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

By Amal A

Intern

(2025-26)

Table of Contents

1 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.

1.1 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.

1.2 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.

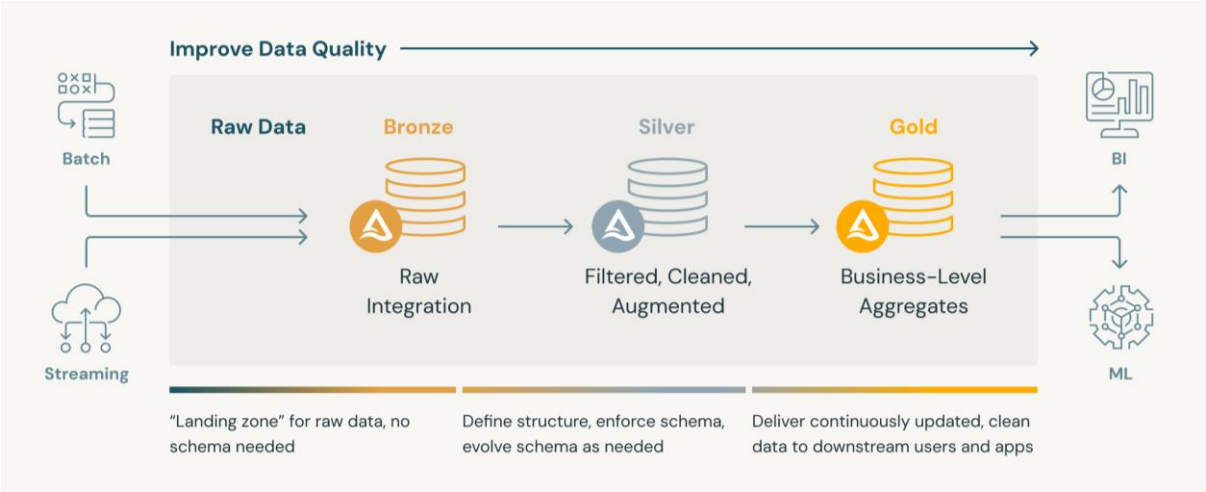

Figure 1. Medalion architecture. (Databricks 2024 b)

2 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.

