

**Scope of Work (SoW) for Transportation & Logistics Data Processing & Analysis**

By Arif

Intern-25

(2025-26)

**Table of Contents**

1. Introduction ...................................................................................... 3
2. Architecture Overview ..................................................................... 5
3. Dataset Description .......................................................................... 9
4. Solution Implementation .................................................................. 10
5. Git Repository .................................................................................. 13
6. Challenges & Solutions .................................................................... 14
7. Learnings .......................................................................................... 14
8. Conclusion ........................................................................................ 15
9. **Introduction**

In the modern transportation and logistics landscape, organizations collect massive amounts of data from vehicles, drivers, delivery routes, and operational procedures. The proper use of this data can considerably improve decision-making, lower operational costs, and increase delivery efficiency. However, raw data is frequently chaotic, fragmented, or dispersed across multiple systems, making it difficult to extract relevant insights.

This project aims to create a data processing and analytics platform utilizing Azure Databricks, PySpark, MySQL, and Power BI, following the Medallion Architecture methodology. The Medallion Architecture offers a structured approach to incrementally refining data from raw (Bronze), cleaned and enriched (Silver), and finally business-level aggregated data (Gold).

This project's main purpose was to automate the ingestion, transformation, and display of transportation and logistics data, allowing business users to track key performance indicators (KPIs) such as delivery efficiency, route optimization, and fleet performance. This was accomplished within a span of 30 hours by utilizing cloud-native tools and scalable data processing methodologies, with all results eventually presented in an interactive Power BI dashboard.

* 1. **Basic Concepts**

**ETL (Extract, Transform, Load):** the process of extracting data from source systems, transforming and refining it to satisfy specific requirements, and finally sending it to a specified destination for further use.

**Data pipeline:** an automated operation that executes and coordinates data transformation and transportation across multiple systems seamlessly.

**Parquet:** Parquet is a columnar data format optimized for storing and accessing data efficiently, specifically tailored to handle large-scale data workloads effectively.

**Azure:** Microsoft’s cloud platform offering cloud computing and storage services.

**Databricks:** a cloud computing and analytics platform. Databricks provisions compute for data processing, provides user interface for data operations.

**Azure Data Factory (ADF):** is one of the services in Azure cloud platform. ADF is used to extract data from source systems and orchestrate data pipelines.

**Power BI:** a business analytics tool developed by Microsoft that allows users to visualize data, share insights, and create interactive dashboards and reports.

**PySpark**: A python API for Apache Spark, enabling Python developers to leverage the power of Spark for large-scale distributed data processing, machine learning, and real-time analytics.

**MySQL:** an open-source relational database management system that enables users to store, organize, and retrieve data efficiently.

**Azure Blob Storage:** a Microsoft cloud-based solution designed for storing large amounts of unstructured data, such as text, images, videos, and log files.

**Azure Data Lake Storage Gen2**: Microsoft's cloud-based solution for big data analytics, combining high-performance hierarchical file storage with the scalability of Azure Blob Storage. It is designed to handle both structured and unstructured data, enabling efficient data processing and integration for advanced analytics workflows.

**Automation in Azure Data Factory (ADF):** it refers to the process of streamlining and orchestrating data workflows, enabling the execution of ETL tasks without manual intervention. It leverages triggers, schedules, and pipelines to automate data movement, transformation, and integration across various sources and destinations efficiently.

* 1. **Medallion Architecture**

Medallion Architecture is a data design pattern with its core idea being to logically organize data into layers. Classic Medallion Architecture consists of three layers: bronze, silver, gold. Data is processed, cleaned, and moved between layers using data pipelines. The quality and structure of data increases from layer to layer.

* **Bronze Layer**: Stores raw data directly ingested from the source system.
* **Silver Layer**: Contains cleaned and transformed data from the bronze layer.
* **Gold Layer**: Holds refined, use-case-specific data for business needs.

