



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

Aftab Tamboli (INT-06)

Biz-Metric India Pvt Ltd

2025

<b>Author(s)</b>
Aftab Tamboli
<b>Degree</b>
Master's Degree of Statistics
<b>Report/Thesis Title</b>
Scope of Work (SoW) for Transportation & Logistics Data Processing & Analysis
<b>Number of pages and appendix pages</b>
29 + 2
<p>With the growing demand for data-driven decision-making in the transportation and logistics sector, organizations are turning toward modern, scalable data solutions. This project focuses on the development of an end-to-end data platform using the Medallion Architecture (Bronze, Silver, and Gold layers) for a logistics company, aiming to streamline operations such as route optimization, fleet performance monitoring, and delivery tracking. The goal is to automate the data pipeline using Azure Data Factory, Azure Data Lake Gen2, and Databricks for efficient data ingestion, transformation, and storage, while using MySQL and Power BI to deliver actionable insights to business stakeholders. The initial sections provide theoretical foundations and introduce core architecture and tools. The implementation section details the data journey from raw source files to refined insights, highlighting processing steps, governance practices, and the use of Apache Spark and Delta Lake. The project concludes with a reflection on the results, evaluating the objectives achieved, and discussing both the strengths and potential improvements of the implemented solution.</p>
<b>Key words</b>
Medallion Architecture, Data Engineering, Data Transformation, Business Intelligence, Microsoft Azure

## **Table of Contents**

1. Introduction
2. Software & Tools
3. Medallion Architecture Overview
4. Project Implementation
  1. Microsoft Azure Setup
  2. Storage Setup
    - Azure Blob Storage
    - Azure Data Lake Storage Gen2
  3. Database Setup
  4. Azure Databricks Service
  5. Azure Data Factory
5. Pipeline Implementation
  1. Connections and Datasets
  2. Bronze Layer Implementation
  3. Silver Layer Implementation
  4. Gold Layer Implementation
6. Power BI Dashboard
  1. Key Performance Indicators
  2. Visualizations
  3. Examples
7. Conclusion

## 1. Introduction

This project focuses on processing and analyzing transportation and logistics data using PySpark, MySQL, and Power BI, following the Medallion Architecture. The goal is to ingest, clean, transform, and aggregate data related to transportation routes, vehicle performance, delivery times, and shipment statuses to generate meaningful business insights.

The final output is a Power BI dashboard displaying key metrics like route optimization, delivery efficiency, and fleet performance. The entire project was implemented on the Microsoft Azure platform using Medallion Architecture for data processing and automating the complete data pipeline from raw data ingestion to the final Gold layer.

A key feature of this implementation is complete automation - a customer or data engineer only needs to add data into the raw files. Once the raw file is updated, the pipeline is triggered automatically, and insights are displayed on the Power BI dashboard without any manual intervention.

## 2. Software & Tools

The project utilizes Microsoft Azure to set up the following tools:

- **Azure Blob Storage (Storage Account)**: Storing raw data
- **Azure Data Lake Storage Gen2 (Storage Account)**: Containing medallion architecture
- **PySpark**: Data processing & transformation
- **Azure Databricks Service (Jupyter Notebook)**: Running PySpark scripts interactively
- **Azure Database for MySQL Server**: Storing transformed data & final aggregated results
- **Azure Data Factory**: Automating the entire pipeline
- **Power BI**: Data visualization & reporting

## 3. Medallion Architecture Overview

The project follows the Medallion Architecture, which consists of three structured layers:

- **Bronze Layer**: Stores raw data with minimal processing (Parquet files)
- **Silver Layer**: Cleansed, transformed, and enriched data (Parquet & MySQL tables)
- **Gold Layer**: Aggregated data for reporting & dashboards (MySQL tables)

## 4. Project Implementation

### 4.1 Microsoft Azure Setup

Microsoft Azure is a cloud computing platform used for building, deploying, and managing applications. It offers services like data storage, analytics, AI, networking, and security. Azure is flexible, scalable, and supports hybrid solutions, making it ideal for businesses and developers across various industries.

Here's how the home page of Microsoft Azure looks:

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the search bar, there are icons for creating a resource, Education, Azure Databricks, Data factories, SQL databases, Azure Database for MySQL, MySQL Server - Azure Arc, Storage accounts, Resource groups, and More services. The 'Azure services' section also includes a 'Create a resource' button. Below this is a 'Resources' section with tabs for 'Recent' and 'Favorite'. It lists several resources: bizblobstore (Storage account), bizlakegen (Storage account), BizRG (Resource group), BizWorkSpace (Azure Databricks Service), bizdatafactory (Data factory (V2)), bizserver (Azure Database for MySQL flexible server), and Azure for Students (Subscription). Each item has a small icon, its name, type, and last viewed time. Below the resources is a 'See all' link. Further down are sections for 'Navigate' (Subscriptions, Resource groups, All resources, Dashboard) and 'Tools'. On the far right, there's a vertical sidebar with a user profile and other navigation links.

A resource group named "BizRG" contains all storage accounts, Data Factory, Database for MySQL Server, and Databricks Services. From here, you can directly access any tools, service, or software utilized for the project.

This is the dashboard for resource group - BizRG:

The screenshot shows the Microsoft Azure resource group dashboard for 'BizRG'. At the top, there's a search bar and a Copilot button. The dashboard title is 'BizRG >'. Below the title, there are tabs for 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Resource visualizer', 'Events', 'Settings', 'Cost Management', 'Monitoring', 'Automation', and 'Help'. The 'Overview' tab is selected. It displays basic information: Subscription (move) : Azure for Students, Subscription ID : f939db4f-5842-4c48-945a-0b3358614845, Deployments : 6 Succeeded, Tags (edit) : Add tags, Location : Southeast Asia. Below this is a 'Resources' section with a table of resources. The table has columns for Name, Type, and Location. The resources listed are: bizblobstore (Storage account, Southeast Asia), bizdatafactory (Data factory (V2), Southeast Asia), bizlakegen (Storage account, Southeast Asia), bizserver (Azure Database for MySQL flexible server, Southeast Asia), and BizWorkSpace (Azure Databricks Service, Southeast Asia). There are also buttons for filtering, grouping, and switching between grid and list views. At the bottom, there are navigation links for '< Previous', 'Page 1 of 1', 'Next >', and a 'Give feedback' button.

## 4.2 Storage Setup

### 4.2.1 Azure Blob Storage

A storage account called "bizblobstore" was created as Azure Blob Storage. Within bizblobstore, a new container named "raw" was created to store the initial four raw data files provided by the company in CSV format. These raw files will be copied later and converted into parquet format for storage in the bronze layer.

This is the Azure blob storage – bizblobstore:

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

afabtambolee@outloo...  
DEFAULT DIRECTORY (AFTABTAM...)

Home > bizblobstore

bizblobstore | Containers

Storage account

Search

+ Container Change access level Restore containers Refresh Delete Give feedback

Overview Activity log Tags Diagnose and solve problems Access Control (IAM) Data migration Events Storage browser Storage Mover Partner solutions Resource visualizer Data storage Containers File shares Queues Tables Security + networking Data management Settings Monitoring Monitoring (classic) Automation

Name raw Last modified 2/4/2025, 2:31:13 pm Anonymous access level Private Lease state Available

The raw files in the container are:

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

afabtambolee@outloo...  
DEFAULT DIRECTORY (AFTABTAM...)

Home > bizblobstore | Containers > raw

Container

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot Give feedback

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: raw

Search blobs by prefix (case-sensitive)

Show deleted blobs

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
delivery_data.csv	2/4/2025, 2:34:00 pm	Hot (Inferred)		Block blob	38.09 KiB	Available
driver_data.csv	2/4/2025, 2:34:00 pm	Hot (Inferred)		Block blob	6.79 KiB	Available
route_data.csv	2/4/2025, 2:34:00 pm	Hot (Inferred)		Block blob	62.35 KiB	Available
vehicle_data.csv	2/4/2025, 2:34:00 pm	Hot (Inferred)		Block blob	7.82 KiB	Available

#### 4.2.2 Azure Data Lake Storage Gen2

Another storage account named "bizlakegen" was created as Azure Data Lake Storage Gen2 for implementing the medallion architecture (bronze layer, silver layer, gold layer). As we progress, bizlakegen will be used to establish connections with the Azure Data Factory. In bizlakegen, three containers were created: bronze, silver, and gold.

Name	Last modified	Anonymous access level	Lease state
bronze	2/4/2025, 2:37:20 pm	Private	Available
gold	2/4/2025, 2:37:34 pm	Private	Available
silver	2/4/2025, 2:37:28 pm	Private	Available

In the bronze container, four directories were created to store the parquet format of each file and the generated logs:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
Delivery	3/4/2025, 11:14:21 am	-	-	-	-	***
Driver	3/4/2025, 11:14:29 am	-	-	-	-	***
Route	3/4/2025, 11:14:42 am	-	-	-	-	***
Vehicle	3/4/2025, 11:14:53 am	-	-	-	-	***

The silver and gold containers are initially empty as they will contain the processed data once the pipeline is triggered.

#### 4.3 Database Setup

An Azure Database for MySQL Flexible Server was created for storing transformed data and final aggregated results:

The screenshot shows the Azure portal interface for managing a MySQL Flexible Server named 'bizserver'. Key details include:

- Subscription: Azure for Students
- Subscription ID: f939db4f-5842-4c48-945a-0b3358614845
- Resource group: BizRG
- Status: Ready
- Location: Southeast Asia
- Configuration: Burstable\_81ms\_1vCores\_2GiB RAM\_20 storage\_360 IOPS
- MySQL version: 8.0
- Availability zone: 2
- Created on: 2025-04-03 03:19:28.9152861 UTC

The 'Getting started' section includes:

- MySQL Flexible server learning center (Start button)
- Configure network access for your server (Start button)
- Setup a sample database schema (Start button)

This was connected to MySQL Workbench (already installed on the local system) using server name, hostname, password, etc. This connection enables running SQL queries for the project.

The MySQL Workbench interface is shown, featuring a dark-themed UI. The main screen displays the welcome message: "Welcome to MySQL Workbench". Below the message, a brief description of MySQL Workbench's capabilities is provided. At the bottom of the main window, there are links to "Browse Documentation >", "Read the Blog >", and "Discuss on the Forums >".

The left sidebar is titled "MySQL Connections" and contains two entries:

- Local instance MySQL80
- Azure bizserver Connection

A red arrow points from the "Azure bizserver Connection" entry towards the main content area, indicating the connection being used.

#### 4.4 Azure Databricks Service

Azure Databricks Service named "BizWorkSpace" was set up to write PySpark commands/code in databricks notebooks for transformations, operations, or aggregations. Later, this will be connected to Azure Data Factory to automate the databricks notebook in the data pipeline.

**BizWorkSpace** Azure Databricks Service

**Overview**

**Essentials**

- Status : Active
- Resource group : [BizRG](#)
- Location : Southeast Asia
- Subscription : [Azure for Students](#)
- Subscription ID : f939db4f-5842-4c48-945a-0b3358614845
- Tags (edit) : [Add tags](#)

Managed Resource Group : [databricks-rg-BizWorkSpace-ydhmtk3ainms](#)

URL : <https://adb-2457935583419999.19.azuredata.databricks.net>

Pricing Tier : [Trial \(Premium - 14-Days Free DBUs\) \(Click to change\)](#)

**Launch Workspace**

**Upgrade to Premium**

**Documentation** [Documentation](#)

**Getting Started** [Getting Started](#)

**Import Data from File** [Import Data from File](#)

**Import Data from Azure Storage** [Import Data from Azure Storage](#)

**Notebook** [Notebook](#)

**Admin Guide** [Admin Guide](#)

**Link Azure ML workspace** [Link Azure ML workspace](#)

## 4.5 Azure Data Factory

Finally, Azure Data Factory (ADF) named "bizdatafactory" was created to orchestrate and automate the movement and transformation of data across different layers of the project. This is where everything happens, from ingestion of raw data to preparing the final gold layer.

**bizdatafactory** Data factory (V2)

**Overview**

**Essentials**

- Resource group ([move](#)) : [BizRG](#)
- Status : Succeeded
- Location : Southeast Asia
- Subscription ([move](#)) : [Azure for Students](#)
- Subscription ID : f939db4f-5842-4c48-945a-0b3358614845

Type : Data factory (V2)

Getting started : [Quick start](#)

**Azure Data Factory Studio**

**Launch studio**

**Quick Starts**

**Tutorials**

**Template Gallery**

**Training Modules**

**Monitoring**

**PipelineRuns** 100  
80

**ActivityRuns** 100  
80

## 5. Pipeline Implementation

### 5.1 Connections and Datasets

Here are the connections made with Azure Data Factory:

The screenshot shows the Azure Data Factory interface with the 'Linked services' section selected. The left sidebar contains navigation links for General, Connections (selected), Integration runtimes, Microsoft Purview, Source control, Author, Security, Workflow orchestration manager, and Apache Airflow. The main area displays a table of linked services with the following data:

Name	Type	Related	Annotations
BizBlobStoreConnection	Azure Blob Storage	4	
BizLakeGenConnection	Azure Data Lake Storage Gen2	10	
BizServerConnection	Azure Database for MySQL	1	
BizWorkSpaceConnection	Azure Databricks	1	

Fourteen datasets were created in the data factory to establish connections with particular activities in the data pipeline:

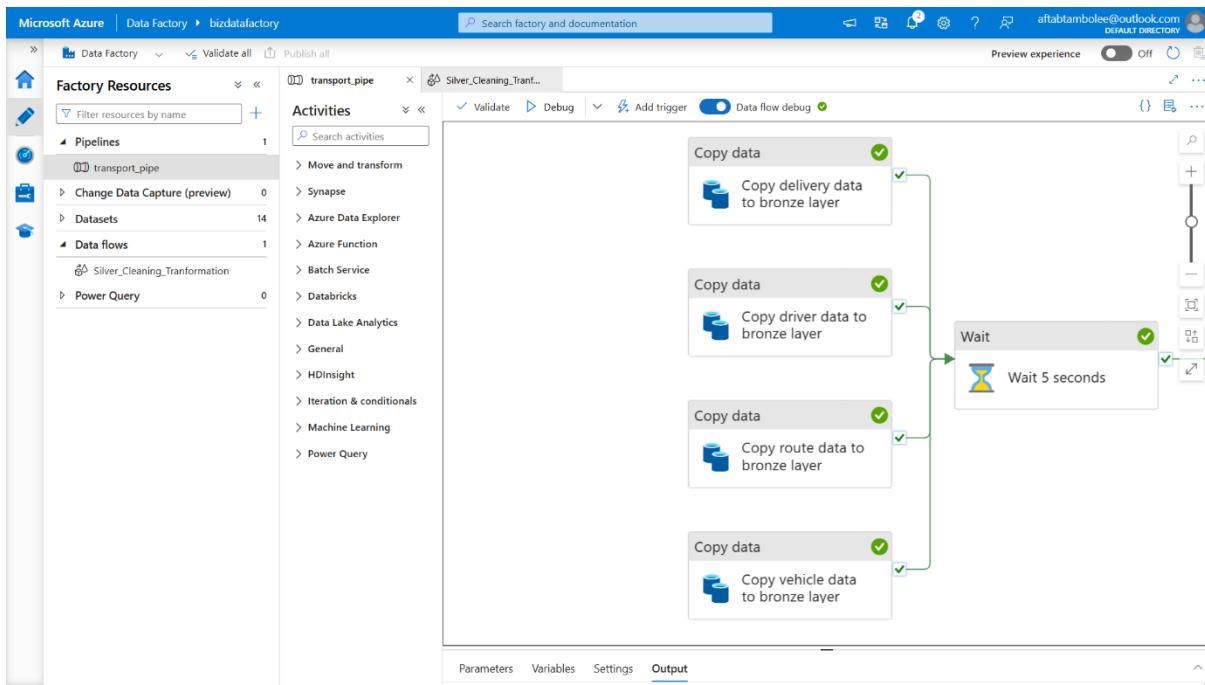
-  DeliveryParFile
-  DeliveryRawCSV
-  DeliveryRawParquet
-  DriverParFile
-  DriverRawCSV
-  DriverRawParquet
-  RouteParFile
-  RouteRawCSV
-  RouteRawParquet
-  SilverDump
-  SilverTable
-  VehicleParFile
-  VehicleRawCSV
-  VehicleRawParquet

## 5.2 Bronze Layer Implementation

A new pipeline was created in bizdatafactory to construct the complete data flow. The pipeline begins with the ingestion of raw data and proceeds to preparing the gold layer.

First, copy activity was used in the pipeline to copy the raw CSV data files from the raw container in bizblobstore to the bronze container in bizlakegen. The data remains unchanged but is converted to parquet format.

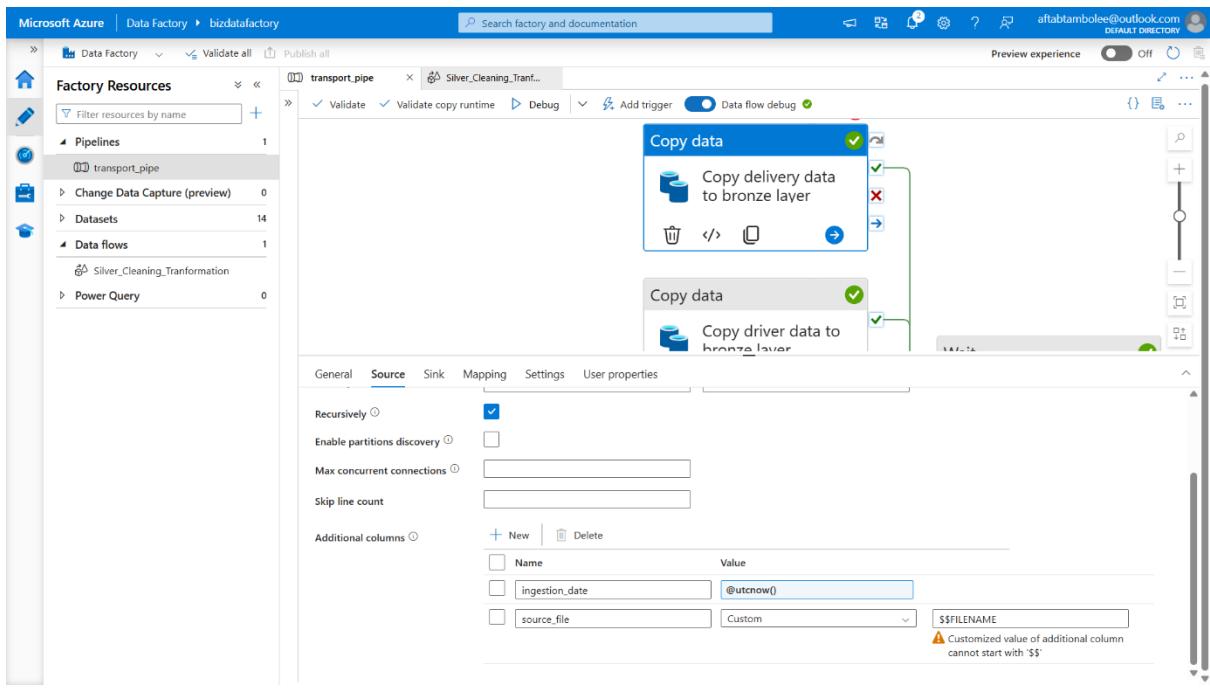
Since there are 4 files, 4 copy activities were used:



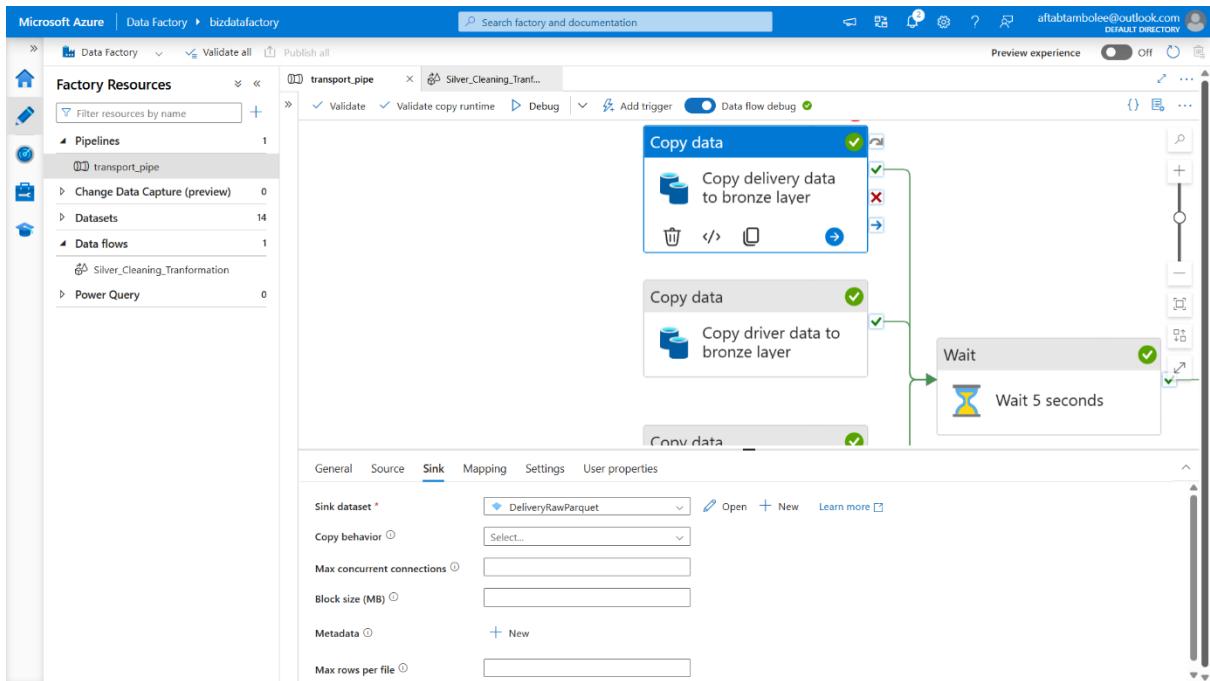
For each copy activity, the source was set as the raw CSV file from bizblobstore:

This screenshot shows the 'Source' tab configuration for one of the 'Copy data' activities in the pipeline. The 'Source dataset' is set to 'DeliveryRawCSV'. The 'File path type' is 'File path in dataset', selected with a radio button. The 'Start time (UTC)' and 'End time (UTC)' fields are empty. The 'Recursively' checkbox is checked. The 'Enable partitions discovery' checkbox is unchecked. The 'Max concurrent connections' field is empty. Other tabs visible include 'General', 'Sink', 'Mapping', 'Settings', and 'User properties'.

Two audit columns (ingestion\_date, source\_file) were added to each bronze table to track data lineage:



The sink was set to the appropriate directory in the bronze container to store each raw parquet file and their generated logs:



Logging was enabled at the INFO level in each copy activity, with the log folder path set to the respective file directory in the bronze container:

The parquet files were successfully generated in their respective directories along with their logs:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
copyactivity-logs	5/4/2025, 11:16:50 am				-	
delivery_data.parquet	5/4/2025, 11:16:55 am	Hot (Inferred)		Block blob	27.83 KIB	Available

The raw data contains some null/empty values. The total raw data with 2 audit columns is shown below (example for delivery data; similar structure applies to driver, route, and vehicle data):

The screenshot shows a Jupyter Notebook interface. In the code cell [3], the following Python code is run:

```
import pandas as pd
delivery_par = pd.read_parquet(r"D:\Downloads\delivery_data.parquet")
display(delivery_par)
```

The resulting DataFrame is displayed below:

	delivery_id	vehicle_id	route_id	driver_id	delivery_date	delivery_time	distance_covered	delivery_status	ingestion_date	source_file
0	1	204	64	32	23-01-2025	None	None	Completed	2025-04-05T05:46:38.5664428Z	delivery_data.csv
1	2	288	1197	108	06-08-2024	12	None	Failed	2025-04-05T05:46:38.5664428Z	delivery_data.csv
2	3	335	161	158	03-09-2024	12	500	None	2025-04-05T05:46:38.5664428Z	delivery_data.csv
3	4	463	1248	231	17-05-2024	None	142.52	Completed	2025-04-05T05:46:38.5664428Z	delivery_data.csv
4	5	383	900	200	10-01-2025	12	84.86	Failed	2025-04-05T05:46:38.5664428Z	delivery_data.csv
...	...	...	...	...	...	...	...	...	...	...
995	996	270	729	228	29-09-2024	4.32	94.4	None	2025-04-05T05:46:38.5664428Z	delivery_data.csv
996	997	166	1954	73	02-04-2024	1.21	None	None	2025-04-05T05:46:38.5664428Z	delivery_data.csv
997	998	218	67	276	18-03-2025	12	None	Failed	2025-04-05T05:46:38.5664428Z	delivery_data.csv
998	999	468	327	153	None	12	158.74	None	2025-04-05T05:46:38.5664428Z	delivery_data.csv
999	1000	490	1042	251	None	None	500	None	2025-04-05T05:46:38.5664428Z	delivery_data.csv

1000 rows × 10 columns

## Generated logs example:

The screenshot shows the Microsoft Azure Storage Explorer interface. On the left, a blob container named "bronze" is selected. On the right, a log file named "copyactivity-logs/Copy delivery data to bronze layer/f6ce4e48-02..." is displayed in a text editor.

The log file content is as follows:

```
1 Timestamp,Level,OperationName,OperationItem,Message
2 2025-04-05 05:46:52.826735,Info,FileRead,"delivery_data.csv","Start to read file: (""Path"":""delivery_data.csv"")"
3 2025-04-05 05:46:53.1236184,Info,FileWrite,"delivery_data.parquet","Start to write file."
4 2025-04-05 05:46:53.1392374,Info,FileRead,"delivery_data.csv","Complete reading file successfully. "
5 2025-04-05 05:46:55.8267656,Info,FileWrite,"delivery_data.parquet","Complete writing file. File is successful."
6
```

At this point, the bronze layer is successfully completed with each parquet file and its logs:

- `delivery_data_bronze.csv` → `delivery_data.parquet`
- `vehicle_data_bronze.csv` → `vehicle_data.parquet`
- `route_data_bronze.csv` → `route_data.parquet`
- `driver_data_bronze.csv` → `driver_data.parquet`

All four copy activities were merged with a wait activity applying a 5-second delay to ensure all files have enough time to be transferred to their destinations before the pipeline automatically performs the silver layer transformation activities.

### 5.3 Silver Layer Implementation

For the silver layer preparation, the data was cleaned, structured, and enriched for analysis by performing:

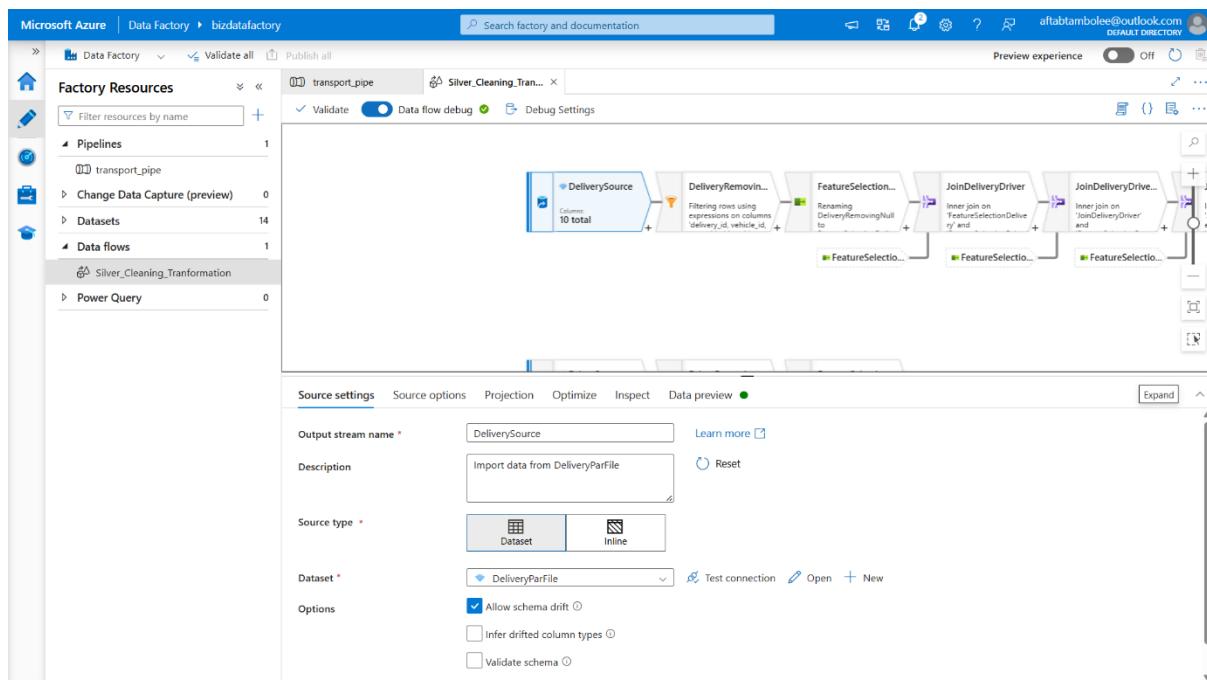
- **Data Cleaning:** Removing records with null values in critical columns
- **Joining Related Tables:** Adding vehicle details, driver details, and route descriptions
- **Computing Additional Fields:** e.g., fuel\_consumed = distance\_covered / fuel\_efficiency
- **Storing Data** in both Parquet and MySQL for further aggregation

Dataflow in Azure Data Factory was used for this purpose, for several reasons:

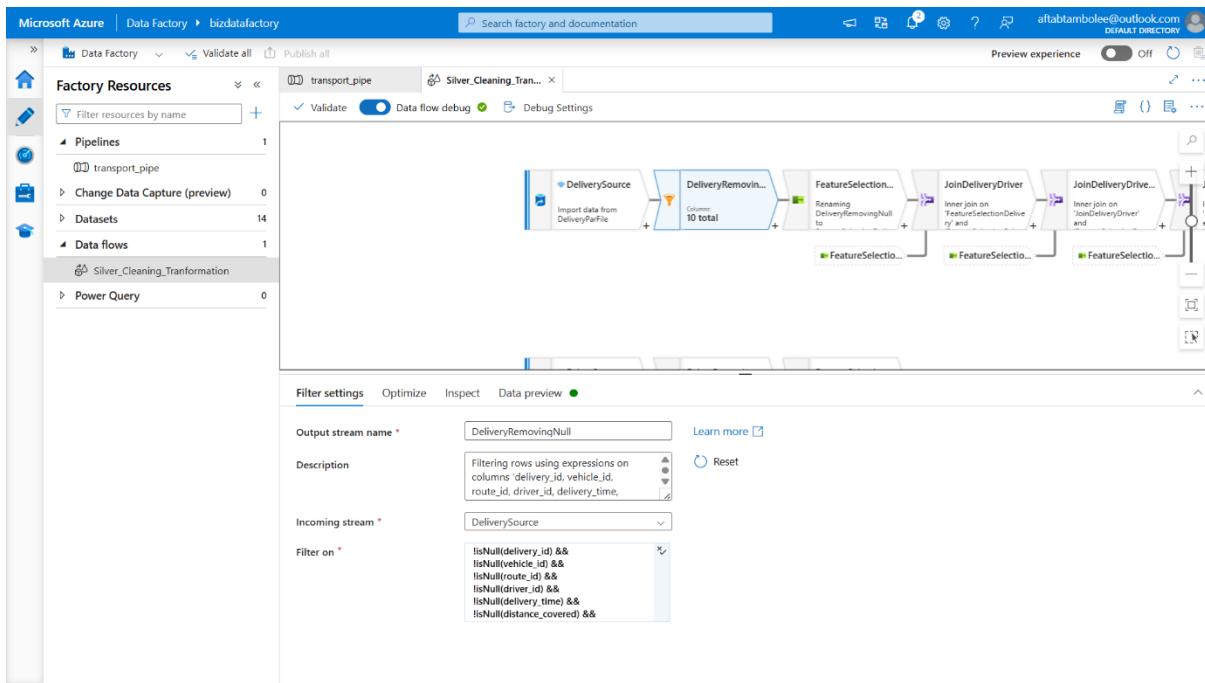
1. It provides freedom to generate parquet files with desired naming conventions, which is not possible in Databricks PySpark notebook
2. It offers easier transformation (cleaning, joining, computing) with graphical representation of the flow
3. It avoids the need to make connections with ADLS from Databricks

Dataflow implementation:

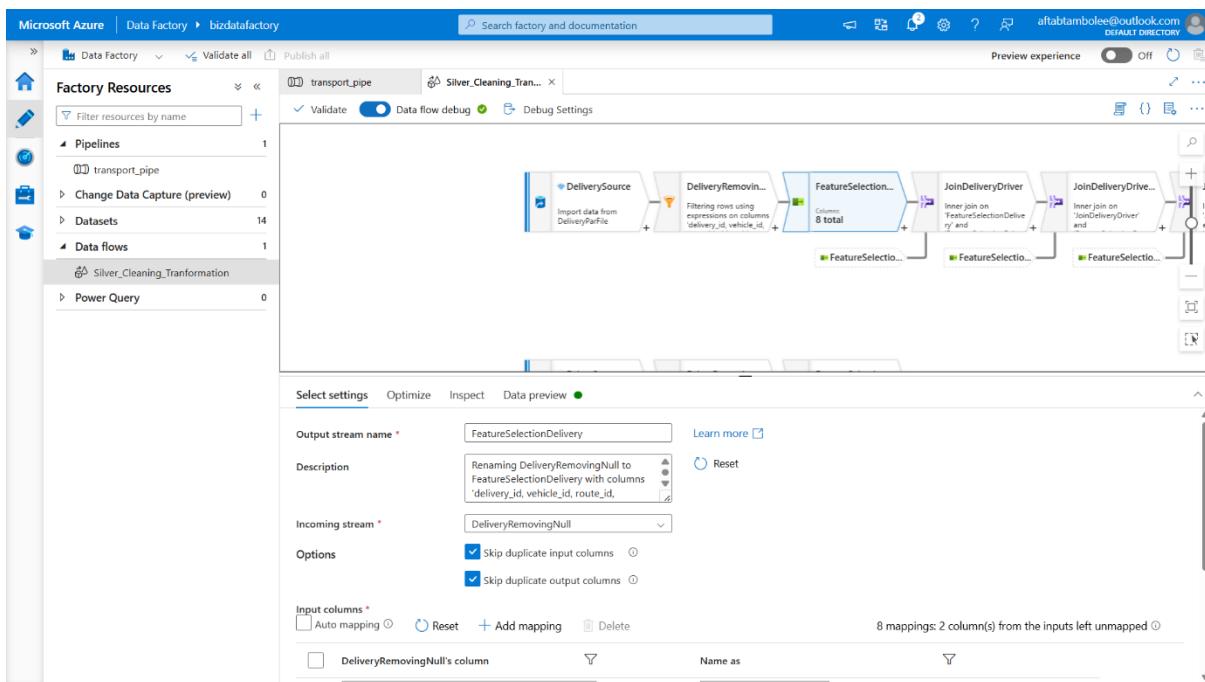
1. Four source activities were used to take the parquet datasets for each bronze file/table:



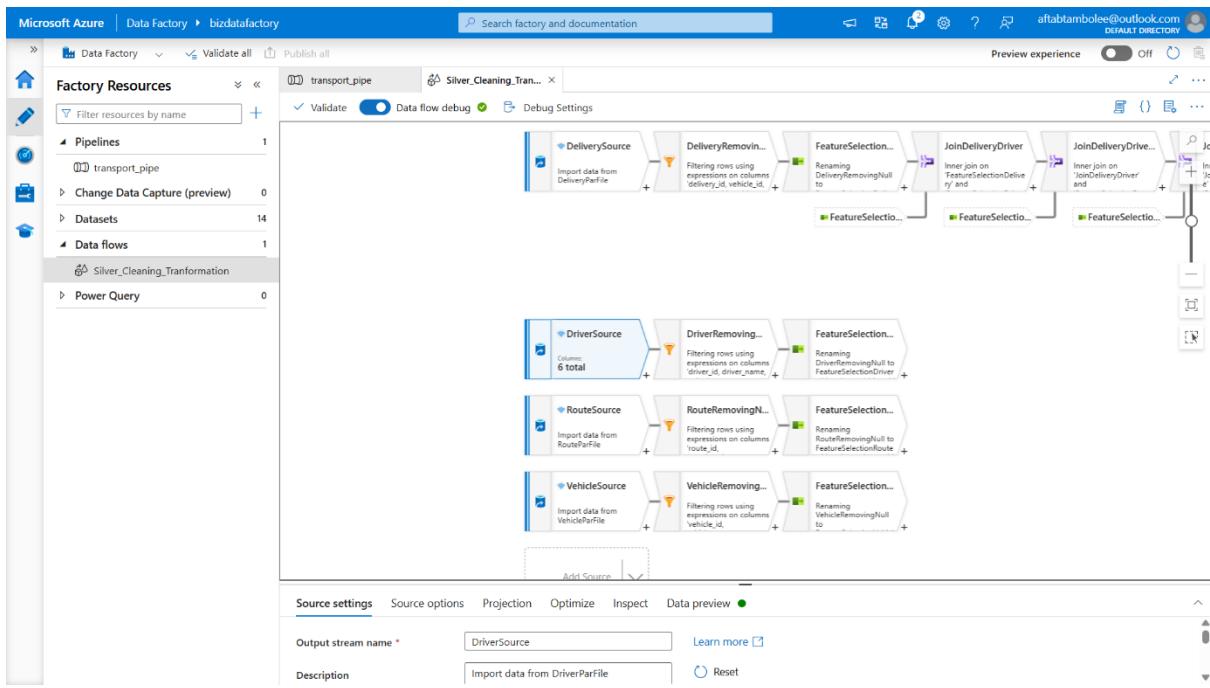
2. Filter activity was used to remove null values only from critical columns for each parquet dataset:



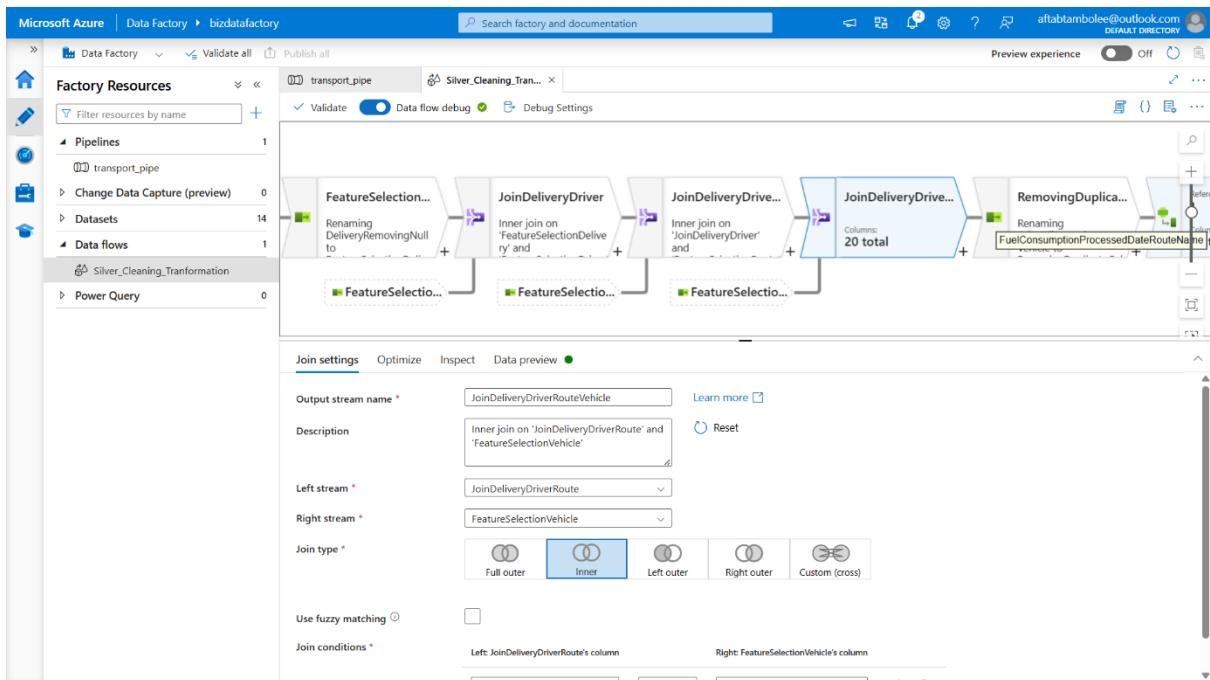
3. Select activity was used to remove unwanted columns (feature selection):



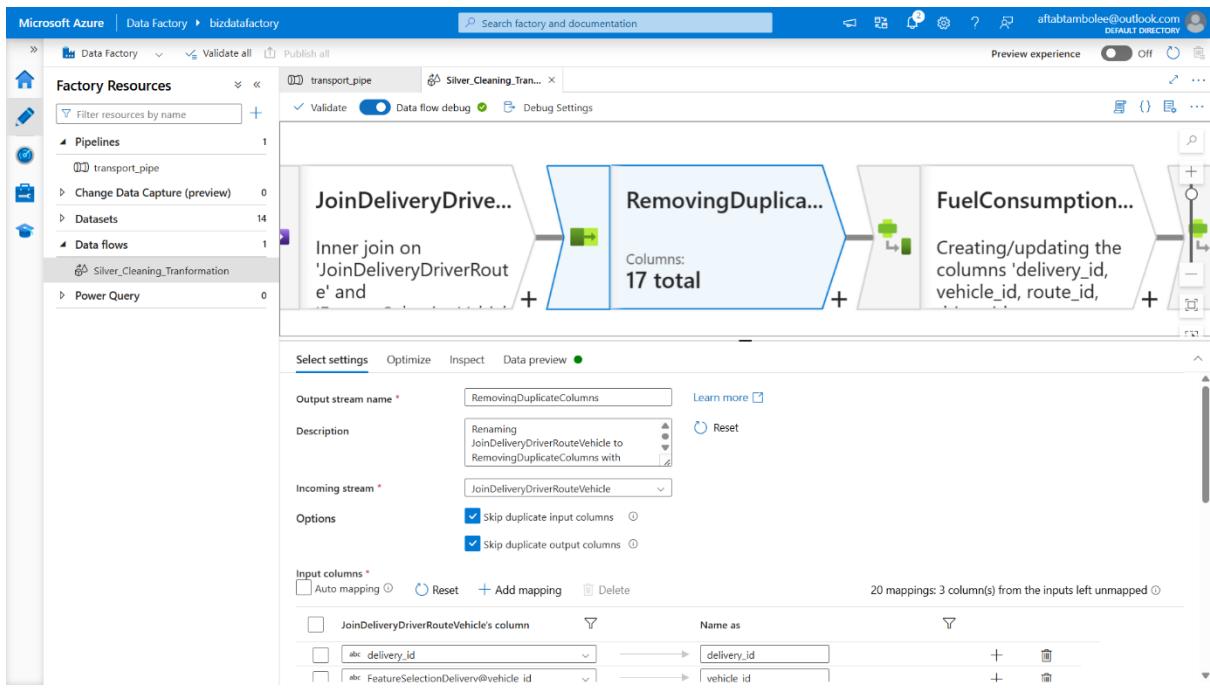
4. These three steps were performed for each of the bronze tables/files/datasets (delivery, driver, route, and vehicle):



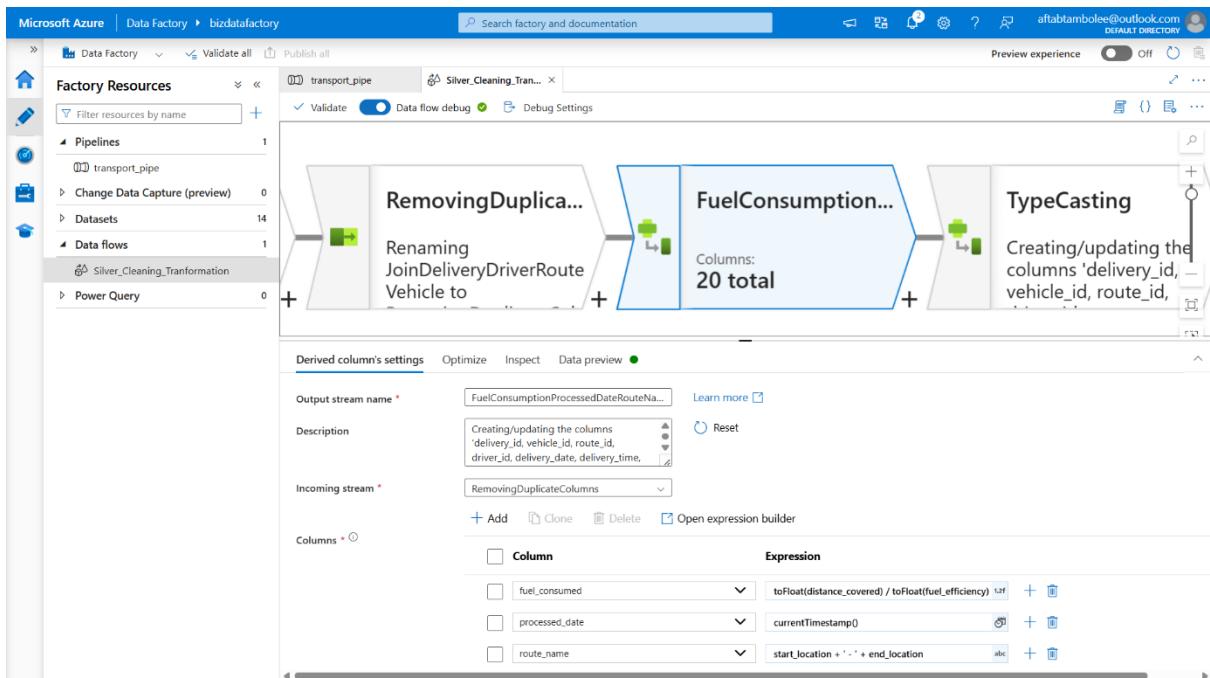
5. Join activity was applied to combine the four tables using primary keys and foreign keys with inner join:



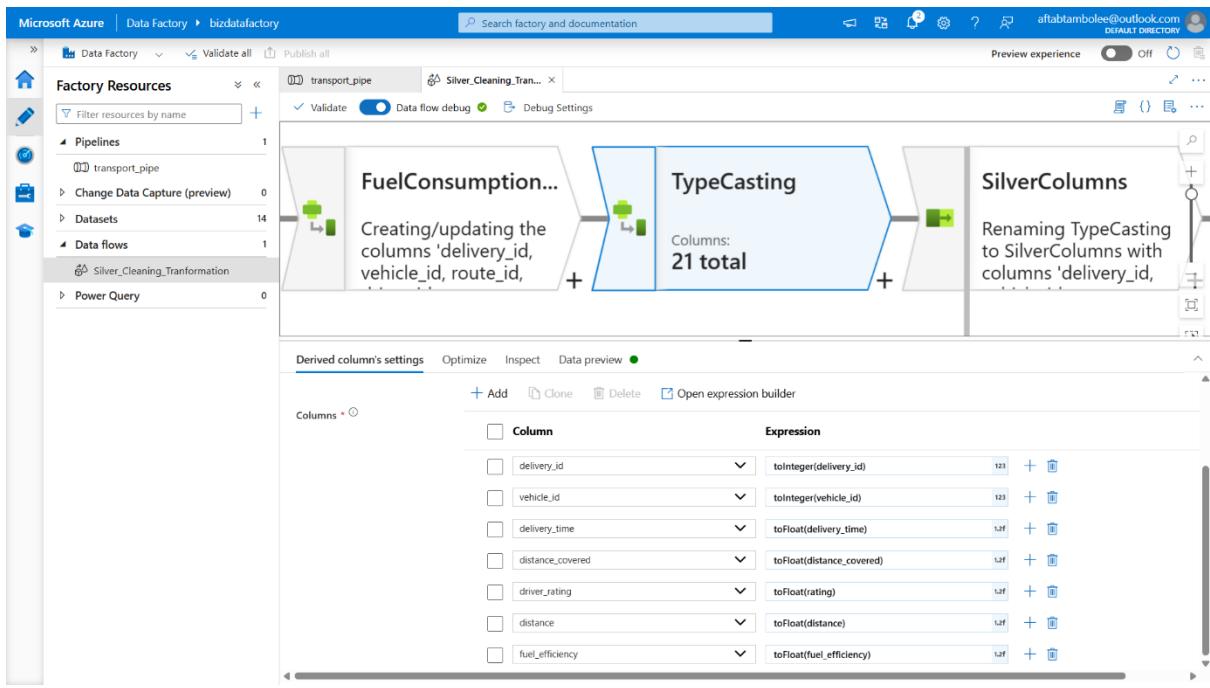
6. During joining, duplicate columns were removed using select activity:



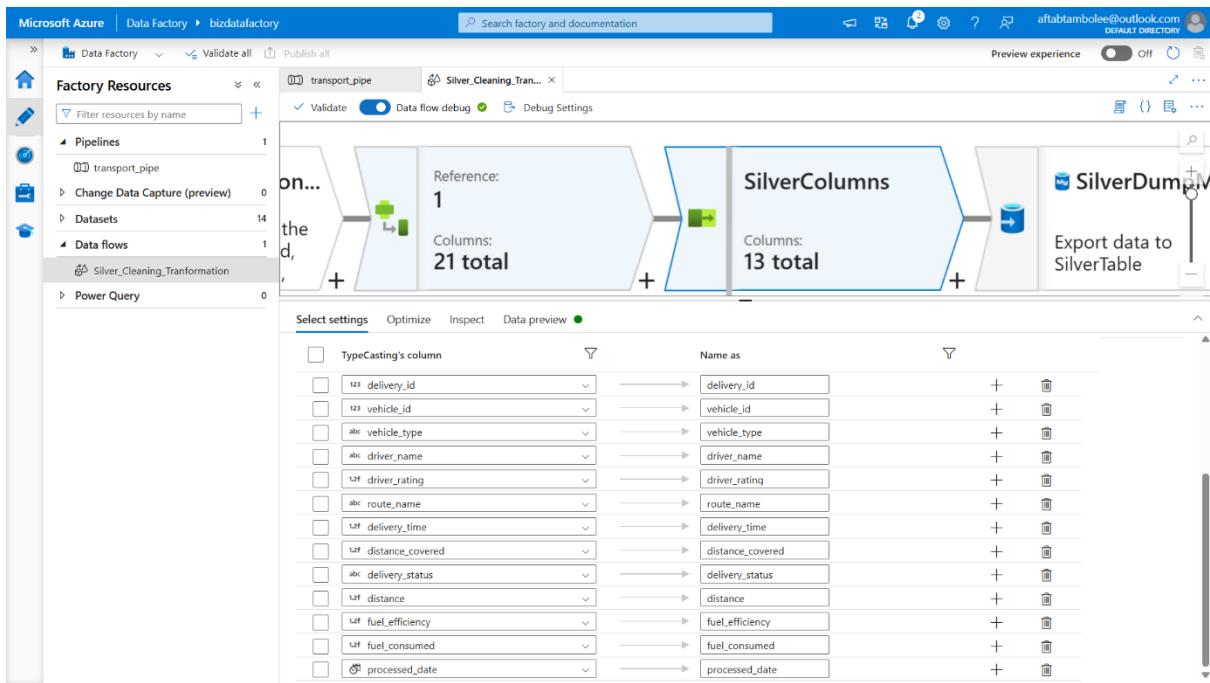
7. Additional columns were added for computing fuel\_consumed, processed\_date, and route\_name:



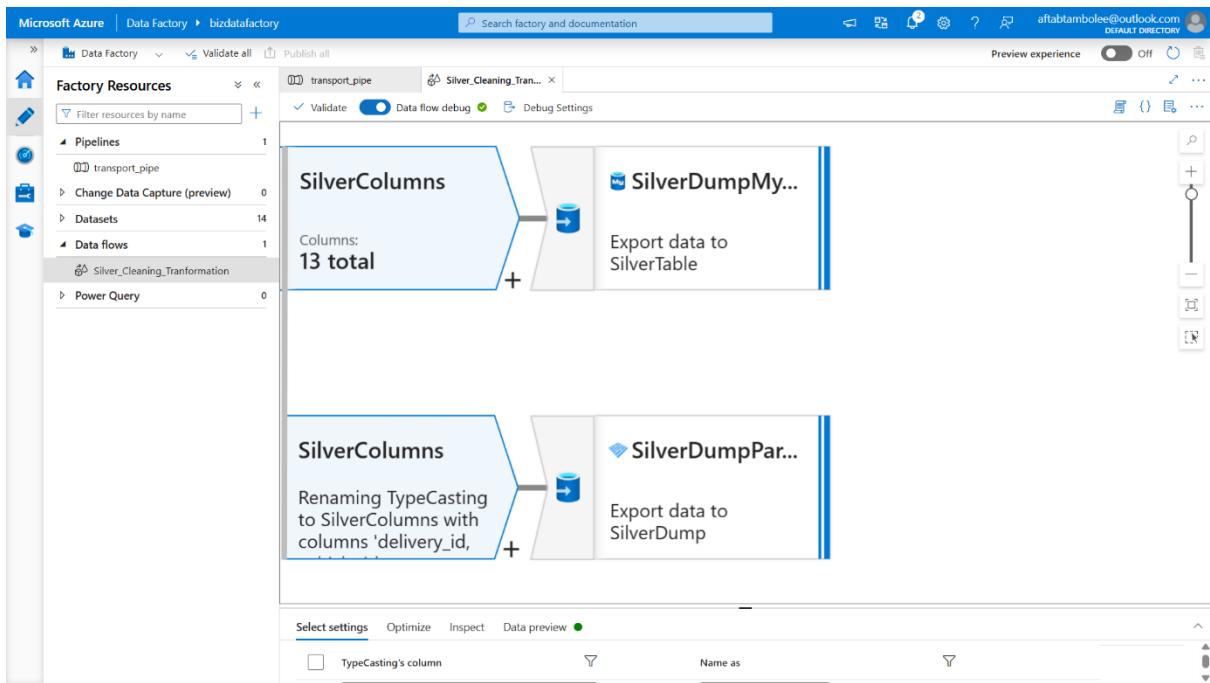
8. Type casting was performed on relevant columns to ensure correct data types for aggregation operations in the gold layer:



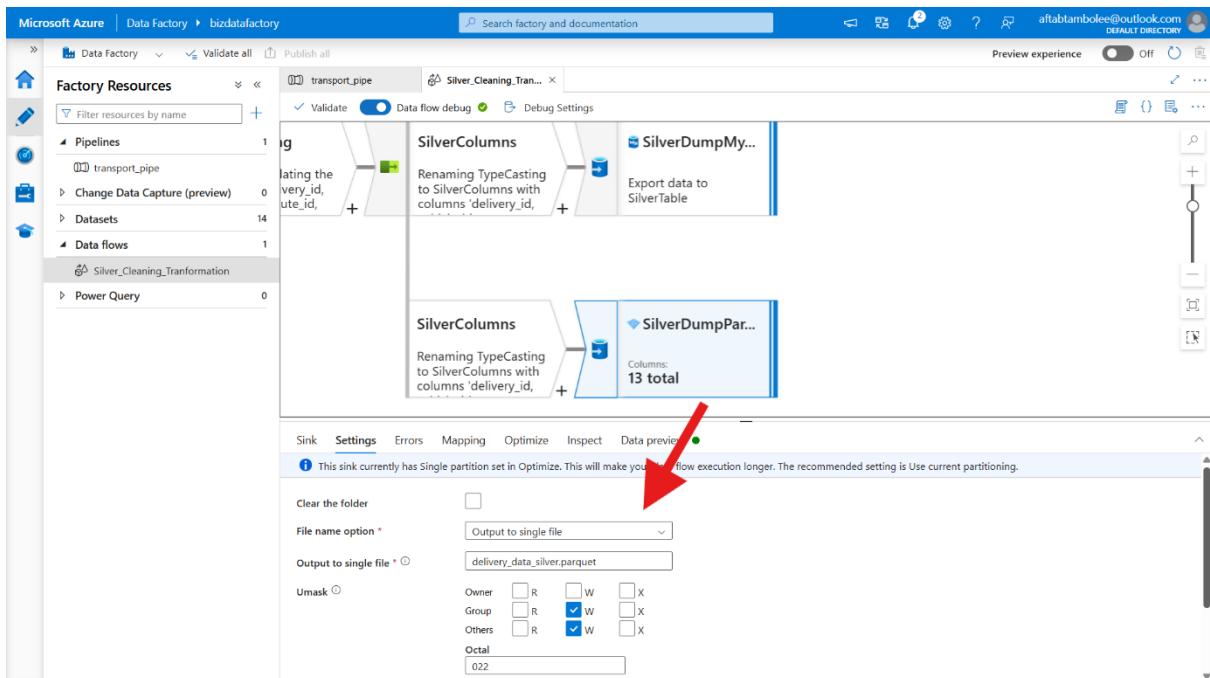
9. Select activity was used again to retain only important columns for the silver table:



10. Two sink activities were used to store the prepared silver table in MySQL database (bizserver) and Data Lake Gen 2 (bigenlake)'s silver container:



Custom naming for the parquet file is shown here:



The MySQL connection was established between Azure Database for MySQL server and MySQL Workbench. Two databases were created for storing silver table and gold table, with the silver\_db database/schema used to sink the dataflow's silver table:

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

aftabtambolee@outlook.com DEFAULT DIRECTORY

Home > bizserver

bizserver | Databases

Azure Database for MySQL flexible server

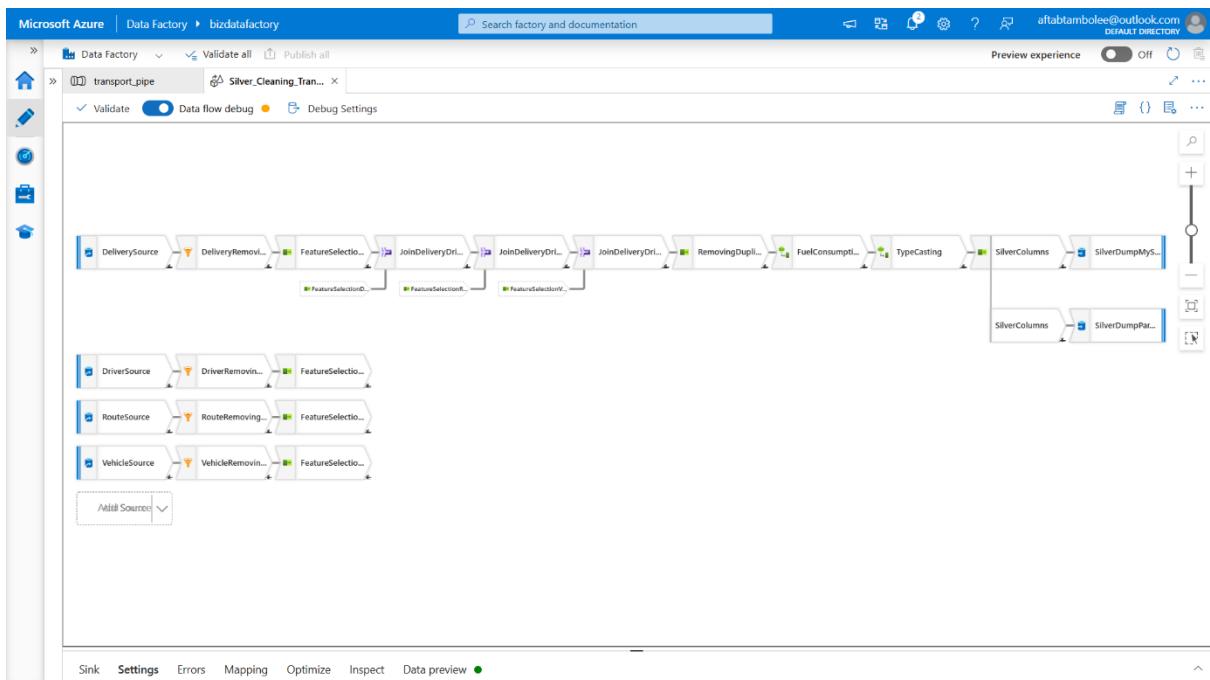
+ Add Delete Refresh Feedback

You can create, view and delete MySQL databases on this server. Note that you cannot delete any system databases such as mysql, sys, information\_schema, performance\_schema. You can connect to the databases using MySQL client tools.

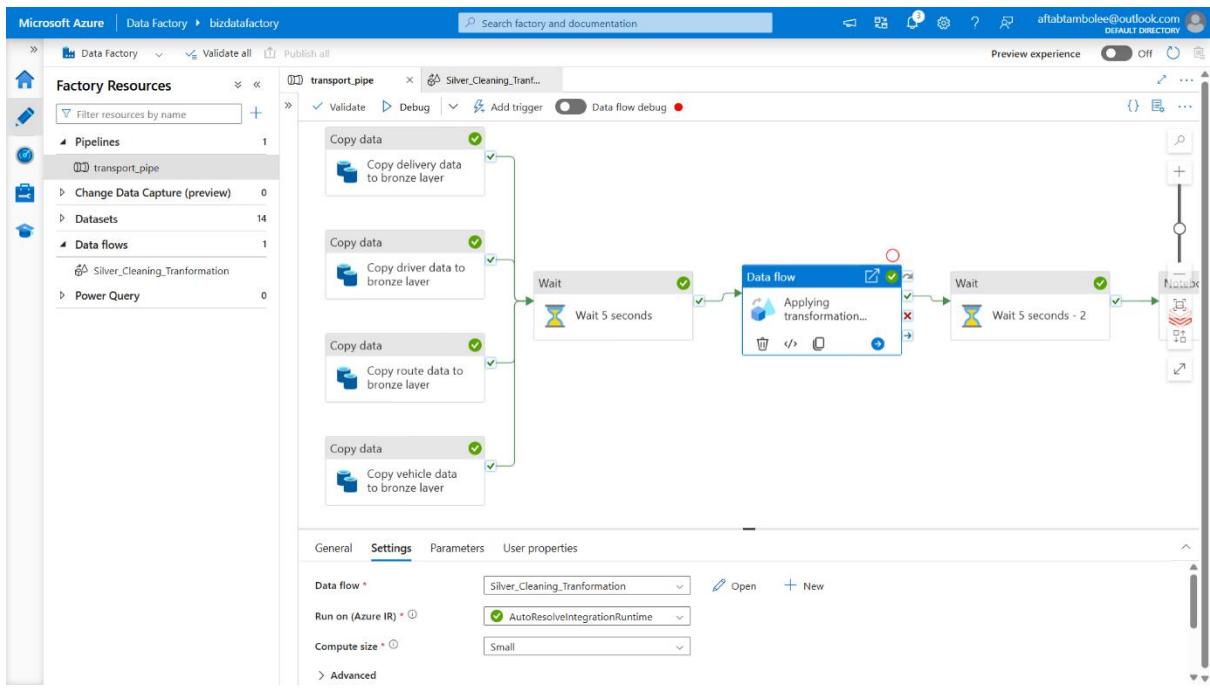
Name	Character set	Collation	Schema type
mysql	utf8mb4	utf8mb4_0900...	System
information_schema	utf8mb3	utf8mb3_gener...	System
performance_schema	utf8mb4	utf8mb4_0900...	System
sys	utf8mb4	utf8mb4_0900...	System
<b>silver_db</b>	utf8mb4	utf8mb4_0900...	User
<b>gold_db</b>	utf8mb4	utf8mb4_0900...	User

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Learning center Resource visualizer Settings Compute + storage Networking Databases Connect Server parameters Replication Maintenance High availability Backup and restore Advisor recommendations Locks Power Platform Security

The completed dataflow:



The dataflow was deployed in the data pipeline:



The final silver layer dataset/silver table:

Silver container (parquet):

```
[7]: silver_table = pd.read_parquet(r'D:\Downloads\delivery_data_silver.parquet')
display(silver_table)
```

	delivery_id	vehicle_id	vehicle_type	driver_name	driver_rating	route_name	delivery_time	distance_covered	delivery_status	distance	fuel_efficiency	fuel_cons
0	5	383	Truck	Joseph Wilson	4.2	New Rogerton - North Jennifer	12.00	84.860001	Failed	693.0	12.96	6.51
1	6	313	Van	Elizabeth Stout	1.4	North Susan - Belport	12.00	500.000000	Completed	372.0	13.48	37.05
2	52	405	Car	Mr. Patrick Adams III	1.4	North John - Robinbury	4.72	500.000000	Completed	121.0	9.86	50.71
3	93	311	Van	Mrs. Lisa Clark	4.9	South Tylerchester - Luisland	12.00	500.000000	Failed	470.0	8.71	57.41
4	130	100	Truck	Alexander Marsh	3.8	Joshuabury - Nicolestad	12.00	500.000000	Failed	898.0	8.05	62.11
...	...	...	...	...	...	...	...	...	...	...	...	...
58	911	465	Car	Ashley Kirk	3.0	Michellemouth - West Melindaborough	4.27	129.110001	Failed	643.0	5.10	25.31
59	931	221	Car	Anthony Alexander	4.2	Shawborough - North Kimberly	12.00	74.860001	Completed	658.0	10.91	6.81
60	934	61	Van	Paul Wu	3.1	Lake Amy - Lake Matthew	2.66	500.000000	Failed	313.0	6.75	74.01
61	952	233	Car	Angela Taylor	4.7	North Mark - East Alexanderhaven	12.00	500.000000	Completed	506.0	10.55	47.31
62	994	462	Truck	Paul Bautista	2.3	Lake Danielle - Johnsonhaven	1.52	148.619995	Completed	810.0	10.93	13.51

63 rows × 13 columns

MySQL table:

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' are visible. The 'Navigator' pane on the left lists 'Schemas' (silver\_db, gold\_db) and 'Tables' (gold\_db, silver\_db). The main area shows a query editor with the following code:

```

19    });
20
21
22 •  SELECT * FROM silver_db.delivery_data_silver;
23 •  TRUNCATE TABLE silver_db.delivery_data_silver;

```

Below the code is a 'Result Grid' showing the data being inserted into the gold table. The columns are: delivery\_id, vehicle\_id, vehicle\_type, driver\_name, driver\_rating, route\_name, delivery\_time, distance\_covered, delivery\_status, distance, fuel\_efficiency, fuel\_consumed, and processed\_date. The data consists of approximately 20 rows of delivery information.

## 5.4 Gold Layer Implementation

For the gold layer, the transformed silver table was used to apply aggregations and create a new gold table in both parquet format and MySQL. Azure Databricks service was utilized for this purpose.

A new cluster named "BizCluster" was created in Databricks:

The screenshot shows the Azure Databricks Compute page. The sidebar on the left includes 'New', 'Workspace', 'Recents', 'Catalog', 'Workflows', 'Compute' (selected), 'Marketplace', 'SQL', 'SQL Editor', 'Queries', 'Dashboards', 'Genie', 'Alerts', 'Query History', 'SQL Warehouses', 'Data Engineering', 'Job Runs', 'Data Ingestion', 'Pipelines', 'Machine Learning', 'Playground', 'Experiments', 'Features', 'Models', and 'Serving'. The main area shows the 'BizCluster' configuration:

- Compute**: Simple form: OFF
- Configuration** tab selected, showing:
  - Policy**: Personal Compute
  - Access mode**: Single user or group access (Dedicated (formerly: Single user) selected)
  - Performance** section: 16.3 ML (includes Apache Spark 3.5.2, Scala 2.12), Use Photon Acceleration (unchecked), Node type: Standard\_DS3\_v2, 14 GB Memory, 4 Cores, Terminate after 60 minutes of inactivity (checked)
  - Tags**: No custom tags, Automatically added tags
  - Advanced options**
- Summary** section:
  - 1 Driver, 14 GB Memory, 4 Cores
  - Runtime: 16.3-x-cpu-mi-scala2.12
  - Unity Catalog: Standard\_DS3\_v2
  - 0.75 DBU/h

This cluster was attached to the PySpark notebook:

```

transportation_gold Python 2 days ago (1s)
File Edit View Run Help Last edit was 2 days ago

from pyspark.sql import SparkSession

# Create Spark session
spark = SparkSession.builder.appName("GoldLayerAggregation").getOrCreate()

# Azure MySQL connection properties
mysql_url = "jdbc:mysql://bizzserver.mysql.database.azure.com:3306/silver_db"
mysql_properties = {
    "user": "afstab",
    "password": "Amyra@0457", # Replace with actual password
    "driver": "com.mysql.cj.jdbc.Driver",
    "sslMode": "REQUIRED"
}

# Load Silver Layer Data from MySQL
silver_df = spark.read.jdbc(
    url=mysql_url,
    table="delivery_data_silver",
    properties=mysql_properties
)

# Show sample data
display(silver_df)

```

(1) Spark Jobs

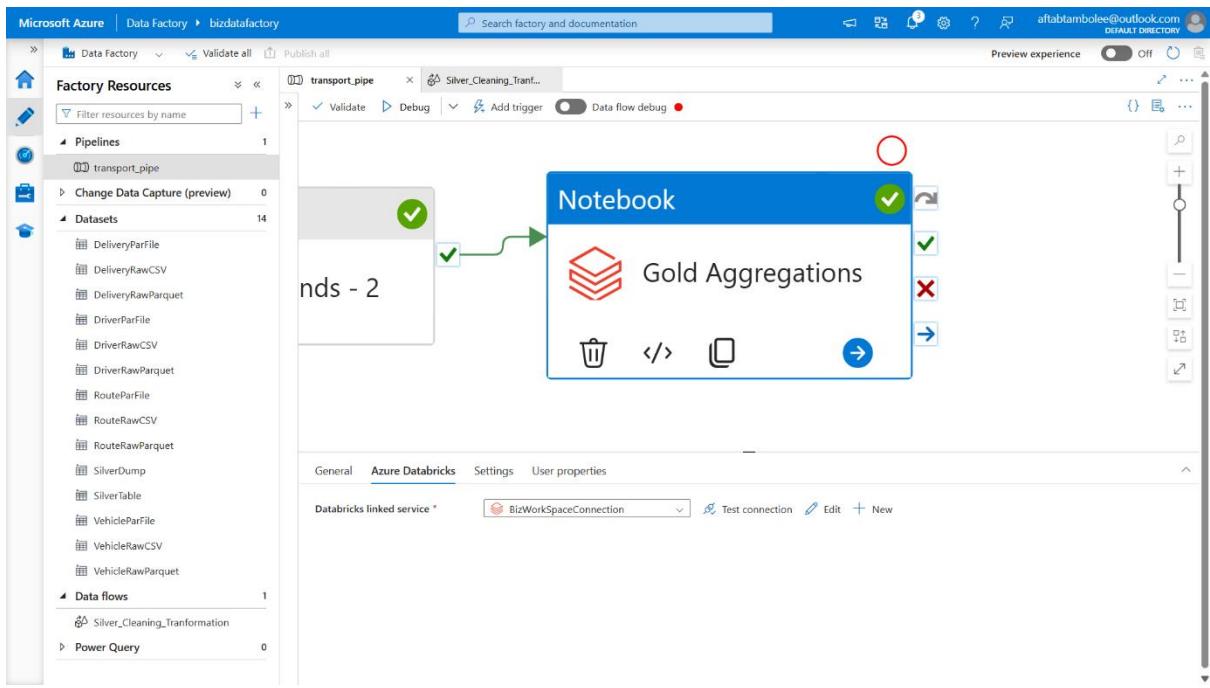
silver\_df: pyspark.sql.dataframe.DataFrame = [delivery\_id: integer, vehicle\_id: integer ... 11 more fields]

Table	+					
<code>i<sup>2</sup>s delivery_id</code>	<code>i<sup>2</sup>s vehicle_id</code>	<code>a<sup>0</sup>c vehicle_type</code>	<code>a<sup>0</sup>c driver_name</code>	<code>1.2 driver_rating</code>	<code>a<sup>0</sup>c route_name</code>	<code>1.2 delivery_time</code>

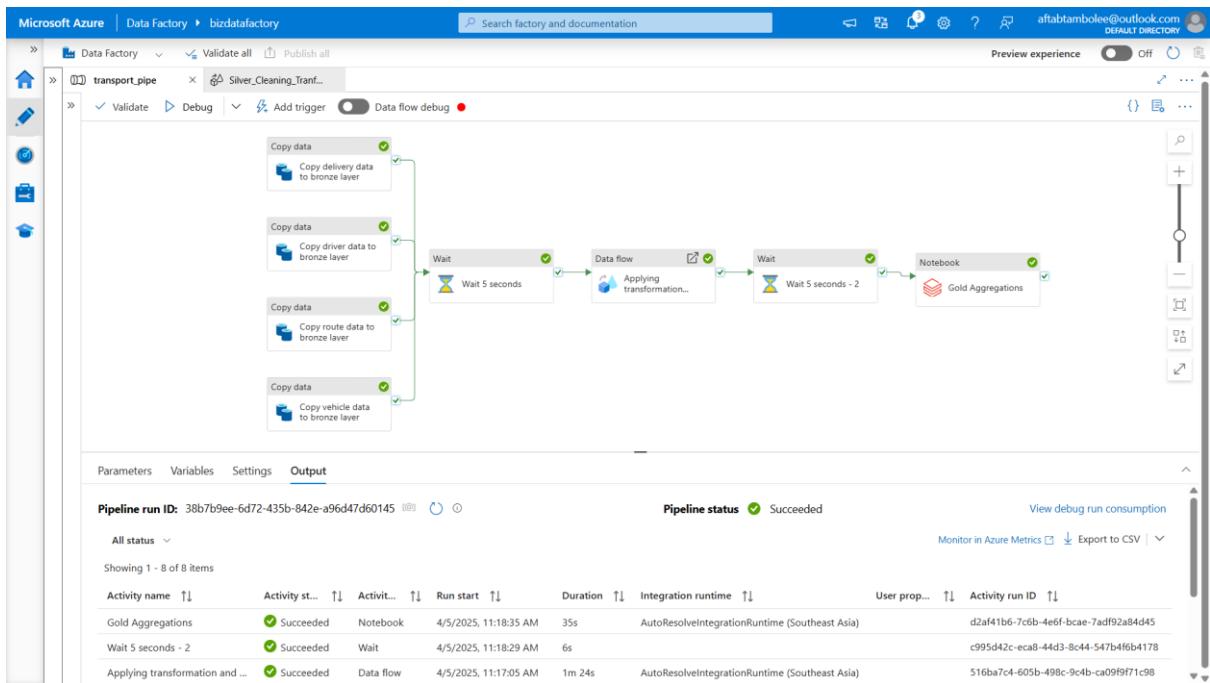
The notebook implementation included:

1. Creating a Spark session named "GoldLayerAggregation"
2. Establishing an Azure MySQL connection to import the silver table
3. Loading the silver table
4. Performing three aggregated operations to calculate:
  - Route optimization Analysis
  - Fleet performance
  - Driver performance
5. Joining the above three tables to create a single gold table
6. Arranging columns in the required sequential format
7. Connecting to MySQL's gold\_db database to transfer the gold table
8. Connecting to Azure Lake Gen 2's bizlakegen storage to transfer the table to gold container in parquet format along with the logging file

The notebook was deployed in the data pipeline:



The complete data pipeline in Azure Data Factory:



The gold table in MySQL (gold\_db.transportation\_gold):

The screenshot shows the MySQL Workbench interface with a connection to 'Azure bizerver Connection'. In the 'Navigator' pane, there are two databases: 'gold\_db' and 'silver\_db'. The 'gold\_db' schema contains a single table named 'transportation\_gold'. A query is being run against this table:

```

16
17
18
19   SELECT * FROM gold_db.transportation_gold;
20 •  SELECT COUNT(*) AS total_rows FROM gold_db.transportation_gold;

```

The results grid displays the following data:

route_name	total_deliveries	avg_delivery_time	avg_fuel_consumed	vehicle_id	total_distance	fuel_efficiency	driver_name	total_deliveries_by_driver	driver_rating	delivery_status	report_date
West Kimberly - Jennerburgh	1	12	62.5	458	500	8	Dawn Hardy	2	3.3	Failed	2025-04-05
Jayport - Wallerbury	1	1.19	22.999	255	159.58	7.03	Sean Bernard	1	4.6	Completed	2025-04-05
Erdbury - East Jesse	1	4.2	87.193	472	500	5.7	Rebecca Flores	1	4.5	Completed	2025-04-05
Gomezport - Reveshaven	1	2.72	11.6241	183	111.01	9.55	Jason Ayers MD	1	3.4	Failed	2025-04-05
Coleside - South Jessica	1	12	55.371	44	500	9.03	Mary Watson	1	1.9	Completed	2025-04-05
Barajaschester - East Lindsey	1	4.16	68.9655	417	500	7.25	Raven Jenkins	1	1.8	Completed	2025-04-05
Levishire - Vollport	1	12	11.4116	409	65.16	5.71	Grey Wood	2	2	Failed	2025-04-05
Dayleton - Cartermouth	1	2.08	52.8541	22	500	9.46	Sean Ortiz	1	1.7	Completed	2025-04-05
North Mark - East Alexanderhaven	1	12	47.3934	433	500	10.55	Angela Taylor	1	4.7	Completed	2025-04-05
Southernport - Port	1	3.77	30.932	496	500	7.5	Kathy Ford	1	3.1	Completed	2025-04-05
Jaredburgh - Port Donemouth	1	12	24.5543	416	189.72	7.79	Matthew Drake	1	3.5	Completed	2025-04-05
Christopherburgh - West Alcia	1	1.93	70.7214	435	500	7.07	Heather Bryant	2	3.1	Failed	2025-04-05
South Shanaland - Kevinshaw	1	1.89	70.1362	146	500	7.13	Michael Bell	2	1	Completed	2025-04-05
Lake Travissire - West Daniellene...	1	12	13.6471	250	136.88	10.03	Sheila Hayes	1	3.4	Completed	2025-04-05
Johnsonfort - East Rebecca	1	12	13.1742	258	145.18	11.02	Rhonda Nelson...	1	3.7	Failed	2025-04-05
Vazquezton - Port Molivien	1	1.41	11.9072	309	173.13	14.54	Ashley Lucas	1	4.1	Failed	2025-04-05
New Rogerton - North Jennifer	1	12	6.54784	383	84.86	12.96	Joseph Wilson	1	4.2	Failed	2025-04-05
Lake Amy - Lake Matthew	1	2.66	74.0741	61	500	6.75	Paul Wu	1	3.1	Failed	2025-04-05
Lake Danielle - Johnsonhaven	1	1.52	13.5974	462	148.62	10.93	Paul Bustista	1	2.3	Completed	2025-04-05
Port Brendatown - Williamsouth	1	4.31	61.1247	231	500	8.18	Robert Callahan	1	4.8	Failed	2025-04-05
Lake John - Marinburgh	1	3.24	9.18951	9	131.41	14.3	Justin Wells	1	4.4	Completed	2025-04-05
New Micheland - North Monica	1	3.04	12.2075	35	129.4	10.6	Dawn Hardy	2	3.3	Completed	2025-04-05
Lisland - Janshire	1	4.61	48.5909	290	500	10.29	Heather Bryant	2	3.1	Failed	2025-04-05
Walkerview - Port Dasy	1	12	6.51561	173	104.875	14.41	Sara Pham	1	1.2	Completed	2025-04-05
Courtneyburgh - Zoeside	1	4.61	8.04025	173	104.875	14.41	Latasha Le	1	1.6	Failed	2025-04-05

Execution Plan: Read Only

The gold container in parquet format with logging file:

The screenshot shows the Microsoft Azure Storage Explorer interface. A blob named 'transportation\_gold' is selected. The 'Overview' tab is active, showing the authentication method as 'Access key (Switch to Microsoft Entra user account)' and the location as 'gold / transportation\_gold'. The blob type is 'Block blob'. The table below lists the blobs in the container:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[-]						***
_committed_2377593780441041011	5/4/2025, 11:19:06 am	Hot (Inferred)		Block blob	123 B	Available
_started_2377593780441041011	5/4/2025, 11:19:04 am	Hot (Inferred)		Block blob	0 B	Available
_SUCCESS	5/4/2025, 11:19:06 am	Hot (Inferred)		Block blob	0 B	Available
part-00000-tid-2377593780441041011-d0e66051...	5/4/2025, 11:19:06 am	Hot (Inferred)		Block blob	8.07 KIB	Available

The gold container's parquet file:

jupyter reading\_bronze Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python [conda env:base] \* ○ ≡

```
[11]: gold_table = pd.read_parquet(r"D:\Downloads\part-00000-tid-2377593780441041011-d0e66051-158c-4ec2-a272-5cb24d37dd53-18-1-c000.snappy.parquet")
display(gold_table)
```

	route_name	total_deliveries	avg_delivery_time	avg_fuel_consumed	vehicle_id	total_distance	fuel_efficiency	driver_name	total_deliveries_by_driver	driver_rating
0	Barajaschester - East Lindsey	1	4.16	68.96550	417	500.000	7.25	Raven Jenkins	1	1
1	Bergview - Laurenside	1	12.00	74.40480	433	500.000	6.72	Allen Robinson	1	4
2	Brandonport - Shannonview	1	12.00	43.36510	260	500.000	11.53	Natalie White	1	4
3	Buckborough - Randystad	1	12.00	12.76040	338	125.690	9.85	Amy Garcia	1	4
4	Christopherburgh - West Alicia	1	1.93	70.72140	435	500.000	7.07	Heather Bryant	2	3
...	...	...	...	...	...	...	...	...	...	...
58	Vegamouth - Port Dawnshire	1	4.32	65.96310	247	500.000	7.58	Christopher Hahn	1	4
59	Walkerville - Port Daisy	1	12.00	6.51561	173	104.875	14.41	Sara Pham	1	1
60	West Daniel - Doughertymouth	1	12.00	35.11240	450	500.000	14.24	Jennifer Boyle	1	3
61	West Kimberly - Jenniferburgh	1	12.00	62.50000	458	500.000	8.00	Dawn Hardy	2	3
62	West Robert - South Andrew	1	3.38	14.02190	302	147.510	10.52	Albert Lynch	1	2

63 rows × 12 columns

At this point, the medallion architecture implementation is complete, with all steps performed: Raw data → Bronze layer → Silver layer → Gold layer

## 6. Power BI Dashboard

The gold table was then used for reporting and creating a Power BI dashboard. A connection file was downloaded from the Azure Database for MySQL server (bizserver) that automatically creates the connection between MySQL gold\_db table and Power BI:

Azure Portal Screenshot: Databases Page for bizserver

The screenshot shows the Microsoft Azure portal interface. The left sidebar navigation bar is visible, with 'Databases' selected under the 'Compute + storage' section. The main content area displays a list of databases for the 'bizserver' MySQL server. The 'gold\_db' database is selected, and a red arrow points to the 'Open in Power BI' button in the table row.

Name	Character set	Collation	Schema type
mysql	utf8mb4	utf8mb4_0900...	System
information_schema	utf8mb3	utf8mb3_gener...	System
performance_schema	utf8mb4	utf8mb4_0900...	System
sys	utf8mb4	utf8mb4_0900...	System
silver_db	utf8mb4	utf8mb4_0900...	User
gold_db	utf8mb4	utf8mb4_0900...	User

### 6.1 Key Performance Indicators

Five KPIs were created as per documentation:

1. **Total Deliveries:** Sum of total deliveries per route/vehicle/driver
2. **Average Delivery Time:** Average delivery time per route/vehicle/driver
3. **Fuel Efficiency:** Average fuel efficiency per vehicle
4. **Driver Performance:** Average driver rating
5. **Delivery Status:** Completed vs Failed

### 6.2 Visualizations

The dashboard was created with the following visualizations:

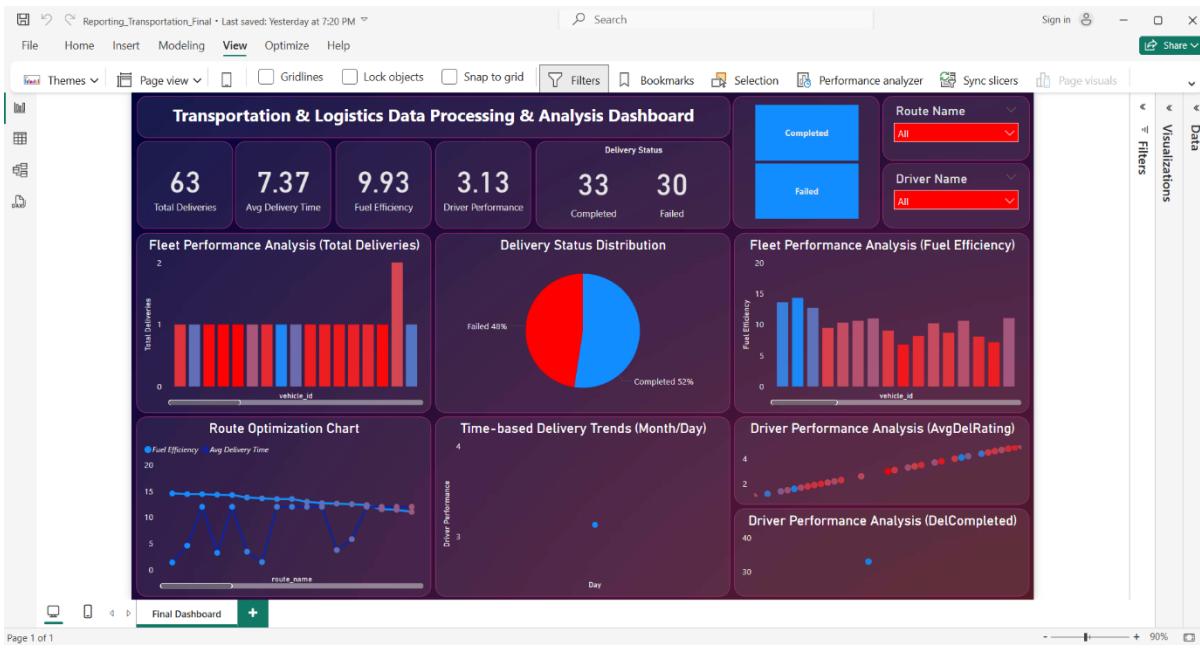
1. **Route Optimization Chart:** Line chart showing average delivery times and fuel consumption per route
2. **Fleet Performance Analysis:** Bar chart visualizing total deliveries per vehicle and fuel efficiency
3. **Driver Performance Analysis:** Scatter plot showing driver performance based on deliveries completed and average ratings
4. **Delivery Status Distribution:** Pie chart showing the distribution of completed and failed deliveries across routes
5. **Time-based Delivery Trends:** Line chart showing monthly/weekly delivery performance trends

The dashboard includes slicers and filters for:

- Delivery status (completed/failed)

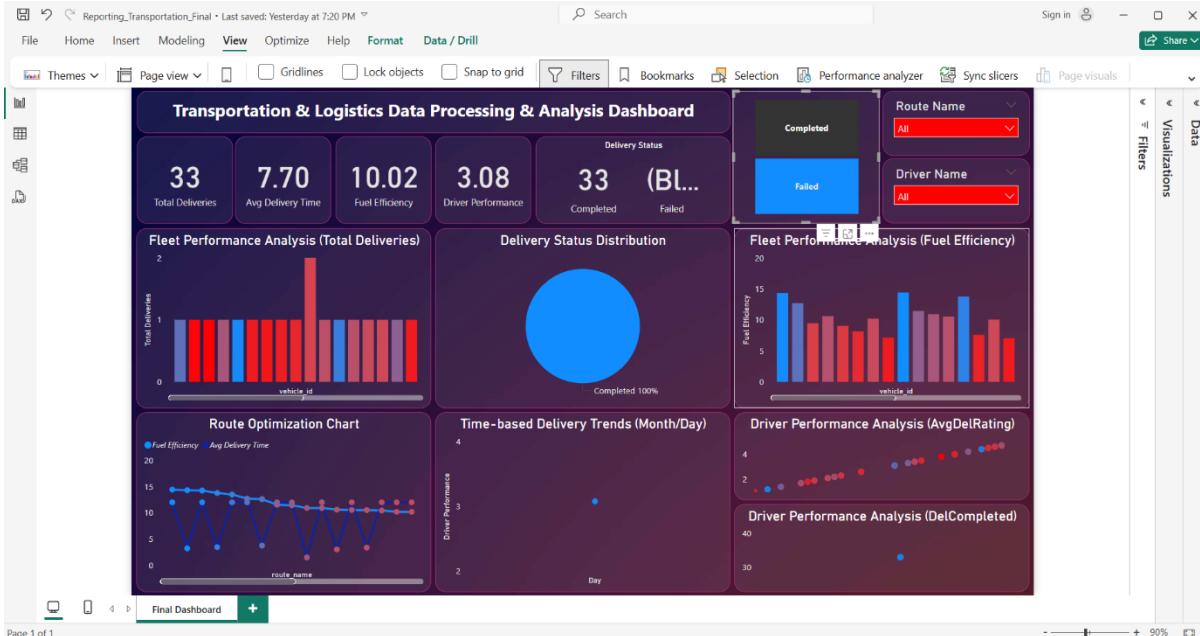
- Route name
- Driver name

The final dashboard:

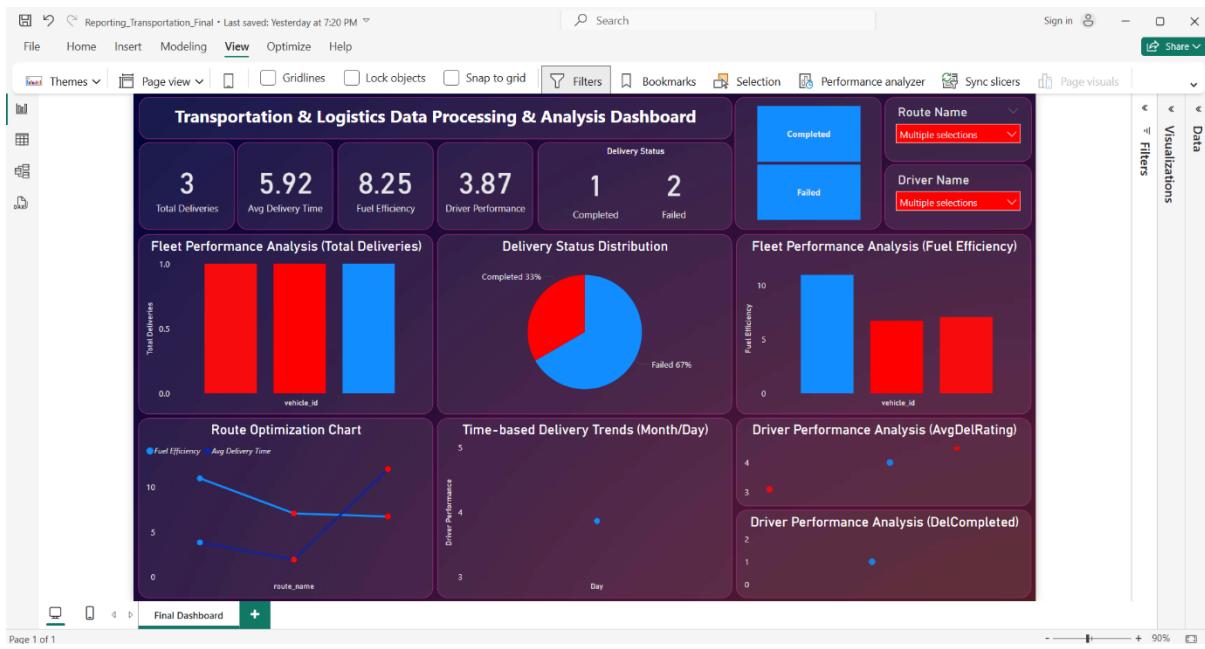


### 6.3 Examples

Example 1: Dashboard filtered by deliveries completed:



Example 2: Dashboard filtered by selected route name and driver name:



## 7. Conclusion

This project successfully implements a complete data pipeline following the medallion architecture on Microsoft Azure. The pipeline automatically processes transportation and logistics data from raw format to insightful visualizations, requiring minimal manual intervention. The solution demonstrates the effective use of various Azure services to create a scalable, automated, and reliable data processing system that provides valuable business insights through Power BI dashboards.