**Batch ETL Pipeline using PySpark on Azure Databricks**

# 1. Introduction

This project demonstrates the design and implementation of a batch ETL pipeline using PySpark in Azure Databricks. The source dataset is NYC Taxi trip data, which is processed from raw CSV format into cleaned, structured, and aggregated formats. The final output is stored in Parquet format in Azure Blob Storage.

# 2. Objective

To build a batch ETL pipeline in Azure Databricks using PySpark which:  
- Reads raw CSV data from Azure Blob Storage  
- Performs data cleaning, filtering, and type casting  
- Applies aggregation   
- Stores Final result in Parquet format back into Azure Blob

# 3. Tools & Technologies

- Azure Databricks  
- PySpark  
- Azure Blob Storage  
- Parquet File Format  
- DBML for ER Diagram

# 4. Dataset Description

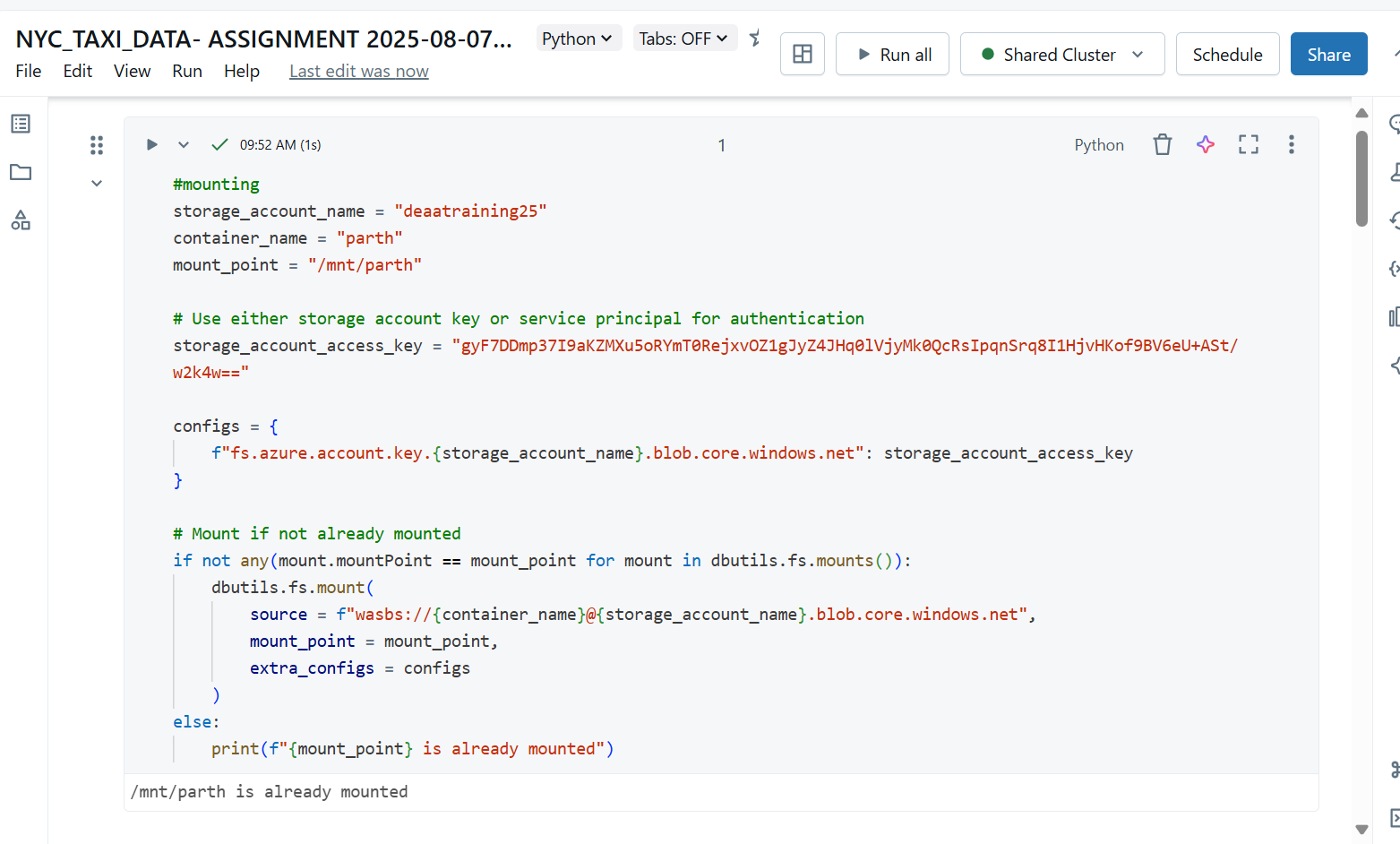
The dataset used is the NYC Taxi trip dataset. The original CSV includes fields such as:  
VendorID, pickup and dropoff timestamps, passenger count, trip distance, location IDs, payment details, fare amount, tip, taxes, surcharges, and total amount.

