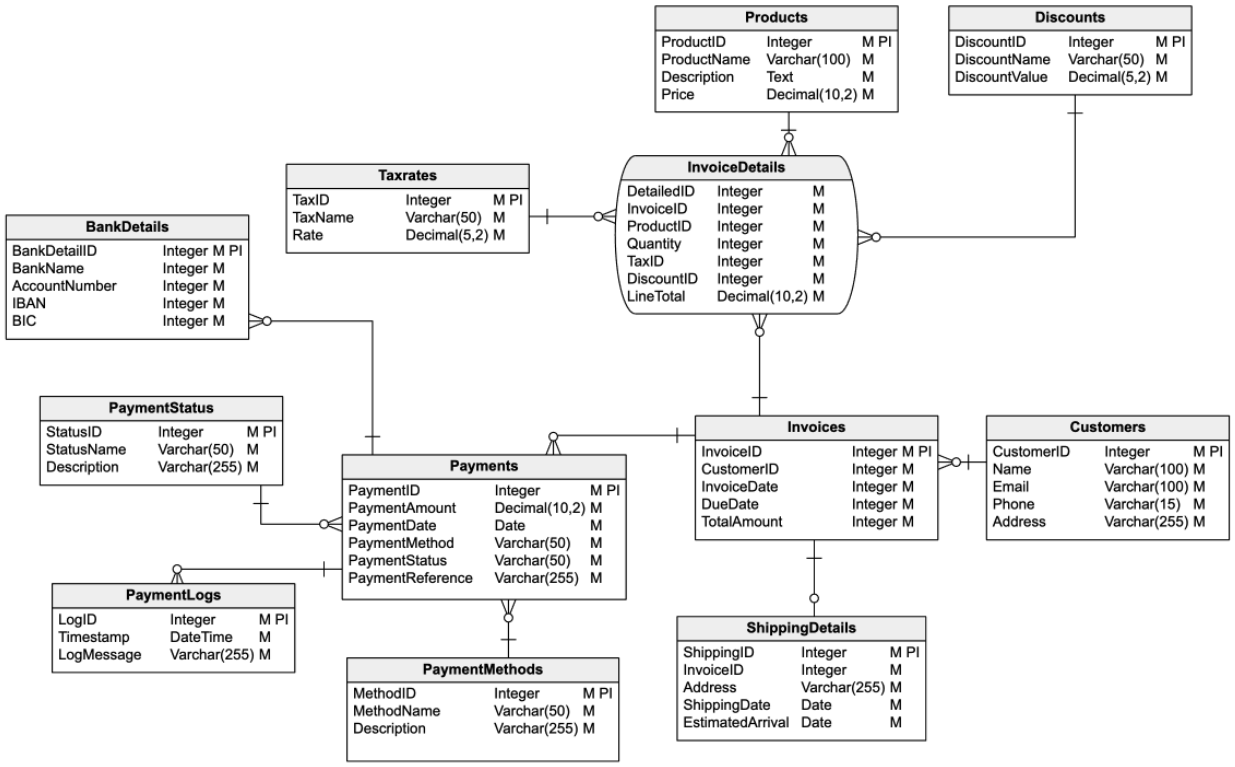
D) Viewer

Ch9. Design & build semantic models

Normalized vs de-normalized data

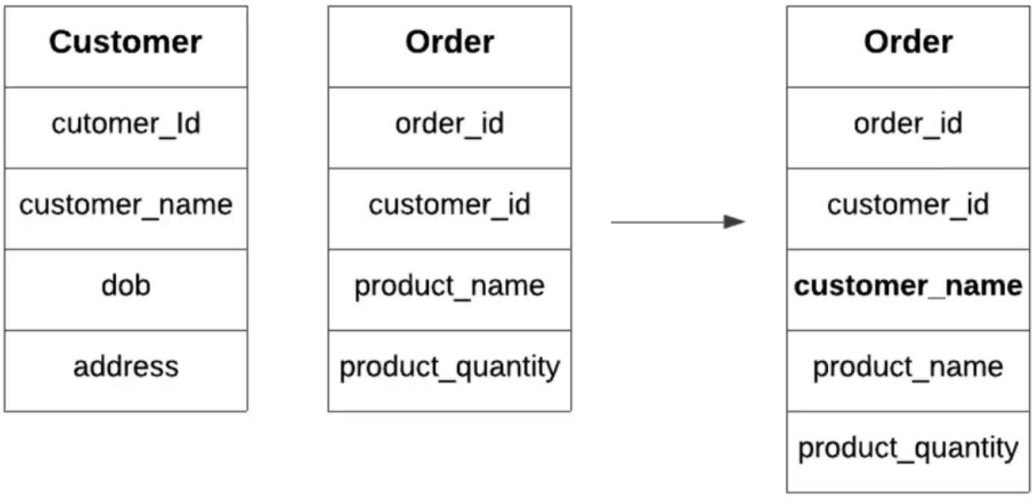
OLTP systems are built for to handle lots of transactions, very fast. The backend database tables are designed to be fully normalized. This means that entities in an OLTP application each have their own table in the backend. It might look something like this, which is an example of an entity-relationship diagram for a billing system. This structure minimizes redundancy - every value is only stored in one place, making it efficient, from a storage perspective. But this kind of heavily normalized structure is less optimal for analytics use cases. OLTP systems are commonly used as a data source for OLAP analytical systems, however, we must de-normalize how the data is structured to make it more useful for analytical workloads.

Fabric Analytics Concepts Notes



Denormalization of data

Put simply, denormalizing data looks like this (below), two or more tables are 'denormalized' into a single table for analytics use cases. This can be done in a Data Warehouse using JOINs.



KQL

Returning a preview sample of 5 records from `StormEvents` table, returning the State, EventType and DamageProperty columns:

```
StormEvents
| take 5
| project State, EventType, DamageProperty
```

Filtering using the WHERE operator:

```
StormEvents
| where State == 'TEXAS' and EventType == 'Flood'
| project StartTime, EndTime, State, EventType, DamageProperty
```

Simply aggregation using the summarise operator:

```
StormEvents
| summarize TotalStorms = count() by State
```

Grouping data into bins:

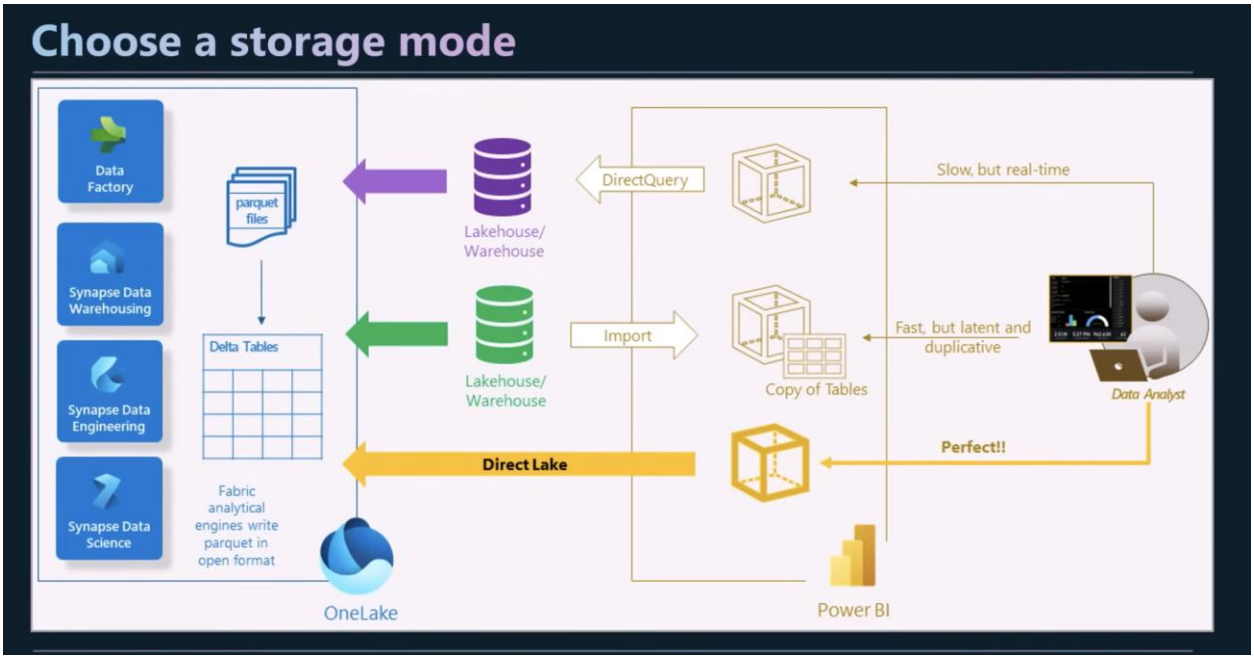
```
StormEvents
| where StartTime between (datetime(2007-01-01) .. datetime(2007-12-31))
    and DamageCrops > 0
| summarize EventCount = count() by bin(StartTime, 7d)
```

Calculating the max, min, avg and sum:

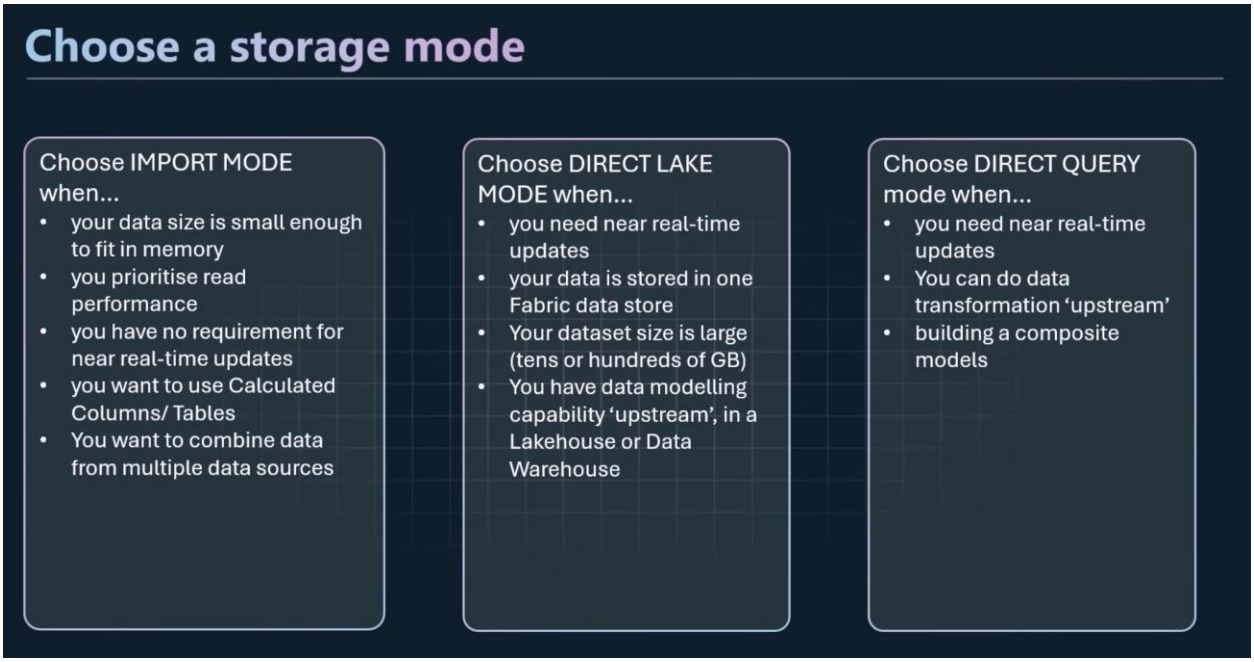
Fabric Analytics Concepts Notes

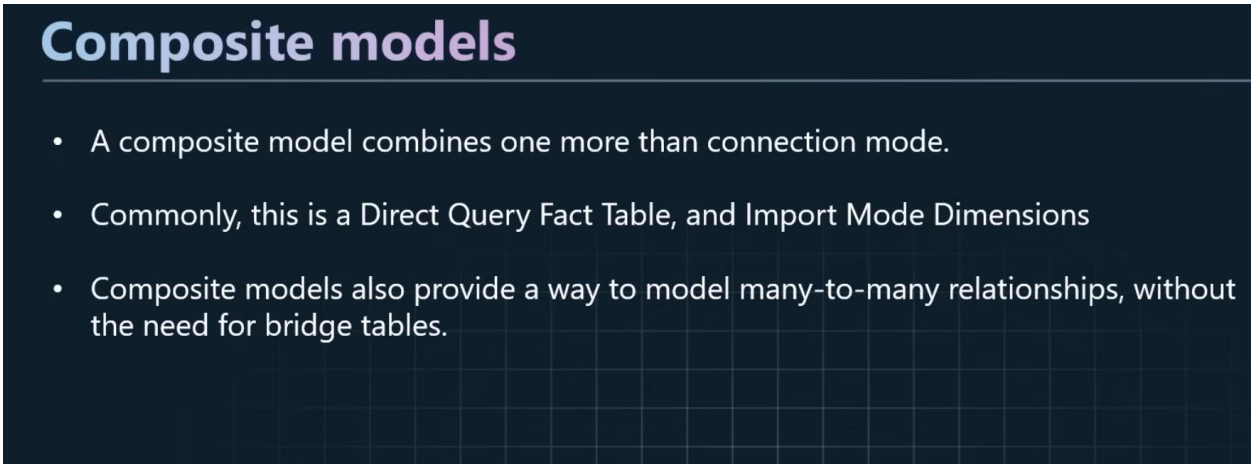
```
StormEvents
| where DamageCrops > 0
| summarize
    MaxCropDamage=max(DamageCrops),
    MinCropDamage=min(DamageCrops),
    AvgCropDamage=avg(DamageCrops)
    by EventType
| sort by AvgCropDamage
```

Storage modes



Fabric Analytics Concepts Notes





Fact table has lots of data (million to billions of rows).Fact table is updated very often (hundreds of records every second in some oltp transactions). Direct mode is good for fact. As data is changing very fast in fact, direct mode gives near real time access to that changing / fresh data.

Dim tables might change a lot slower & they also have fewer rows than fact. So Dim table's can use import mode. Let's say products dim table is updated once per day.

Aggregations (Power BI feature)

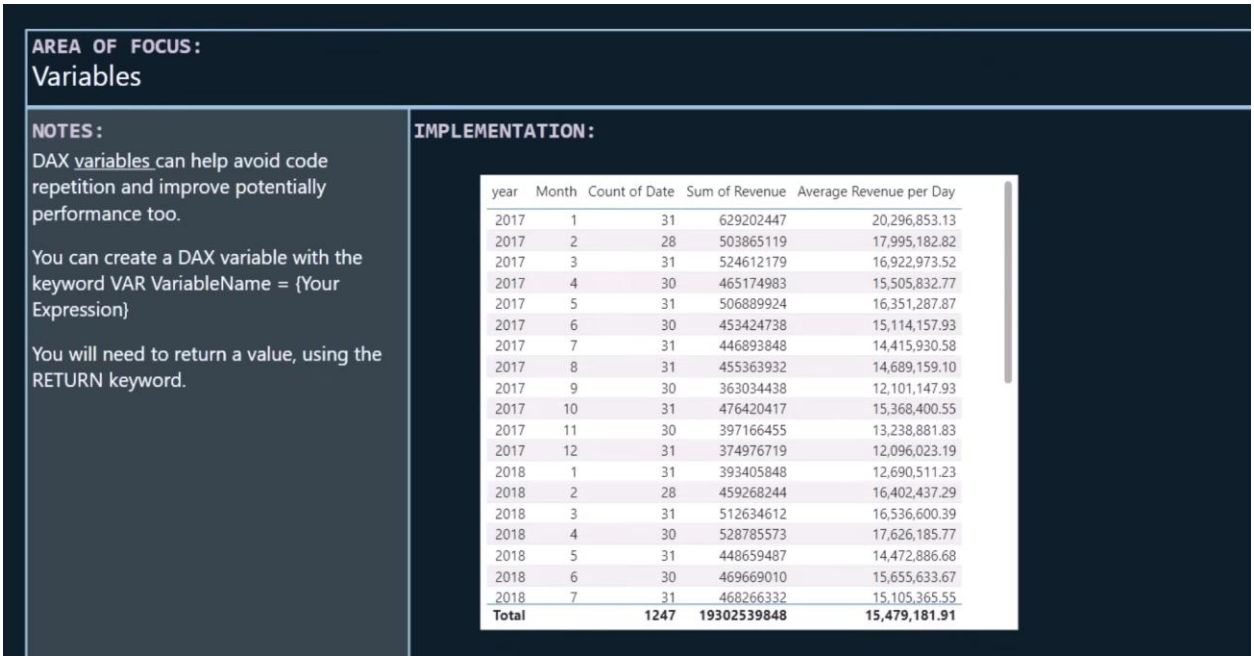
- Aggregations are a specific feature within Power BI for aggregating large datasets and caching the aggregation.
- The goal is to improve performance in large data models.
- Aggregations can also be used in composite models.
- Aggregations can either be created in your data source, or you can use Import Mode and create the aggregation in the Power Query engine.
- User-defined aggregations – using the 'Manage aggregations' dialog in Power BI Desktop to define aggregations for aggregation columns with summarization, detail table, and detail column properties
- Automatic aggregations (Premium subscriptions only) - use state-of-the-art machine learning (ML) to continuously optimize DirectQuery semantic models for maximum report query performance