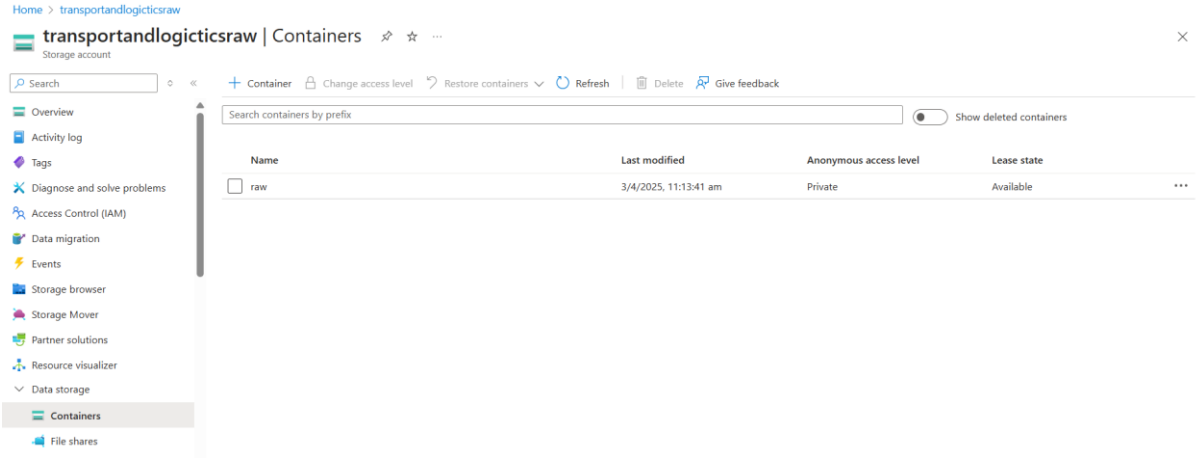

Figure 2. Source data Azure blob storage

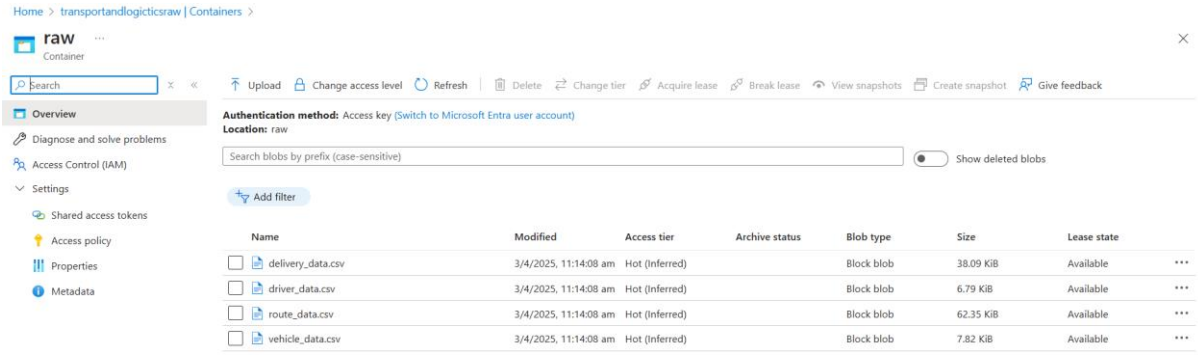

Figure 3. raw container with source csv files

- **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.

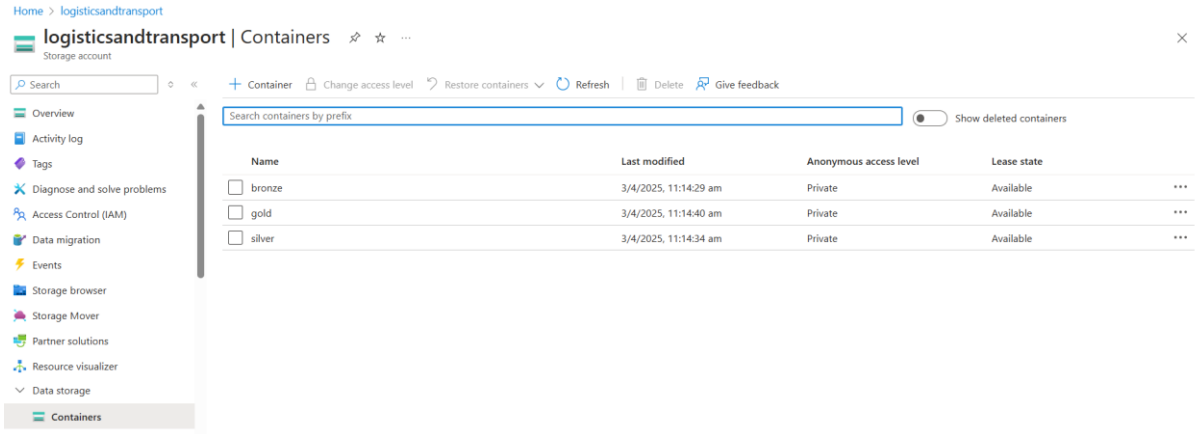

Figure 4. ADLS Gen2 Storage

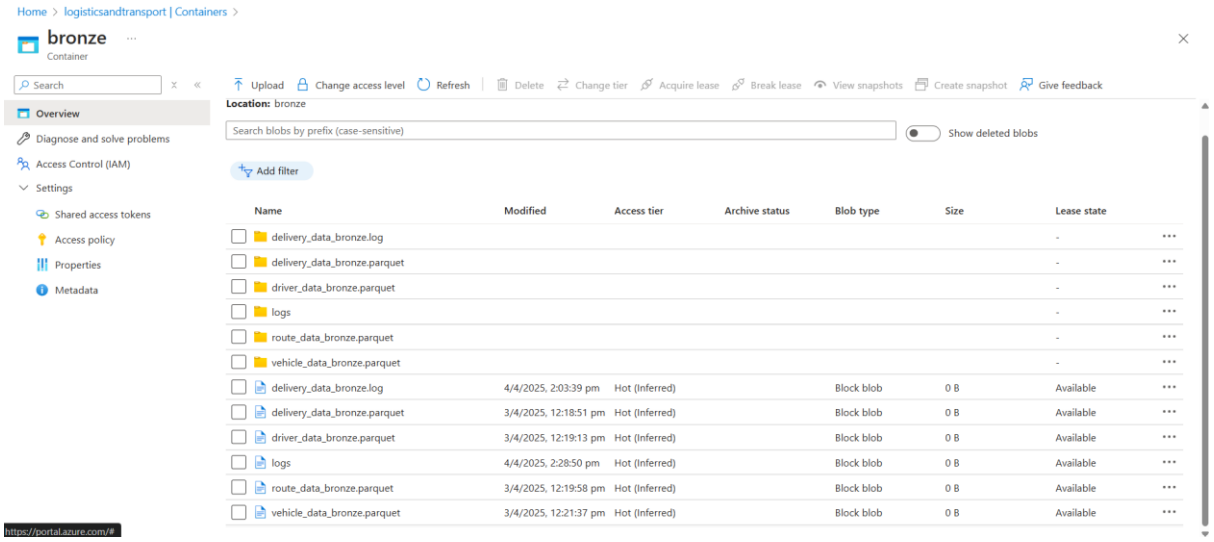

Figure 5. ADLS bronze container

- **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.

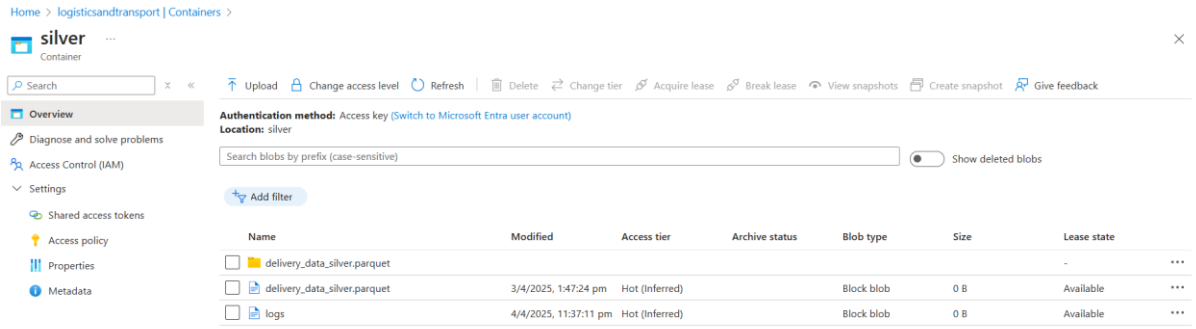

Figure 6. Silver container

Field	Type	Null	Key	Default	Extra
my_row_id	bigint unsigned	NO	PRI	NULL	auto_increment INVISIBLE
delivery_id	int	YES		NULL	
vehicle_id	int	YES		NULL	
vehicle_type	longtext	YES		NULL	
driver_name	longtext	YES		NULL	
rating	float	YES		NULL	
route_name	longtext	YES		NULL	
distance	float	YES		NULL	
delivery_time	float	YES		NULL	
distance_covered	float	YES		NULL	
delivery_status	longtext	YES		NULL	
fuel_consumed	double	YES		NULL	
processed_date	timestamp	NO		NULL	

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).

Field	Type	Null	Key	Default	Extra
my_row_id	bigint unsigned	NO	PRI	NULL	auto_increment INVISIBLE
route_name	longtext	YES		NULL	
total_deliveries	bigint	NO		0	
avg_delivery_time	double	YES		NULL	
avg_fuel_consumed	double	YES		NULL	
vehicle_id	int	YES		NULL	
total_distance	double	YES		NULL	
fuel_efficiency	double	YES		NULL	
driver_name	longtext	YES		NULL	
delivery_status	longtext	YES		NULL	
total_deliveries_by...	bigint	NO		0	
driver_rating	double	YES		NULL	
report_date	date	NO		NULL	

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.

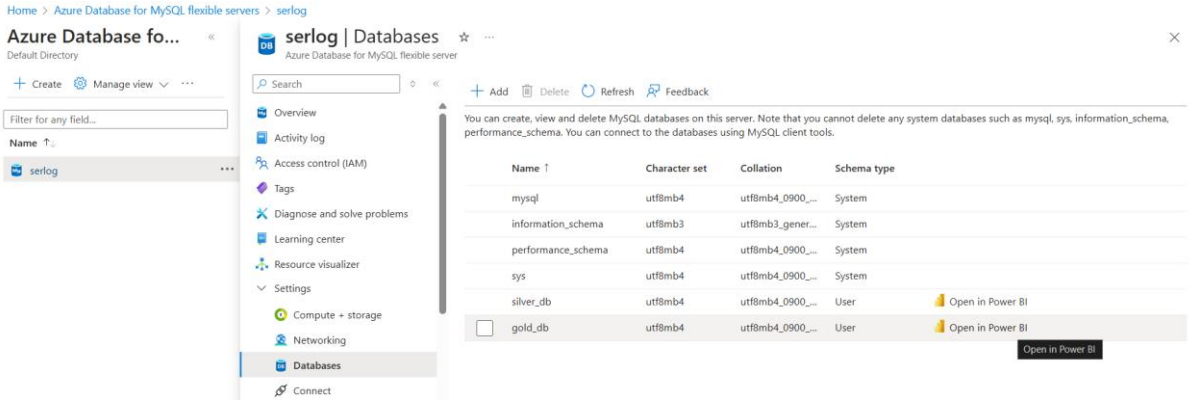

Figure 9. Downloading (.pbids) file of the database from Azure Databases for MySQL flexible servers

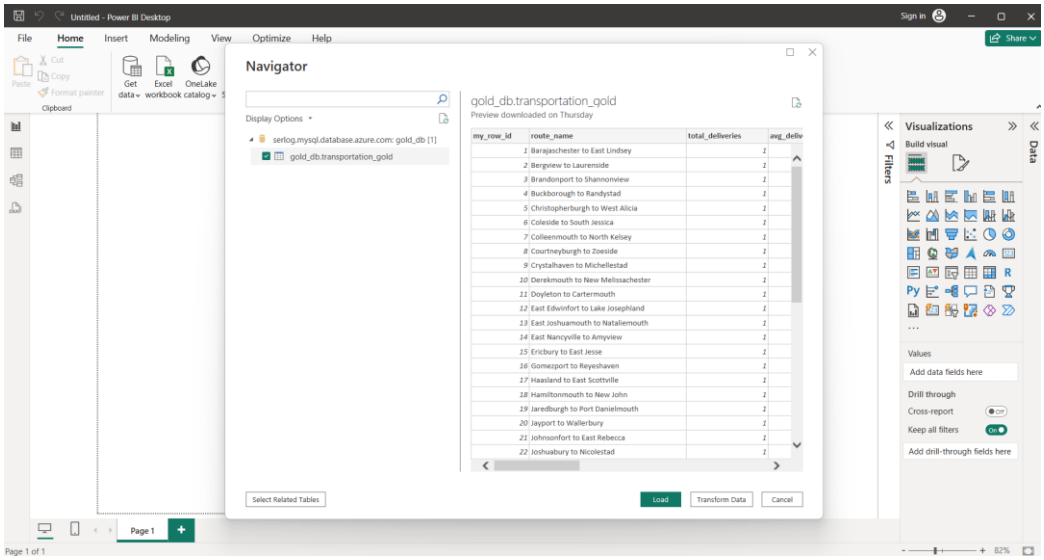

Figure 10. the downloaded gold_db.pbids file loaded into power bi for visualization.

3 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.

3.1 Schemas

delivery_data.csv

Column Name	Type	Description
delivery_id	int	Unique Delivery ID (Primary Key)
vehicle_id	int	Foreign Key to Vehicle
route_id	int	Foreign Key to Route
driver_id	int	Foreign Key to Driver
delivery_date	date	Date of the delivery
delivery_time	float	Time taken for delivery (in hours)
distance_covered	float	Distance covered in kilometers
delivery_status	varchar	Status of delivery (Completed/Failed)
source_file	varchar	Name of the source file

driver_data.csv

Column Name	Type	Description
driver_id	int	Unique Driver ID (Primary Key)
driver_name	varchar	Name of the driver
experience_years	int	Years of experience
rating	float	Driver rating (1-5)

route_data.csv

Column Name	Type	Description
route_id	int	Unique Route ID (Primary Key)
start_location	varchar	Starting location of the route
end_location	varchar	End location of the route
distance	float	Distance of the route in kilometers

vehicles_data.csv

Column Name	Type	Description
vehicle_id	int	Unique Vehicle ID (Primary Key)
vehicle_type	varchar	Type of vehicle (e.g., Truck, Van)
fuel_efficiency	float	Fuel efficiency (km per liter)
capacity	int	Maximum capacity (in tons)

4 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.

4.1 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.

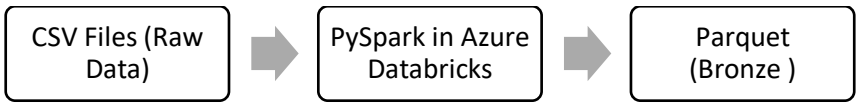

Figure 11. Ingestion Flow (bronze layer)

4.2 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.

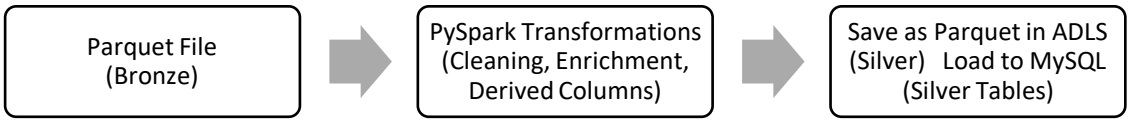

Figure 12. Processing Flow (silver layer)

4.3 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.