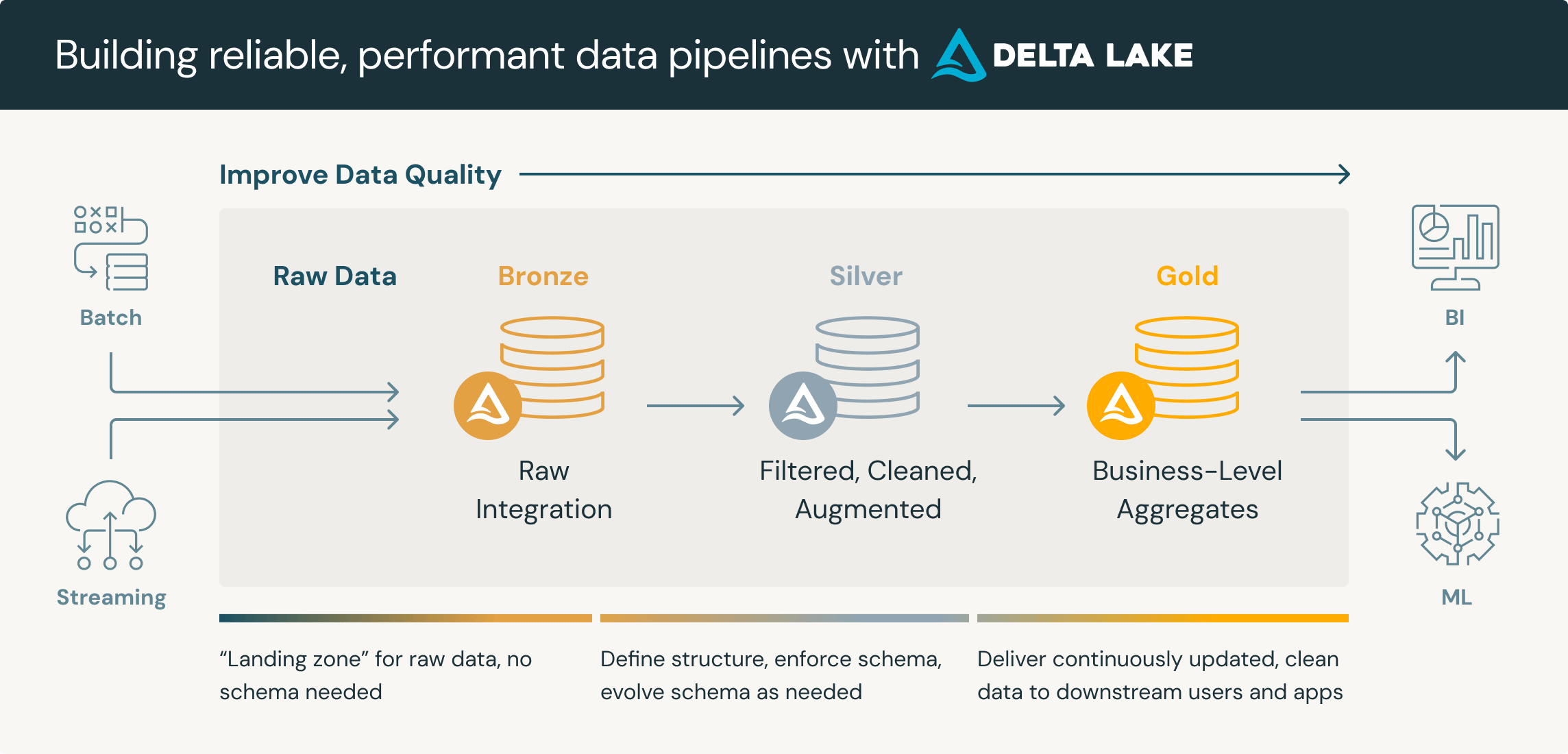


Figure 1. Medalion architecture. (Databricks 2024 b)

1. **Architecture Overview**

This project uses the Medallion Architecture concept as discussed above, which divides the data pipeline into three fundamental layers: bronze, silver, and gold. Each layer represents a stage of data refinement, enabling scalable, maintainable, and traceable data processing.

In this project the Medallion architecture is in the Azure Data Lake Storage Gen2 storage but since the source file was of csv and the raw file needed to be of parquet for better processing, the source file was stored in a new Azure blob storage and was transformed and brought into ADLS container bronze using PySpark Databricks notebook.