**Data Set Source: https://www.kaggle.com/datasets/anandaramg/taxi-trip-data-nyc**

# 5. ETL Pipeline Steps

Step-by-step implementation of the ETL pipeline:

1. Mount or access Azure Blob Storage to read the uploaded CSV file.



1. Read the raw CSV data using PySpark's DataFrame API with `inferSchema` and `header` options.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Use `. with Column()` to type cast each column to the appropriate datatype and filter invalid rows.

A screenshot of a computer program

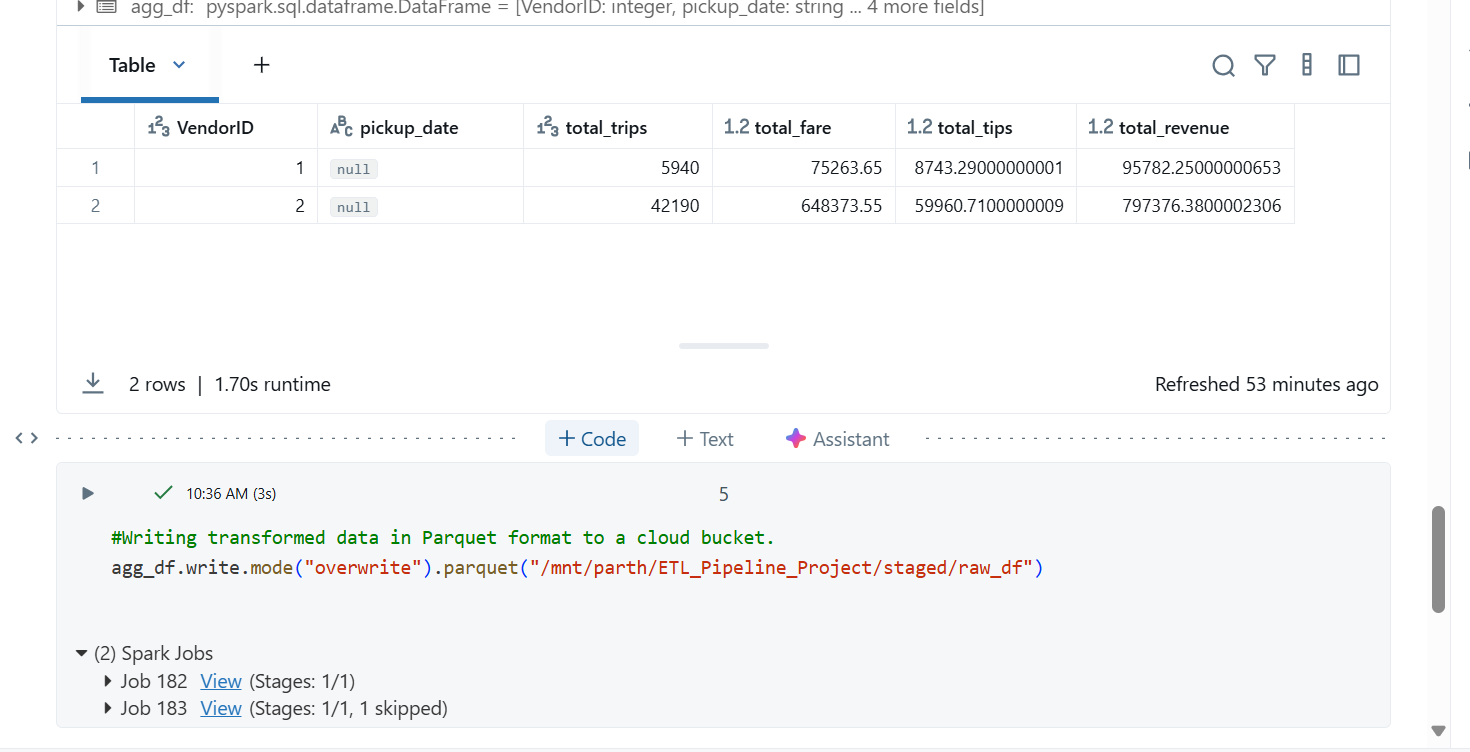
AI-generated content may be incorrect.

