Bootcamp Project 1 - Data Pipeline for Customer Account Analysis

Objective:

The project aims to design and implement a robust data pipeline for processing customer account data. This includes copying data from a backend team's storage account, performing necessary transformations using ADF and upserting (inserting or updating) data from a file stored in Azure Data Lake Storage ADLS GOLD Storage into sql database table iThe pipeline aims to ensure efficient, accurate, and scalable data processing to support downstream analytics and reporting needs.

Project Steps

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

1. Copy Activity: Configure an Azure Data Factory (ADF) copy activity to transfer data from the backend team's storage account to your designated raw(bronze) container in the data lake.
2. Source: Specify the backend team's storage account details and the location of the data to be copied.
    a. accounts.csv
    b. customers.csv
    c. loan_payments.csv
    d. loans.csv
    e. transactions.csv
3. Sink: Define your data lake storage account and the raw(bronze) container where the copied data will be placed. Reference: This step is similar to copying a dataset from Kaggle: https://www.kaggle.com/datasets/varunkumari/ai-bank-dataset

Step 2: Use ADF Dataflows to remove the duplicates

1. Use ADF Dataflows (use aggregate transformation/windows/assert)
    a. Read Data: Read data from five different sources within the raw(bronze) container.
    b. Data Cleaning: Implement logic to identify and remove any hanging or irrelevant data from the sources.
    c. Data Transformation: Apply necessary transformations to prepare the data for further processing. This might involve schema changes(avoid inferSchema), data type conversions, or handling missing values (parquet or delta)

Step 3: Dataflows using SCD Type technique (SCD 1 and SCD 2)

1. Use Dataflows in pipeline
    a. Place data into SQL DB
    b. Schedule the Pipelines

i. Local to Bronze Layer

ii. Bronze to Silver Layer

iii. Silver to Gold Layer

Step 4: Use Power BI for Data Visualization

- Use tables as a source to PowerBI Reports and build visuals on top of it
- Publish the developed report into Fabric Workspace

Repeat the Steps using the automated triggers.

Additional Considerations

- Dynamic Parameters: Incorporate dynamic parameters within your data pipelines to enable configuration changes without modifying the pipeline code itself. This enhances flexibility and simplifies pipeline management.
- Key Vault: Utilize Azure Key Vault to securely store sensitive information used in your pipelines, such as connection strings or credentials.

Deliverables

1. Documentation: Create a well-structured document outlining the steps involved in the data pipeline, including screenshots and detailed explanations. [Include Screenshots]
2. Code Snippets: Share relevant code snippets from your Databricks notebooks (SQL and/or PySpark) and pipeline activity configurations (JSON) in a dedicated repository (e.g., GitHub).
3. Follow the Naming convention while saving your project documentations

"Your Name_Project no._Project Name"

Example file name: Sruthi_BC001_Dataset

and share in your individual group

1. Draw an architecture diagram (Use Draw.io), and use connections between them
2. Create a Master Pipeline using Execute Pipeline Task
    a. Call child pipelines related to 3 layers in sequence