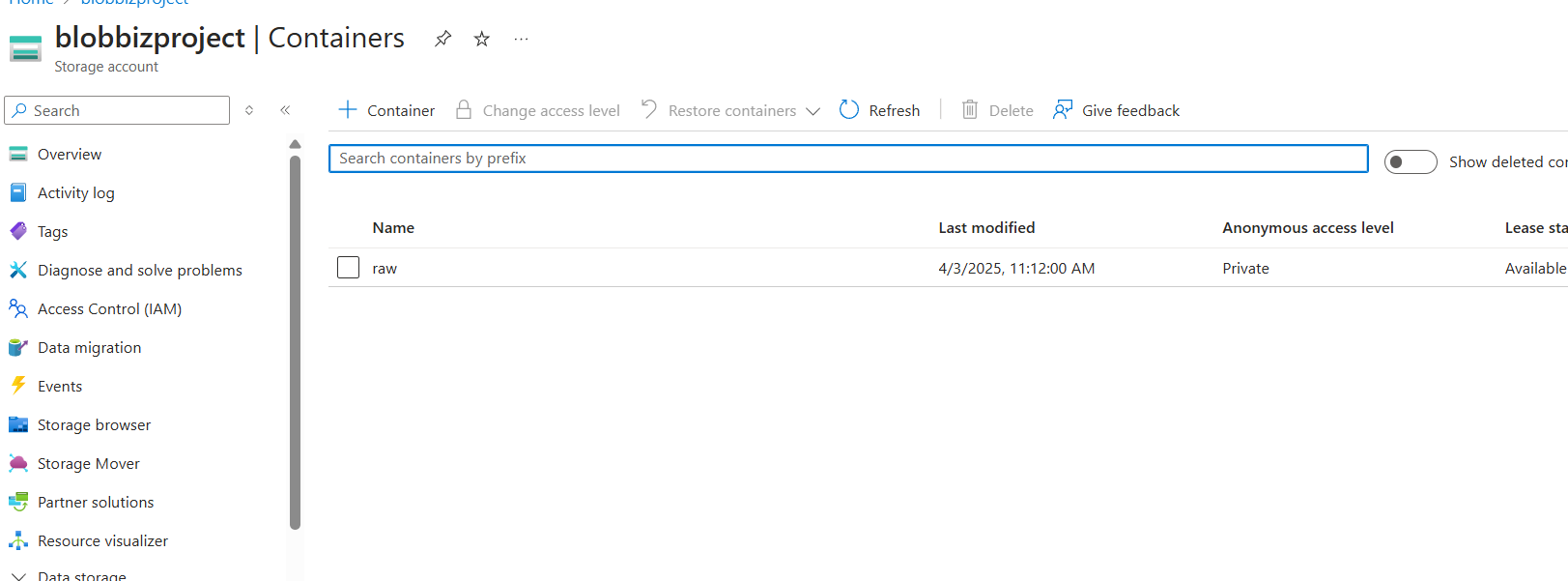


Figure 2. Source data Azure blob storage

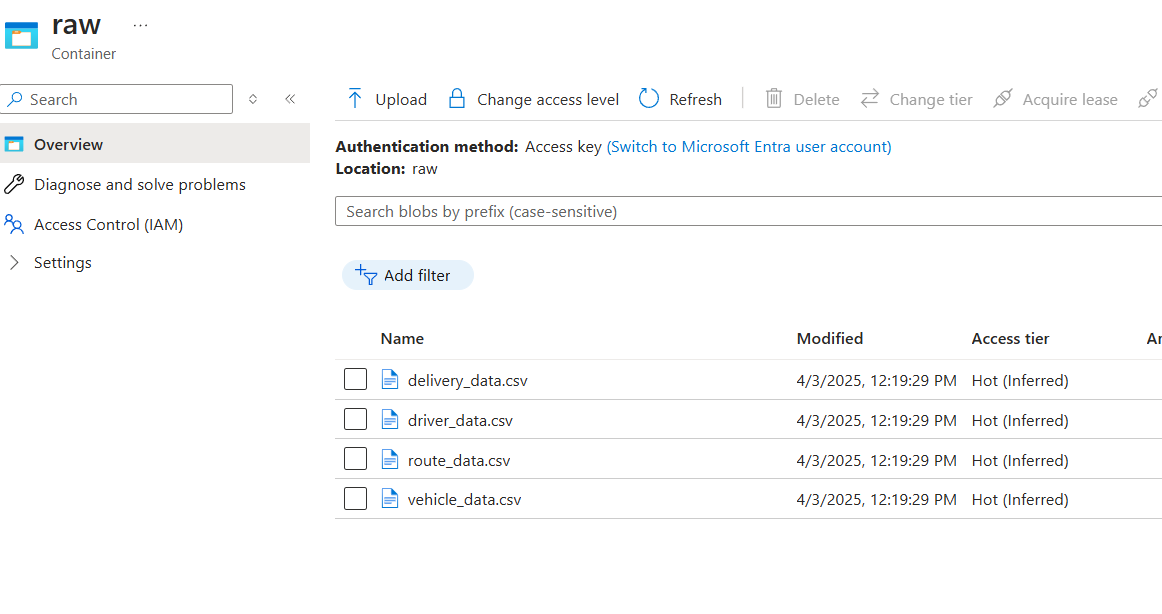


Figure 3. raw container with source csv files

* **Bronze Layer:**

The Bronze layer contains raw, unaltered data. There are no data types established, and the data is identical to the source system except that it is in parquet format.

The bronze layer data is used to recreate all subsequent levels. This layer frequently contains meta-data, such as the timestamp when the data was entered, the original file name, or the streaming source name.   
The bronze layer can contain numerous tables, each with data at a different transformation step.

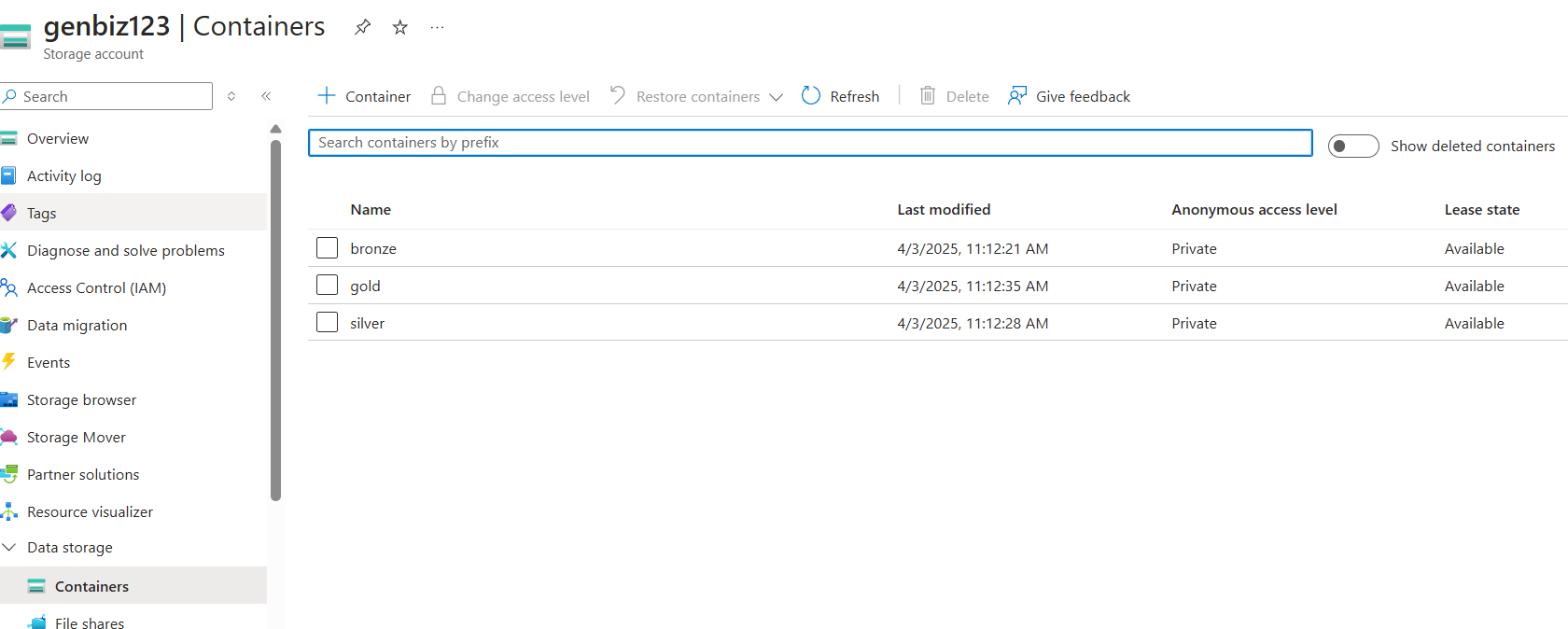


Figure 4. ADLS Gen2 Storage

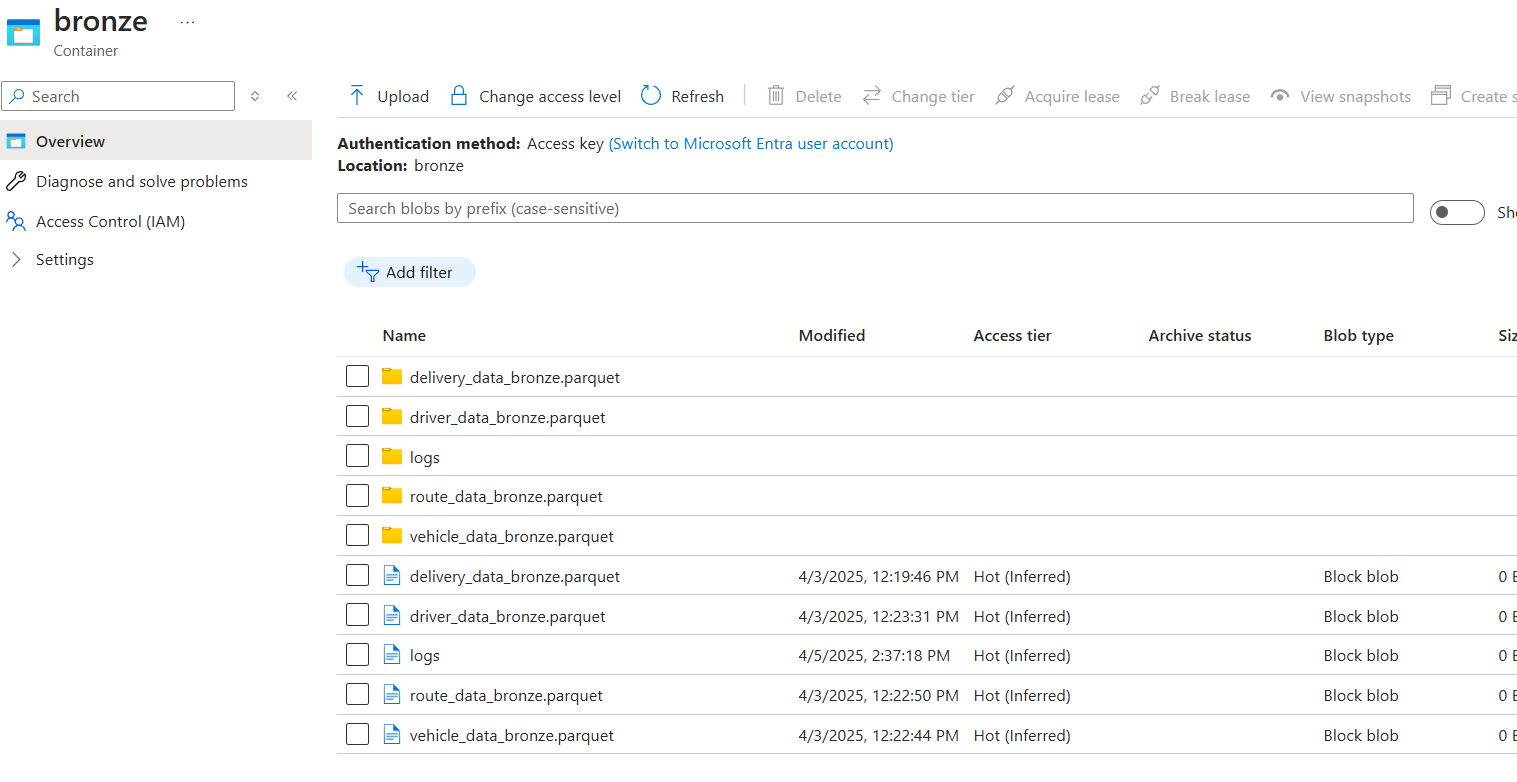


Figure 5. ADLS bronze container

* **Silver Layer:**

The silver layer contains data that has been conformed, cleansed, and converted in PySpark Databricks. This includes transformation and cleaning processes such as deleting duplicates, resolving faulty or incorrect data and assigning data types and the separation of data into columns. Silver layer has data in two formats one is parquet and the other is SQL as Deatricks notebook writes the transformed data as parquet file to silver ADLS container and to MySQL database silver \_db as delivery\_data\_silver table.

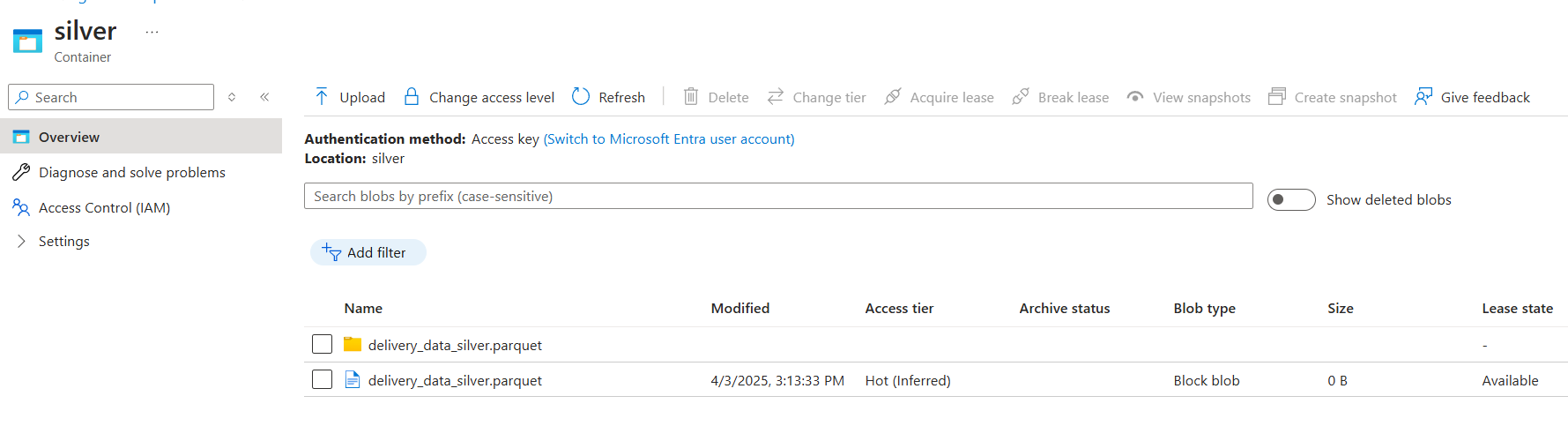


Figure 6. Silver container

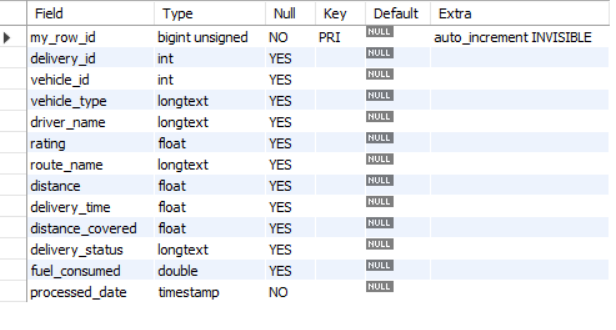


Figure 7. Table delivery\_data\_silver from database silver\_db

* **Gold Layer:**

The gold layer contains final aggregated dataset in MySQL for Power BI reports, computing key business metrics.

**Operations Performed in MySQL (Using Silver Layer as Source):**

* + Route Optimization Analysis Total deliveries per route.
  + Average delivery time per route.
  + Average fuel consumption per route.
  + Fleet Performance Total deliveries per vehicle.
  + Average distance covered per vehicle.
  + Average fuel efficiency per vehicle.
  + Driver Performance Total deliveries per driver.
  + Average delivery time per driver.
  + Driver rating analysis (average rating).

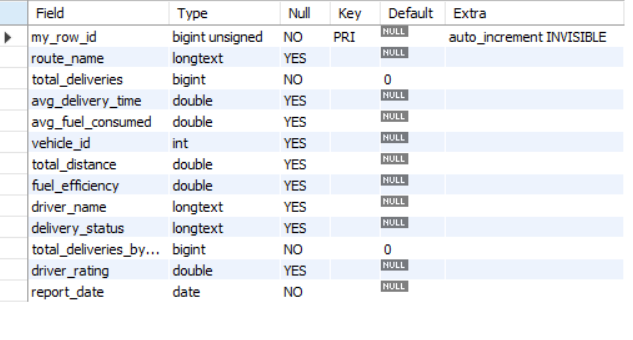


Figure 8. table transportation\_gold from database gold\_db

Finally, the gold\_db is download from Azure flexible server and loaded into power bi for visualization.

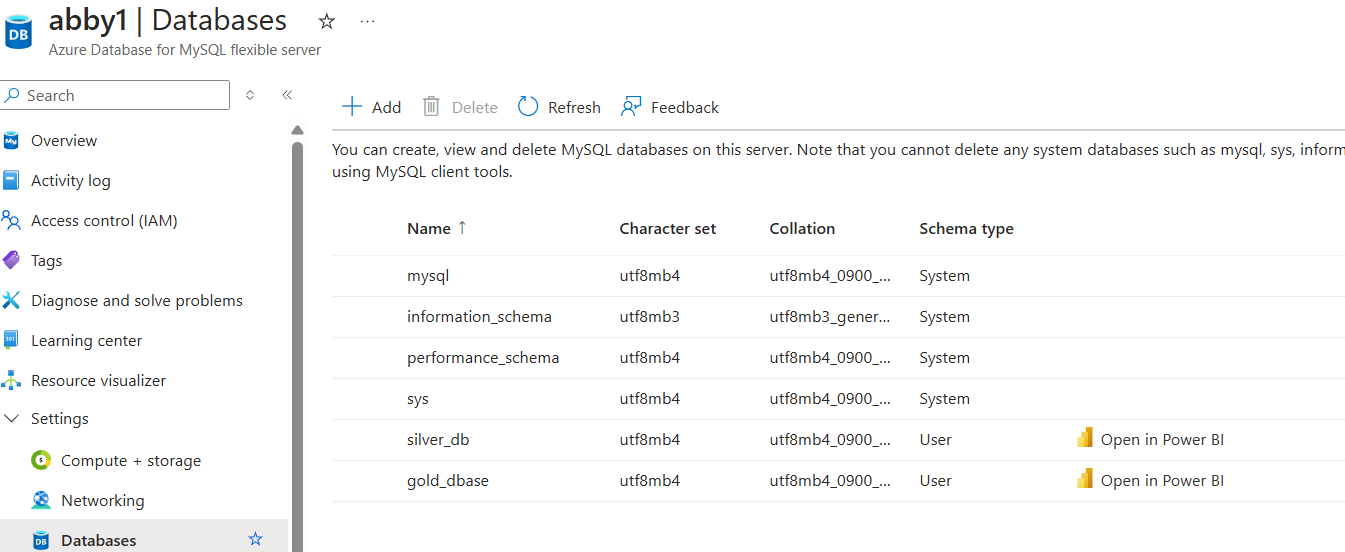


Figure 9. Downloading (.pbids) file of the database from Azure Databases for MySQL flexible servers

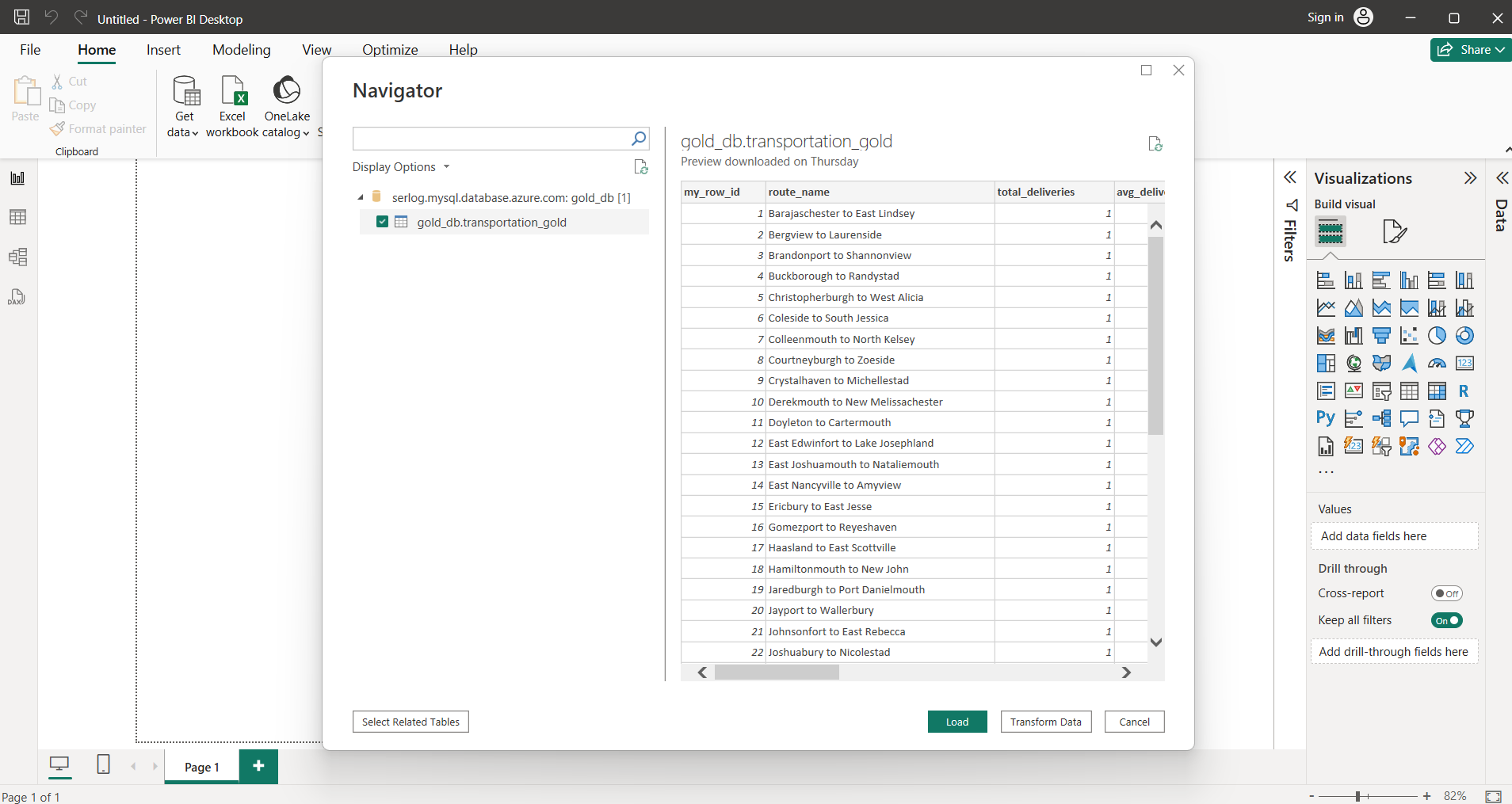


Figure 10. the downloaded gold\_db.pbids file loaded into power bi for visualization.

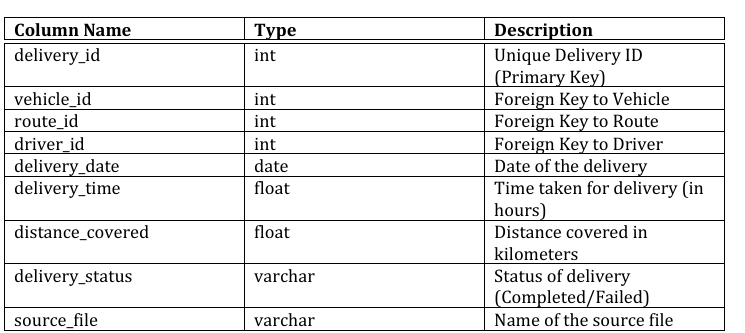
1. **Dataset Description**

The project uses a small but structured set of CSV files representing different aspects of the transportation and logistics process. These datasets are used to simulate real-world logistics scenarios, such as deliveries, vehicle usage, route performance, and driver efficiency.

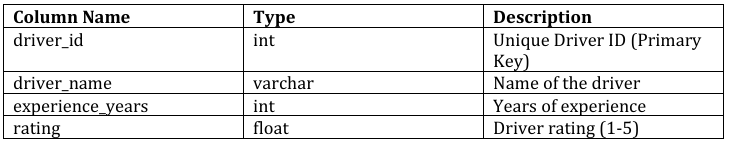
|  |  |
| --- | --- |
| Files | Descriptions |
| delivery\_data.csv | |  | | --- | |  |  |  | | --- | | Delivery logs containing information on each shipment including driver, vehicle, route, timestamps, and status. | |
| driver\_data.csv | Details of vehicles such as vehicle ID, type, capacity, and fuel efficiency. |
| route\_data.csv | Driver information including name, ID, rating, and license details. |
| Vehicle\_data.csv | Description of transportation routes, distances, start/end locations. |

* 1. **Schemas**

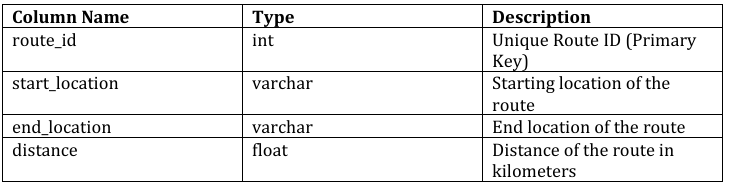
**delivery\_data.csv**



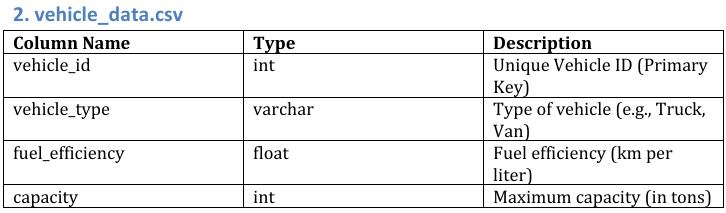
**driver\_data.csv**

****

**route\_data.csv**

****

**vehicles\_data.csv**



1. **Solution Implementation**

The Medallion Architecture was used to build the project solution. The entire process was orchestrated with Azure tools and big data technologies to enable scalability and automation.

* 1. **Data Ingestion -Bronze Layer**

The raw data files were provided in CSV format, representing various aspects of a logistics operation, such as delivery\_data.csv, driver\_data.csv, vehicle\_data.csv, route\_data.csv.

These files were ingested using PySpark in Azure Databricks and stored in ADLS the Bronze layer from raw in Azure blob storage.

During ingestion:

* The schema was inferred.
* Additional metadata fields like ingestion date and source file were added to maintain an audit trail.
* Data was written in Parquet format to ensure efficient storage and querying.

This ingestion step laid the foundation for reliable, traceable data processing.

Figure 11. Ingestion Flow (bronze layer)

* 1. **Data Processing -Silver Layer**

In the Silver layer, the goal was to produce clean, enriched datasets ready for business analysis. This involved:

* Null Handling: Records with missing critical fields (like driver\_id, vehicle\_id, or route\_id) were filtered out.
* Data Type Casting: Fields such as dates and numerical values were correctly cast for consistency.
* Join Operations:
  + delivery\_data was enriched by joining it with driver\_data, vehicle\_data, and route\_data.
  + This added contextual information such as driver names, vehicle types, and route names.
* Derived Columns:
  + Route\_name was derived by joining start and end location columns from route\_data
  + fuel\_consumed was calculated using distance\_covered / fuel\_efficiency.
  + delivery\_duration was derived from timestamps for further time-based analysis.

The processed silver data was written both to Parquet files in ADLS and to MySQL table using JDBC for easy BI access.

Figure 12. Processing Flow (silver layer)

* 1. **Aggregation and Business Logic – Gold Layer**

The Gold layer focused on generating business-level aggregates that would power KPIs in the dashboard. This was achieved through SQL queries executed in MySQL on top of the silver table.

These aggregations were stored in a final fact table called transportation\_gold within the gold\_db schema in MySQL.

Figure 13. Aggregation Flow (gold layer)

* 1. **Dashboarding with Power BI**

The final step was to present these insights through a professional, interactive Power BI dashboard, connected to the Gold MySQL table. The dashboard included:

**KPIs:**

* Total Deliveries
* Avg Delivery Time
* Fuel Consumption Rate
* Route-wise Delivery Count
* Driver Ratings

**Visuals:**

* Line Chart: Route Optimization over time
* Bar Chart: Vehicle usage and fuel efficiency
* Pie Chart: Delivery Status distribution
* Table: Driver-wise delivery stats
* Slicer & Filters: Region, Date, Vehicle Type

The Power BI dataset was scheduled to refresh automatically, syncing with the most recent Gold-layer data via MySQL.

Figure 14. Dashboard Flow (Power BI)

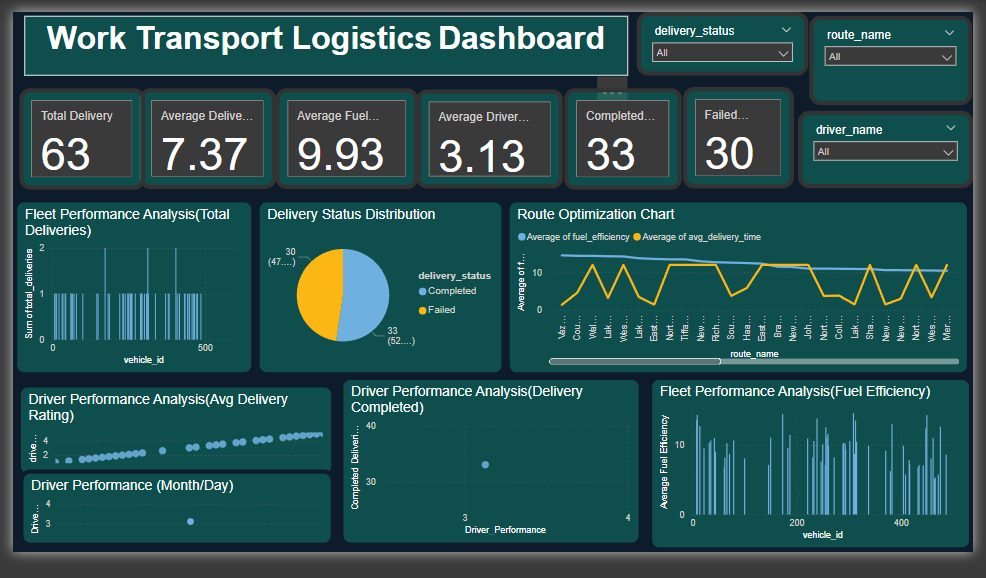


Figure 15. Final Dashboard with KPIs and visuals

* 1. **Automation using Azure Data Factory (ADF)**

To ensure end-to-end automation, Azure Data Factory pipelines were created to orchestrate:

1. Trigger data ingestion from CSVs into Bronze.
2. Launch Databricks notebooks for silver and gold processing.
3. Load final gold data into MySQL.

This made the pipeline fully automated, requiring no manual intervention once deployed.

Figure 16. Overall Pipeline Overview

1. **Git Repository**

All source files, scripts, notebooks, and SQL queries used during the implementation of this project are stored and version-controlled in a Git repository. This ensures collaboration, reproducibility, and code backup throughout the project.

transportation-logistics-project/

Git Rep link- <https://github.com/Aarif09/Work_Logistics_Project1>

**6. Challenges & How I Solved Them**

Working on this project came with its fair share of technical bumps, especially since it involved cloud platforms, big data tools, and dashboard integration. But each challenge turned out to be a great learning moment and helped me build some real problem-solving muscles.

Some of the main challenges I faced:

* Figuring out which Azure regions actually support MySQL Flexible Server (surprisingly tricky).
* Connecting MySQL Flexible Server to MySQL Workbench for running queries smoothly.
* Running into issues due to missing libraries when writing data into MySQL.
* Getting Power BI to talk to MySQL without throwing errors.

How I tackled them:

* I dove into recent blogs, community forums, and Azure’s latest updates to find regions where MySQL Flexible Server was supported.
* Followed online documentation closely to troubleshoot connection issues with MySQL Workbench.
* Solved the library issue by manually installing the missing one directly on the Databricks cluster.
* For the Power BI error, I installed the required .NET components to enable smooth MySQL connectivity.

**7. What I Learned**

As a fresher, this was my first real-world project—and honestly, it was a game changer. It gave me exposure to working with large datasets, industry-grade tools, and end-to-end data engineering processes.

Here are some key takeaways:

* Understood the layered data architecture—Bronze, Silver, Gold—and how it helps in structuring data from raw to business-ready.
* Learned how to build and manage data pipelines, including handling schema changes and dependencies.
* Got hands-on with PySpark notebooks for ingesting, cleaning, and transforming data.
* Used MySQL and Spark SQL to write logic for KPIs using joins, aggregations, and custom business rules.
* Created an interactive Power BI dashboard connected to the Gold layer to present insights in a user-friendly way.

**8. Wrapping It Up**

This project was a great introduction to building a cloud-based data and analytics solution, specifically for a transportation and logistics use case. I worked across multiple tools—Azure Databricks, Data Lake Storage, MySQL, Power BI, and Azure Data Factory—to create a complete, automated ETL pipeline using the Medallion architecture.

**8.1 What I Achieved**

* Pulled in raw logistics data from CSV files and structured it into a data lake.
* Built PySpark pipelines for cleaning and transforming the data.
* Wrote MySQL logic to calculate key business metrics.
* Developed a live Power BI dashboard with real-time KPIs.
* Automated everything using Azure Data Factory so the entire process runs on its own.

**8.2 Why It Matters for Stakeholders**

This solution helps different teams in the logistics space:

* Operations Managers can now track delivery times, delays, and fuel usage easily.
* Route Planners get insights from historical data to optimize delivery paths.
* Fleet Managers can analyze driver performance and vehicle usage trends.
* Executives can make informed decisions by viewing real-time business metrics through the dashboard.

**8.3 My Personal Takeaways**

* Learned how to work with modern data tools and cloud services.
* Faced and solved real technical challenges—definitely boosted my confidence.
* Got a full view of the data lifecycle—from raw ingestion to business insights.
* Realized the importance of structured workflows, clear documentation, and effective communication.

Overall, this project was not just about tech—it taught me how to think like a data engineer and collaborate like a professional. It was a solid start to my journey in the world of data and analytics, and I’m excited for what’s next!