**Case Study 2: Retail Sales Data Processing (using Azure DataBricks)**

**Client:** XYZ Retail Inc.

**Objective:** To enhance the data processing pipeline by incorporating Azure Databricks for advanced data transformations and using Azure Data Factory for orchestration.

**Background:** XYZ Retail Inc. wants to process retail sales data by performing complex transformations with Azure Databricks . The goal is to load raw data from Blob Storage, transform it using Databricks, and store the processed data back in Blob Storage.

**Requirements:**

1. **Data Storage:**
   * **Source Container:** source-data in Azure Blob Storage for raw CSV files.
   * **Target Container:** transformed-data in Azure Blob Storage for data processed by Azure Databricks.
2. **Data Processing:**
   * **Transformation with Databricks:** Use PySpark and SQL in Azure Databricks to perform complex data transformations and aggregations.
   * **Data Movement:** Move data between Blob Storage and Databricks and store the processed data back in Blob Storage.

**High-Level Steps:**

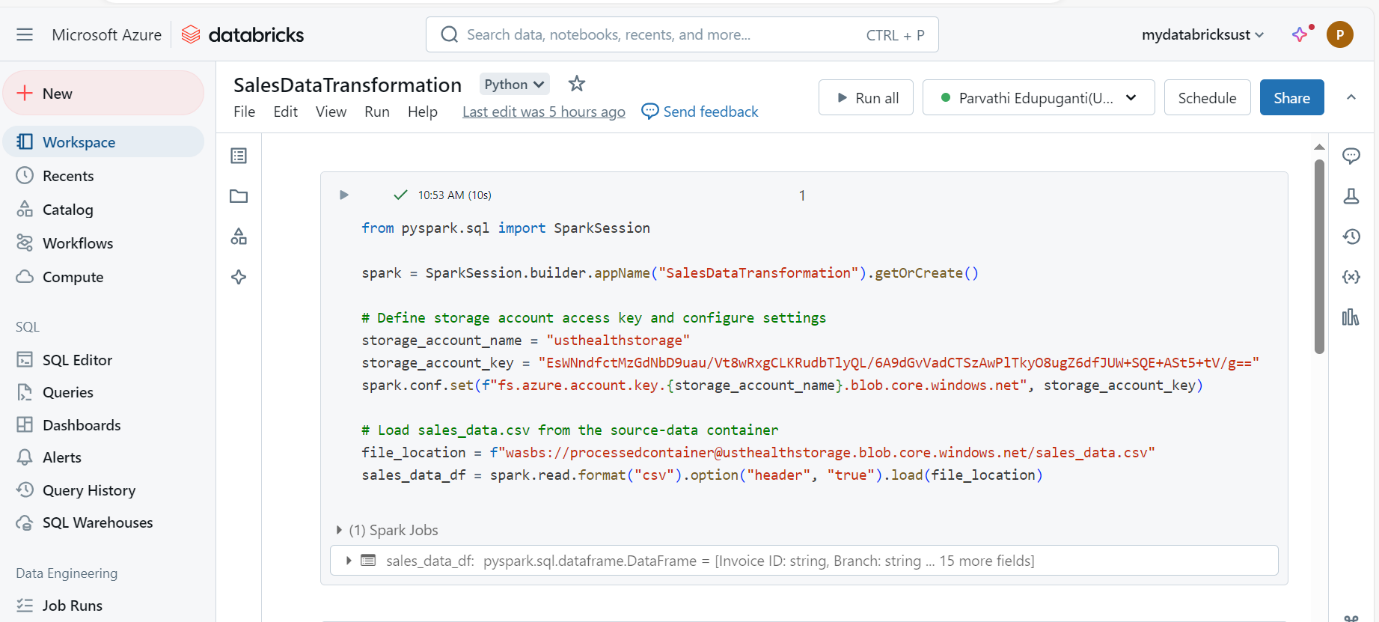
