Transportation & Logistics Data Processing Project – Documentation

📌 Project Summary

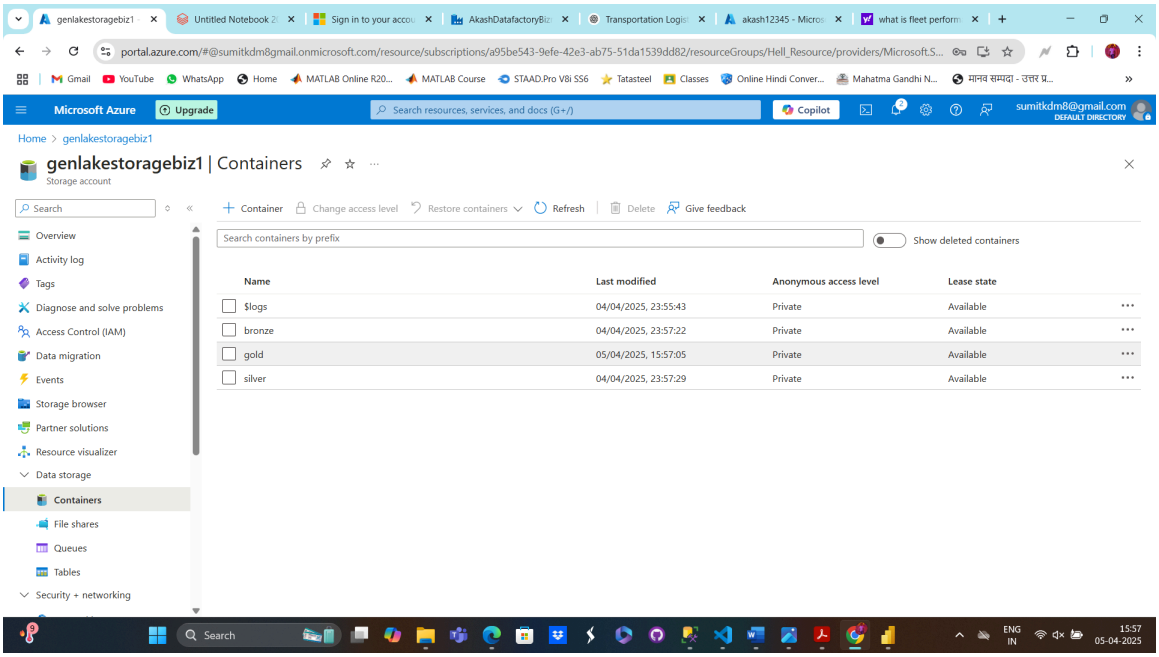
This project focuses on analyzing data from the transportation and logistics domain to generate meaningful insights related to deliveries, routes, vehicles, and drivers. The overall objective is to:
- Improve route efficiency
- Monitor fleet and driver performance
- Track and reduce fuel consumption
- Visualize operations through dashboards

To achieve this, we use a structured data pipeline built using PySpark for data processing, MySQL for storing cleaned and aggregated data, and Power BI for visual reporting. The pipeline follows the Medallion Architecture model consisting of Bronze (raw data), Silver (cleaned and enriched), and Gold (aggregated for reporting) layers.

🧰 Tools & Technologies Used

Tool/Technology	Purpose/Usage
PySpark	Used for reading, cleaning, transforming, and enriching large CSV datasets.
Jupyter Notebook	To write and run PySpark scripts interactively.
MySQL	Stores Silver (clean) and Gold (aggregated) layer data for further reporting.
Power BI	Connects to MySQL Gold layer to build dashboards and visualizations.
CSV Files	Raw input data files containing delivery, route, vehicle, and driver information.

🏗️ Project Architecture – Medallion Approach



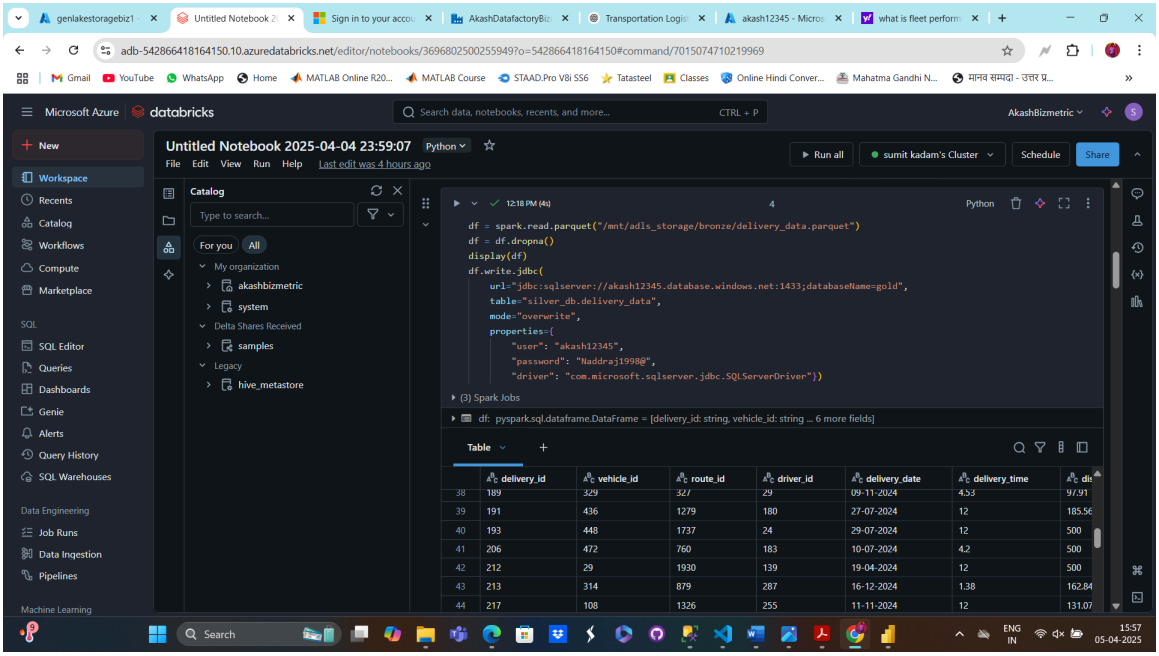
1. Bronze Layer – Raw Data Ingestion

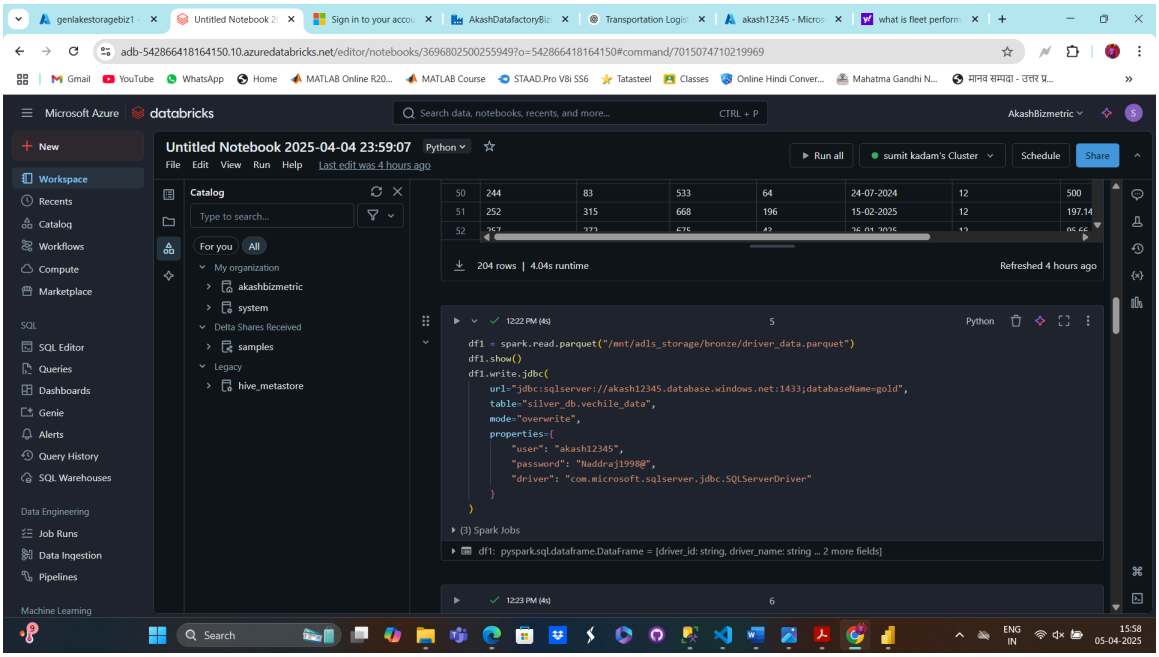
The Bronze layer is the foundation where raw data is ingested. CSV files from multiple sources are loaded and converted to a uniform Parquet format. No major transformation is applied at this layer. Metadata fields such as ingestion date and source filename are added for auditing.

Expected files:
- delivery_data.csv
- vehicle_data.csv
- route_data.csv
- driver_data.csv

2. Silver Layer – Cleaned & Enriched Data

In this layer, the raw data is cleaned by removing records with nulls, invalid formats, or duplicates. The tables are then enriched by joining related datasets such as vehicle information, route data, and driver details


Untitled Notebook 2025-04-04 23:59:07
df = spark.read.parquet("/mnt/adls_storage/bronze/delivery_data.parquet")
df = df.dropna()
display(df)
df.write.jdbc(
url="jdbc:sqlserver://akash12345.database.windows.net:1433;databaseName=gold",
table="silver_db.delivery_data",
mode="overwrite",
properties={
"user": "akash12345",
"password": "Naddraj1998@",
"driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"})
(3) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [delivery_id: string, vehicle_id: string ... 6 more fields]
Table
delivery_id vehicle_id route_id driver_id delivery_date delivery_time


Untitled Notebook 2025-04-04 23:59:07
204 rows | 4.04s runtime
Refreshed 4 hours ago
df1 = spark.read.parquet("/mnt/adls_storage/bronze/driver_data.parquet")
df1.show()
df1.write.jdbc(
url="jdbc:sqlserver://akash12345.database.windows.net:1433;databaseName=gold",
table="silver_db.vechile_data",
mode="overwrite",
properties={
"user": "akash12345",
"password": "Naddraj1998@",
"driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
}
)
(3) Spark Jobs
df1: pyspark.sql.dataframe.DataFrame = [driver_id: string, driver_name: string ... 2 more fields]

adb-5428664181641 50.10.azuredatabricks.net/editor/notebooks/3696802500255949?o=5428664181641 50#command/7015074710219969

Untitled Notebook 2025-04-04 23:59:07 Python

File Edit View Run Help Last edit was 4 hours ago

Run all | sumit kadam's Cluster | Schedule | Share

df1: pyspark.sql.dataframe.DataFrame = [driver_id: string, driver_name: string ... 2 more fields]

12:23 PM (4s) 6

```
df2 = spark.read.parquet("/mnt/adls_storage/bronze/route_data.parquet")
df2.show()
df2.write.jdbc(
    url="jdbc:sqlserver://akash12345.database.windows.net:1433;databaseName=gold",
    table="silver_db.route_data",
    mode="overwrite",
    properties={
        "user": "akash12345",
        "password": "Naddraj1998@",
        "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
    }
)
```

(3) Spark Jobs

df2: pyspark.sql.dataframe.DataFrame = [route_id: string, start_location: string ... 2 more fields]

12:23 PM (4s) 7

```
df3 = spark.read.parquet("/mnt/adls_storage/bronze/vehicle_data.parquet")
df3.show()
df3.write.jdbc(
    url="jdbc:sqlserver://akash12345.database.windows.net:1433;databaseName=gold",
```

adb-5428664181641 50.10.azuredatabricks.net/editor/notebooks/3696802500255949?o=5428664181641 50#command/7015074710219969

Untitled Notebook 2025-04-04 23:59:07 Python

File Edit View Run Help Last edit was 4 hours ago

Run all | sumit kadam's Cluster | Schedule | Share

12:23 PM (4s) 7

(3) Spark Jobs

df3: pyspark.sql.dataframe.DataFrame = [vehicle_id: string, vehicle_type: string ... 2 more fields]

12:23 PM (1s) 8

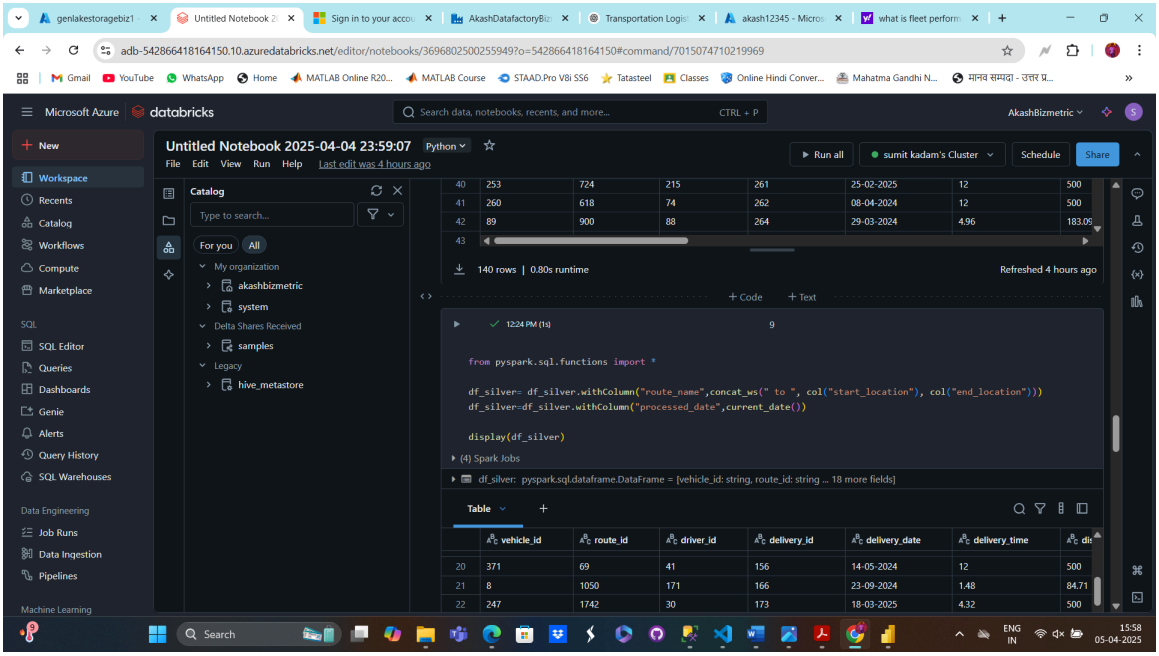
```
df_silver=df.join(df1, "driver_id", "inner") \
    .join(df2, "route_id", "inner") \
    .join(df3, "vehicle_id", "inner") \
    .withColumn("fuel_consumed", col("distance_covered") / col("fuel_efficiency"))

display(df_silver)
```

(4) Spark Jobs

df_silver: pyspark.sql.dataframe.DataFrame = [vehicle_id: string, route_id: string ... 16 more fields]

Table

	vehicle_id	route_id	driver_id	delivery_id	delivery_date	delivery_time	dis
10	360	1297	43	78	14-10-2024	2.59	187.71
11	192	960	173	84	21-02-2025	12	107.4
12	311	1578	193	93	12-11-2024	12	500
13	435	127	216	107	27-03-2024	5.01	500
14	218	818	254	123	01-08-2024	12	500
15	30	1237	298	126	05-10-2024	12	77.09
16	100	757	225	130	02-03-2025	12	500


Untitled Notebook 2025-04-04 23:59:07
Python
File Edit View Run Help Last edit was 4 hours ago
Run all
sumit kadam's Cluster
Schedule
Share
Catalog
Workspace
Recents
Catalog
Workflows
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Pipelines
Machine Learning
My organization
akashbizmetric
system
Delta Shares Received
samples
Legacy
hive_metastore
140 rows | 0.80s runtime
Refreshed 4 hours ago
+ Code
+ Text
12:24 PM (1s)
from pyspark.sql.functions import *
df_silver= df_silver.withColumn("route_name",concat_ws(" to ", col("start_location"), col("end_location")))
df_silver=df_silver.withColumn("processed_date",current_date())
display(df_silver)
(4) Spark Jobs
df_silver: pyspark.sql.dataframe.DataFrame = [vehicle_id: string, route_id: string ... 18 more fields]
Table
vehicle_id
route_id
driver_id
delivery_id
delivery_date
delivery_time

adb-542866418164150.10.azuredatabricks.net/editor/notebooks/3696802500255949?o=542866418164150#command/7015074710219969

Microsoft Azure | databricks

Untitled Notebook 2025-04-04 23:59:07 Python

File Edit View Run Help Last edit was 4 hours ago

Run all | sumit kadam's Cluster | Schedule | Share

12:24 PM (1s) 10

```
df_silver=df_silver.select("delivery_id","vehicle_type","driver_name","route_name","delivery_time",
"distance_covered","delivery_status","fuel_consumed","processed_date")
display(df_silver)
```

(4) Spark Jobs

df_silver: pyspark.sql.dataframe.DataFrame = [delivery_id: string, vehicle_type: string ... 7 more fields]

Table

	delivery_id	vehicle_type	driver_name	route_name	delivery_time
1	5	Truck	Joseph Wilson	New Rogerton to North Jennifer	12
2	6	Van	Elizabeth Stout	North Susan to Bellport	12
3	9	Van	Mary Watson	Norriston to East Karenville	12
4	10	null	Cheryl Davis	Garrettstad to Sheilaport	1.39
5	16	Truck	Alan Steele	Lake Denise	2.3
6	34	Van	Paul Foster	Pagechester	12
7	52	Car	Mr. Patrick Adams III	North John to Robinbury	4.72
8	60	Bus	Spencer Randolph	Lake Michele	12
9	77	Truck	Sara Pham	Port Laurenton to Francistown	2.54
10	78	Bus	Jeffrey Lindsey	Elizabethfort to Angelaland	2.59
11	84	Car	Roy Price	Walkerton to Jeffreyport	12
12	93	Van	Mrs. Lisa Clark	South Tylerchester to Luisland	12
13	107	Truck	null	East Melanie to Garretthaven	5.01

adb-542866418164150.10.azuredatabricks.net/editor/notebooks/3696802500255949?o=542866418164150#command/7015074710219969

Microsoft Azure | databricks

Untitled Notebook 2025-04-04 23:59:07 Python

File Edit View Run Help Last edit was 4 hours ago

Run all | sumit kadam's Cluster | Schedule | Share

12:24 PM (12s) 12

```
)
```

True

12:25 PM (4s) 13

```
df_silver.write.mode("overwrite").parquet("/mnt/adls_storage/silver/silver_data.parquet")
```

(4) Spark Jobs

12:25 PM (3s) 14 Python

```
df_silver.write.jdbc(
    url="jdbc:sqlserver://akash12345.database.windows.net:1433;databaseName=gold",
    table="silver_db.Silver_table",
    mode="overwrite",
    properties={
        "user": "akash12345",
        "password": "Naddraj1998@",
        "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
    }
)
```

(4) Spark Jobs

Calculated fields such as fuel consumed are added to improve analysis. Data is stored in both Parquet format and a MySQL table called `silver_db.delivery_data_silver`.

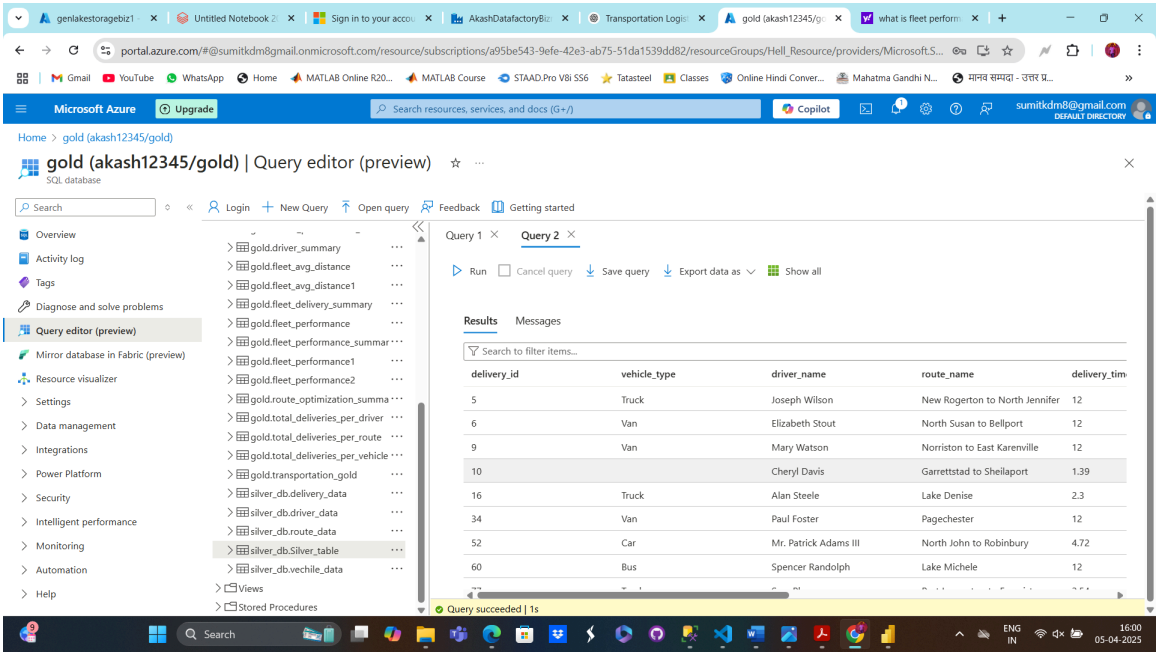
Example Calculation:
fuel_consumed = distance_covered / fuel_efficiency

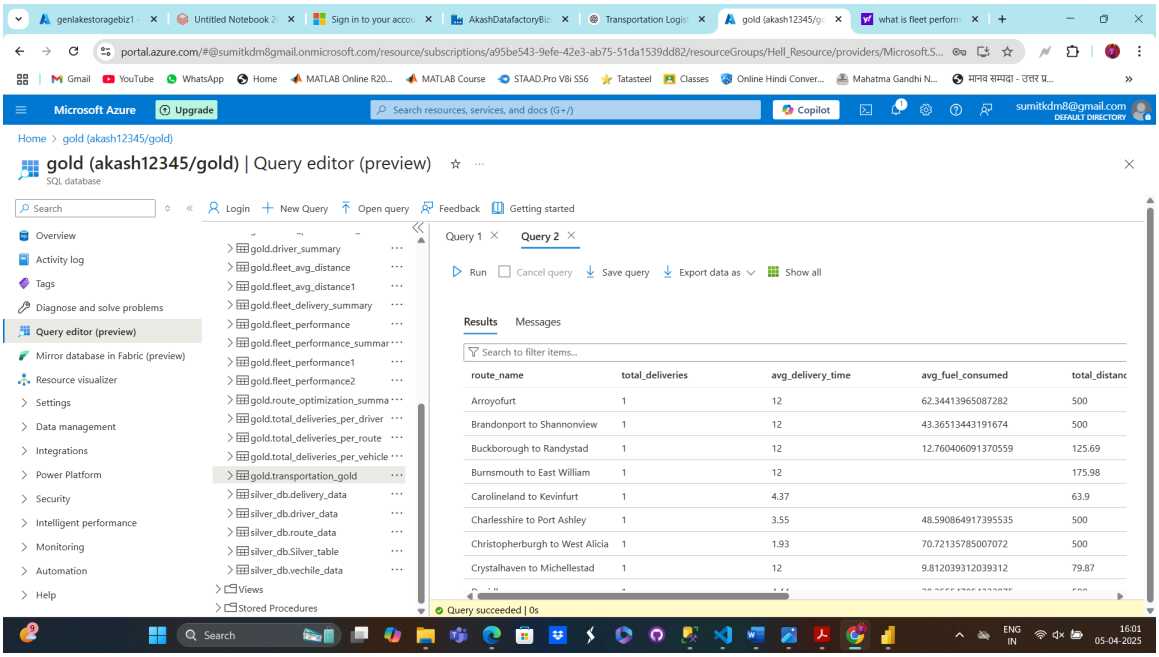
[Space for Silver Layer diagram/image]

3. Gold Layer – Aggregated Data for Reporting

This layer prepares data for final reporting. Key metrics are calculated by aggregating the Silver layer data. Examples include:
- Total number of deliveries per route, driver, and vehicle
- Average delivery time
- Total fuel consumption and efficiency
- Driver ratings and performance over time

Stored in MySQL table: `gold_db.transportation_gold`


Microsoft Azure
Upgrade
Copilot
sumitkdm8@gmail.com
DEFAULT DIRECTORY
Home > gold (akash12345/gold)
gold (akash12345/gold) | Query editor (preview)
SQL database
Login
New Query
Open query
Feedback
Getting started
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Data management
Integrations
Power Platform
Security
Intelligent performance
Monitoring
Automation
Help
gold.driver_summary
gold.fleet_avg_distance
gold.fleet_avg_distance1
gold.fleet_delivery_summary
gold.fleet_performance
gold.fleet_performance1
gold.fleet_performance2
gold.total_deliveries_per_driver
gold.total_deliveries_per_route
gold.total_deliveries_per_vehicle
gold.transportation_gold
silver_db.delivery_data
silver_db.driver_data
silver_db.route_data
silver_db.Silver_table
silver_db.vechile_data
Views
Stored Procedures
Query 1
Query 2
Run
Cancel query
Save query
Export data as
Show all
Results
Messages
Search to filter items...
delivery_id
vehicle_type
driver_name
route_name
5 Truck Joseph Wilson New Rogerton to North Jennifer 12
6 Van Elizabeth Stout North Susan to Bellport 12
9 Van Mary Watson Norriston to East Karenville 12
10 Cheryl Davis Garrettstad to Sheilaport 1.39
16 Truck Alan Steele Lake Denise 2.3
34 Van Paul Foster Pagechester 12
52 Car Mr. Patrick Adams III North John to Robinbury 4.72
60 Bus Spencer Randolph Lake Michele 12
Query succeeded | 1s
16:00
05-04-2025

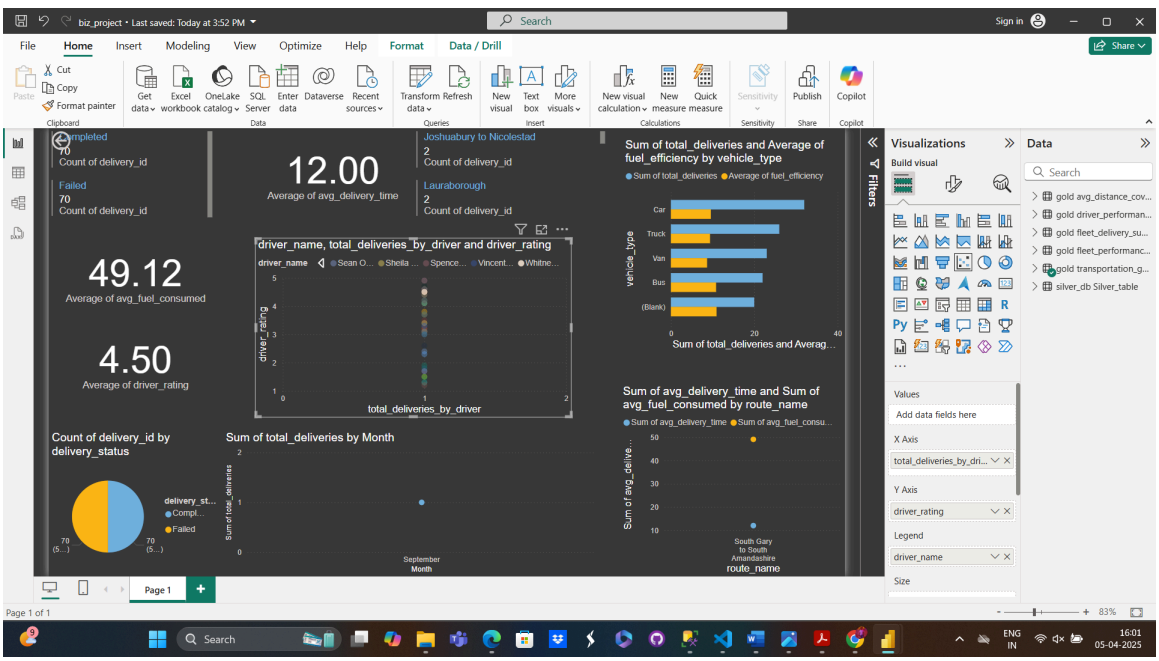

Microsoft Azure
Upgrade
Copilot
sumitkdm8@gmail.com
DEFAULT DIRECTORY
Home > gold (akash12345/gold)
gold (akash12345/gold) | Query editor (preview)
SQL database
Login
New Query
Open query
Feedback
Getting started
Query 1
Query 2
Run
Cancel query
Save query
Export data as
Show all
Results
Messages
Search to filter items...
route_name
total_deliveries
avg_delivery_time
avg_fuel_consumed
Arroyofurt 1 12 62.34413965087282 500
Brandonport to Shannonview 1 12 43.36513443191674 500
Buckborough to Randystad 1 12 12.760406091370559 125.69
Burnsmouth to East William 1 12 175.98
Carolineland to Kevinfurt 1 4.37 63.9
Charlesshire to Port Ashley 1 3.55 48.590864917395535 500
Christopherburgh to West Alicia 1 1.93 70.72135785007072 500
Crystalhaven to Michellestad 1 12 9.812039312039312 79.87
Query succeeded | 0s
16:01
05-04-2025

📊 Power BI Dashboard – Visual Insights

Power BI is used to build interactive visual dashboards based on the Gold layer data stored in MySQL. It includes key performance indicators (KPIs), trend lines, and comparative analysis for better decision-making.

Connected Data Source: MySQL – gold_db.transportation_gold

Key Visuals and KPIs:
1. Line Chart – Average delivery time & fuel usage per route
2. Bar Chart – Total deliveries per vehicle and their efficiency
3. Scatter Plot – Driver performance vs ratings
4. Pie Chart – Completed vs Failed deliveries
5. Line Graph – Delivery trends by week/month



✅ Final Deliverables

1. MySQL Databases:
   - Silver layer table: silver_db.delivery_data_silver
   - Gold layer table: gold_db.transportation_gold

2. Power BI Dashboard:
   - Includes visual KPIs, filters, and export options.

3. Documentation:

- Detailed Word or PDF file explaining each step in the pipeline.

4. Git Repository:
   - