**Datawarehouse Final Project**

**Stage 1: Data Ingestion**

Dataset used for this project:

<https://www.kaggle.com/datasets/tforsyth/4-year-historical-sales-data/data>

Initially, the dataset is uploaded to GitHub:

A screenshot of a computer

Description automatically generated

Created a resource group to centralize all resources, allowing for quick access and efficient administration. This resource group help me easily arrange and manage multiple resources, improving monitoring and access control for enhanced efficiency:

A screenshot of a computer

Description automatically generated

Initially Created an Azure storage account to store the data imported from GitHub. It provides scalable, durable, and highly accessible storage solutions, assuring the dependability and accessibility of stored data. This also enables easy integration of data pipelines and rapid data processing and analysis inside the Azure environment:

A screenshot of a computer

Description automatically generated

Created Bronze, Silver, Gold Containers Inside Storage account:

* Bronze – To store raw data to keep its original form and provide flexibility for future analysis.
* Silver – To assure data quality through validation and basic transformations and preparing it for downstream usage.
* Gold – To store finalized, optimized data, ready for use by end users, with improved performance for efficient analysis and decision-making.

A screenshot of a computer

Description automatically generated

Setting up a data factory, which is a fully managed data integration service that orchestrates and automates data transfer and transformation:

A screenshot of a computer

Description automatically generated

Created a pipeline in Azure Data Factory to seamlessly import data from GitHub and store it in the data lake created in the previous step. This pipeline orchestrates data transfer, making data ingestion operations more automated and efficient.

A screenshot of a computer

Description automatically generated

Files created in the bronze container after running the pipeline created in previous step:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Stage 2: Data Ingestion**

Created an Azure Databricks service to help with the transformation process, ensuring that raw data in the bronze layer is cleaned and ready for further analysis. Using Azure Databricks for transformation because it offers faster data transformation operations:

A screenshot of a computer

Description automatically generated

Created the Cluster in data bricks to utilize computational resources to run the notebooks:

A screenshot of a computer

Description automatically generated

Mounting Databricks with all the containers we created in Azure's data lake to enable seamless data access and integration:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Created a notebook ‘Bronze to Silver’ to perform data transformations on the data in 'bronze' container and then store the transformed data in the ‘silver' container:

A screenshot of a computer

Description automatically generated

Result after running Bronze to Silver notebook:

A screenshot of a computer

Description automatically generated

Created another notebook ‘Silver to Gold’ to create facts and dimensions to design data warehouse and then store the transformed datasets in the ‘Gold' container:

A screenshot of a computer

Description automatically generated

Result after running Silver to Gold notebook:

A screenshot of a computer

Description automatically generated

Integrating data bricks Notebooks into the pipeline we previously developed in Data Factory allows us to conduct sophisticated data transformations smoothly within the current pipeline, streamlining data processing.

A screenshot of a computer

Description automatically generated

**Stage 3: Data Loading**

Finally, I created a serverless SQL database in Azure Synapse Analytics to store views generated from final tables saved in gold containers. The serverless design of Azure Synapse Analytics enables efficient querying and analysis of stored data when we are using the data for reporting purposes.

A screenshot of a computer

Description automatically generated

Created a storage procedure to create the views:

A screenshot of a computer

Description automatically generated

Created new pipeline in Synapse analytics to create view:

A screenshot of a computer

Description automatically generated

Result after loading data in SQL database created on Synapse Analytics:

A computer screen shot of a computer screen

Description automatically generated

.

**Stage 4: Data Reporting Using Power BI**

Imported data from azure synapse analytics SQL database on to PowerBI to create Reports & Dashboards. Below are the final dashboards.

Insights:

A screenshot of a computer

Description automatically generated

Forecast:

A screenshot of a computer

Description automatically generated