1. Aggregate the cleaned data to compute total rides per vendor per day using `group By ()` and `count ()`.

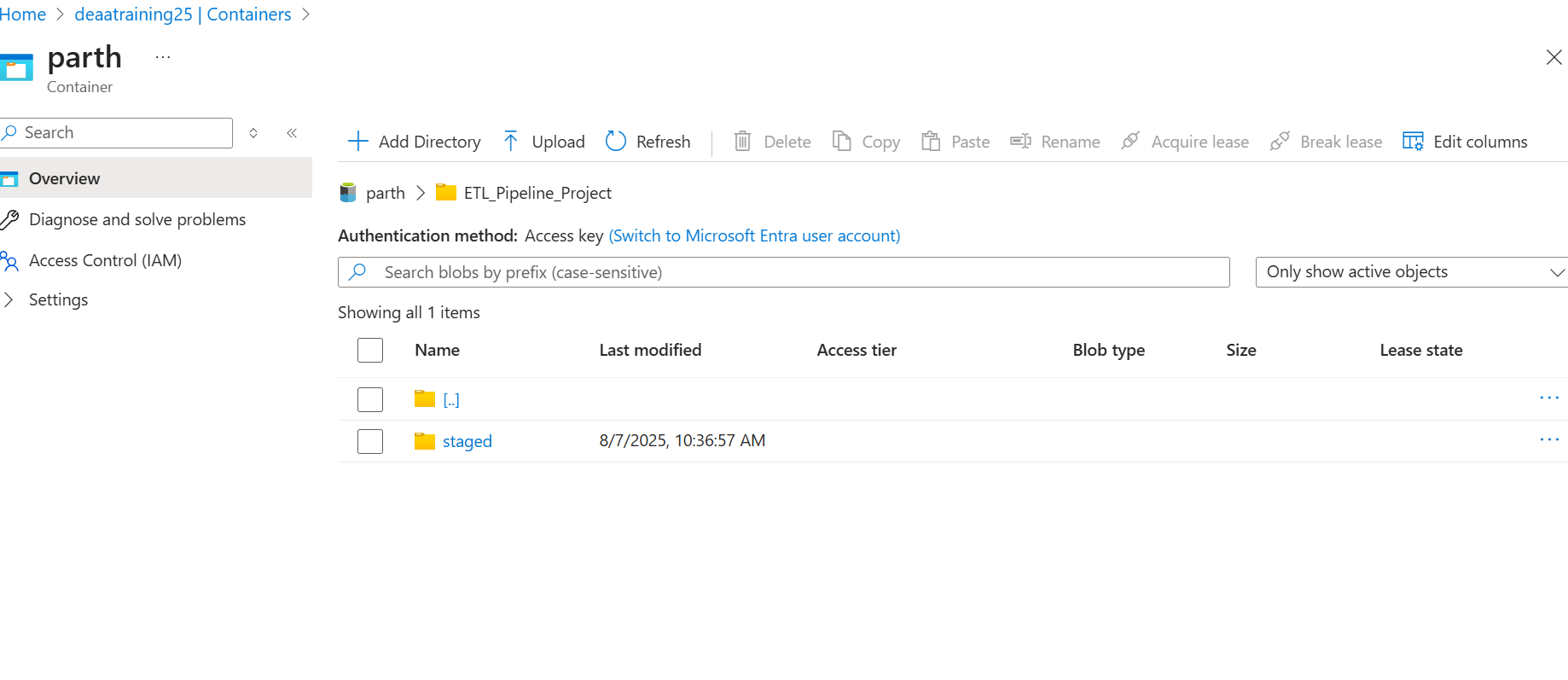
A screenshot of a computer code

AI-generated content may be incorrect.

1. Store the final aggregated output in Parquet format into Azure Blob Storage.



1. Checking in Azure Cloud Storage

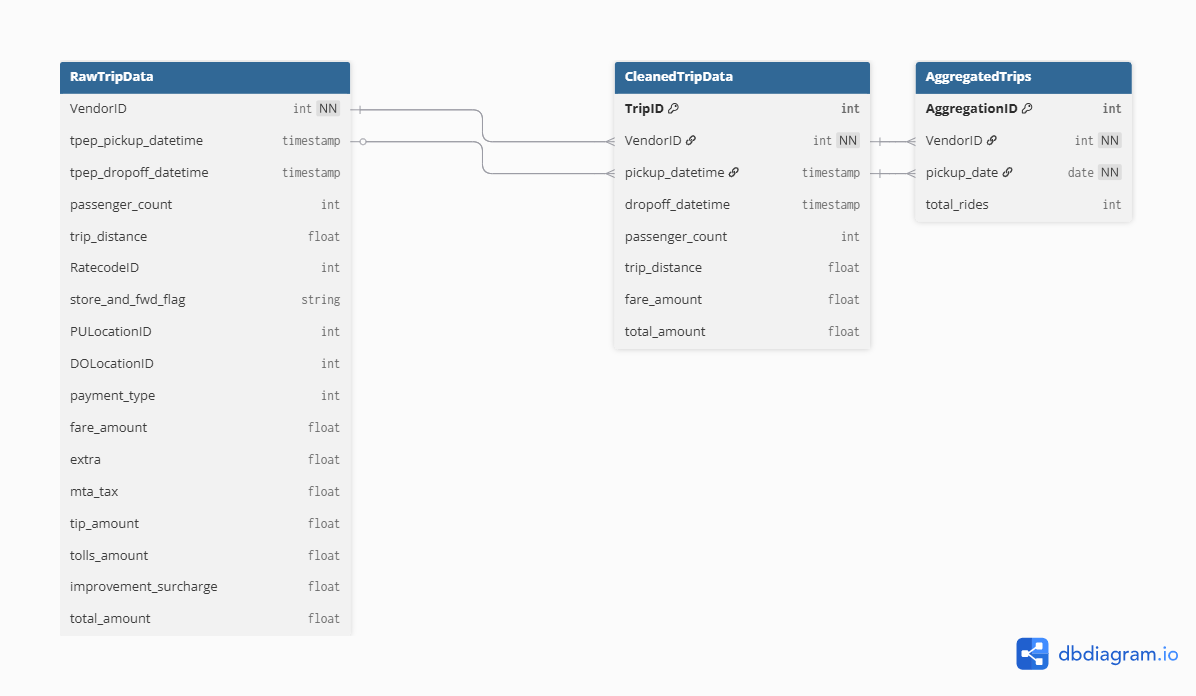


# 6. Code Overview

The PySpark code includes reading the file, cleaning using `.withColumn()`, filtering bad rows, aggregating data, and writing output in Parquet. All operations are performed in Databricks Notebook cells.

# 7. ER Diagram

The ER Diagram consists of three main entities:  
- RawTripData: The initial raw data table with all columns from CSV  
- CleanedTripData: Filtered and type-casted data  
- AggregatedTrips: Aggregated total rides per vendor per day  
  
Relationships:  
- CleanedTripData is derived from RawTripData  
- Aggregated Trips references Cleaned Trip Data



# 8. Conclusion

This ETL project successfully demonstrates how to use PySpark on Azure Databricks to process large-scale data, perform cleaning and aggregation, and efficiently store it in a cloud-based system. The use of Parquet files ensures fast access for analytics or reporting.