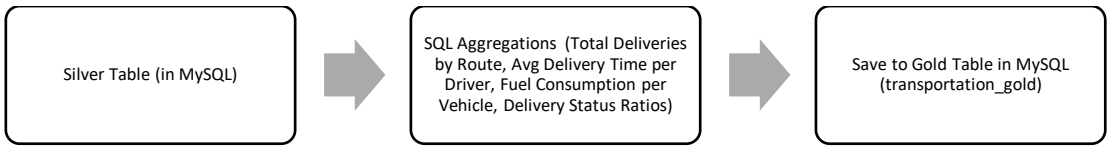

Figure 13. Aggregation Flow (gold layer)

4.4 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.

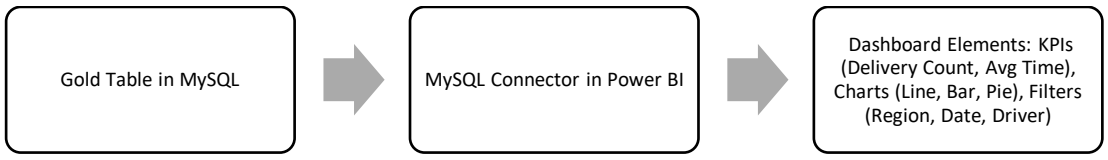

Figure 14. Dashboard Flow (Power BI)

Figure 15. Final Dashboard with KPIs and visuals

4.5 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

5 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/Atherius/Scope_of_Work_Transportation_Logistics/

6 Challenges & Solutions

Throughout the implementation of this project, I encountered various challenges while working with big data technologies, cloud services, and dashboard integration. These challenges provided valuable learning experiences and helped me develop troubleshooting and problem-solving skills.

- **Challenges:**
  - Finding apt location that provides MySQL flexible server service in Azure.
  - Connecting MySQL flexible server with workbench for querying.
  - Library unavailability caused error while writing to MySQL.
  - Connecting MySQL to Power BI prompted Error.

- **Solutions:**
  - Going through latest blogs and updates to find locations that provide MySQL flexible server service in Azure.
  - Understanding and rectifying mistakes through documentations available online.
  - Fixing library unavailability by directly installing library into cluster.
  - Downloaded and installed missing .NET essentials for MySQL connectivity to Power BI.

7 Learnings

As a fresher working on my first industrial internship project, this experience was a significant milestone in my professional journey. It gave me the opportunity to work on real-world data, apply modern tools, and follow industry best practices throughout the end-to-end data engineering lifecycle.

- Understood the importance of a layered architecture (Bronze, Silver, Gold) for organizing raw, clean, and business-ready data.
- Gained practical knowledge of designing data pipelines, managing dependencies, and handling schema evolution.
- developed PySpark notebooks for data ingestion, cleaning, and transformation.
- Performed complex joins, aggregations, and business rule logic to generate KPIs using MySQL and Spark SQL.
- Designed and published an interactive Power BI dashboard connected to the gold layer.

8 Conclusion

This project successfully delivered a complete, cloud-based data pipeline and analytics solution for a transportation and logistics use case. Through the integration of Azure Databricks, Data Lake Storage, MySQL, Power BI, and Azure Data Factory, I implemented an end-to-end ETL architecture following the Medallion pattern (Bronze, Silver, Gold).

8.1 **Achievements**

- Ingested raw logistics data from CSV into a structured data lake.
- Built PySpark pipelines for data cleaning, enrichment, and transformation.
- Designed MySQL-based aggregation logic to compute key business metrics.
- Developed a dynamic Power BI dashboard showcasing real-time KPIs.
- Automated the entire workflow using Azure Data Factory.

8.2 **Impact on Logistics Stakeholders**

The solution enables:

- Operations Managers to monitor delivery performance, delays, and fuel usage.
- Route Planners to identify optimization opportunities using historical trends.
- Fleet Managers to analyze driver efficiency and vehicle utilization.
- Executives to view key metrics through an interactive dashboard for data-driven decision-making.

8.3 **My Takeaways**

- Learned modern data engineering tools.
- Handled real-world datasets and solved practical technical challenges.
- Gained confidence in working with cloud platforms and automation tools.
- Developed a deeper understanding of both backend data architecture and frontend visualization.

This project not only enhanced my technical abilities but also taught me the value of structured workflows, documentation, and stakeholder communication. It has been a rewarding first step into the data industry, and a strong foundation for my future career in data engineering and analytics.