End-to-End Sales Data Analytics Platform on Azure

Trainee Name: Swathi Baskaran

Table of Contents

Project Statement

The goal of this project is to design and implement an end-to-end cloud-based data analytics platform that ingests raw sales, customer, and product data from CSV files, processes and transforms it into a structured data warehouse using Azure Data Factory and Azure Synapse Analytics, orchestrates the workflows and deployments with Azure DevOps, and delivers interactive business intelligence dashboards through Power BI. This solution provides real-time and historical insights into sales performance, customer behavior, and product trends, enabling data-driven decision-making.

Project Overview

This project demonstrates the implementation of a modern data engineering and analytics solution on Microsoft Azure.

- **Data Sources:** Sales, Products, and Customers data in CSV format.
- **Data Ingestion:**
  - Azure Data Factory (ADF) pipelines were developed to extract raw data from the Azure Storage Account.
  - Incremental loads and scheduled triggers ensure automated data refresh.
- **Data Transformation and Storage:**
  - Data is cleaned, standardized, and transformed using Azure Synapse Analytics.
  - Fact and dimension tables (FactSales, DimCustomer, DimProduct) form a star schema optimized for analytics.
  - SQL scripts and stored procedures manage transformations.
- **Orchestration & CI/CD:**
  - Azure DevOps manages source control, versioning, and deployment of ADF pipelines, Synapse scripts, and ARM templates.
  - CI/CD pipelines (YAML) ensure seamless deployment across environments.

- **Visualization & Reporting**:
    - Power BI dashboards provide interactive analytics with KPIs such as total sales, sales by product, customer segmentation, and time-based sales trends.
    - DAX measures ensure accurate aggregation and business logic representation.
- **Business Value:**
    - Enables management to analyze customer purchasing patterns, product performance, and revenue trends.
    - Provides a scalable, automated, and reusable cloud data analytics solution.

Prerequisities

1. **Azure Subscription** with permission to create and manage resources.
2. Basic knowledge of **SQL, ETL concepts**, and **data warehousing.**
3. Familiarity with **Azure Portal, Azure Storage** and **Power BI.**
4. **Power BI** installed and configured for building and publishing reports.
5. Access to **Azure DevOps** for source control and deployment.
6. Source CSV files available:
    - sales.csv (transactional sales data)
    - products.csv (product catalog)
    - customers.csv (customer details)

Azure Resources Used for this Project

- **Azure Storage Account** – Stores raw CSV data files
- **Azure Data Factory (ADF)** – Ingests and transforms raw data through pipelines.
- **Azure Synapse Analytics (Synapse Studio)** – Acts as the data warehouse, hosts star schema (FactSales, DimCustomer, DimProduct)
- **Azure DevOps** – Provides version control, CI/CD pipelines, and deployment automation.
- **Power BI** – Used for building and publishing dashboards for reporting.

Project Objectives

- Ingest CSV files from Azure Storage into a cloud-based system.
- Automate ETL processes using Azure Data Factory pipelines.
- Build a star schema data warehouse in Synapse (FactSales, DimCustomer, DimProduct).
- Implement CI/CD using Azure DevOps for managing ADF pipelines, Synapse scripts, and templates.
- Develop interactive dashboards in Power BI for sales, customer, and product analytics.
- Enable automation and scalability for handling larger datasets and scheduled refreshes.

Tools Used

- **Azure Data Factory (ADF)**
    - Cloud-based ETL tool
    - Extracts data from Azure Storage
    - Transforms and loads into Synapse Analytics
- **Azure Synapse Analytics**
    - Cloud data warehouse
    - Stores transformed data in star schema (FactSales, DimCustomer, DimProduct)
    - Used for running SQL queries and managing tables
- **Azure Storage Account**
    - Landing zone for raw CSV files (sales.csv, products.csv, customers.csv)
    - Source for ADF pipelines
- **Azure DevOps**
    - Source control for JSON pipeline definitions, SQL scripts, and YAML files
    - CI/CD pipeline automation for deployments
- **Power BI**
    - Used for data modeling, visualization, and creating interactive dashboards from Synapse data.

Execution Overview

1. **Data Ingestion**
   - Raw CSV files (sales.csv, products.csv, customers.csv) are uploaded to the Azure Storage Account.
   - Azure Data Factory (ADF) pipelines are triggered to pick up these files.

2. **Data Transformation & Loading**
   - ADF Pipelines transform the raw data (Data type conversions, formatting, cleaning).
   - Transformed data is loaded into Azure Synapse Analytics tables.
   - The star schema is created with:
     - FactSales (Transactional Sales Data)
     - DimCustomer (Customer Details)
     - DimProduct (Product Details)

3. **Data Storage & Querying**
   - Synapse Analytics acts as the central data warehouse
   - SQL scripts define and manage schema, relationships, and business rules.

4. **Source Control & Deployment**
   - All ADF pipeline definitions, Synapse SQL scripts, and configurations are stored in Azure DevOps.
   - YAML Pipelines in DevOps handle CI/CD automation, ensuring consistent deployments across environments.

5. **Data Visualization**
   - Power BI connects directly to Synapse Analytics.
   - Data models are built linking FactSales with DimCustomer and DimProduct.
   - Interactive dashboards are created to show:
     - Sales by product
     - Sales by customer
     - Overall sales performance trends

6. End-to-End Flow

- New CSV files land in Storage → ADF ingests and loads → Synapse stores in warehouse → Power BI visualizes → DevOps manages version control & automation.

Implementation – Tasks Performed

1. **Data Preparation**
   - Collected and organized source data files: sales.csv, products.csv, and customers.csv.
   - Uploaded raw data files into the Azure Storage Account (Blob container).
2. **Pipeline Development in ADF**
   - Created Linked Services to connect ADF with Azure Storage and Synapse Analytics.
   - Designed Datasets representing input CSV files and output tables.
   - Built Data Pipelines to:
     - Extract CSV data from storage
     - Transform the data into structured format
     - Load the transformed data into Synapse Analytics
3. **Data Warehouse Setup in Synapse Analytics**
   - Designed and created a star schema with:
     - FactSales table (Transactional sales data)
     - DimCustomer table (customer attributes)
     - DimProduct table (product details)
   - Executed SQL scripts for table creation, constraints, and data loading.
4. **Integration with Azure DevOps**
   - Configured Git repository in Azure DevOps to store JSON pipeline definitions, SQL scripts, and ARM templates.
   - Implemented CI/CD pipelines using YAML for automated deployments.
5. **Data Visualization in Power BI**

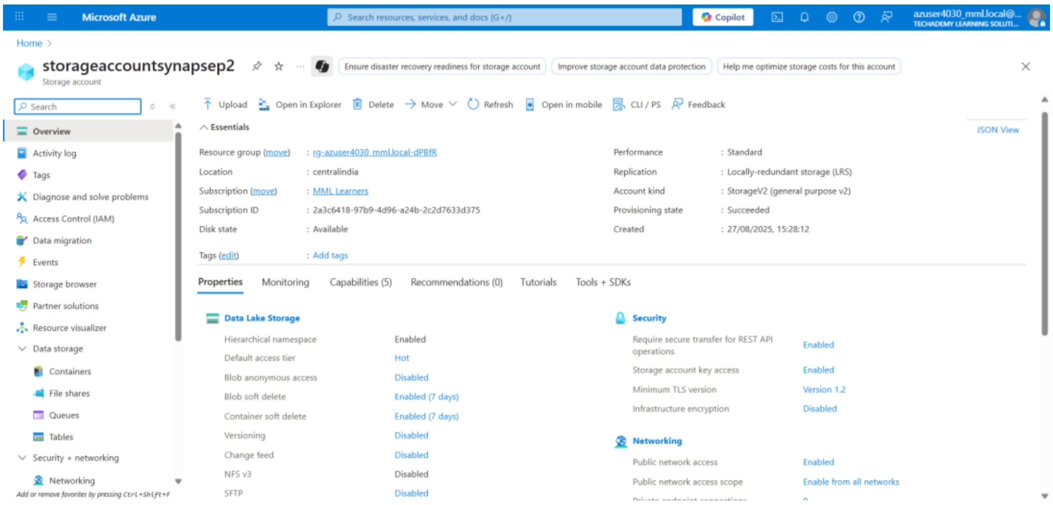
- Connected Power BI to Synapse Analytics as the data source.
- Built relationships between FactSales, DimCustomer, and DimProduct.
- Designed dashboards and reports showing key insights such as:
  - Total sales by Product
  - Total sales by Customer
  - Sales Performance trends

6. **Testing and Validation**
   - Verified correctness of ETL pipelines by comparing source and target data.
   - Validated star schema relationships and Power BI reports for accuracy.
7. **Final Deployment**
   - Published final dashboards in Power BI for reporting.
   - Ensured automation of pipeline runs for regular data refresh.

Practical Implementation on Azure Portal

Step 1: Create Azure Storage Account

- Provision a Storage Account for raw and processed data



storageaccountsynapsep2

Storage account

Essentials

Resource group (move) : rg-azuser4030_mml.local-dPBfR

Location : centralindia

Subscription (move) : MML Learners

Subscription ID : 2a3c6418-97b9-4d96-a24b-2c2d7633d375

Disk state : Available

Tags (edit) : Add tags

Performance : Standard

Replication : Locally-redundant storage (LRS)

Account kind : StorageV2 (general purpose v2)

Provisioning state : Succeeded

Created : 27/08/2025, 15:28:12

Data Lake Storage

Hierarchical namespace	Enabled
Default access tier	Hot
Blob anonymous access	Disabled
Blob soft delete	Enabled (7 days)
Container soft delete	Enabled (7 days)
Versioning	Disabled
Change feed	Disabled
NFS v3	Disabled
SFTP	Disabled

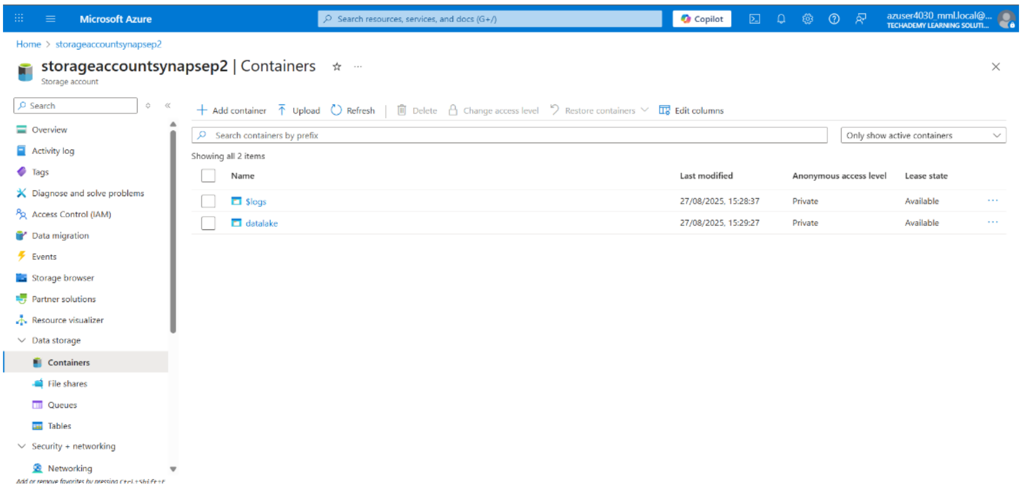
Security

Require secure transfer for REST API operations	Enabled
Storage account key access	Enabled
Minimum TLS version	Version 1.2
Infrastructure encryption	Disabled