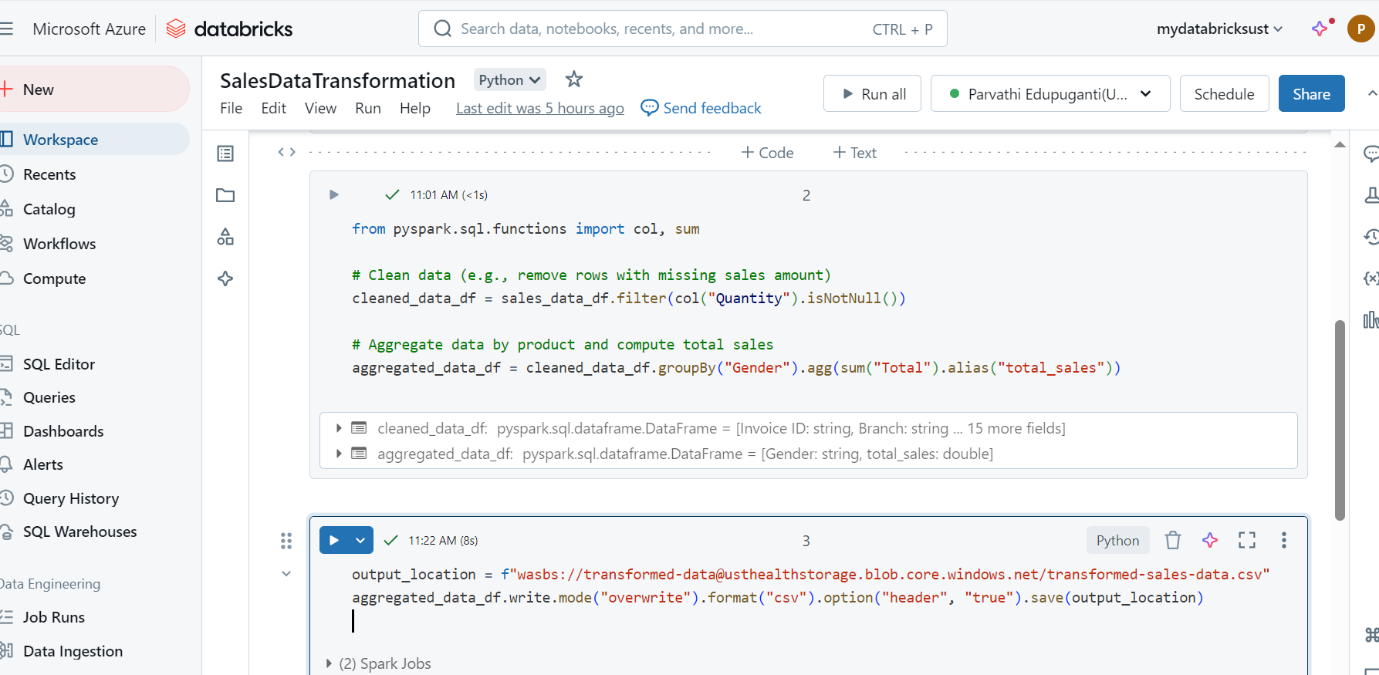
**1. Set Up Azure Blob Storage**

1. **Create Storage Accounts:**
   * **Storage Account Name:** xyzretailstorage
   * **Resource Group:** RetailDataGroup
   * **Region:** Choose an appropriate region (e.g., East US)
2. **Create Containers:**
   * **Source Container:** source-data
   * **Target Container:** transformed-data
3. **Upload Sample Data:**
   * Upload sales\_data.csv to the source-data container.

**2. Set Up Azure Databricks**

1. **Create an Azure Databricks Workspace:**
   * **Workspace Name:** RetailDatabricks
   * **Resource Group:** RetailDataGroup
   * **Region:** Same as storage account (e.g., East US)
2. **Create a Cluster:**
   * Set up a cluster in Azure Databricks with appropriate configurations for processing the data.
3. **Create a Databricks Notebook:**
   * **Notebook Name:** SalesDataTransformation
   * Use PySpark or SQL to read the CSV file from Azure Blob Storage, perform transformations (e.g., cleaning, aggregating), and write the transformed data back to the transformed-data container.

Case study2:

1. Azure databricks:
2. 
3. 
4. o/p:

