Data Transformation using Databricks:

Introduction:

You are using Azure Databricks for data transformation.

The tutorial focuses on creating compute clusters and mounting Azure Data Lake storage for data access.

Getting Started:

Access Databricks from the Azure portal.

Launch the Databricks workspace, which will be used for data transformation logic.

Databricks Workspace Components:

Workspace: Create notebooks for transformation logic.

Repo: Used for Git integration.

Recent: Lists recently accessed notebooks.

Data: Create databases and tables.

Compute: Create Spark clusters for data transformation.

Workflow: Used for creating jobs but won't be used in this tutorial.

Creating a Compute Cluster:

Go to the "Compute" tab.

Configure cluster details, including name, policy, node type, and more.

Enable "Terminate after 15 minutes of inactivity" to save costs.

Enable "Credential pass-through" for user-level data access.

Creating a Notebook:

Use the "Workspace" tab to create a new notebook.

Name it "storage Mount" and choose Python as the default language.

Select the compute cluster you created earlier.

Mounting Data Lake Storage:

Use Python code to mount Azure Data Lake storage.

Update the container name, storage account name, and mount point in the code.

Run the code to mount the storage.

Accessing Data in the Container:

Use DBUtils.fs.ls to list files in the mounted container.

You can now access data in the container using Databricks.

Mounting Silver and Gold Containers:

Repeat the mounting process for the silver and gold containers.

You now have Mount points for all three containers.

Level 1 Transformation: Bronze to Silver

In the first level of transformation, the goal is to extract data from the bronze zone (raw data) and transform it into the silver zone, where it is cleaned and structured. This transformation is relatively minimal because the data from the SQL Server database is already in a structured format. The key transformation task in this stage is converting date-time columns to date format.

Transformation Requirements

Convert date-time columns to date format.

Apply this transformation to all tables in the bronze zone where date columns are found.

Implementation

Data Retrieval: Connect to the bronze container in the data lake and list the available tables.

Iterative Transformation: Iterate through each table in the bronze container.

Load the data into a PySpark DataFrame.

Identify date columns and transform them from date-time format to date format.

Generate an output path pointing to the silver container.

Write the transformed data in Delta format to the silver container.

Delta Format: Store the transformed data in the Delta format for better versioning and schema handling.

Level 2 Transformation: Silver to Gold

In the second level of transformation, the data from the silver zone is further refined and prepared for reporting purposes. In this project, the primary focus is on standardizing column names according to a specific naming convention.

Transformation Requirements

Standardize column names to follow a naming convention (e.g., Pascal case with underscores).

Apply this transformation to all tables in the silver zone.

Implementation

Data Retrieval: Connect to the silver container in the data lake and list the available tables.

Iterative Transformation: Iterate through each table in the silver container.

Load the transformed data into a PySpark DataFrame.

Standardize column names according to the naming convention.

Generate an output path pointing to the gold container.

Write the transformed data in Delta format to the gold container.

Databricks Workspace and Notebooks

The entire data transformation process is implemented using Databricks notebooks.

The notebooks are organized into two levels of transformation: bronze to silver and silver to gold.

PySpark is used extensively for data manipulation and transformation.

Magic commands are used to switch between programming languages, such as SQL, Scala, and R, within the same notebook.

Creating Databricks Notebooks

In your Databricks workspace, create two notebooks: "Bronze to Silver" and "Silver to Gold."

These notebooks will be used to perform data transformations.

Data Transformation

In your notebooks, write Spark code to perform data transformations. The transformations can include data cleansing, aggregation, and enrichment.

Ensure that the data is transformed from the "Bronze" layer to the "Silver" layer and then to the "Gold" layer.

Use Delta Lake to store intermediate and final results, as it provides version tracking and reliability.

Scheduling and Monitoring

Schedule notebook runs using Databricks jobs.

Monitor notebook runs in real-time within the Databricks workspace.

Implement error handling and logging for robust data transformations.

Accessing Sensitive Values

Store sensitive values like access tokens securely in Azure Key Vault.

Access these values from Databricks notebooks using Key Vault integration for enhanced security.