Networking

Public network access	Enabled
Public network access scope	Enable from all networks

- Create a container called **'datalake'** in the storage account for the raw CSV files in the Azure Storage Account



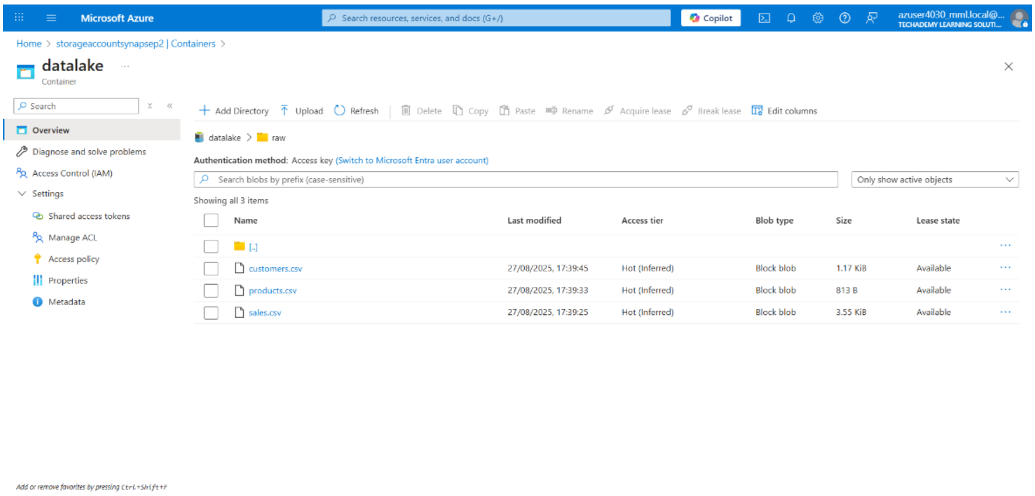
storageaccountsynapsep2 | Containers

Storage account

Showing all 2 items

Name	Last modified	Anonymous access level	Lease state
$logs	27/08/2025, 15:28:37	Private	Available
datalake	27/08/2025, 15:29:27	Private	Available

- Upload CSV files into the source container



datalake

Container

datalake > raw

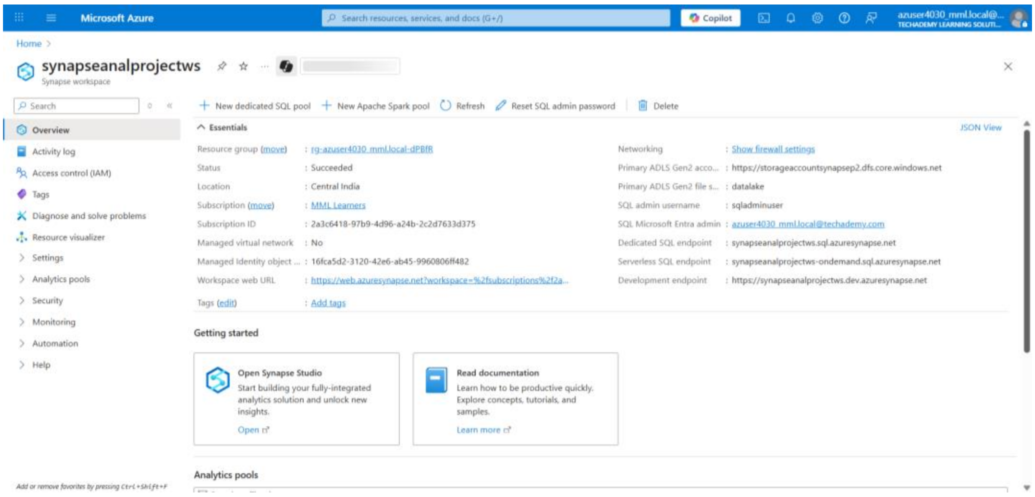
Authentication method: Access key (Switch to Microsoft Entra user account)

Showing all 3 items

Name	Last modified	Access tier	Blob type	Size	Lease state
[..]					
customers.csv	27/08/2025, 17:39:45	Hot (Inferred)	Block blob	1.17 KiB	Available
products.csv	27/08/2025, 17:39:33	Hot (Inferred)	Block blob	813 B	Available
sales.csv	27/08/2025, 17:39:25	Hot (Inferred)	Block blob	3.55 KiB	Available

Step 2: Create an Azure Synapse Analytics Workspace

- Creating an Azure Synapse Studio



synapseanalprojectws
Synapse workspace

Essentials

Resource group (move) : rg-azuser4030_mml.local-dPBfR
Status : Succeeded
Location : Central India
Subscription (move) : MML Learners
Subscription ID : 2a3c6418-97b9-4d96-a24b-2c2d7633d375
Managed virtual network : No
Managed Identity object ... : 16fca5d2-3120-42e6-ab45-9960806ff482
Workspace web URL : https://web.azuresynapse.net?workspace=%2fsubscriptions%2f2a...
Tags (edit) : Add tags

Networking : Show firewall settings
Primary ADLS Gen2 acco... : https://storageaccountsynapsep2.dfs.core.windows.net
Primary ADLS Gen2 file s... : datalake
SQL admin username : sqladminuser
SQL Microsoft Entra admin : azuser4030_mml.local@techademy.com
Dedicated SQL endpoint : synapseanalprojectws.sql.azuresynapse.net
Serverless SQL endpoint : synapseanalprojectws-ondemand.sql.azuresynapse.net
Development endpoint : https://synapseanalprojectws.dev.azuresynapse.net

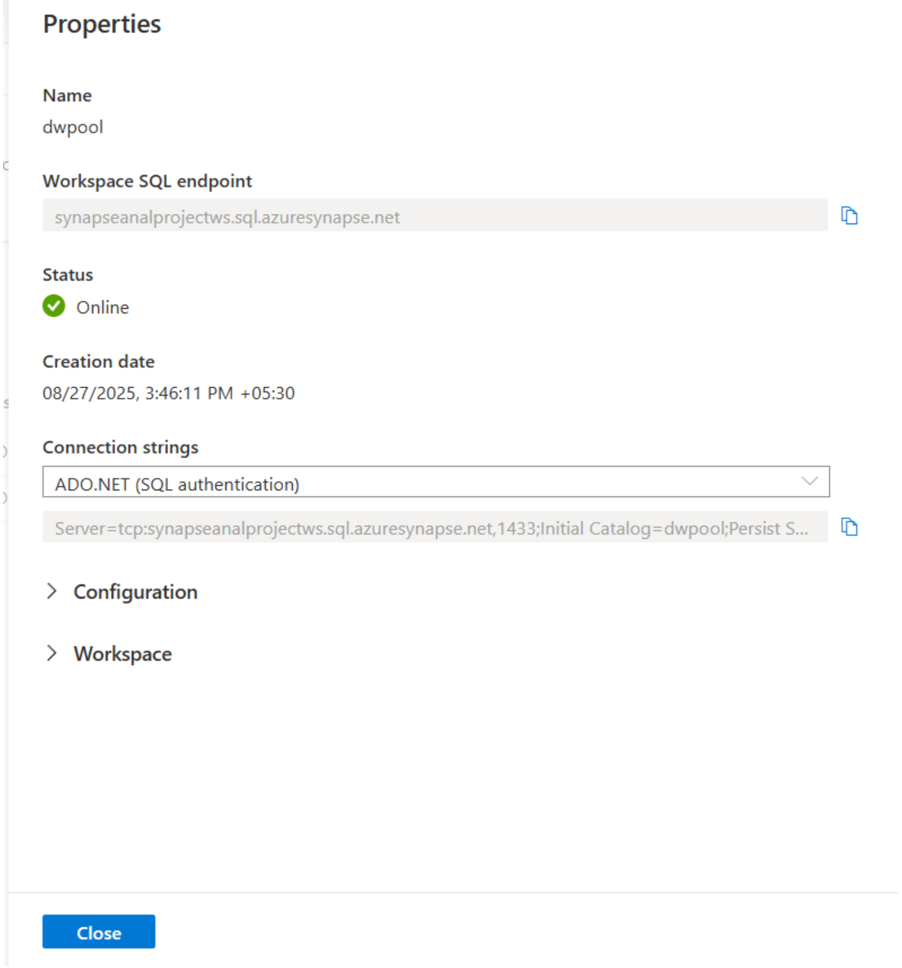
Getting started

Open Synapse Studio
Start building your fully-integrated analytics solution and unlock new insights.
Open

Read documentation
Learn how to be productive quickly. Explore concepts, tutorials, and samples.
Learn more

Analytics pools

- Creating a SQL Pool in Azure Synapse Studio



Properties

Name
dwpool

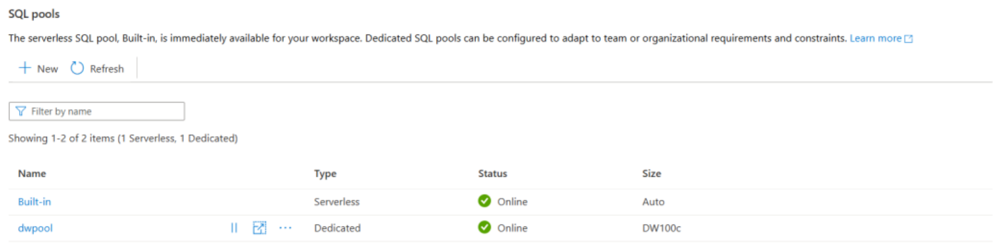
Workspace SQL endpoint
synapseanalprojectws.sql.azuresynapse.net

Status
Online

Creation date
08/27/2025, 3:46:11 PM +05:30

Connection strings
ADO.NET (SQL authentication)
Server=tcp:synapseanalprojectws.sql.azuresynapse.net,1433;Initial Catalog=dwpool;Persist S...

- Monitoring the SQL pool created



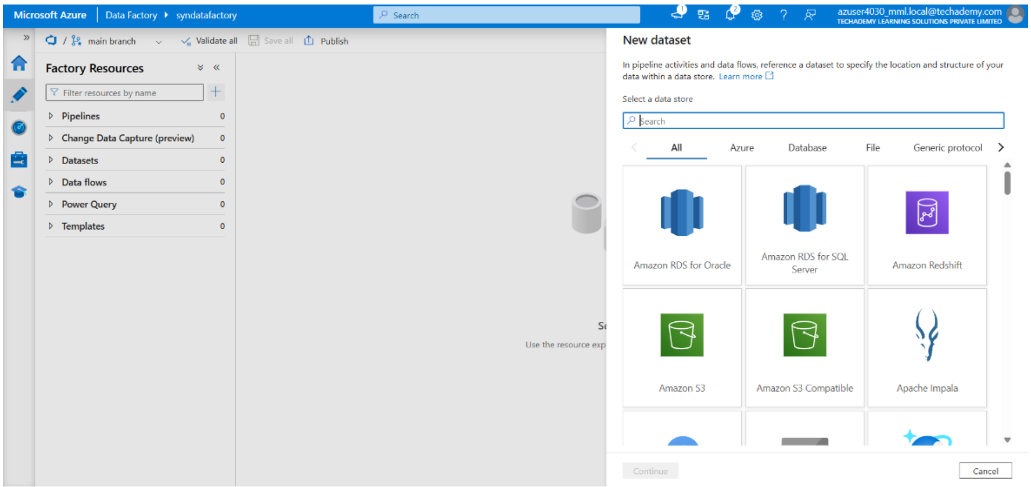
The serverless SQL pool, Built-in, is immediately available for your workspace. Dedicated SQL pools can be configured to adapt to team or organizational requirements and constraints. Learn more

Showing 1-2 of 2 items (1 Serverless, 1 Dedicated)

Name	Type	Status	Size
Built-in	Serverless	✓ Online	Auto
dwpool	Dedicated	✓ Online	DW100c

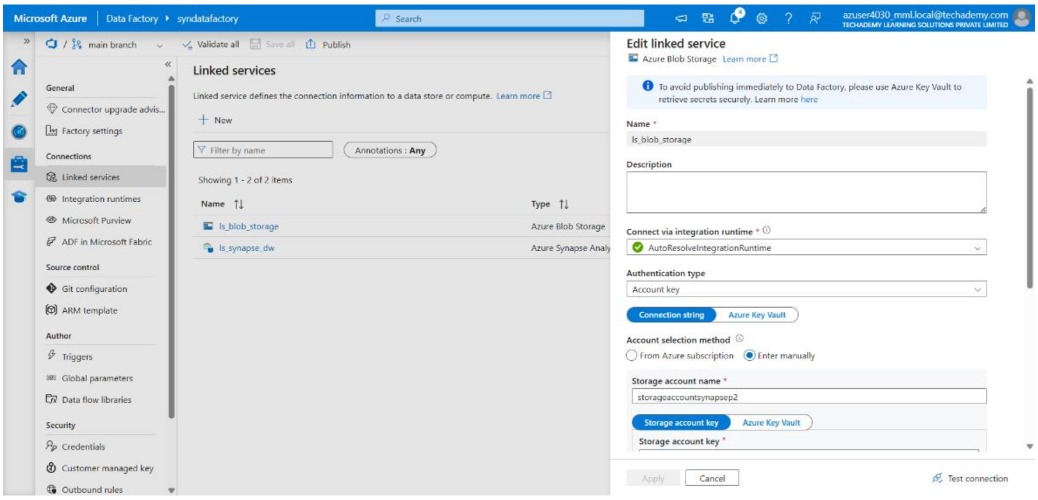
Step 3: Create Azure Data Factory Studio

- Creating datasets within Azure Data Factory (ADF)



In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

- Creating linked service for Azure Blob Storage



Linked service defines the connection information to a data store or compute. Learn more

Showing 1 - 2 of 2 items

Name	Type
ls_blob_storage	Azure Blob Storage
ls_synapse_dw	Azure Synapse Analy

Edit linked service

Azure Blob Storage Learn more

To avoid publishing immediately to Data Factory, please use Azure Key Vault to retrieve secrets securely. Learn more here

Name *: ls_blob_storage

Description:

Connect via integration runtime *: AutoResolveIntegrationRuntime

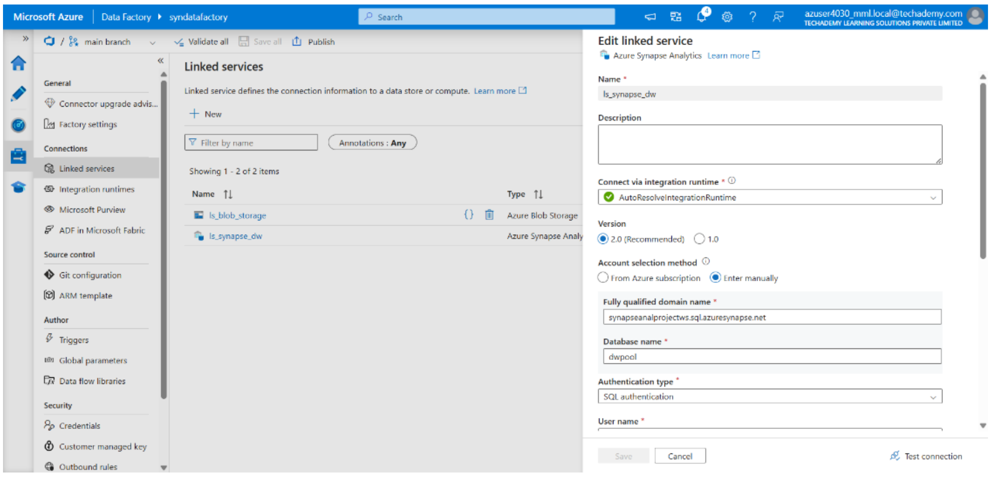
Authentication type: Account key

Account selection method: ○ From Azure subscription ● Enter manually

Storage account name *: storageaccountsynapsep2

Storage account key *

- Create linked service for Azure Synapse Analytics



Linked services

Linked service defines the connection information to a data store or compute. Learn more

+ New

Filter by name | Annotations : Any

Showing 1 - 2 of 2 items

Name	Type
ls_blob_storage	Azure Blob Storage
ls_synapse_dw	Azure Synapse Analy

Edit linked service

Azure Synapse Analytics Learn more

Name *: ls_synapse_dw

Description:

Connect via integration runtime *: AutoResolveIntegrationRuntime

Version: ● 2.0 (Recommended) ○ 1.0

Account selection method: ○ From Azure subscription ● Enter manually

Fully qualified domain name *: synapseanalprojectws.sql.azuresynapse.net

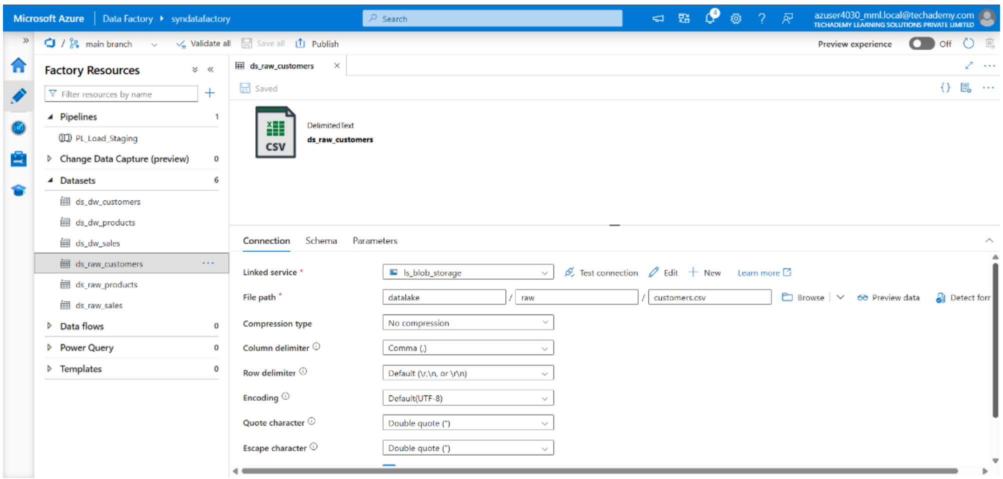
Database name *: dwpool

Authentication type *: SQL authentication

User name *

Save | Cancel | Test connection

- Connecting Azure Blob Storage Linked Service to the raw dataset



Linked service *: ls_blob_storage | Test connection | Edit | + New | Learn more

File path *: datalake / raw / customers.csv | Browse | Preview data | Detect forr

Compression type: No compression

Column delimiter: Comma (,)

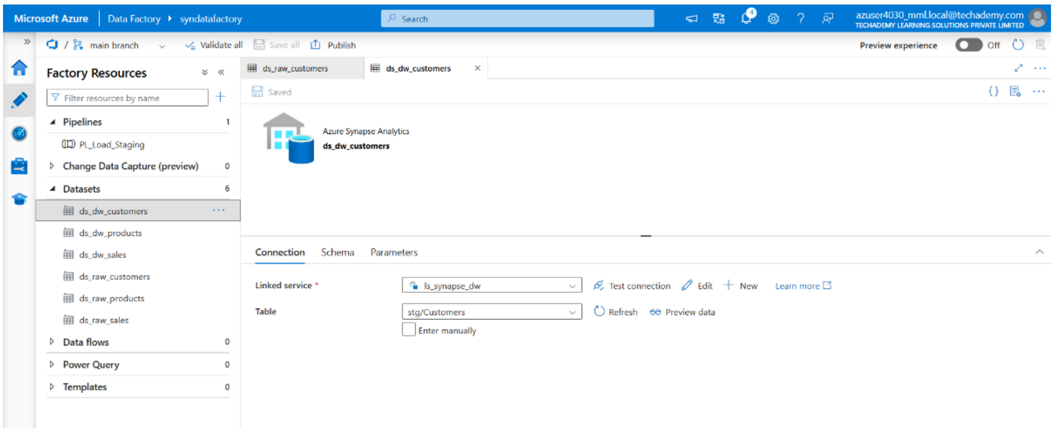
Row delimiter: Default (\r,\n, or \r\n)

Encoding: Default(UTF-8)

Quote character: Double quote (")

Escape character: Double quote (")

- Connecting Azure Synapse Analytics Linked Service to the data warehouse dataset



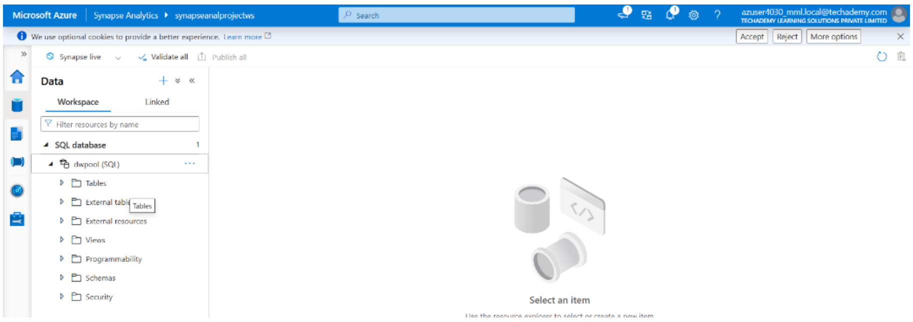
Linked service *: ls_synapse_dw | Test connection | Edit | + New | Learn more

Table: stg/Customers | Refresh | Preview data

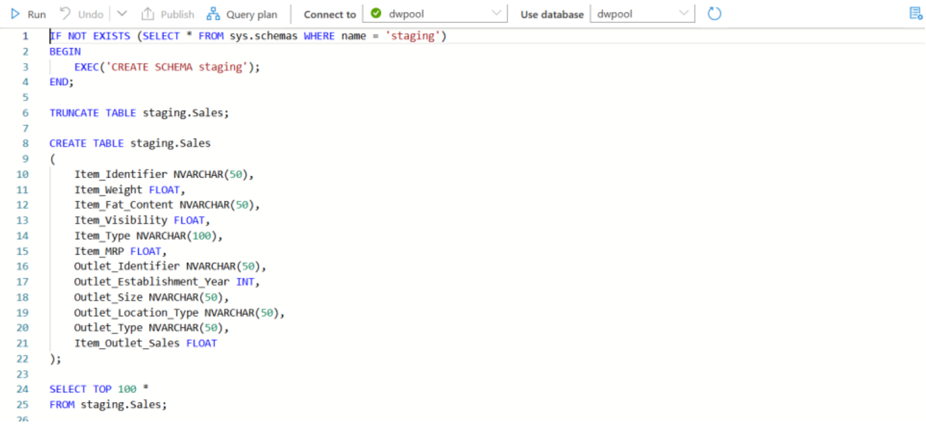
□ Enter manually

Step 4: Creating SQL table in Synapse Studio

- Connecting to the SQL Database



- Creating staging tables in Synapse Studio

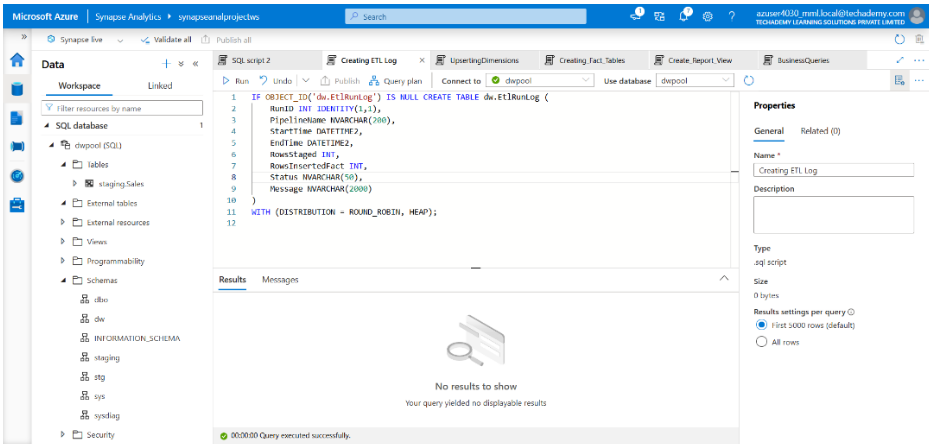


Connect to ● dwpool

Use database dwpool

```sql
IF NOT EXISTS (SELECT * FROM sys.schemas WHERE name = 'staging')
BEGIN
    EXEC('CREATE SCHEMA staging');
END;

TRUNCATE TABLE staging.Sales;

CREATE TABLE staging.Sales
(
    Item_Identifier NVARCHAR(50),
    Item_Weight FLOAT,
    Item_Fat_Content NVARCHAR(50),
    Item_Visibility FLOAT,
    Item_Type NVARCHAR(100),
    Item_MRP FLOAT,
    Outlet_Identifier NVARCHAR(50),
    Outlet_Establishment_Year INT,
    Outlet_Size NVARCHAR(50),
    Outlet_Location_Type NVARCHAR(50),
    Outlet_Type NVARCHAR(50),
    Item_Outlet_Sales FLOAT
);

SELECT TOP 100 *
FROM staging.Sales;
```

- Creating ETL Logs in Synapse Studio



Connect to ● dwpool

Use database dwpool

```sql
IF OBJECT_ID('dw.EtlRunLog') IS NULL CREATE TABLE dw.EtlRunLog (
    RunID INT IDENTITY(1,1),
    PipelineName NVARCHAR(200),
    StartTime DATETIME2,
    EndTime DATETIME2,
    RowsStaged INT,
    RowsInsertedFact INT,
    Status NVARCHAR(50),
    Message NVARCHAR(2000)
)
WITH (DISTRIBUTION = ROUND_ROBIN, HEAP);
```

Properties

General Related (0)

Name * Creating ETL Log

Description

Type .sql script

Size 0 bytes

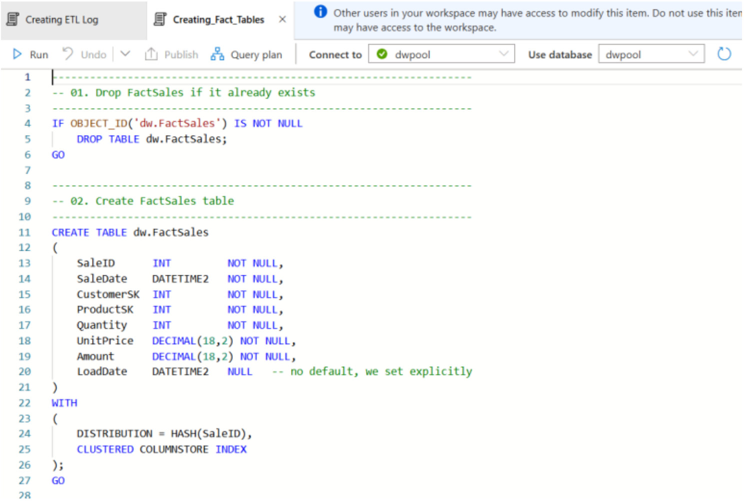
Results settings per query

● First 5000 rows (default)

○ All rows

00:00:00 Query executed successfully.

- Creating Fact table for Sales



Other users in your workspace may have access to modify this item. Do not use this iten may have access to the workspace.

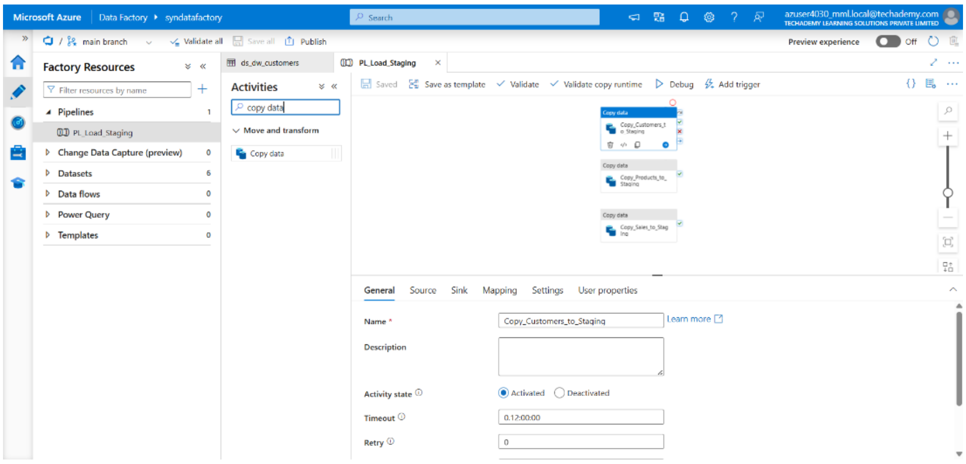
Connect to: dwpool

Use database: dwpool

```sql
----------------------------------------------------------------------
-- 01. Drop FactSales if it already exists
----------------------------------------------------------------------
IF OBJECT_ID('dw.FactSales') IS NOT NULL
    DROP TABLE dw.FactSales;
GO

----------------------------------------------------------------------
-- 02. Create FactSales table
----------------------------------------------------------------------
CREATE TABLE dw.FactSales
(
    SaleID      INT          NOT NULL,
    SaleDate    DATETIME2    NOT NULL,
    CustomerSK  INT          NOT NULL,
    ProductSK   INT          NOT NULL,
    Quantity    INT          NOT NULL,
    UnitPrice   DECIMAL(18,2) NOT NULL,
    Amount      DECIMAL(18,2) NOT NULL,
    LoadDate    DATETIME2    NULL    -- no default, we set explicitly
)
WITH
(
    DISTRIBUTION = HASH(SaleID),
    CLUSTERED COLUMNSTORE INDEX
);
GO
```

Step 5: Creation of Pipeline

- Creating a pipeline to copy data from raw files into staging tables in Azure Data Factory (ADF)



copy data

Name * Copy_Customers_to_Staging

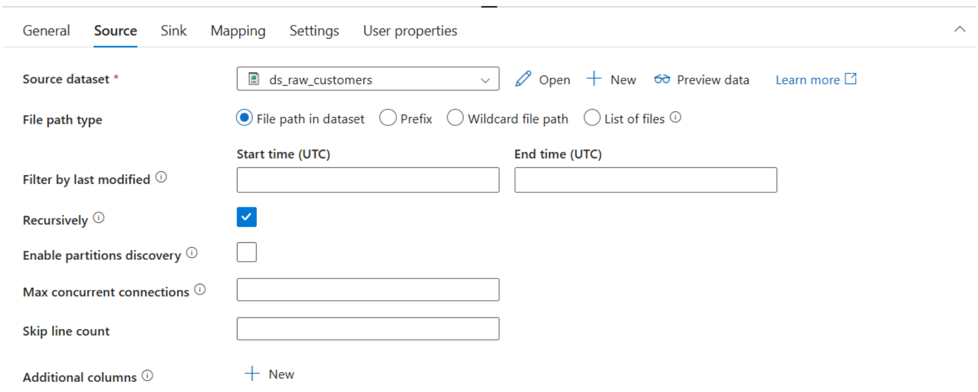
Description

Activity state ● Activated ○ Deactivated

Timeout 0.12:00:00

Retry 0

- Setting the source for the pipeline



Source dataset * ds_raw_customers

File path type ● File path in dataset ○ Prefix ○ Wildcard file path ○ List of files

Filter by last modified: Start time (UTC) End time (UTC)