Large format semantic models

- Large format semantic models provide a highly compressed in-memory cache for optimized query performance, enabling fast user interactivity
- Normally used in your semantic models are > 10GB in size.
- Available on Premium P SKUs, Embedded A SKUs, and with Premium Per User (PPU)
- Used commonly when connecting third-party tools via the XMLA endpoint
- On-demand loading (same feature as Direct Lake)

Fabric Analytics Concepts Notes

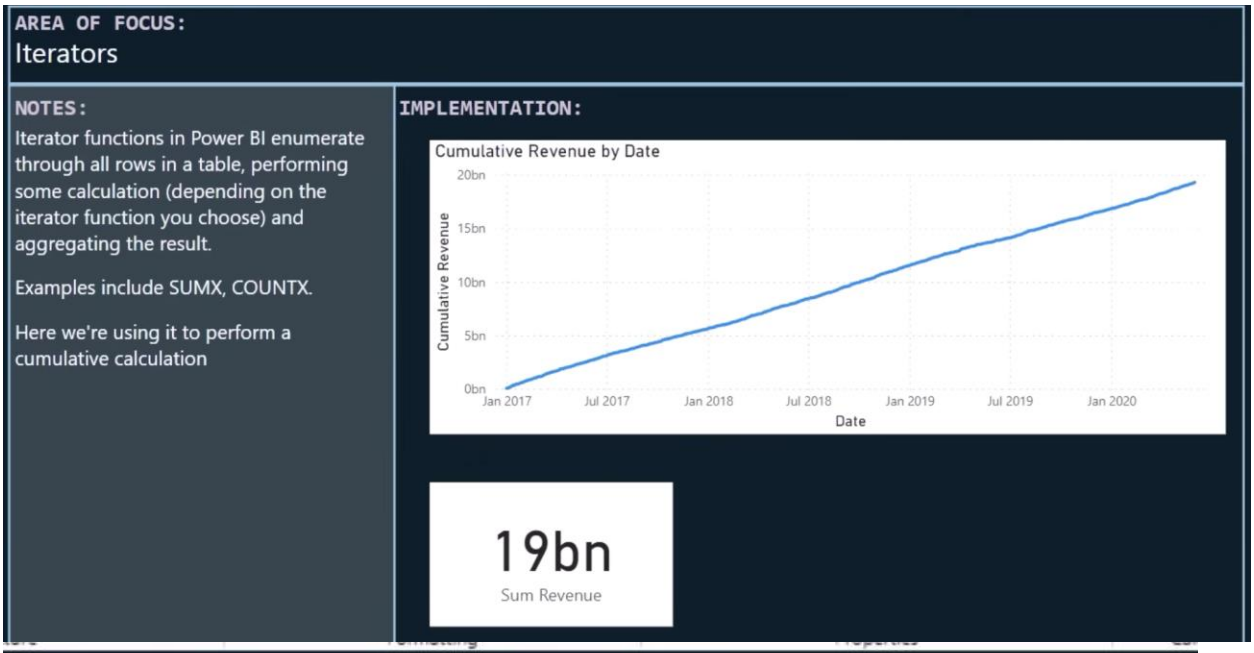
DAX Variables, functions & parameters



year	Month	Count of Date	Sum of Revenue	Average Revenue per Day
2017	1	31	629202447	20,296,853.13
2017	2	28	503865119	17,995,182.82
2017	3	31	524612179	16,922,973.52
2017	4	30	465174983	15,505,832.77
2017	5	31	506889924	16,351,287.87
2017	6	30	453424738	15,114,157.93
2017	7	31	446893848	14,415,930.58
2017	8	31	455363932	14,689,159.10
2017	9	30	363034438	12,101,147.93
2017	10	31	476420417	15,368,400.55
2017	11	30	397166455	13,238,881.83
2017	12	31	374976719	12,096,023.19
2018	1	31	393405848	12,690,511.23
2018	2	28	459268244	16,402,437.29
2018	3	31	512634612	16,536,600.39
2018	4	30	528785573	17,626,185.77
2018	5	31	448659487	14,472,886.68
2018	6	30	469669010	15,655,633.67
2018	7	31	468266332	15,105,365.55
Total		**1247**	**19302539848**	**15,479,181.91**

```
1 Average Revenue per Day =
2 VAR TotalRevenue = SUM('revenue'[Revenue])
3 VAR TotalDays = DISTINCTCOUNT('date'[Date])
4 RETURN
5     IF(TotalDays > 0, TotalRevenue / TotalDays, BLANK())
```

Fabric Analytics Concepts Notes




```
1 Cumulative Revenue =
2 SUMX(
3     FILTER(
4         ALL('date'[Date]),
5         'date'[Date] <= MAX('date'[Date])
6     ),
7     [Sum Revenue]
8 )
```

Fabric Analytics Concepts Notes

AREA OF FOCUS:

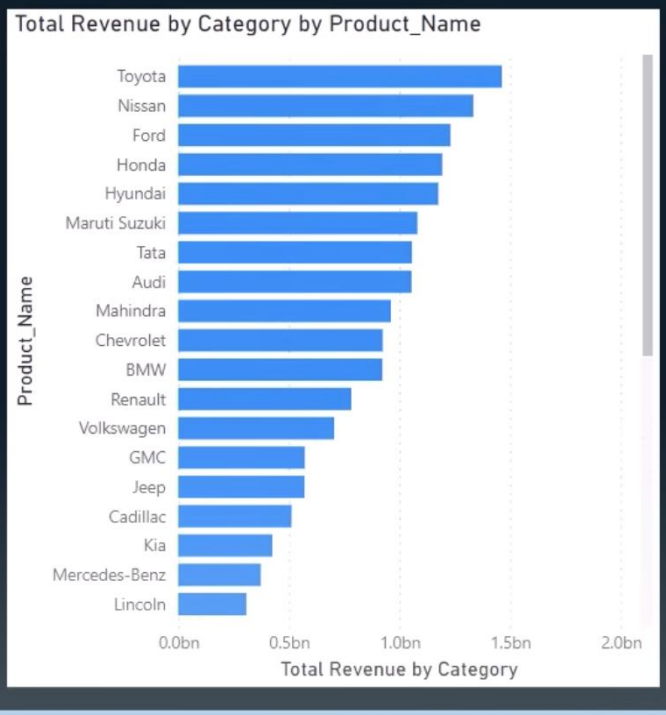
Table filtering

NOTES:

The FILTER function returns a table that represents a subset of another table or expression.

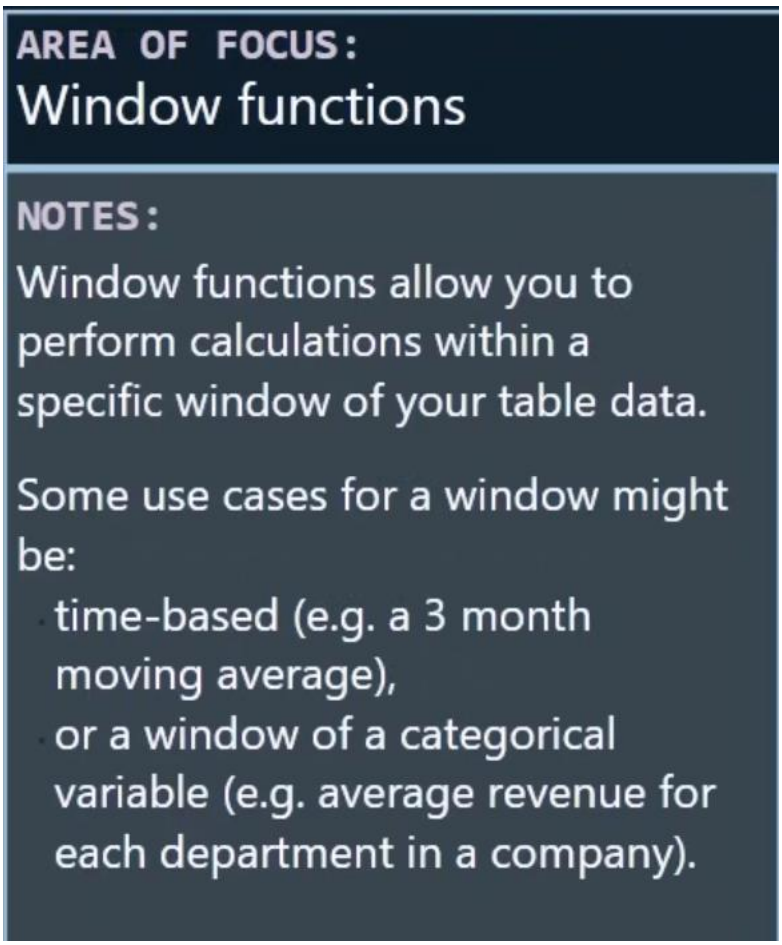
For example:
FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")

IMPLEMENTATION:




```
1 Total Revenue by Category =
2 CALCULATE(
3     SUM('revenue'[Revenue]),
4     FILTER(
5         'products',
6         'products'[Product_Name] = SELECTEDVALUE('products'[Product_Name])
7     )
8 )
```

Fabric Analytics Concepts Notes



Applies to: ✅ Calculated column ✅ Calculated table ✅ Measure ✅ Visual calculation

Returns multiple rows which are positioned within the given interval.

Syntax

DAX Copy

```
WINDOW ( from[, from_type], to[, to_type][, <relation> or <axis>][, <o
```

Parameters

Expand table

Term	Definition
from	Indicates where the window starts. It can be any DAX expression that returns a scalar value. The behavior depends on the <from_type> parameter: - If <from_type> is REL, the number of rows to go back (negative value) or forward (positive value) from the current row to get the first row in the window.
from_type	Modifies behavior of the <from> parameter. Possible values are ABS (absolute) and REL (relative). Default is REL.
to	Same as <from>, but indicates the end of the window. The last row is included in the window.
to_type	Same as <from_type>, but modifies the behavior of <to>.
relation	(Optional) A table expression from which the output rows are returned. If specified, all columns in <partitionBy> must come from it or a related

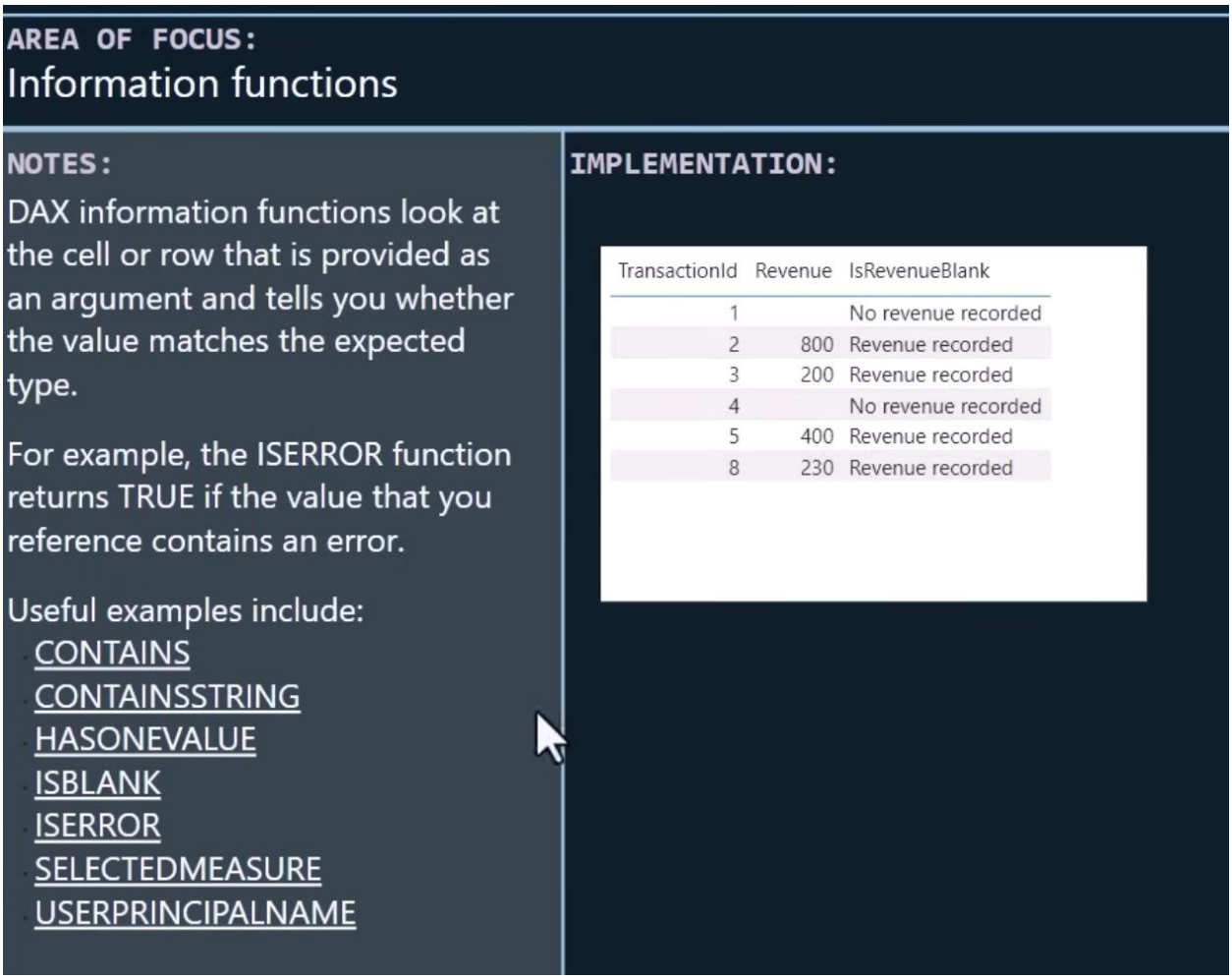
Example 1 - measure

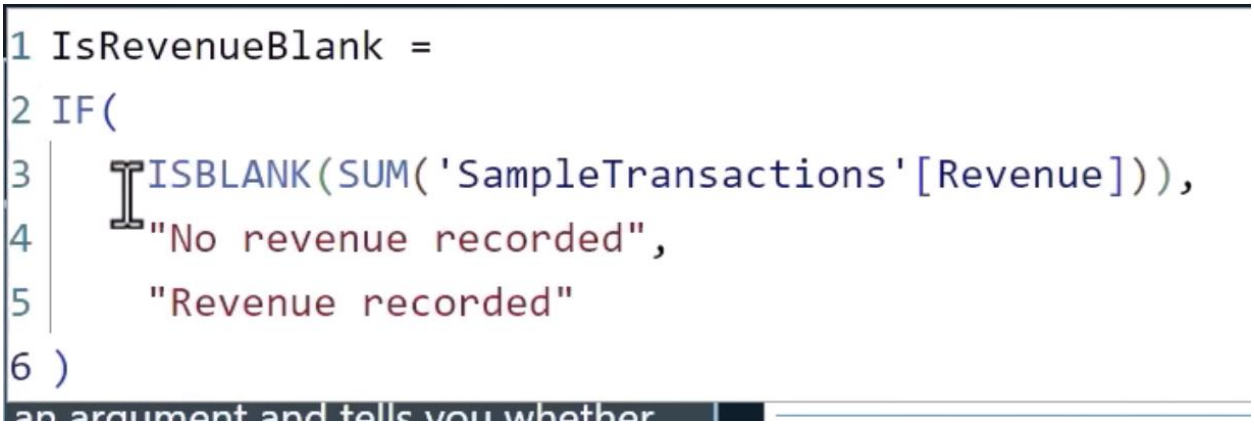
The following measure:

DAX Copy

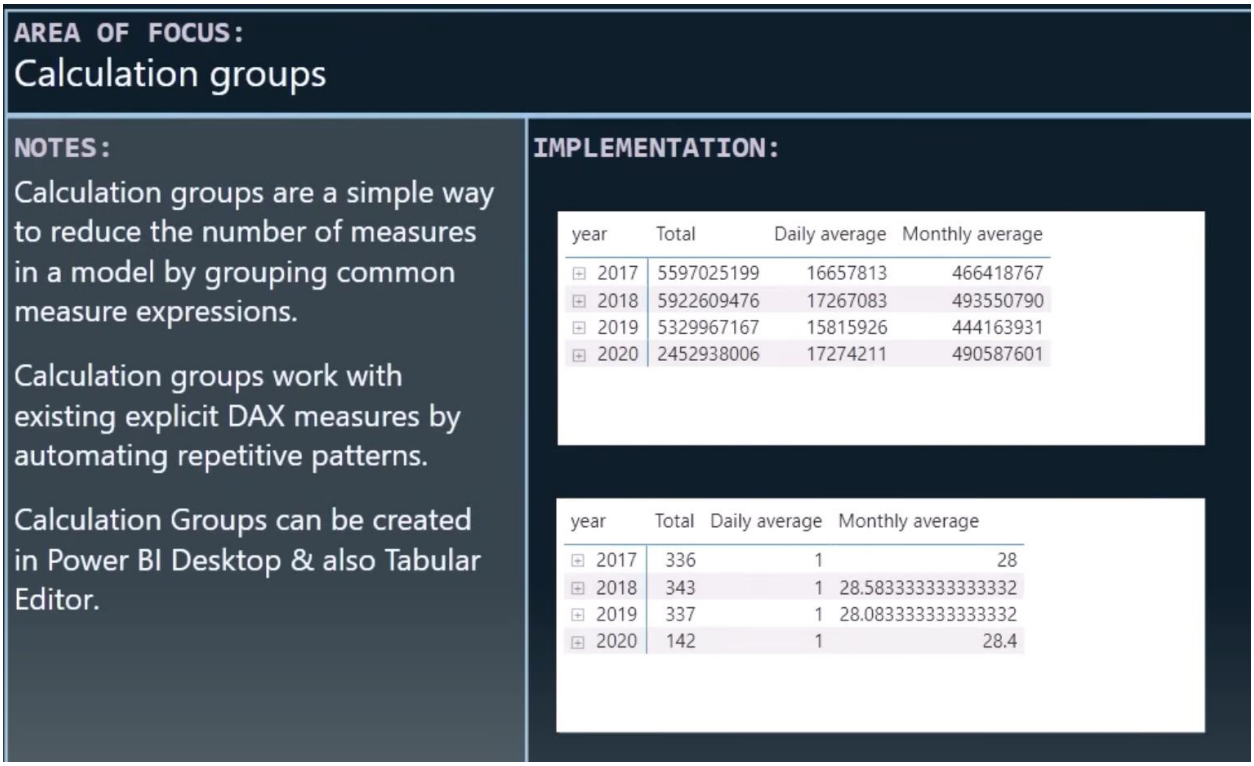
```
3-day Average Price =
AVERAGEX(
    WINDOW(
        -2,REL,0,REL,
        SUMMARIZE(ALLSELECTED('Sales'), 'Date'[Date], 'Product'[Produc
        ORDERBY('Date'[Date]),
        KEEP,
        PARTITIONBY('Product'[Product])
    ),
    CALCULATE(AVERAGE(Sales[Unit Price]))
)
```

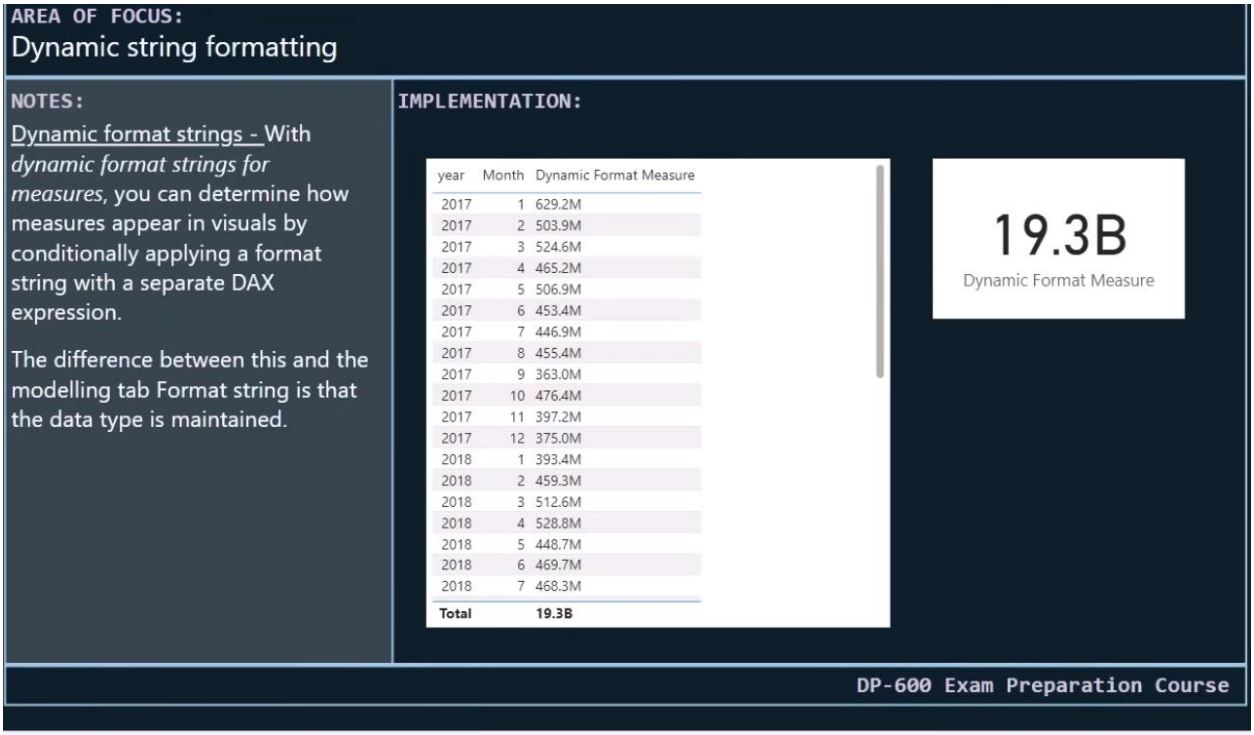
Fabric Analytics Concepts Notes





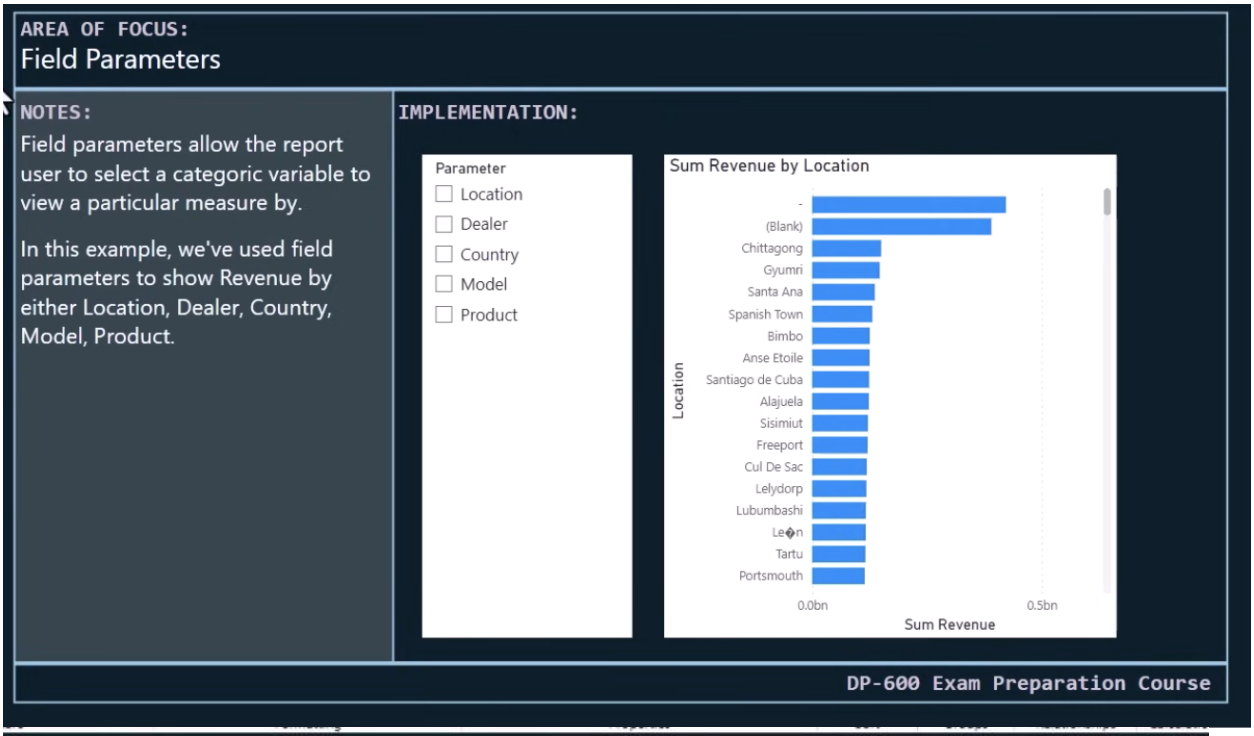
Fabric Analytics Concepts Notes





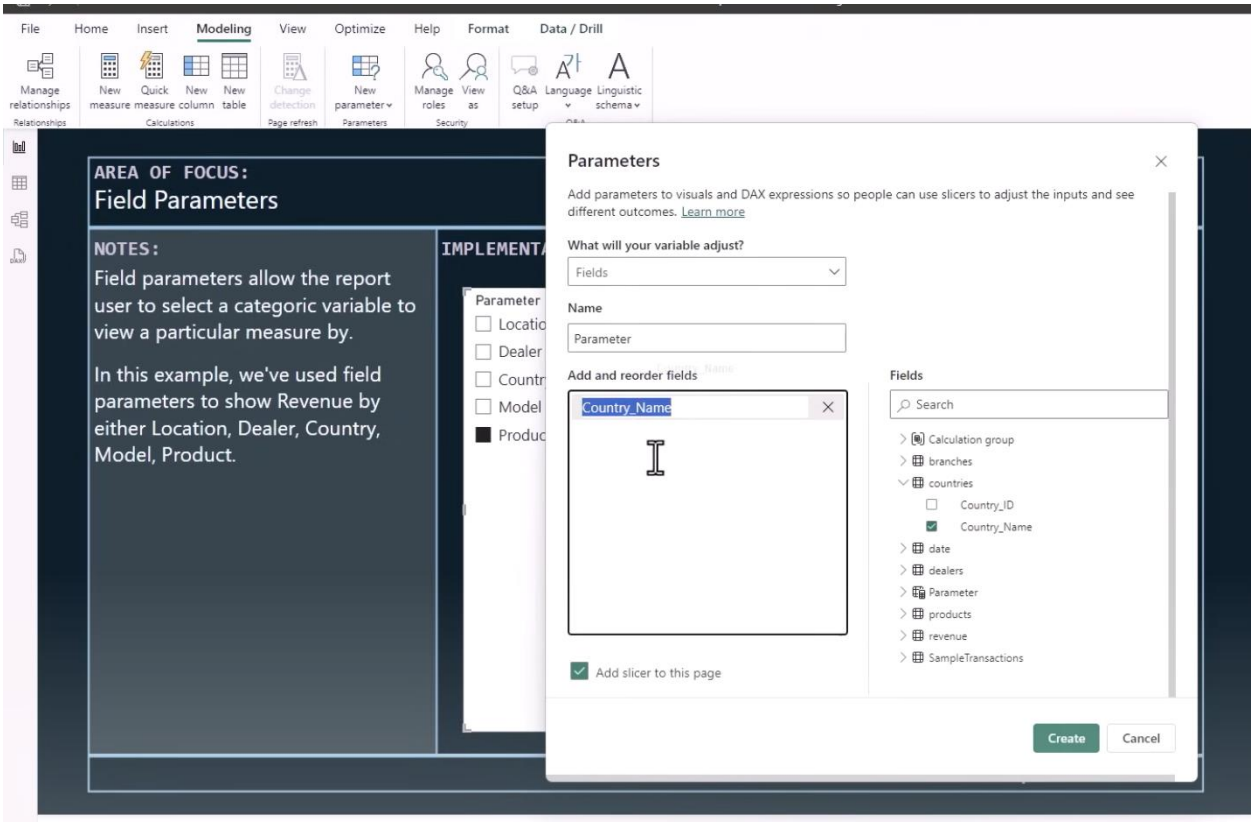
Fabric Analytics Concepts Notes

```
Dynamic Format Measure =
VAR SumRev = [Sum Revenue]
RETURN
    SWITCH(
        TRUE(),
        SumRev < 1000, FORMAT(SumRev, "0.00"),
        SumRev < 1000000, FORMAT(SumRev/1000, "0.0K"),
        SumRev < 1000000000, FORMAT(SumRev/1000000, "0.0M"),
        FORMAT(SumRev/1000000000, "0.0B")
    )
```




```
Parameter = {
    ("Location", NAMEOF('dealers'[Location_NM]), 0),
    ("Dealer", NAMEOF('dealers'[Dealer_NM]), 1),
    ("Country", NAMEOF('countries'[Country_Name]), 2),
    ("Model", NAMEOF('products'[Model_Name]), 3),
    ("Product", NAMEOF('products'[Product_Name]), 4)
}
```

Fabric Analytics Concepts Notes



The DAX expression AVERAGEX is an example of:

A) An information function

B) A calculation group

C) Table filtering

D) A window function

E) An iterator function

AVGX iterates over each row of a specified table, computes the expression, and then averages the results.

Fabric Analytics Concepts Notes

Question

On-demand loading (loading only the data that is needed for a query) is a feature of which TWO of the following:

A) Import mode

B) Direct Lake mode

C) Direct Query mode

D) Large format semantic models

E) The XMLA endpoint

Question

Dynamic format strings overcome which significant limitation that comes from using the DAX FORMAT function?

A) The Format function is slow on large datasets

B) The Format function returns a string value, so the values can't be used in chart visuals.

C) The Format function can't handle date locale whilst formatting.

D) The Format function can't be used with Field Parameters

The DAX expressions SELECTEDMEASURE() is most likely found in the construction of which of the following:

A) Calculation item (Calculation Groups)

B) Field parameters

C) An iterator function

D) Large format semantic models

E) A window function

```
-- Calculation Item: "YTD"
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

Calculation Groups with SELECTEDMEASURE() are ideal for:

- Time intelligence (e.g., YTD, MoM).
- Dynamic formatting (e.g., "%" vs. "$").
- Scenario analysis (e.g., "Actuals" vs. "Forecast").

Which of the following is an irreversible operations (can't be changed afterwards)?

A) Changing the cross-filtering of a relationship to bidirectional

B) Changing the Storage mode of a table to Import

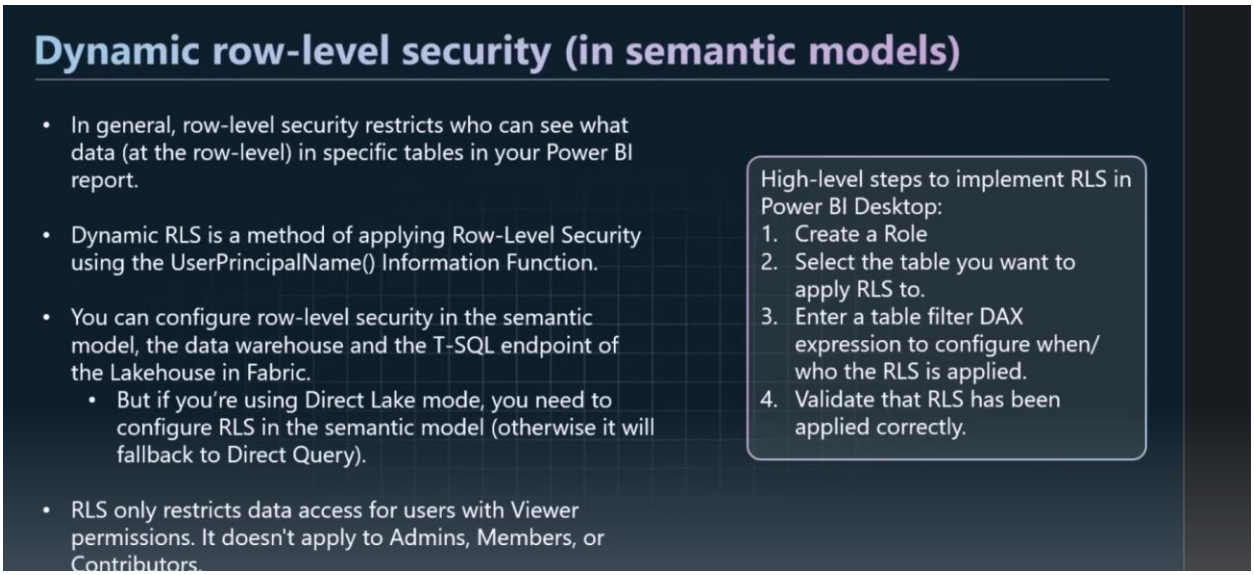
C) Naming a Calculation Group

D) Converting a semantic model into a large-format semantic model

E) Creating a window function

When a table's storage mode is changed to Import in Power BI or Analysis Services, the data is physically copied into the model. If it was previously in Direct Query or Direct Lake mode, switching to Import discards the real-time query connection behavior. While technically you can switch it back in some tools, doing so may lose configurations or break dependencies, and in many enterprise-level environments, it's considered a one-way change due to downstream impacts.

Ch10. Secure & Optimize Semantic Models

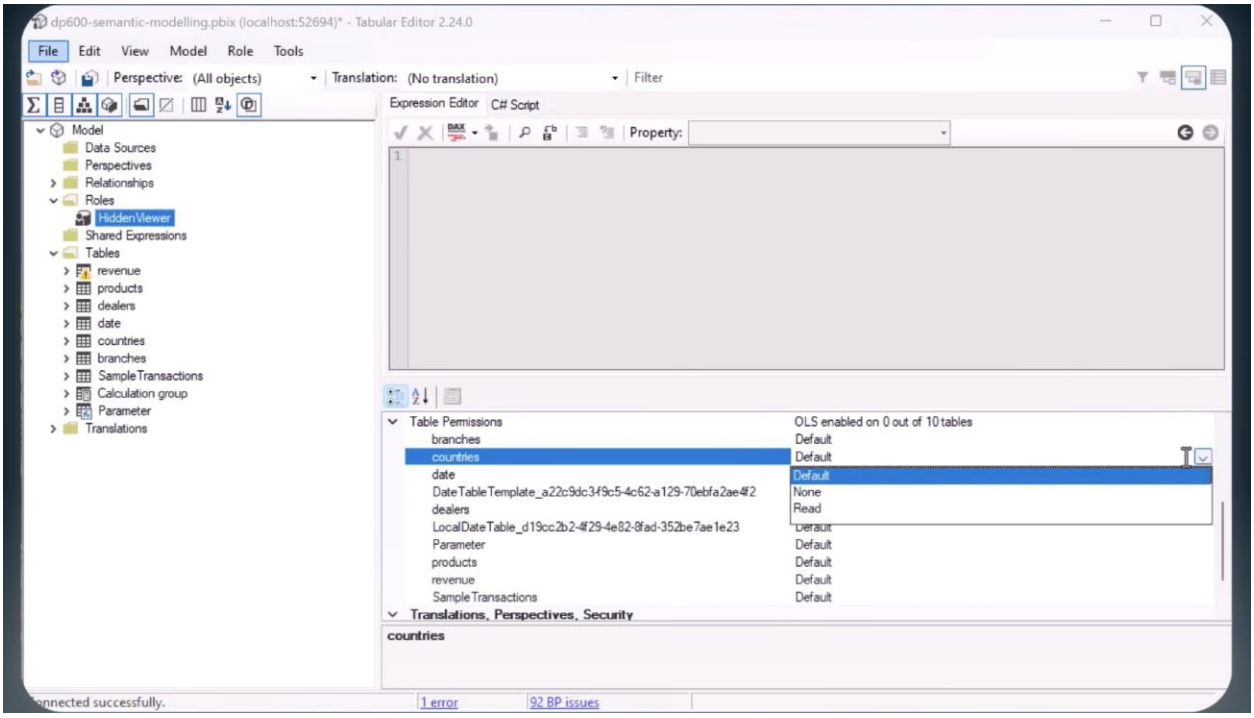


Object-level security (using Tabular Editor)

- With object-level security, you (as a model author) can configure who can view tables and specific columns within a Power BI report.
- To configure object-level security, you need to use an external tool, such as Tabular Editor.
- Similarly to RLS, OLS only restricts data access for users with Viewer permissions. It doesn't apply to Admins, Members, or Contributors.

High-level steps to implement OLS in Tabular Editor:

1. Create a Role (in Power BI Desktop), if you haven't already
2. Open Tabular Editor, find the Role, click on Table Properties for that Role
3. Set the permissions for the table to *None* or *Read*
4. Publish the report to the Service, and add people/groups to the Role.

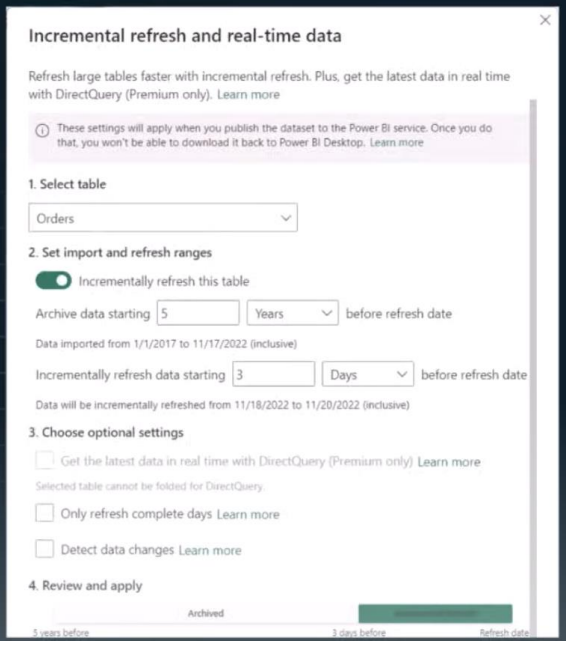


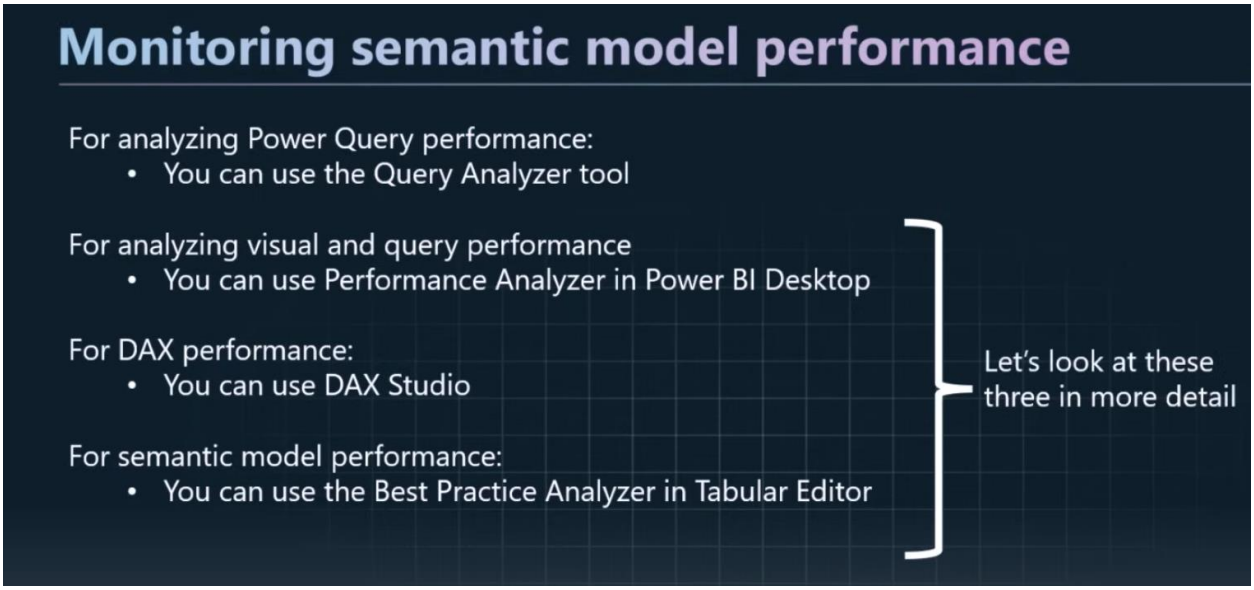
Incremental refresh (Power BI)

- Typically used on large Fact tables, incremental refresh allows you to pull in only the data that has changed in a given time period, rather than a full load of the data from source. This has the following benefits:
  - Fewer traditional refreshes are required
  - Refreshes are a lot quicker (only pulling in what is new)
  - Resource consumption is reduced
  - Refreshes can be more reliable (no long-running connections needed to your data source)
- Currently, incremental refresh is only possible within Power BI (not in any of the Fabric ETL items like a Dataflow Gen2 or Data Pipeline).
- Incremental refresh in Power BI is available for Power BI Premium licenses only (PPU or a Premium capacity subscription)
- Incremental refresh policies are defined in Power BI Desktop

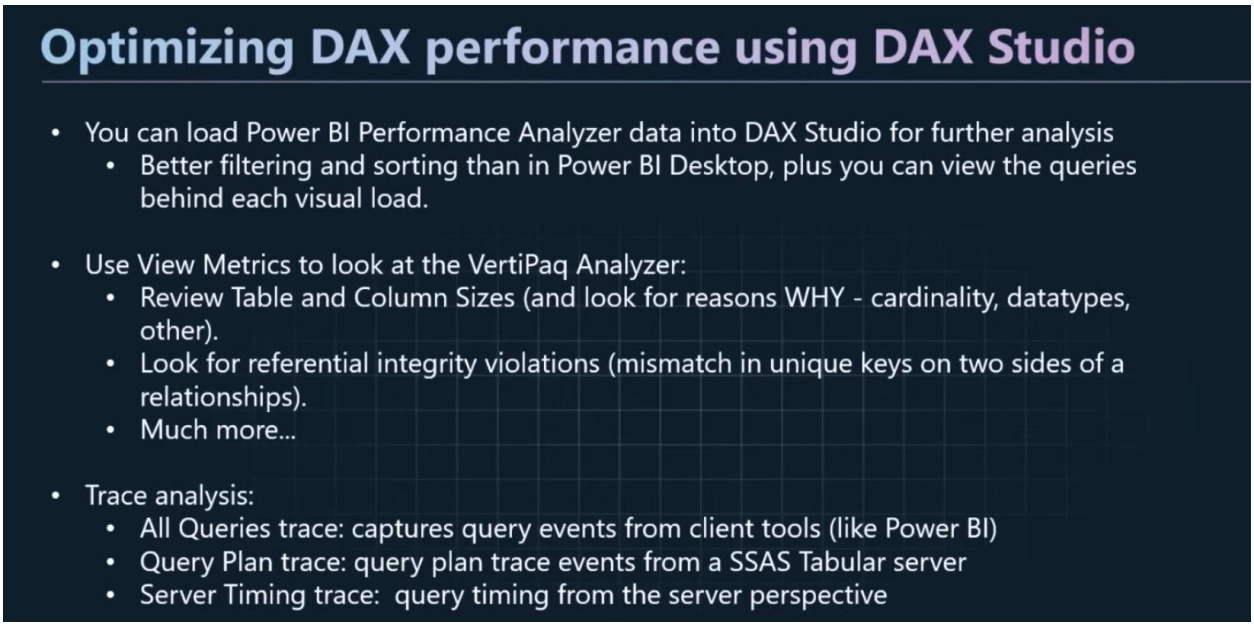
Implementing incremental refresh (Power BI)

- Create RangeStart and RangeEnd parameters
- In Power Query, apply custom date filters on your table's date column, using the RangeStart and RangeEnd parameters
- Define your incremental refresh policy

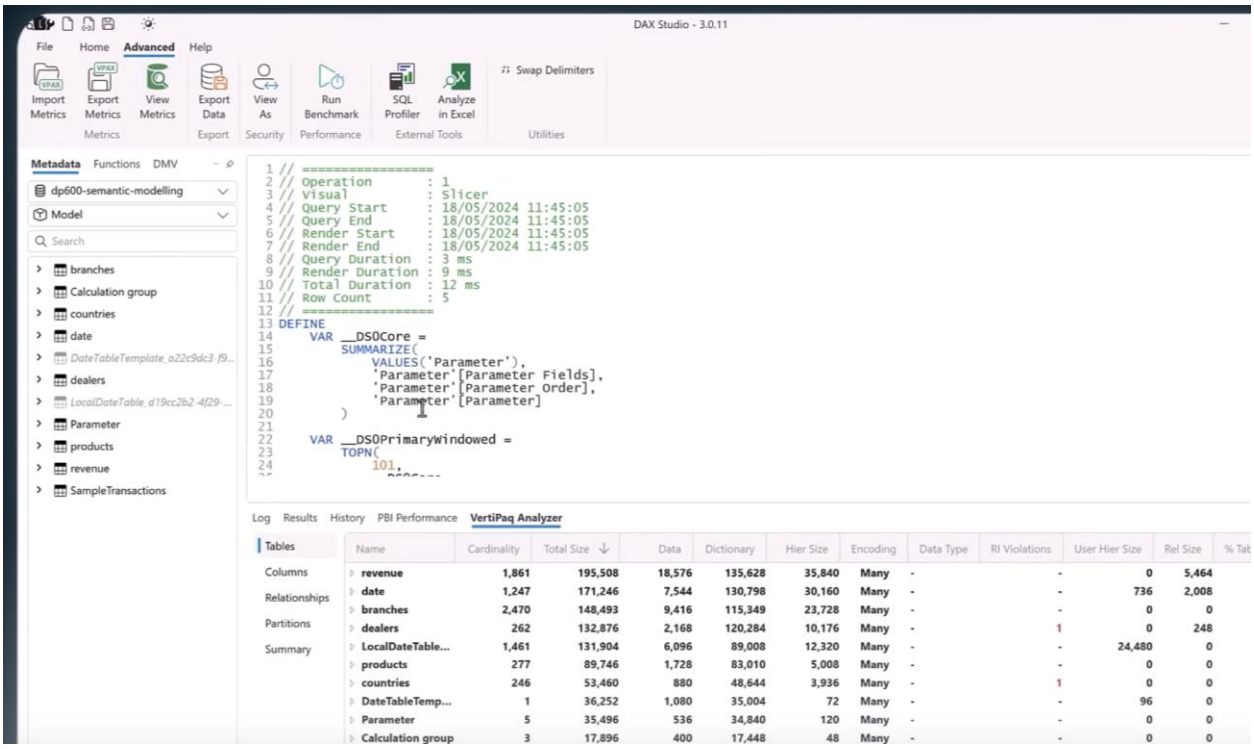


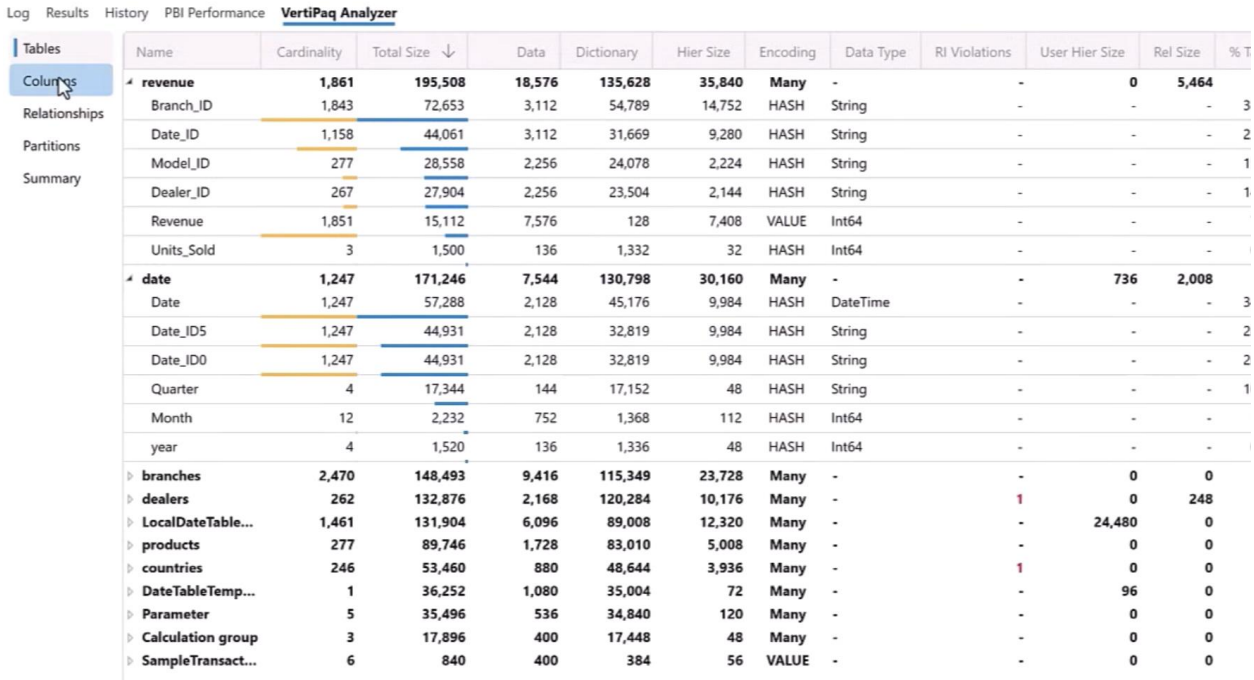

Monitoring semantic model performance
For analyzing Power Query performance:
• You can use the Query Analyzer tool
For analyzing visual and query performance
• You can use Performance Analyzer in Power BI Desktop
For DAX performance:
• You can use DAX Studio
For semantic model performance:
• You can use the Best Practice Analyzer in Tabular Editor
Let's look at these three in more detail

DAX Studio


Optimizing DAX performance using DAX Studio
• You can load Power BI Performance Analyzer data into DAX Studio for further analysis
• Better filtering and sorting than in Power BI Desktop, plus you can view the queries behind each visual load.
• Use View Metrics to look at the VertiPaq Analyzer:
• Review Table and Column Sizes (and look for reasons WHY - cardinality, datatypes, other).
• Look for referential integrity violations (mismatch in unique keys on two sides of a relationships).
• Much more...
• Trace analysis:
• All Queries trace: captures query events from client tools (like Power BI)
• Query Plan trace: query plan trace events from a SSAS Tabular server
• Server Timing trace: query timing from the server perspective

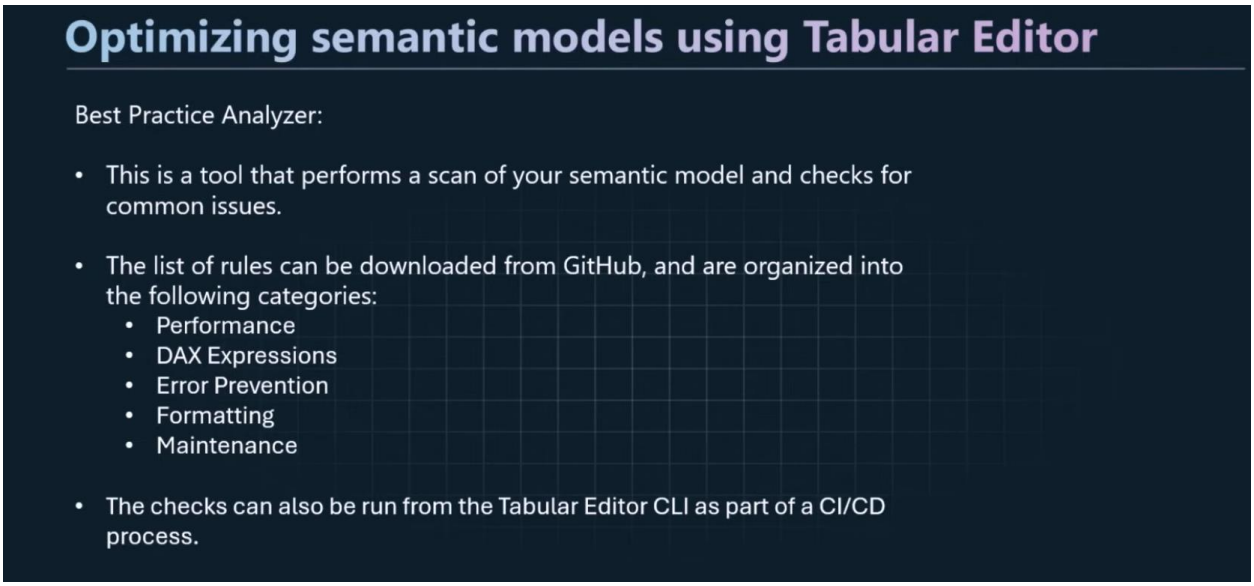
Fabric Analytics Concepts Notes

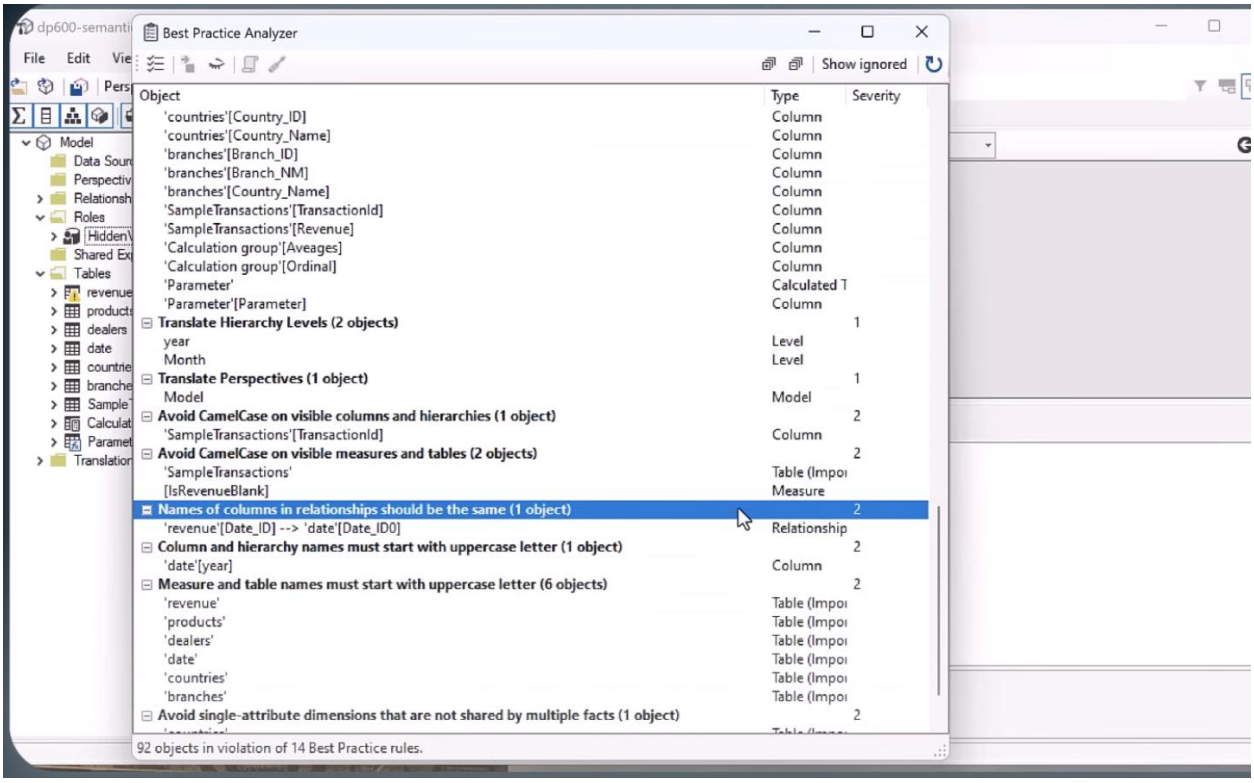




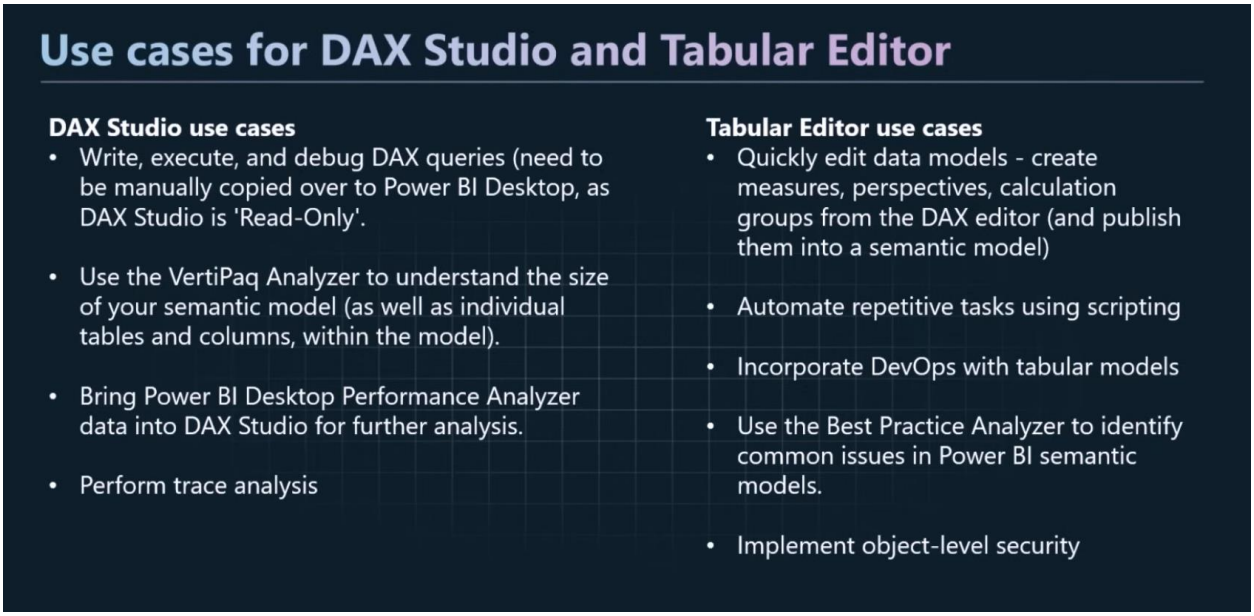
Fabric Analytics Concepts Notes

Tabular Editor





Fabric Analytics Concepts Notes



DAX Studio: The DAX Query & Performance Tool
Primary Purpose: Debugging, optimizing, and analyzing DAX queries/data models.
Key Use Cases:

1. **DAX Query Development**
    - Write, test, and debug complex DAX queries in a dedicated IDE.
    - *Limitation*: Queries must be manually copied to Power BI Desktop (read-only connection).
2. **Performance Analysis**
    - **VertiPaq Analyzer**: Inspect semantic model size (tables/columns) to optimize storage (e.g., identify high-cardinality columns).
    - **Trace Analysis**: Capture query execution plans to diagnose performance bottlenecks.
3. **Integration with Power BI**
    - Import Performance Analyzer data for deeper investigation.

Tabular Editor: The Model Management & Automation Tool
Primary Purpose: Advanced modeling, automation, and DevOps for semantic models.
Key Use Cases:

1. **Model Editing**
    - Rapidly create/edit measures, calculation groups, and perspectives without waiting for Power BI refreshes.
    - *Example*: Build time-intelligence calculation groups (YTD, QoQ%) efficiently.
2. **Automation & Scripting**
    - Use C# scripts to automate repetitive tasks (e.g., bulk measure edits, metadata updates).
3. **DevOps & Best Practices**
    - **Best Practice Analyzer**: Flag issues (e.g., unused columns, missing descriptions).
    - **Version Control**: Integrate with Git for collaborative model development.
4. **Security**

- Configure object-level security (OLS) for row-level security (RLS) models.

When to Use Which Tool?

Scenario	DAX Studio	Tabular Editor
Debugging a slow DAX measure	✅	⚠️ (Limited)
Creating calculation groups	❌	✅
Analyzing data model size	✅	❌
Automating measure creation	❌	✅
Testing query performance	✅	❌

Pro Tip:

- **Combine Both**: Use Tabular Editor to build/optimize your model, then DAX Studio to fine-tune query performance.

Your goal is to analyze Performance Analyzer data in DAX Studio to find the visual in your report with the longest total load time (ms).

Put the following steps in the correct order to achieve this:

1. In Power BI Performance Analyzer, click Start Recording
2. Click 'Refresh Visuals' or interact with the report. Then click Stop Recording.
3. Export the Performance Data JSON file
4. Import the Performance Data JSON in DAX Studio
5. On the PBI Performance tab, sort by Total Ms (descending) to find the longest refresh time.

Step-by-Step Process

1. **Start Recording in Power BI Performance Analyzer**
   - In Power BI Desktop, go to the *View* tab → *Performance Analyzer* → Click **Start Recording**.
   - *Purpose*: Begins capturing metrics for all report interactions/refreshes.
2. **Refresh Visuals or Interact with the Report**
   - Click **Refresh Visuals** (or interact with filters/slicers).
   - After completing actions, click **Stop Recording**.
   - *Why?*: Generates performance data for all visuals during the session.
3. **Export the Performance Data as JSON**
   - In Performance Analyzer, click the **Export** button to save the data as a .json file.
   - *Key Point*: This file contains timing details (e.g., DAX query duration, visual rendering).

4. **Import the JSON File into DAX Studio**
    - Open DAX Studio → Navigate to the *Tools* tab → Select **Import Performance Data**.
    - Load the exported .json file.
    - *Outcome*: DAX Studio parses the data into a queryable format.
5. **Sort by Total Ms (Descending) in DAX Studio**
    - Go to the *PBI Performance* tab in DAX Studio.
    - Sort the **Total Ms** column in descending order.
    - *Result*: The visual with the highest value is your performance bottleneck.

You want to use the Best Practice Analyzer tool within Tabular Editor to assess your DAX performance.

Which of the following Severity codes for BPA rule violations indicates an Error:

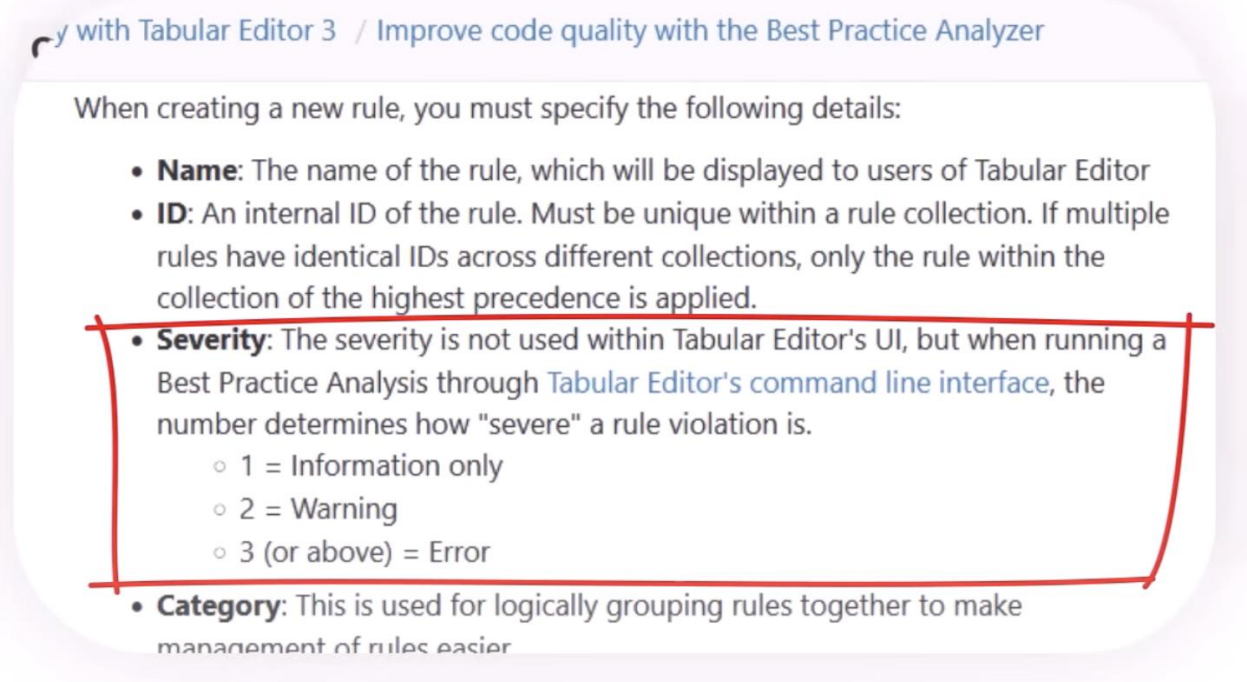
A. Level 0

B. Level 1

C. Level 2 and above

D. Level 2

E. Level 3 and above



y with Tabular Editor 3 / Improve code quality with the Best Practice Analyzer

When creating a new rule, you must specify the following details:

- **Name**: The name of the rule, which will be displayed to users of Tabular Editor
- **ID**: An internal ID of the rule. Must be unique within a rule collection. If multiple rules have identical IDs across different collections, only the rule within the collection of the highest precedence is applied.
- **Severity**: The severity is not used within Tabular Editor's UI, but when running a Best Practice Analysis through Tabular Editor's command line interface, the number determines how "severe" a rule violation is.
    - 1 = Information only
    - 2 = Warning
    - 3 (or above) = Error
- **Category**: This is used for logically grouping rules together to make management of rules easier

Fabric Analytics Concepts Notes

When implementing Dynamic Row-Level Security, which DAX Information Function should you use to filter the data in a specific table based on the logged-in user's email address?

A. USER()

B. USEROBJECTID()

C. USEREMAIL()

D. USERPRINCIPALNAME()

E. USERNAME()

In DAX Studio, which of the following records queries that are generated by a client tool like Power BI Desktop?

A. Query Plan trace

B. SQL Profiler

C. All Queries trace

D. Server Timings trace

E. VertiPaq Analyzer

1. **All Queries Trace**:
    - **Purpose**: Captures *every* query sent to the semantic model by client tools (e.g., Power BI Desktop, Excel).
    - **Use Case**: Identify which visuals/filters generate DAX queries and analyze their frequency/performance.
2. **Why Other Options Are Incorrect**:
    - **A) Query Plan Trace**: Shows the *execution plan* for a *single* query (useful for optimization, not logging all queries).
    - **B) SQL Profiler**: Legacy SQL Server tool (not native to DAX Studio).
    - **D) Server Timings Trace**: Measures *storage engine* vs. *formula engine* processing times (doesn't log queries).
    - **E) VertiPaq Analyzer**: Inspects data model structure (size/compression), not query logging.

Fabric Analytics Concepts Notes

The first step in implementing Incremental Refresh in Power BI is to:

A. Add a RangeStart and a RangeEnd column to your dataset.

B. Add RefreshStart and RefreshEnd parameters to your Power BI Desktop project

C. Add a RefreshStart and a RefreshEnd column to your dataset.

D. Add RangeStart and RangeEnd parameters to your Power BI Desktop project

For analyzing Power Query performance:

- You can use the Query Diagnostics tool

For analyzing visual and query performance

- You can use Performance Analyzer in Power BI Desktop

For DAX performance:

- You can use DAX Studio

For semantic model performance:

- You can use Tabular Editor

There are a number of methods you can use for improving DAX performance, we will look at using DAX Studio.

DAX Studio use cases:

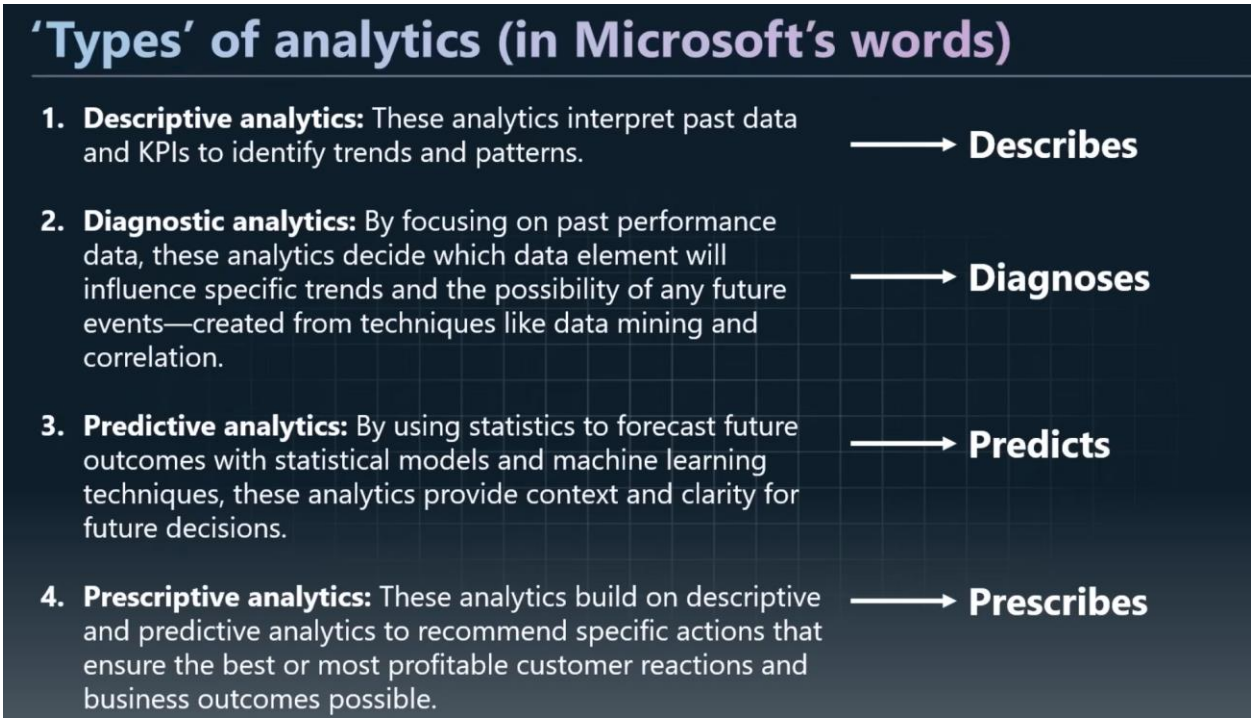
- Use the VertiPaq Analyzer to understand the size of your semantic model (as well as individual tables and columns, within the model).
- Bring Power BI Desktop Performance Analyzer data into DAX Studio for further analysis.
- Trace analysis:
    - All Queries trace: captures query events from client tools (like Power BI)
    - Query Plan trace: query plan trace events from a SSAS Tabular server
    - Server Timing trace:
- Analyze DAX performance using Query Profiling
- Write, execute, and debug DAX queries (need to be manually copied over to Power BI Desktop, as DAX Studio is 'Read-Only'.
- Charting and Pivot table for visualization

DAX Studio for DAX performance improvements

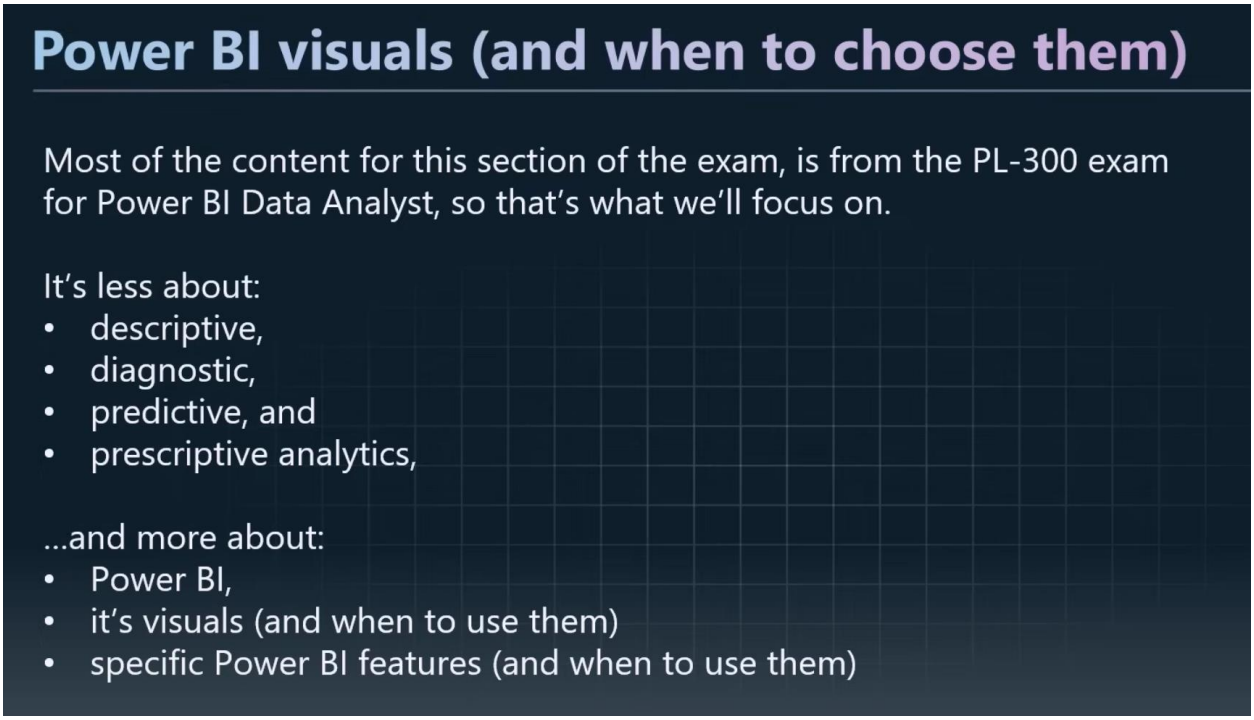
- You can load Power BI Performance Analyzer data into DAX Studio for further analysis

Ch11. Perform exploratory analytics

Fabric Analytics Concepts Notes



PBI Visuals



Fabric Analytics Concepts Notes

Visuals (when to choose/ not to choose)

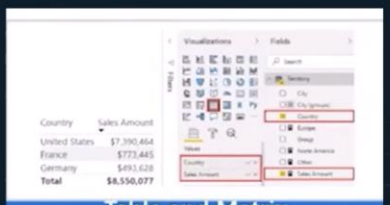


Table and Matrix Visualizations

- ✅ good for visualising fine-grained details, or for drill-throughs.
- ✅ can be used to display aggregate information (example: Monthly Revenue)
- ❌ difficult to spot long-term trends

Bar and Column Charts

- ✅ when you have one categoric variable and one numeric variable (example: Revenue by Region)
- ✅ can be 'stacked' to show more than one categoric variable
- ❌ not well suited to time-series information

Line and Area Charts

- ✅ well-suited for time-series information
- ✅ legend can be used to show how a metric changes within each category
- ❌ area chart can be difficult to interpret

Card Visualization

- ✅ good for KPI metrics
- ✅ can also include % change metrics, which give context
- ❌ don't give longer-term trends (can be coupled with a spark line to show 'momentum'

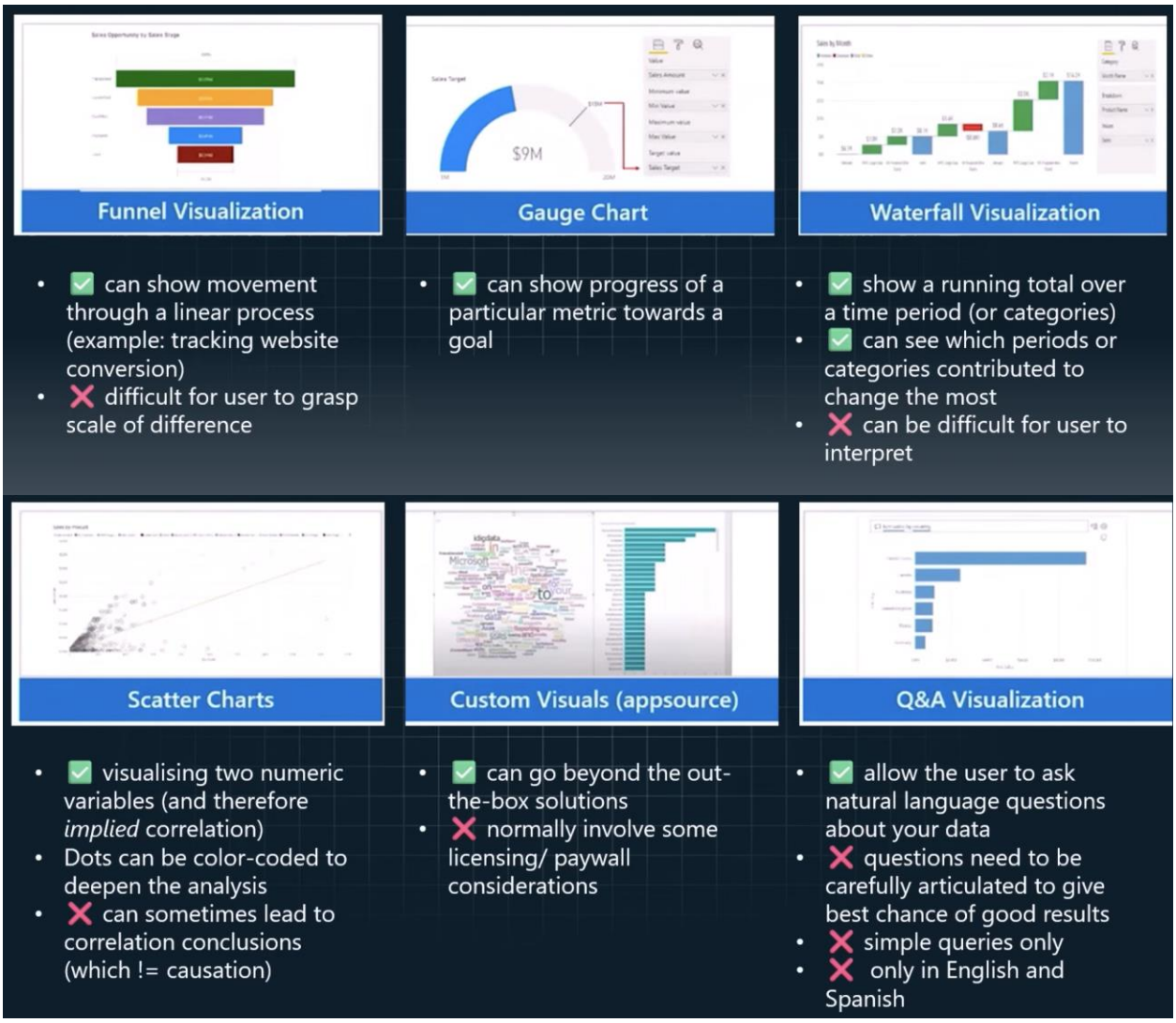
Pie chart, Donut Chart, Tree Maps

- ✅ can be used to show ratios
- ❌ difficult for human brain to compare between categories (area vs linear)
- ❌ presenting ratios obscures overall numbers
- ❌ cannot see trends over time

Combo Charts

- ✅ typically used to visualize more than one metric on Y-axis
- ✅ ❌ allows user to come to conclusion about 'correlation' (good or bad?)
- ❌ can be difficult for user to interpret (which axis shows which metric), often needs a Legend to explain

Fabric Analytics Concepts Notes



Fabric Analytics Concepts Notes

1. Correlation

- **Definition**: A statistical relationship where two variables change together.
- **Key Points**:
  - **Does not imply causation** (e.g., ice cream sales and drowning incidents both rise in summer)
  - Measured by correlation coefficients (e.g., Pearson's *r*: `−1` to `+1`).
- **Example**:
  - *Observation*: Higher social media usage correlates with higher anxiety rates.
  - *Reality*: Both may be driven by a third factor (e.g., stress).

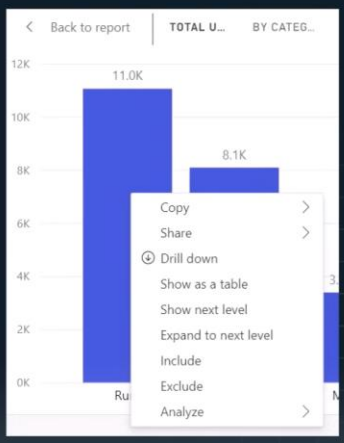
2. Causation

- **Definition**: A direct cause-and-effect relationship where one variable *changes* another.
- **Key Points**:
  - Requires **experimental control** (e.g., randomized trials) or rigorous evidence.
  - Often summarized as *"X causes Y."*
- **Example**:
  - *Observation*: Smoking increases lung cancer risk.
  - *Evidence*: Decades of controlled studies linking tobacco to cancer.

PBI Features

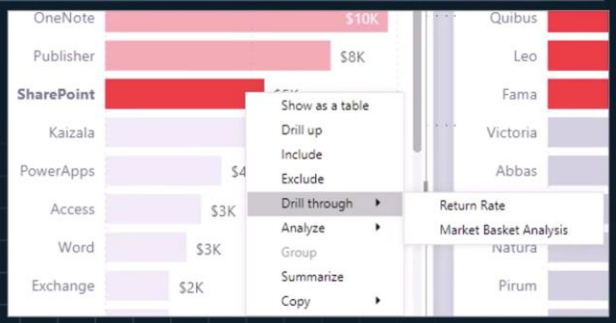
Fabric Analytics Concepts Notes

Drilldown



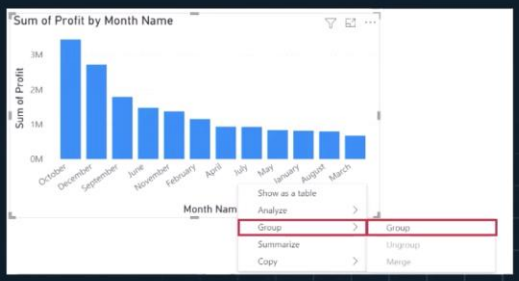
- allows user to explore data through layers of a hierarchy (either explicit or implicit)
- requires end user knowledge of the feature

Drillthrough



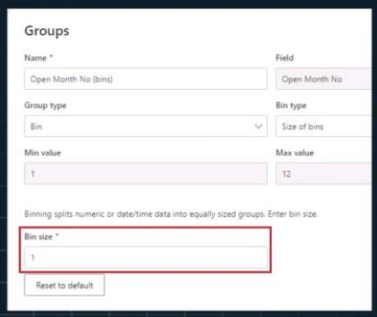
- allows user to drill-through to a separate report page, to show more detailed information for that given category.
- user journey/ navigation needs to be carefully thought
- requires end user knowledge of the feature

Grouping



- allows report author to group two or more categories

Binning

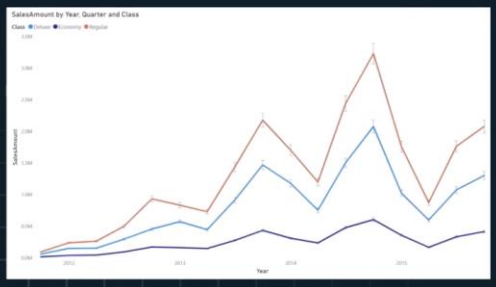


- Allows report author to create 'bins' for continuous variables
- Example: from a continuous 'Salary' variable to Salary Ranges (from $0-$30k, $30k-60k etc)

Other features

Reference lines

- Allows report authors to provide static reference lines across X or Y axis to give user some context.
- Example: average sales for all salespeople

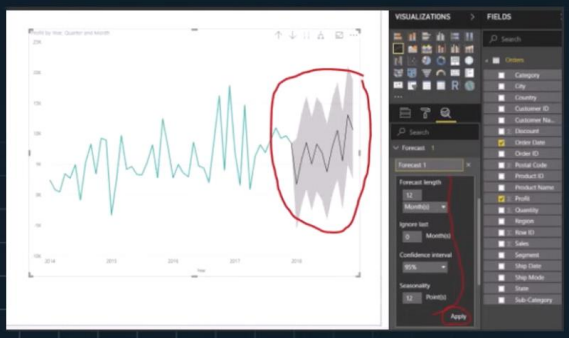


Visualising errors

- An important tool to visualise uncertainty
- Useful for showing uncertainty for a measurement or a prediction
- Can use markers, lines or shaded areas
- Requires data on errors

'What-if' parameters

- Rudimentary scenario analysis
- Requires some data engineering and/or prediction algorithm to prepare a what-if analysis

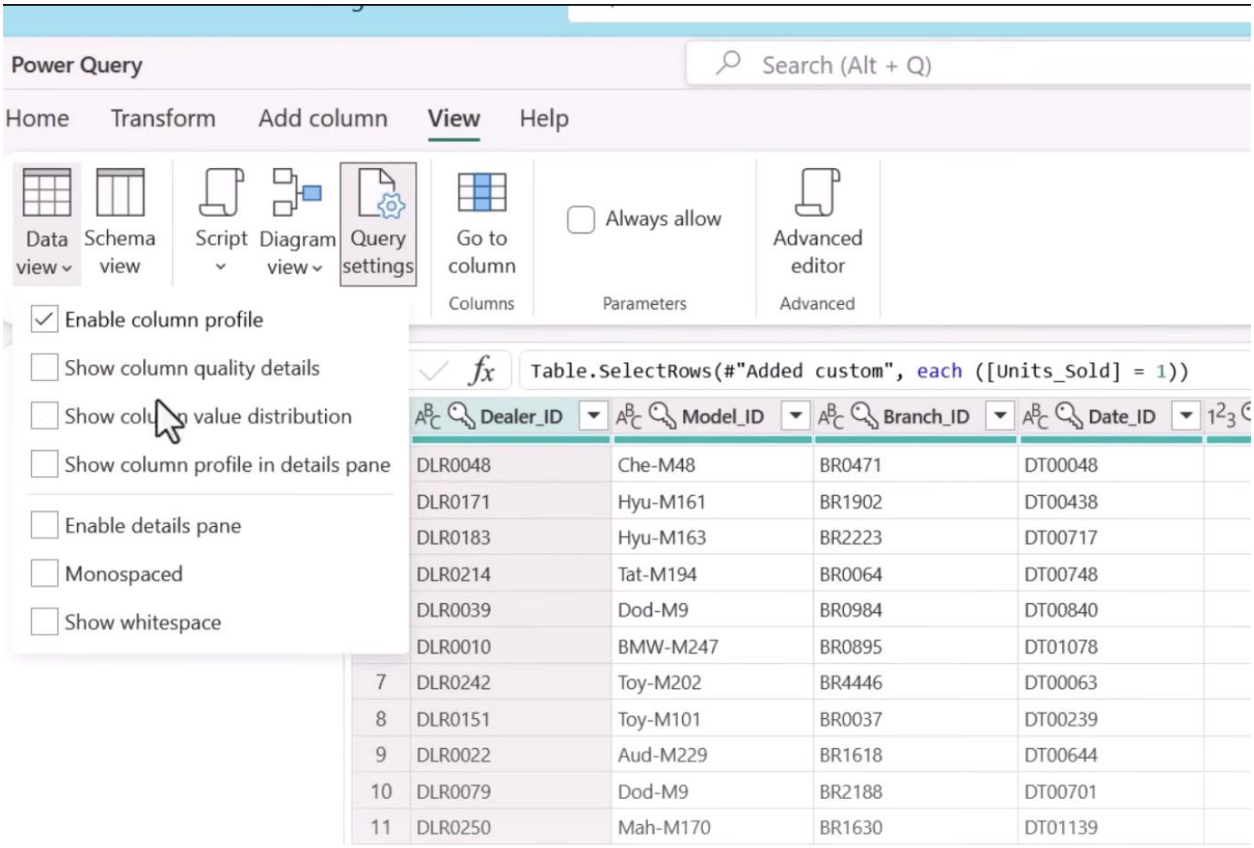


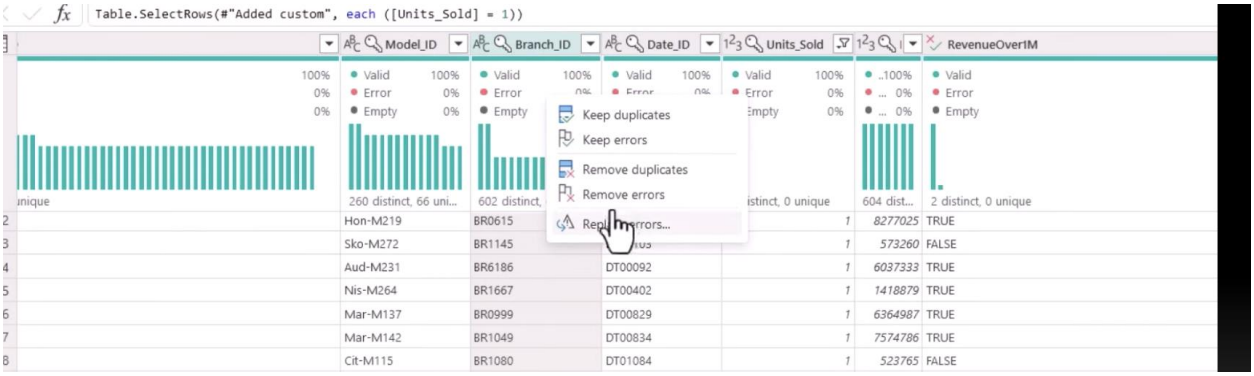
Time-series forecasting

- Gives report authors the ability to add a forecast line to time-series, continuous variables (with error bounds)
- Can be added in the Analytics pane
- Careful!

Data Profiling with Power Query

Fabric Analytics Concepts Notes





Fabric Analytics Concepts Notes

A company annual report shows a net profit of $34 million dollars.

The company has 12 business units, each with their own net profit (or loss) amount.

Which of the following visual types could (best) be used to visually show how each business unit contributed to the overall net profit metric?

A) Line chart

B) Scatter chart

C) Matrix visual

D) Waterfall chart

E) Q & A visual

Why a Waterfall Chart?

1. **Purpose-Built for Contributions**:
    - Clearly shows **cumulative impact** of each business unit (positive or negative) on the total net profit ($34M).
    - Starts with an initial value (often $0), adds/subtracts each unit's profit/loss, and ends at the final total.
2. **Example Visualization**:
    - **First bar**: Starting point ($0).
    - **Middle bars**: Each business unit's profit (green) or loss (red).
    - **Final bar**: Total net profit ($34M).

You want to add measurement error bars on a time-series line chart, to show potential error in each measurement.

Where would you go add this information to your visual?

A) Analytics pane

B) Format visual pane

C) View tab > Show error bars

D) Build visual pane

E) In the Model view

Fabric Analytics Concepts Notes

You want to visually compare two continuous variables (Age and Height of Survey respondents) in one chart.

Which of the following visual types would (best) represent this data?

A) Line chart

B) Stacked bar visual

C) Scatter visual

D) Q & A visual

E) Matrix visual

By default, the Data Profiling tool reviews the top *N* rows of your dataset to show you potential data quality issues.

What is *N*?

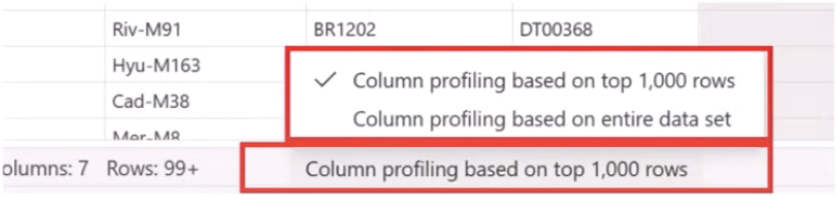
A) 10

B) 100

C) 500

D) 1000

E) 10000



Which of the following features of the Data Profiling tool can help you identify duplicate values in a column that you plan to use as a key to join on?

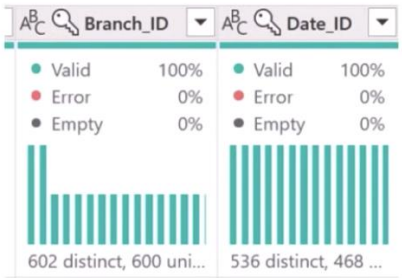
A) Column quality details

B) Column value distribution

C) Column duplicate analysis

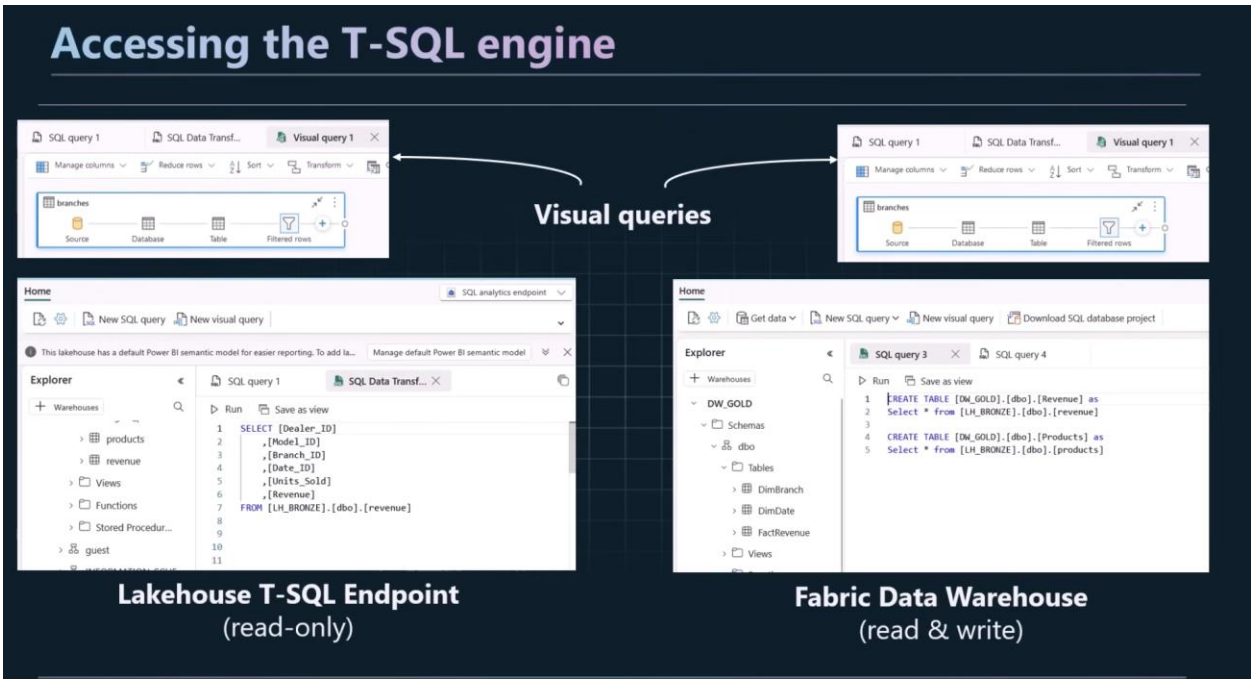
D) Column key constraints

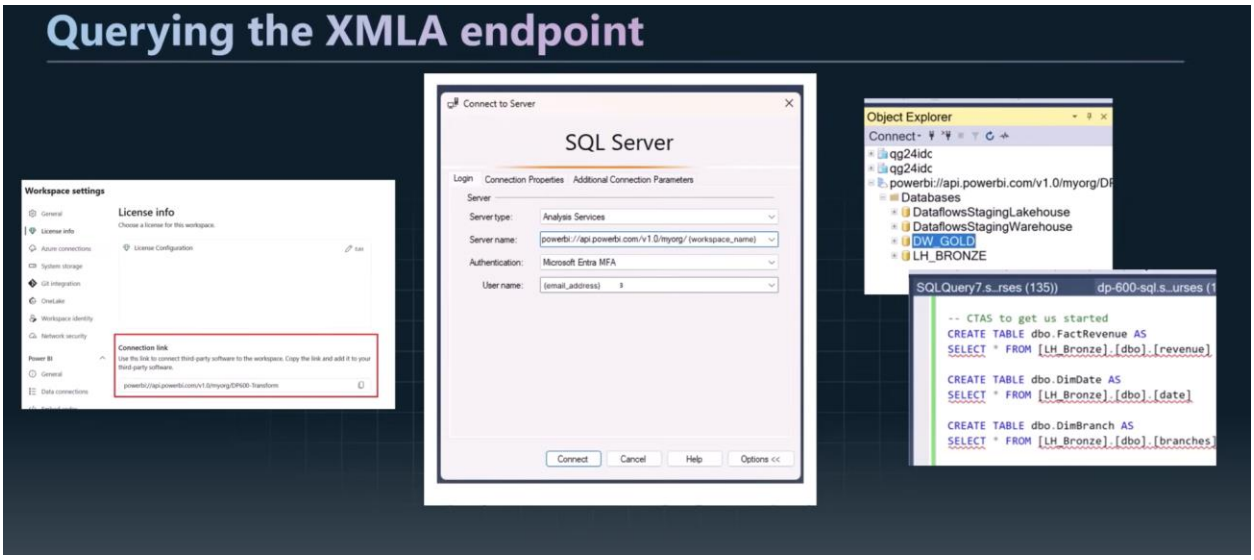
E) Group By



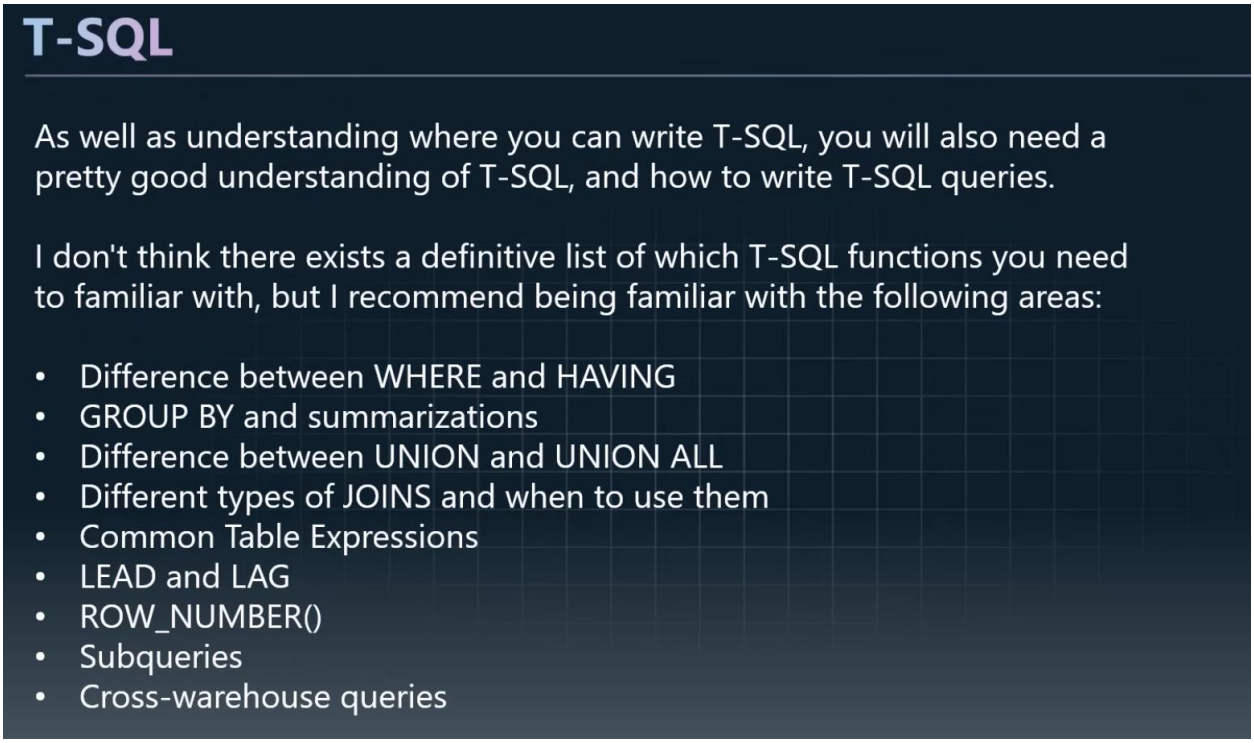
Ch12. Query data using SQL

- Query a lakehouse in Fabric by using SQL queries or the visual query editor
- Query a warehouse in Fabric by using SQL queries or the visual query editor
- Connect to and query datasets by using the XMLA endpoint





Fabric Analytics Concepts Notes



T-SQL Concept	Purpose	Example/Syntax	Key Notes
WHERE vs. HAVING	Filters rows (WHERE) vs. filters aggregates (HAVING).	SELECT ... WHERE salary > 50000 vs. GROUP BY ... HAVING AVG(salary) > 50000	WHERE operates before GROUP BY; HAVING after.
GROUP BY	Groups rows by columns for aggregations (SUM, AVG, COUNT).	SELECT dept, AVG(salary) FROM employees GROUP BY dept	Essential for summary reports.
UNION vs. UNION ALL	Combines result sets (UNION removes duplicates; UNION ALL retains them).	SELECT ... UNION SELECT ...	UNION ALL is faster but less clean.
JOIN Types	Combines tables (INNER, LEFT, RIGHT, FULL, CROSS).	SELECT * FROM A INNER JOIN B ON A.id = B.id	LEFT JOIN keeps all left table rows; FULL keeps all from both.

Fabric Analytics Concepts Notes

T-SQL Concept	Purpose	Example/Syntax	Key Notes
Common Table Expressions (CTEs)	Creates temporary result sets for complex queries.	WITH CTE AS (SELECT ...) SELECT * FROM CTE	Improves readability; reusable in the same query.
LEAD/LAG	Accesses next/previous row values without self-join.	SELECT name, LAG(salary) OVER (ORDER BY date)	Window functions; useful for trend analysis.
ROW_NUMBER()	Assigns sequential numbers to rows (e.g., ranking).	SELECT ROW_NUMBER() OVER (PARTITION BY dept ORDER BY salary)	Often used with pagination or deduplication.
Subqueries	Nested queries (can be in SELECT, FROM, WHERE).	SELECT name FROM employees WHERE salary > (SELECT AVG(salary) FROM employees)	Correlated subqueries reference outer query.
Cross-Warehouse Queries	Queries across multiple databases/warehouses (e.g., Azure Synapse).	SELECT * FROM db1.table1 JOIN db2.table2 ON ...	Requires proper permissions and linked servers/federated queries.

Key Takeaways:

1. **WHERE** filters raw data; **HAVING** filters grouped data.
2. **UNION ALL** > **UNION** for performance (if duplicates don't matter).
3. **CTEs** simplify complex logic; **LEAD/LAG** avoid self-joins.
4. **ROW_NUMBER()** is versatile for rankings/pagination.

Fabric Analytics Concepts Notes

```
-- change the above query to only return the TOP 3 Branches in Spain
SELECT TOP 3 rev.Branch_ID, B.Branch_NM, B.Country_Name, SUM(Revenue) AS TotalRevenue FROM dbo.FactRevenue rev
INNER JOIN dbo.DimBranch B on rev.Branch_ID = B.Branch_ID
WHERE B.Country_Name = ' Spain'
GROUP BY rev.Branch_ID, B.Country_Name, B.Branch_NM
ORDER BY SUM(rev.Revenue) DESC
```

Messages

Branch_ID	Branch_NM	Country_Name	TotalRevenue
BR2047	Volkswagen South Africa Motors	Spain	26346189
BR2050	Volkswagen South Africa Motors	Spain	24546006
BR2048	Volkswagen South Africa Motors	Spain	14758116

For aggregates like sum, we are use having instead of where clause

```
-- what about if we want to filter after the aggregate? I.e. Give me all the Branches that
-- has a Revenue > some amount X
SELECT B.Branch_ID, B.Branch_NM, SUM(Revenue) AS TotalRevenue FROM dbo.FactRevenue rev
INNER JOIN dbo.DimBranch B on rev.Branch_ID = B.Branch_ID
GROUP BY B.Branch_ID, B.Branch_NM
HAVING SUM(Revenue) > 50000000
```

CTE

Final statement of CTE is select stmt to return sth out of CTE

```
-- Common Table Expressions
WITH top_5_rev as (
    SELECT TOP 5 Branch_ID, SUM(Revenue) AS TotalRevenue FROM dbo.FactRevenue
    GROUP BY Branch_ID
    ORDER BY SUM(Revenue) DESC
),

branches as (
    SELECT Branch_ID, Branch_NM from dbo.DimBranch
)

SELECT t5.Branch_ID, b.Branch_NM, t5.TotalRevenue  FROM top_5_rev t5
LEFT JOIN branches b on t5.Branch_ID = b.Branch_ID
```

```sql
-- LAG and LEAD

with revT as (
    SELECT TOP 5 Branch_ID
        , floor(Revenue/1000000) as Rev_M
    FROM dbo.FactRevenue
)

SELECT Branch_ID
    , Rev_M
    , LAG(Rev_M, 2) OVER(ORDER BY Rev_M) as lag_col
from revT
```

Branch_ID	Rev_M
BR0051	1
BR0071	6
BR0121	3
BR0161	2
BR0161	2

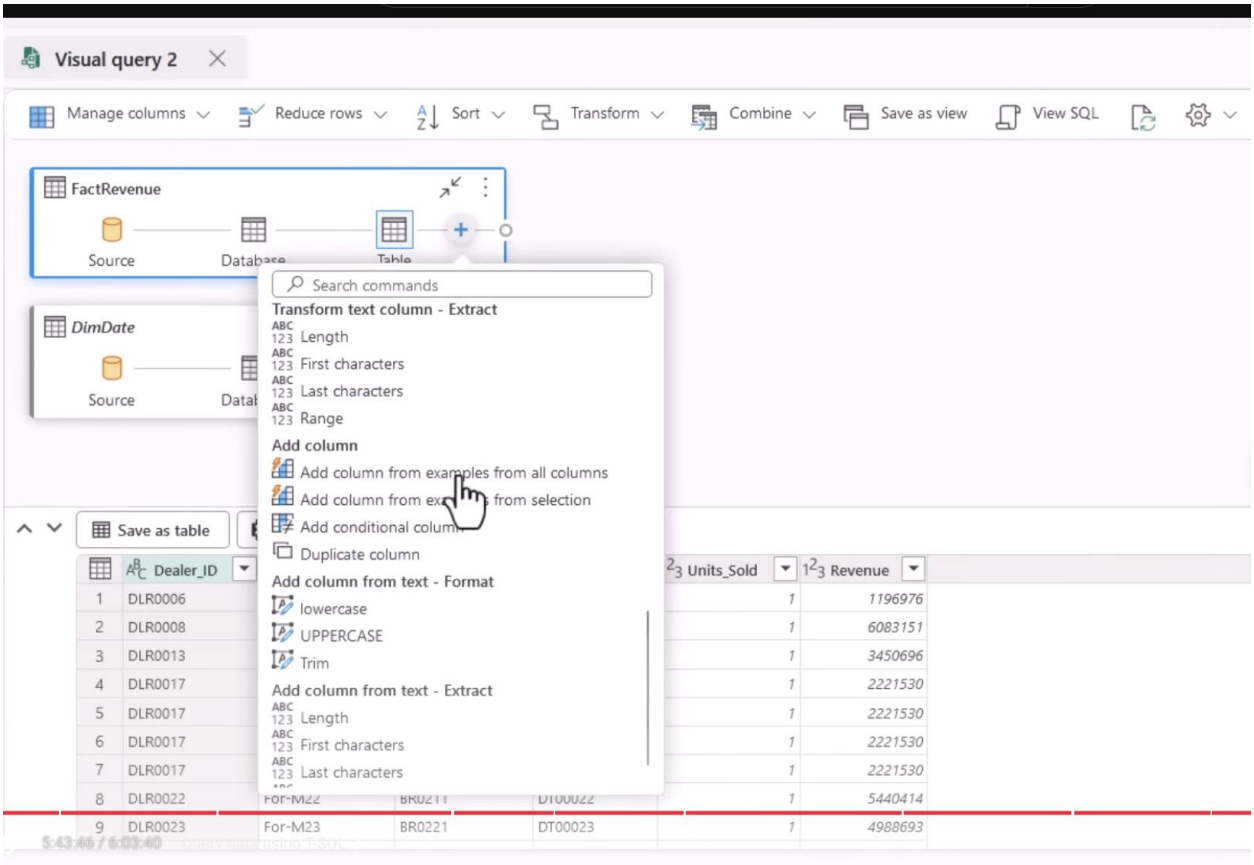
```sql
-- LAG and LEAD

with revT as (
    SELECT TOP 5 Branch_ID
        , floor(Revenue/1000000) as Rev_M
    FROM dbo.FactRevenue
)

SELECT Branch_ID
    , Rev_M
    , LAG(Rev_M, 1) OVER(ORDER BY Rev_M) as lag_col
from revT
```

Messages

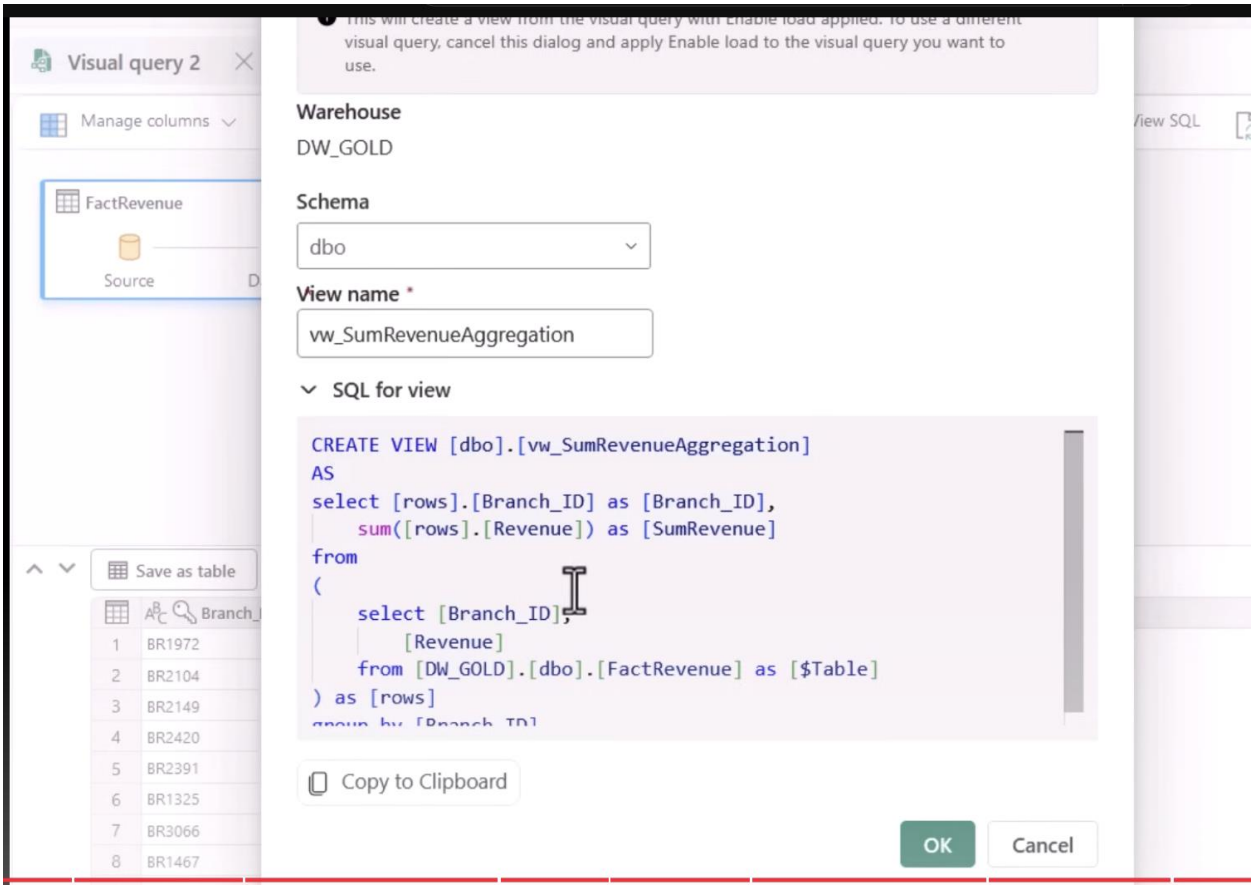
Branch_ID	Rev_M	lag_col
BR0051	1	NULL
BR0161	2	1
BR0161	2	2
BR0121	3	2
BR0071	6	3

```sql
-- Subqueries
SELECT * from (
    SELECT * from dbo.FactRevenue
    WHERE Dealer_ID in ('DLR0001', 'DLR0017')
) sub
```

Fabric Analytics Concepts Notes



Fabric Analytics Concepts Notes



Question
The following SQL script creates the result shown below. What is {FUNCTION}?

```
SELECT
   Branch_ID,
   , Col1
   , {FUNCTION}
From Sales
```

Branch_ID	Col1	Col2
BR0051	1	NULL
BR0161	2	NULL
BR0161	2	1
BR0121	3	2
BR0071	6	2

A) LAG(Col1, 2) OVER(ORDER BY Col1) as Col2

B) LEAD(Col1, 2) OVER(ORDER BY Col1) as Col2

C) LAG(Col1, -2) OVER(ORDER BY Col1) as Col2

D) LEAD(Col1, -2) OVER(ORDER BY Col1) as Col2

It is lag func because col 2 is 2 rows behind col1. NULL at top → Lag
Lead looks ahead in time rather than back in time
Offset is always +ve

Fabric Analytics Concepts Notes

You have the following query analyzing Sales data for various products:

```
SELECT s.ProductName, d.Year, SUM(s.SalesAmount)
FROM Sales s
LEFT JOIN DimDate d ON s.DateKey = d.DateKey
```

Your goal is to analyze the Sales data by ProductName and Year, but only for products that have a yearly Sales Amount of more than $50,000.

How would you complete the query?

A) GROUP BY p.ProductKey, d.DateKey
HAVING SUM(s.SalesAmount) > 50000

B) GROUP BY s.ProductName, d.Year
HAVING SUM(s.SalesAmount) > 50000

C) WHERE s.SalesAmount > 50000
GROUP BY p.ProductKey, d.DateKey

D) WHERE s.SalesAmount > 50000
GROUP BY p.ProductName, d.Year

You are trying to inspect a join between two tables to spot referential integrity violations.

Which of the following T-SQL join types would be easiest to identify keys on both sides of the join that do not have a match on the other side of the join?

A) LEFT JOIN

B) RIGHT JOIN

C) INNER JOIN

D) FULL OUTER JOIN

E) CROSS JOIN

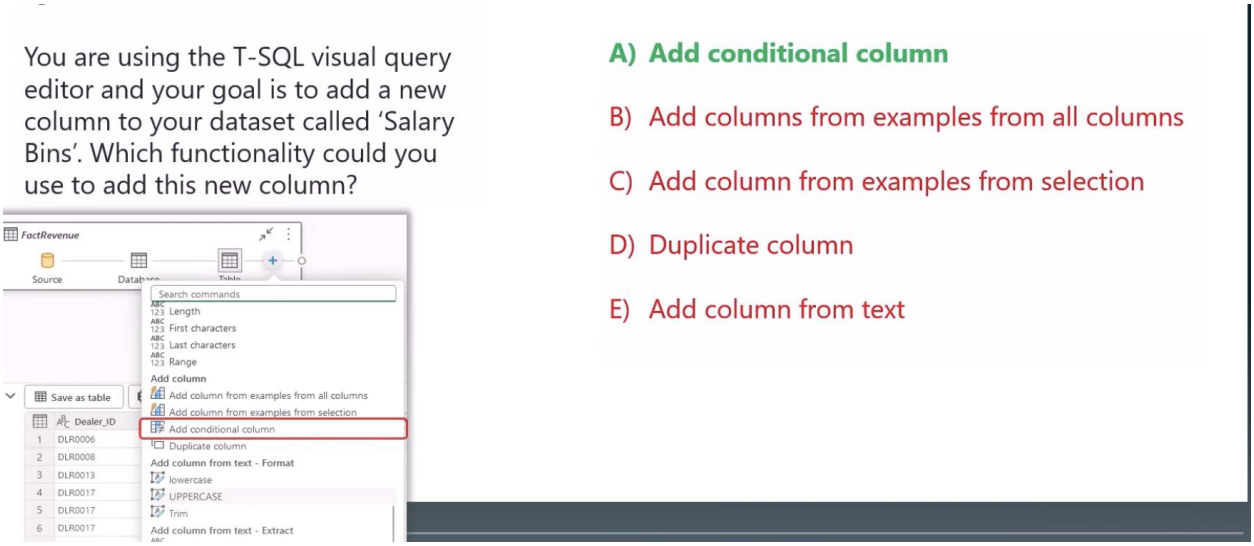
Cross join – tells all different combinations

Referential integrity is a database concept that ensures relationships between tables remain consistent and valid. It guarantees that any foreign key value in a child table must match a primary key value in the parent table (or be NULL).

Key Rules of Referential Integrity

1. **Primary Key (Parent Table)**:
    - Must be unique and non-null (e.g., customer_id in a Customers table).
2. **Foreign Key (Child Table)**:
    - Must reference an existing primary key or be NULL (e.g., customer_id in an Orders table).
3. **Actions Enforced by DBMS**:
    - **On Delete**: What happens to child rows if a parent row is deleted?
        - CASCADE: Delete child rows automatically.
        - SET NULL: Set foreign key to NULL.
        - RESTRICT: Block deletion if child rows exist.
    - **On Update**: Automatically update foreign keys if the parent key changes.

Fabric Analytics Concepts Notes



Reference

https://www.youtube.com/watch?v=Bjk93hi21QM&ab_channel=LearnMicrosoftFabricwithWill