Lab 11 – Pipeline notebook

1. Provision an Azure Synapse Analytics workspace

A screenshot of a computer

AI-generated content may be incorrect.

1. Preview of the source files

A screenshot of a computer

AI-generated content may be incorrect.

1. Import and run the given Python notebook

A screenshot of a computer

AI-generated content may be incorrect.

1. Browse the parquet files created

A screenshot of a computer

AI-generated content may be incorrect.

1. Query the parquet files to verify the transformed data

A screenshot of a computer

AI-generated content may be incorrect.

1. Parameterize the notebook for automation

A screenshot of a computer

AI-generated content may be incorrect.

1. Create and configure pipeline to use notebook

A screenshot of a computer

AI-generated content may be incorrect.

1. Trigger and monitor the pipeline

A screenshot of a computer

AI-generated content may be incorrect.

1. Verify the result of the pipeline

A screenshot of a computer

AI-generated content may be incorrect.

1. Delete Azure Resources

A screenshot of a computer

AI-generated content may be incorrect.

Summary:

Running multiple data flows in parallel can lead to separate Spark clusters for each activity, resulting in unused warm pools. Sequential data flows reuse compute resources, speeding up cluster startup times. Splitting data flows into components simplifies monitoring and debugging. Our source files are CSVs, each containing a year of data, with three files representing three years. We import a Spark transform Python notebook from GitHub into Synapse, attach it to the Spark cluster, and run it. We read the CSV files from the data folder and set a variable for a unique folder name. The initial code block takes time to load due to Spark cluster initialization. After loading the CSV data into a dataframe, we split CustomerName into FirstName and LastName fields and drop CustomerName. We load the transformed dataframe into Parquet format back into the data lake. This Spark pipeline extracts, transforms, and loads data. To automate, we parameterize the folderName variable, create a pipeline, configure the Spark pool and notebook, and trigger the pipeline. Running the pipeline results in new Parquet files in the specified folder.