Recursively ☑

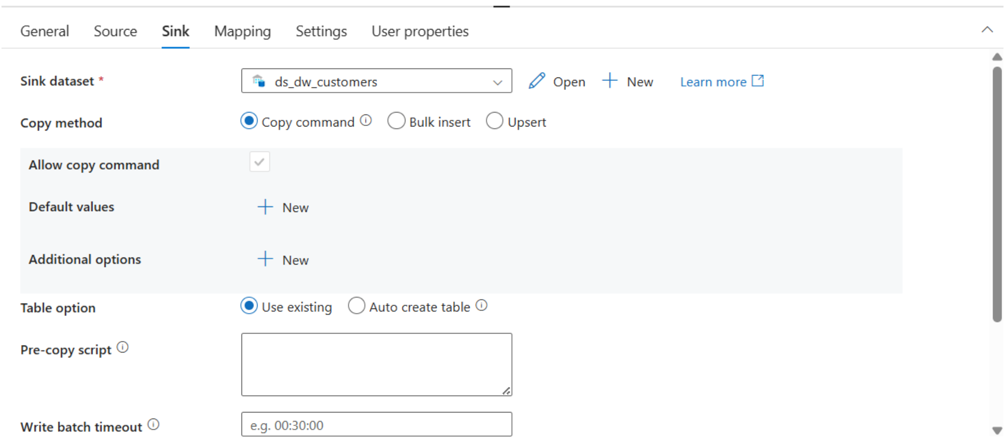
Enable partitions discovery □

Max concurrent connections

Skip line count

Additional columns + New

- Setting the sink for the pipeline



Sink dataset * ds_dw_customers

Copy method ● Copy command ○ Bulk insert ○ Upsert

Allow copy command ☑

Default values + New

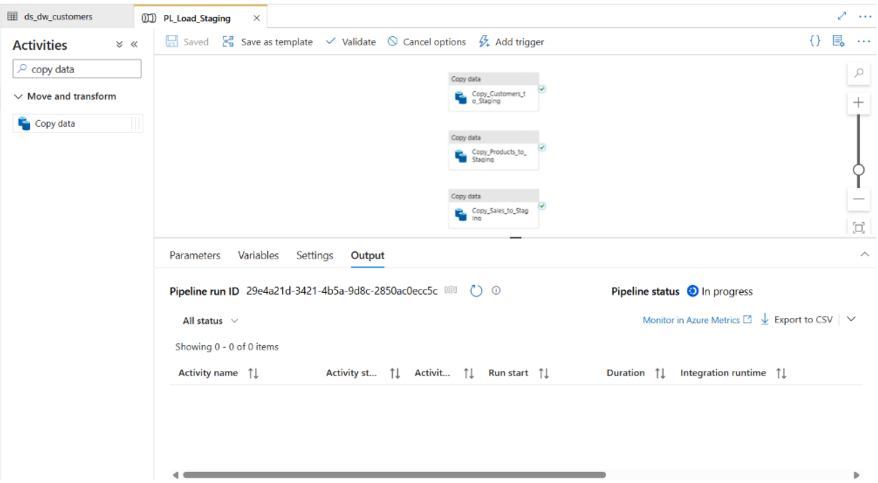
Additional options + New

Table option ● Use existing ○ Auto create table

Pre-copy script

Write batch timeout e.g. 00:30:00

- Executing the Pipeline



Pipeline run ID 29e4a21d-3421-4b5a-9d8c-2850ac0ecc5c

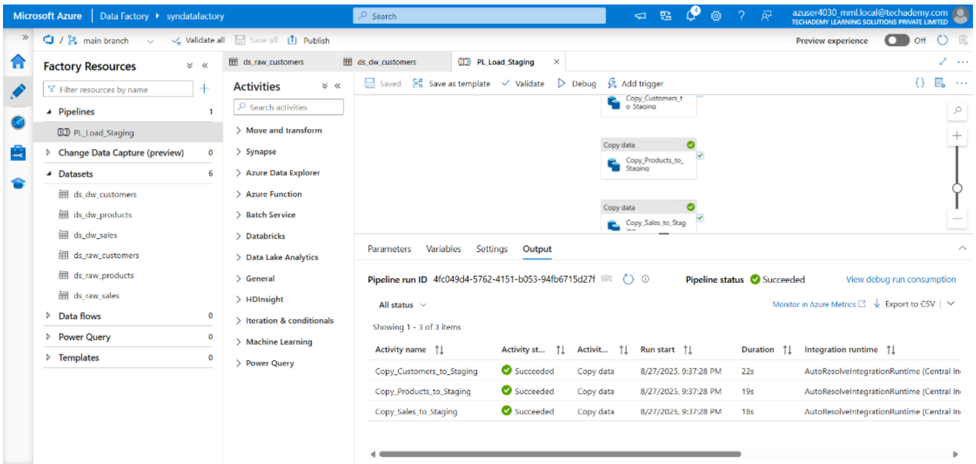
Pipeline status In progress

All status

Showing 0 - 0 of 0 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime

- Successful execution of the pipeline



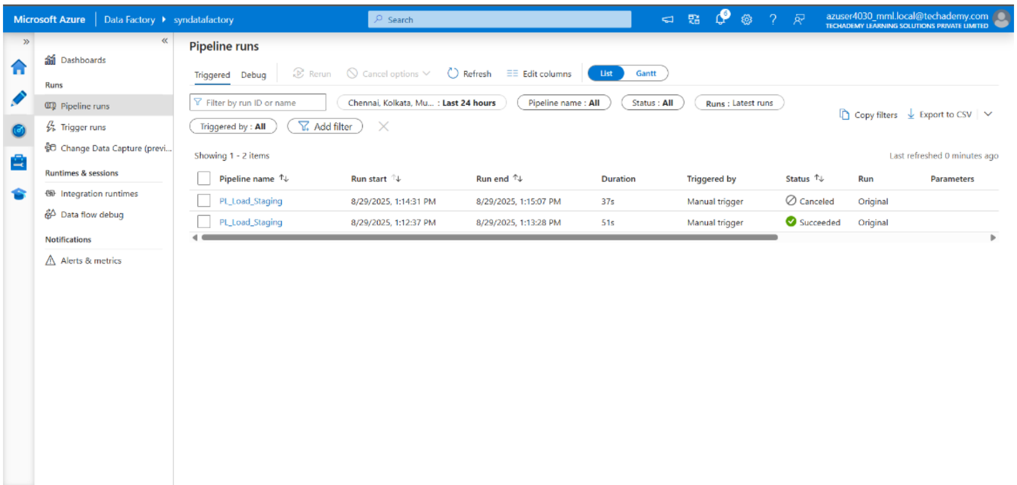
Pipeline run ID 4fc049d4-5762-4151-b053-94fb6715d27f

Pipeline status Succeeded

Showing 1 - 3 of 3 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
Copy_Customers_to_Staging	Succeeded	Copy data	8/27/2025, 9:37:28 PM	22s	AutoResolveIntegrationRuntime (Central In
Copy_Products_to_Staging	Succeeded	Copy data	8/27/2025, 9:37:28 PM	19s	AutoResolveIntegrationRuntime (Central In
Copy_Sales_to_Staging	Succeeded	Copy data	8/27/2025, 9:37:28 PM	18s	AutoResolveIntegrationRuntime (Central In

- Monitoring the Pipeline



Pipeline runs

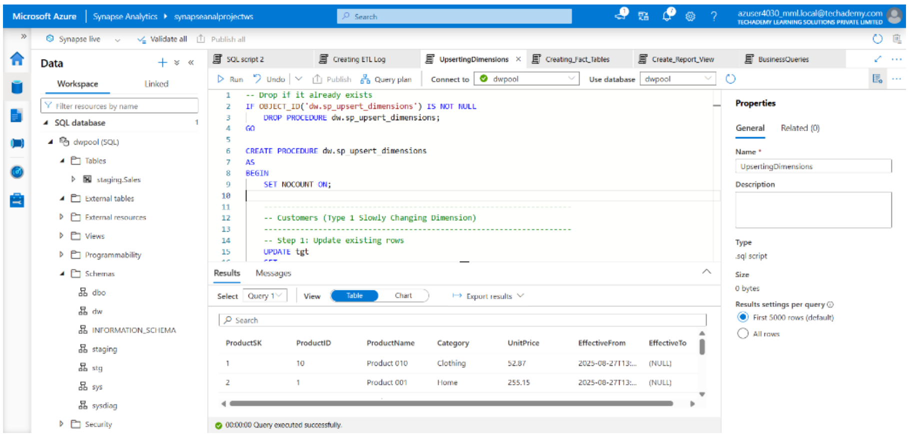
Chennai, Kolkata, Mu... : Last 24 hours | Pipeline name : All | Status : All | Runs : Latest runs | Triggered by : All

Showing 1 - 2 items

Pipeline name	Run start	Run end	Duration	Triggered by	Status	Run	Parameters
PL_Load_Staging	8/29/2025, 1:14:31 PM	8/29/2025, 1:15:07 PM	37s	Manual trigger	Canceled	Original	
PL_Load_Staging	8/29/2025, 1:12:37 PM	8/29/2025, 1:13:28 PM	51s	Manual trigger	Succeeded	Original	

Step 6: Executing SQL Queries

- Upsert Dimensions



```
-- Drop if it already exists
IF OBJECT_ID('dw.sp_upsert_dimensions') IS NOT NULL
    DROP PROCEDURE dw.sp_upsert_dimensions;
GO

CREATE PROCEDURE dw.sp_upsert_dimensions
AS
BEGIN
    SET NOCOUNT ON;

    -------------------------------------------------------------------
    -- Customers (Type 1 Slowly Changing Dimension)
    -------------------------------------------------------------------
    -- Step 1: Update existing rows
    UPDATE tgt
```

ProductSK	ProductID	ProductName	Category	UnitPrice	EffectiveFrom	EffectiveTo
1	10	Product 010	Clothing	52.87	2025-08-27T13:...	(NULL)
2	1	Product 001	Home	255.15	2025-08-27T13:...	(NULL)

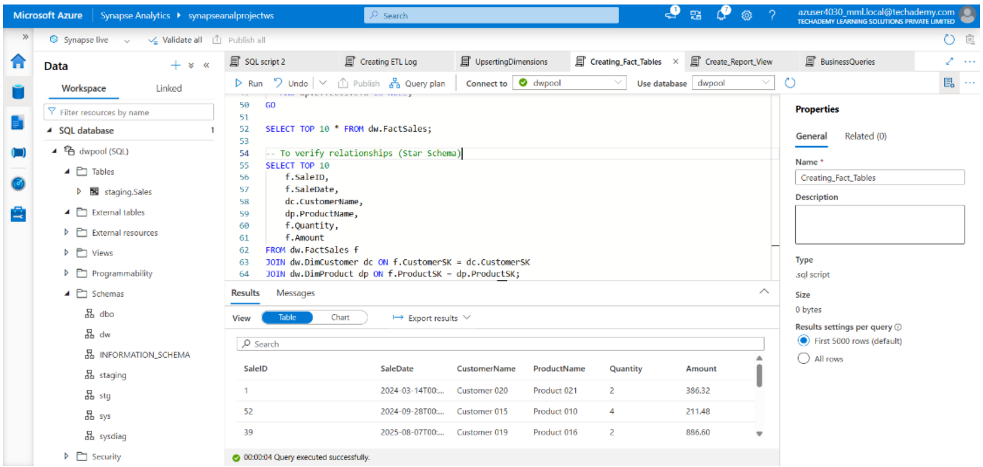
00:00:00 Query executed successfully.

Name: UpsertingDimensions

Type: .sql script

Size: 0 bytes

- Querying the Facts table

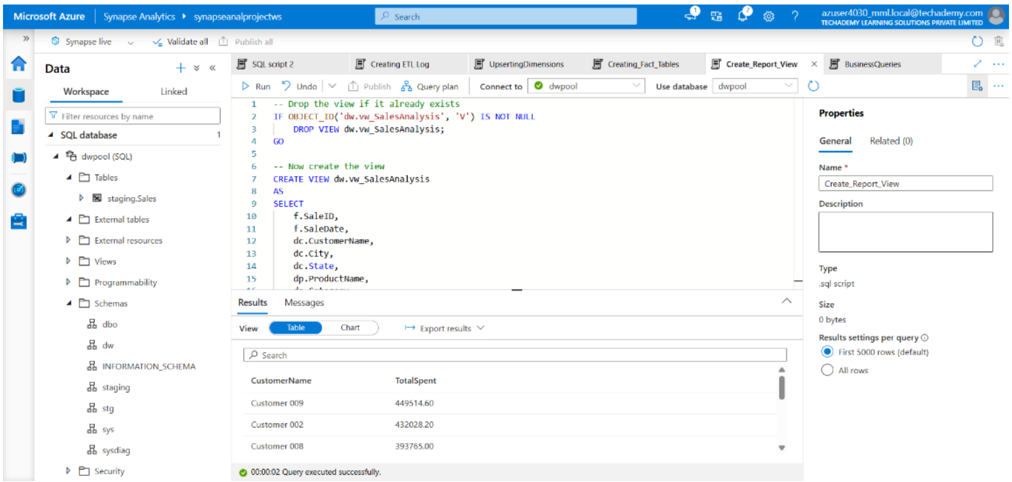


```
GO

SELECT TOP 10 * FROM dw.FactSales;

-- To verify relationships (Star Schema)
SELECT TOP 10
    f.SaleID,
    f.SaleDate,
    dc.CustomerName,
    dp.ProductName,
    f.Quantity,
    f.Amount
FROM dw.FactSales f
JOIN dw.DimCustomer dc ON f.CustomerSK = dc.CustomerSK
JOIN dw.DimProduct dp ON f.ProductSK = dp.ProductSK;
```

SaleID	SaleDate	CustomerName	ProductName	Quantity	Amount
1	2024-03-14T00:...	Customer 020	Product 021	2	386.32
52	2024-09-28T00:...	Customer 015	Product 010	4	211.48
39	2025-08-07T00:...	Customer 019	Product 016	2	886.60

00:00:04 Query executed successfully.

- Creating Report Views

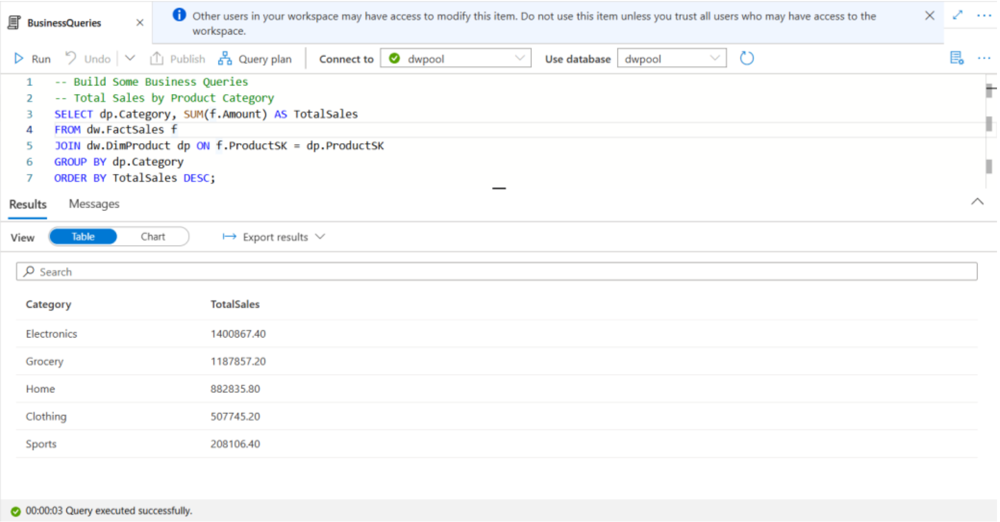


```
-- Drop the view if it already exists
IF OBJECT_ID('dw.vw_SalesAnalysis', 'V') IS NOT NULL
    DROP VIEW dw.vw_SalesAnalysis;
GO

-- Now create the view
CREATE VIEW dw.vw_SalesAnalysis
AS
SELECT
    f.SaleID,
    f.SaleDate,
    dc.CustomerName,
    dc.City,
    dc.State,
    dp.ProductName,
```

CustomerName	TotalSpent
Customer 009	449514.60
Customer 002	432028.20
Customer 008	393765.00

00:00:02 Query executed successfully.

- Business Queries



Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

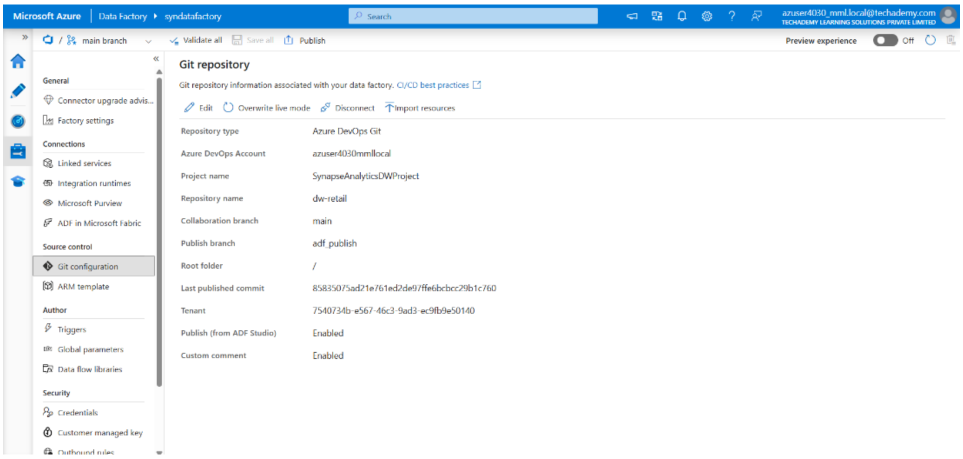
```
-- Build Some Business Queries
-- Total Sales by Product Category
SELECT dp.Category, SUM(f.Amount) AS TotalSales
FROM dw.FactSales f
JOIN dw.DimProduct dp ON f.ProductSK = dp.ProductSK
GROUP BY dp.Category
ORDER BY TotalSales DESC;
```

Category	TotalSales
Electronics	1400867.40
Grocery	1187857.20
Home	882835.80
Clothing	507745.20
Sports	208106.40

00:00:03 Query executed successfully.

Step 7: Implementing Source Control and CI/CD with Azure DevOps

- Git Integration with Azure Data Factory (ADF)

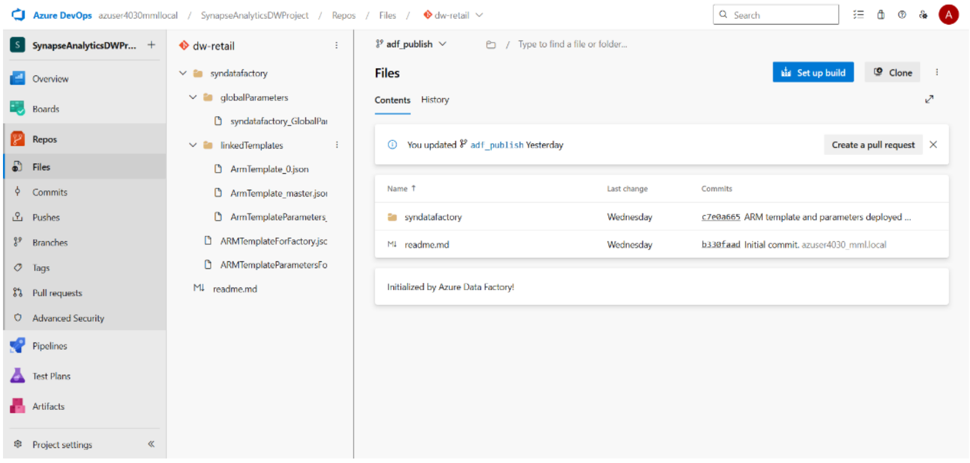


Git repository

Git repository information associated with your data factory. CI/CD best practices

Repository type	Azure DevOps Git
Azure DevOps Account	azuser4030mmllocal
Project name	SynapseAnalyticsDWProject
Repository name	dw-retail
Collaboration branch	main
Publish branch	adf_publish
Root folder	/
Last published commit	85835075ad21e761ed2de97ffe6bcbcc29b1c760
Tenant	7540734b-e567-46c3-9ad3-ec9fb9e50140
Publish (from ADF Studio)	Enabled
Custom comment	Enabled

- Storing SQL Scripts and ARM Templates



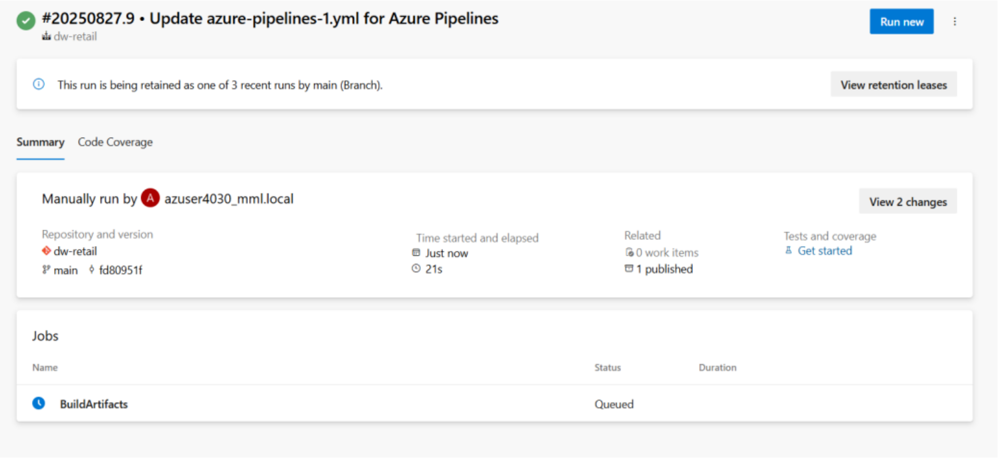
Files

You updated adf_publish Yesterday

Name ↑	Last change	Commits
syndatafactory	Wednesday	c7e0a665 ARM template and parameters deployed ...
readme.md	Wednesday	b330faad Initial commit. azuser4030_mml.local

Initialized by Azure Data Factory!

- CI/CD Pipeline setup



#20250827.9 • Update azure-pipelines-1.yml for Azure Pipelines
dw-retail

This run is being retained as one of 3 recent runs by main (Branch).

Manually run by A azuser4030_mml.local

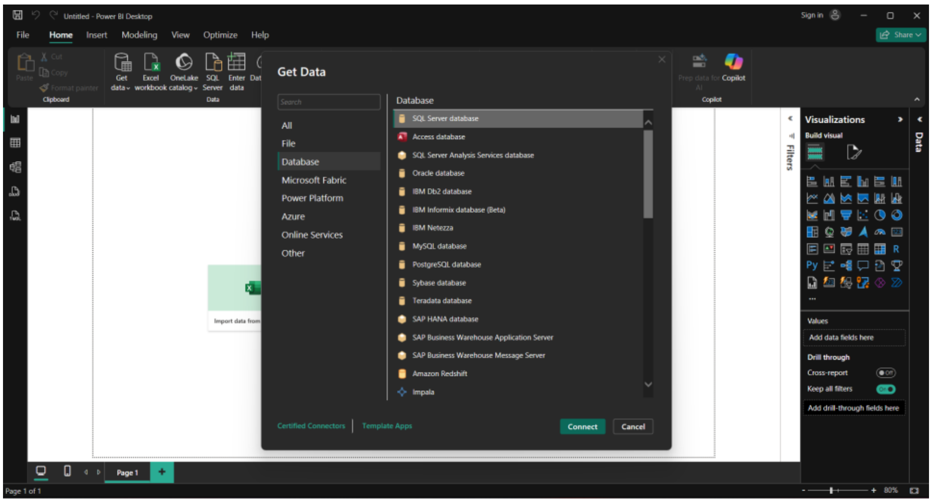
Repository and version	Time started and elapsed	Related	Tests and coverage
dw-retail main fd80951f	Just now 21s	0 work items 1 published	Get started

Jobs

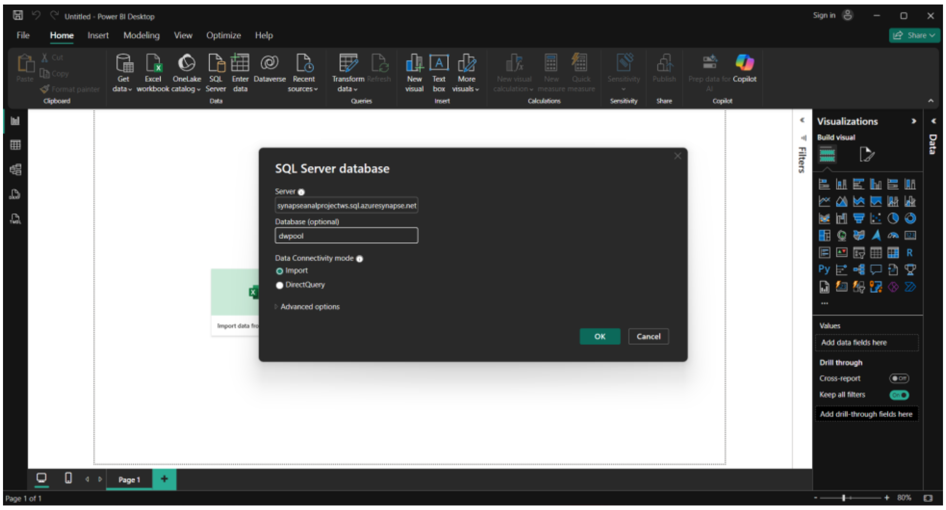
Name	Status	Duration
BuildArtifacts	Queued	

Step 8: Verify Data and Visualize

- Open Power BI > Get Data from SQL Server Database



- Connect to the SQL Server and Database and select the respective tables



SQL Server database

Server: synapseanalprojectws.sql.azuresynapse.net

Database (optional): dwpool

Data Connectivity mode

● Import

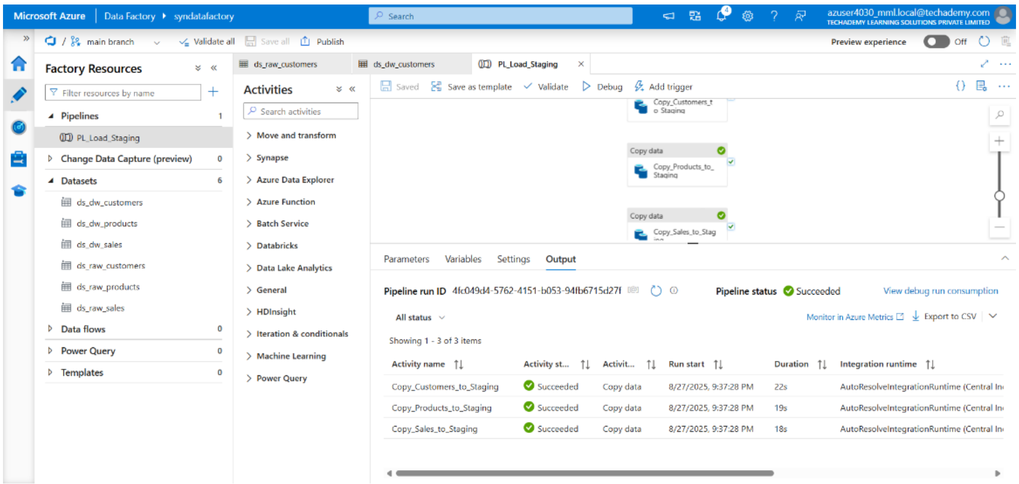
○ DirectQuery

- Perform some analysis and create a report

Successful Output Generated

ADF Pipeline Execution

- Pipeline executed successfully, extracting data from Azure Storage (CSV files).
- Data was transformed and loaded into Azure Synapse Analytics tables.
- ADF run history shows all activities completed without failure.

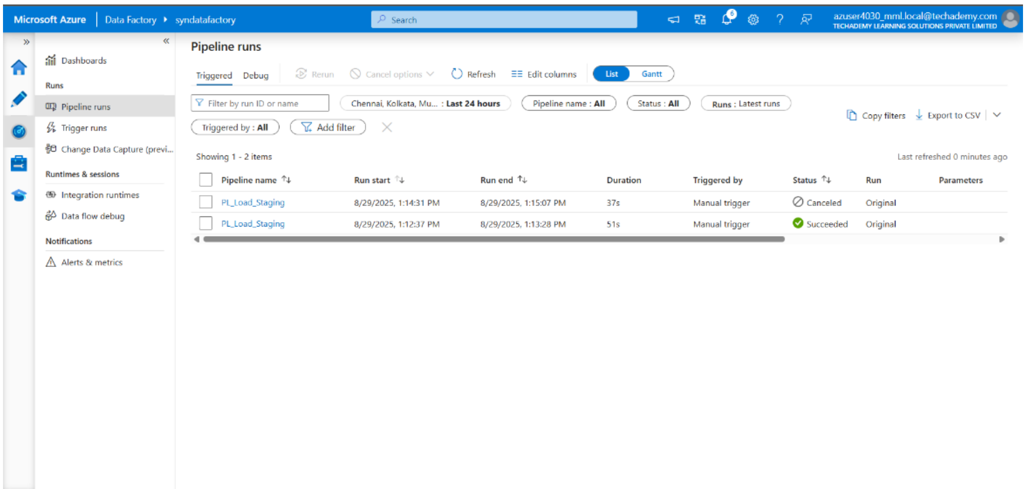


Pipeline run ID 4fc049d4-5762-4151-b053-94fb6715d27f

Pipeline status ✓ Succeeded

Showing 1 - 3 of 3 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
Copy_Customers_to_Staging	✓ Succeeded	Copy data	8/27/2025, 9:37:28 PM	22s	AutoResolveIntegrationRuntime (Central In
Copy_Products_to_Staging	✓ Succeeded	Copy data	8/27/2025, 9:37:28 PM	19s	AutoResolveIntegrationRuntime (Central In
Copy_Sales_to_Staging	✓ Succeeded	Copy data	8/27/2025, 9:37:28 PM	18s	AutoResolveIntegrationRuntime (Central In



Pipeline runs

Chennai, Kolkata, Mu... : Last 24 hours | Pipeline name : All | Status : All | Runs : Latest runs | Triggered by : All

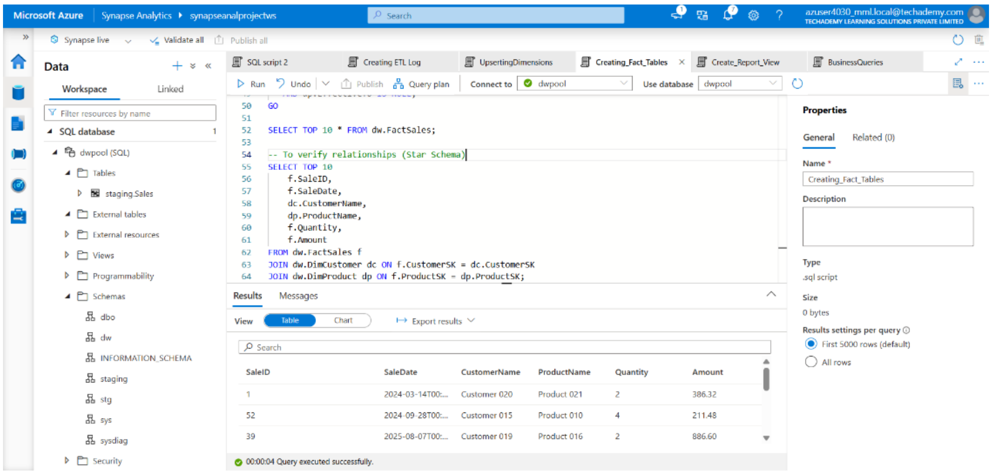
Showing 1 - 2 items

Last refreshed 0 minutes ago

□	Pipeline name	Run start	Run end	Duration	Triggered by	Status	Run	Parameters
□	PL_Load_Staging	8/29/2025, 1:14:31 PM	8/29/2025, 1:15:07 PM	37s	Manual trigger	⊘ Canceled	Original	
□	PL_Load_Staging	8/29/2025, 1:12:37 PM	8/29/2025, 1:13:28 PM	51s	Manual trigger	✓ Succeeded	Original	

Query Validation in Azure Synapse Analytics

- SQL queries executed successfully on FactSales, DimCustomer, and DimProduct.
- Verified data correctness and relationships between dimension and fact tables.
- Confirmed the data model supports analytical use cases.



Connect to: dwpool | Use database: dwpool

```sql
GO

SELECT TOP 10 * FROM dw.FactSales;

-- To verify relationships (Star Schema)
SELECT TOP 10
    f.SaleID,
    f.SaleDate,
    dc.CustomerName,
    dp.ProductName,
    f.Quantity,
    f.Amount
FROM dw.FactSales f
JOIN dw.DimCustomer dc ON f.CustomerSK = dc.CustomerSK
JOIN dw.DimProduct dp ON f.ProductSK = dp.ProductSK;
```

SaleID	SaleDate	CustomerName	ProductName	Quantity	Amount
1	2024-03-14T00:...	Customer 020	Product 021	2	386.32
52	2024-09-28T00:...	Customer 015	Product 010	4	211.48
39	2025-08-07T00:...	Customer 019	Product 016	2	886.60

00:00:04 Query executed successfully.

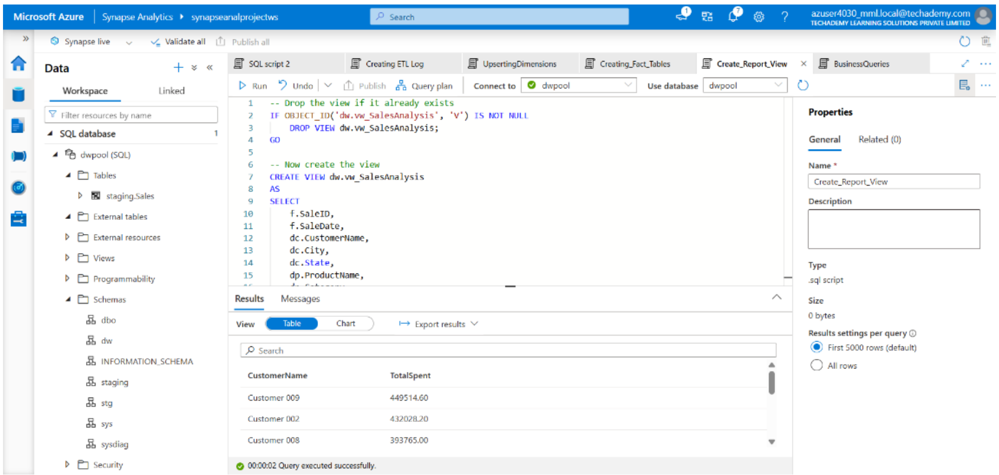
Name *: Creating_Fact_Tables

Description

Type: .sql script

Size: 0 bytes

Results settings per query: ● First 5000 rows (default) ○ All rows



Connect to: dwpool | Use database: dwpool

```sql
-- Drop the view if it already exists
IF OBJECT_ID('dw.vw_SalesAnalysis', 'V') IS NOT NULL
    DROP VIEW dw.vw_SalesAnalysis;
GO

-- Now create the view
CREATE VIEW dw.vw_SalesAnalysis
AS
SELECT
    f.SaleID,
    f.SaleDate,
    dc.CustomerName,
    dc.City,
    dc.State,
    dp.ProductName,
```

CustomerName	TotalSpent
Customer 009	449514.60
Customer 002	432028.20
Customer 008	393765.00

00:00:02 Query executed successfully.

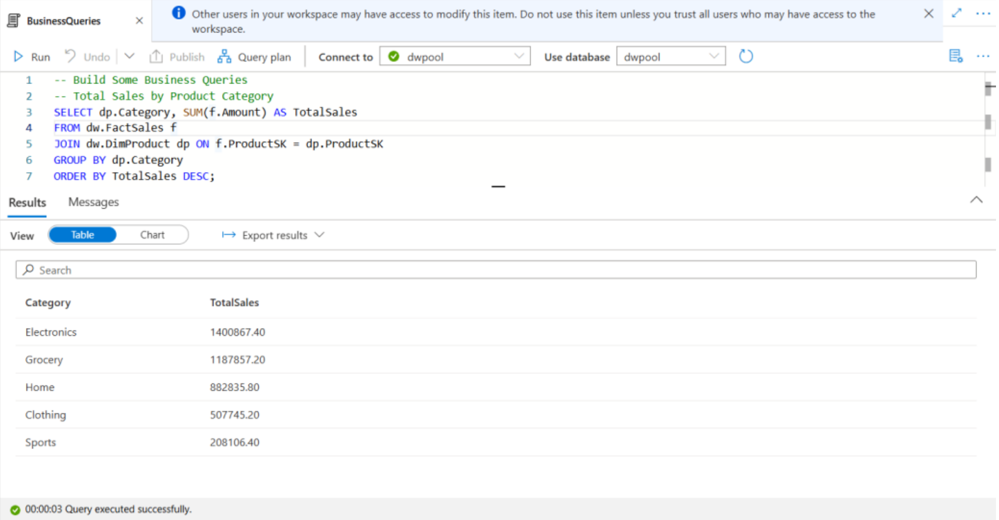
Name *: Create_Report_View

Description

Type: .sql script

Size: 0 bytes

Results settings per query: ● First 5000 rows (default) ○ All rows



Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

Connect to: dwpool | Use database: dwpool

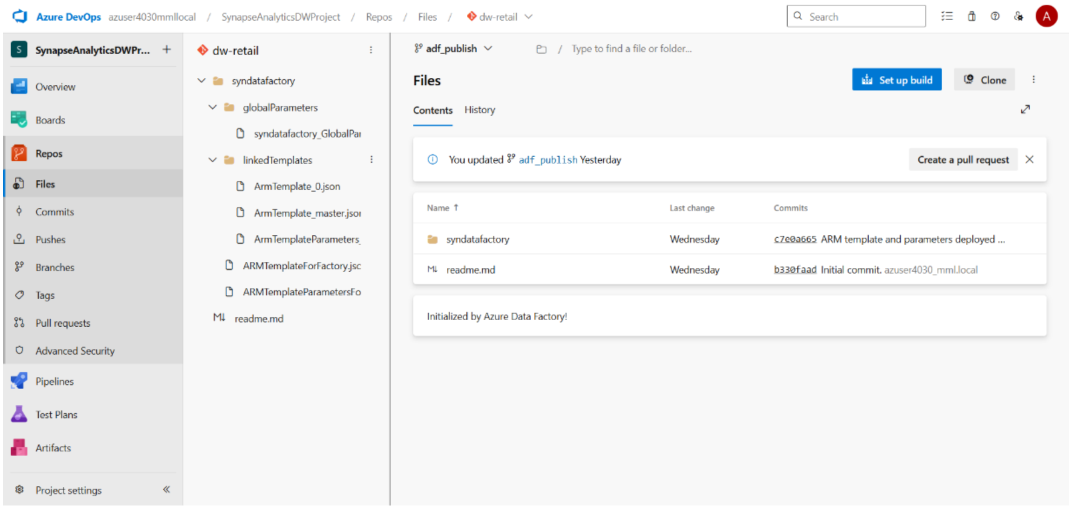
```sql
-- Build Some Business Queries
-- Total Sales by Product Category
SELECT dp.Category, SUM(f.Amount) AS TotalSales
FROM dw.FactSales f
JOIN dw.DimProduct dp ON f.ProductSK = dp.ProductSK
GROUP BY dp.Category
ORDER BY TotalSales DESC;
```

Category	TotalSales
Electronics	1400867.40
Grocery	1187857.20
Home	882835.80
Clothing	507745.20
Sports	208106.40

00:00:03 Query executed successfully.

Azure DevOps Pipeline Execution

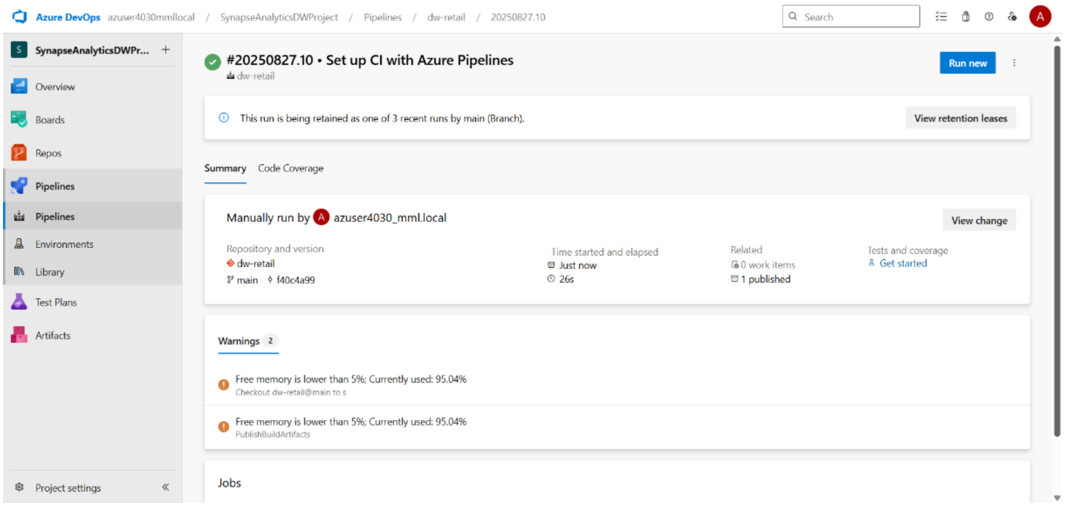
- CI/CD pipeline deployed ADF ARM templates and Synapse SQL scripts from the repo.
- Build stage executed successfully with no errors



You updated adf_publish Yesterday

Name ↑	Last change	Commits
syndatafactory	Wednesday	c7e0a665 ARM template and parameters deployed ...
readme.md	Wednesday	b330faad Initial commit. azuser4030_mml.local

Initialized by Azure Data Factory!



#20250827.10 • Set up CI with Azure Pipelines
dw-retail

This run is being retained as one of 3 recent runs by main (Branch).

Manually run by azuser4030_mml.local

Repository and version: dw-retail, main, f40c4a99
Time started and elapsed: Just now, 26s
Related: 0 work items, 1 published
Tests and coverage: Get started

Warnings 2

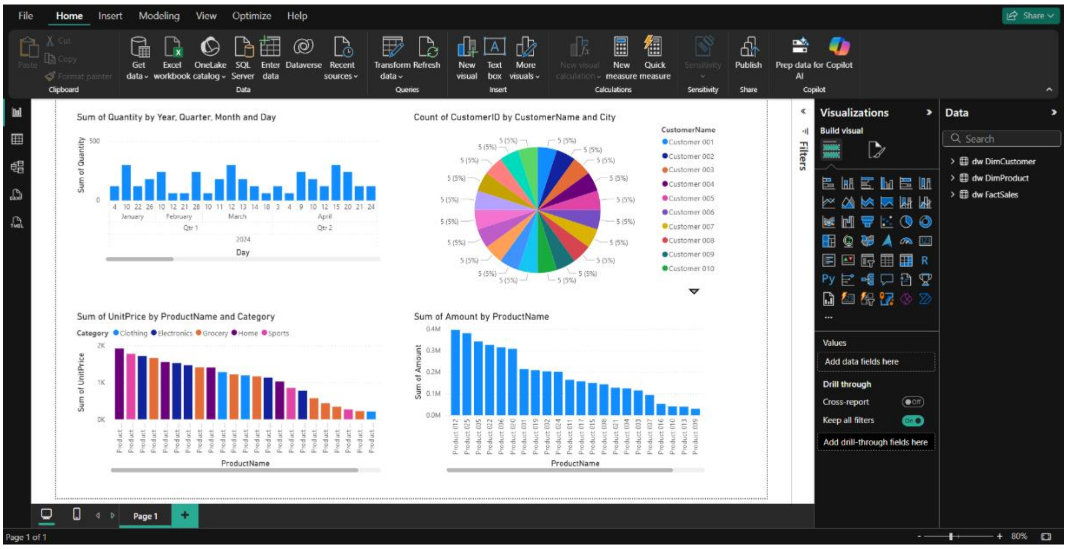
Free memory is lower than 5%; Currently used: 95.04%
Checkout dw-retail@main to s

Free memory is lower than 5%; Currently used: 95.04%
PublishBuildArtifacts

Power BI Visualization

- Power BI connected to Synapse and consumed validated data
- Reports built showing "Sum of Quantity by Year, Quarter, Month and Day", "Count of CustomerID by CustomerName and City", "Sum of Amount by ProductName", "Sum of UnitPrice by ProductName and Category"

- Dashboards confirmed end-to-end functionality of the project.



Strategies for Optimizing the Process

1. **Perform Data Cleaning and Standardization** – Handle nulls, duplicates, inconsistent formats, and ensure schema alignment during ingestion.
2. **Adopt Incremental Data Loads** – Instead of full refreshes, ingest only new or changed data in ADF to save time and resources.
3. **Leverage Staging Layers** – Use Azure Storage as a staging area before transformations in Synapse for smoother workflows.
4. **Design Efficient Data Models** – Implement star schema with Fact and Dimension tables for optimal query and reporting performance.
5. **Use Partitioning and Indexing** – Partition large fact tables and add indexes in Synapse to accelerate query execution.
6. **Automate Deployments with CI/CD** – Utilize Azure DevOps pipelines for automated deployments of ADF, Synapse, and Power BI assets.
7. **Optimize Power BI Models** – Import only necessary fields, use measures instead of calculated columns, and apply row-level security.
8. **Monitor and Scale Resources** – Use Azure Monitor to track performance, and scale Synapse or ADF integration runtimes dynamically based on workload.

9. **Apply Cost Management Practices** – Move cold data to cheaper storage tiers and pause Synapse compute when not in use.
10. **Maintain Governance and Documentation** – Enforce naming conventions, document pipelines, and review processes periodically for continuous optimization.

Conclusion

This project successfully implemented a comprehensive and scalable sales data processing pipeline using Azure Data Factory (ADF), Azure Synapse Analytics, Azure Storage, and Power BI, converting raw CSV files into structured fact and dimension tables to support advanced reporting and analytics.

Successful Implementation of the Data Pipeline

- Azure Data Factory orchestrated the end-to-end workflow, from ingesting raw CSV files (customers, products, sales) from Azure Storage to loading them into Synapse Analytics.
- The pipeline supported both scheduled and on-demand executions, enabling flexible and reliable data movement.

Efficient Data Cleaning, Transformation, and Modeling

- Raw CSV data was validated and cleaned to remove duplicates, handle nulls, and enforce consistency.
- Transformations such as joins, aggregations, and schema alignment were applied before loading data into Synapse.
- A star schema was designed with FactSales and dimension tables (DimCustomer, DimProduct), enabling optimized querying and analysis.

Exploration, Optimization, and Querying in Synapse Analytics

- Synapse enabled execution of analytical SQL queries across large datasets with improved performance.

- Optimization techniques such as partitioning, indexing, and efficient table design were applied.
- The structured model supported quick retrieval of insights like total sales by product, region, or customer segment.

Integration and Reporting

- Azure DevOps pipelines ensured CI/CD, with version-controlled ADF ARM templates, Synapse SQL scripts, and deployment automation.
- Power BI dashboards were developed, connecting directly to Synapse tables, to provide interactive reports and visualizations.
- Business insights such as customer purchasing patterns, top-selling products, and revenue trends were delivered through intuitive visualizations.

As a whole, this project demonstrates a scalable, end-to-end data engineering and analytics solution for sales data, leveraging the synergy between ADF, Synapse, Azure Storage, DevOps, and Power BI. The pipeline ensures reliable ingestion, efficient transformation, governed storage, automated deployments, and business intelligence dashboards enabling data-driven decision-making with accuracy and efficiency.