**Data Pipeline for Customer Account Analysis**

**Bootcamp Project - 1**

**Objective**

The goal of this project was to design and implement a robust data pipeline for processing customer account data. This involved:

* Copying data from the backend team's storage account.
* Performing necessary transformations using Databricks.
* Upserting (inserting or updating) data from a file stored in Azure Data Lake Storage (ADLS) GOLD Storage into a dedicated SQL pool table in Azure Synapse Analytics.
* Ensuring efficient, accurate, and scalable data processing to support downstream analytics and reporting needs.

**Project Implementation Steps**

**Step 1: Data Ingestion (Backend Storage to Raw (Bronze) Container)**

**Actions Taken:**

* Configured an **Azure Data Factory (ADF) copy activity** to transfer datasets from the backend storage to the **bronze container**.
* Source datasets:
  + accounts.csv
  + customers.csv
  + loan\_payments.csv
  + loans.csv
  + transactions.csv
* Sink: Defined the **bronze container** in **ADLS**, ensuring all files were copied successfully.
* Verified the data in the **bronze storage container** via Azure Storage Explorer and Databricks.

**Challenges Encountered & Solutions:**

* Ensured **Managed Identity permissions** were set correctly in IAM for ADF to access the storage.
* Used the **Storage Blob Data Contributor** role for smooth data movement.

**Step 2: Databricks Activity (Incremental/Delta Processing in Silver Layer)**

**Actions Taken:**

* Created a **Databricks notebook** for data processing:
  + Read raw data from the **bronze container**.
  + Cleaned and filtered invalid data (e.g., missing values, incorrect formats).
  + Converted datasets to **Parquet/Delta format** for efficient querying.
  + Saved transformed data in the **silver container**.
* Validated the Silver storage container in **Databricks**.

**Verification:**

* Used dbutils.fs.ls("mnt/silver/") in Databricks to check files.
* Successfully read and displayed data using **PySpark DataFrames**.

**Challenges Encountered & Solutions:**

* **Silver container was empty initially:** Fixed by debugging transformation logic and ensuring correct paths were used.
* **Schema Mismatch Issues:** Enforced explicit schema definitions in PySpark.

**Step 3: Databricks Activity (ETL Processing for Gold Layer)**

**Actions Taken:**

* Implemented **business logic** to transform silver layer data.
* Calculated **total balance across all accounts** for each customer.
* Stored processed data in the **gold container** in **Parquet/Delta format**.
* Verified file presence using dbutils.fs.ls("mnt/gold/").

**Verification:**

* Successfully loaded and validated transformed data.
* Ensured correctness by querying customer balances in Databricks.

**Challenges Encountered & Solutions:**

* **Mounting issues with ADLS Gen2:** Fixed by reconfiguring Databricks access and using the correct storage account keys.

**Step 4: Azure Synapse Analytics (Final Storage & Upserts)**

**Actions Taken:**

* Configured an **external table** in Azure Synapse Analytics to map data from the gold storage.
* Used **OPENROWSET** to verify data accessibility.
* Set up an **upsert mechanism** to merge updates dynamically into the SQL pool table.
* Executed queries to ensure seamless integration and querying from Synapse.

**Verification:**

* Queried data in Synapse Studio (SELECT TOP 10 \* FROM customer\_balance;).
* Confirmed data integrity and correctness.

**Challenges Encountered & Solutions:**

* **Access denied error in Synapse:** Fixed by ensuring the **Synapse workspace identity had correct IAM roles**.
* **Missing files in ADLS Gold container:** Resolved by rerunning the Databricks pipeline.

**Final Deliverables**

* **Fully functional data pipeline** integrating ADF, Databricks, ADLS, and Synapse.
* **Correctly processed data** stored in **Bronze, Silver, and Gold layers**.
* **Upsert logic implemented** for real-time data updates in Synapse.
* **Documented all steps with screenshots**.
* **Code Snippets:**
  + PySpark scripts for ETL.
  + SQL queries for validation.
  + ADF JSON configurations.

**Conclusion**

This project successfully implemented an end-to-end **Azure Data Engineering pipeline** for **Customer Account Analysis**. The pipeline efficiently handled **data ingestion, transformation, storage, and analytics**, enabling **scalable and real-time insights**. All challenges were resolved, and the pipeline is now **ready for production deployment**.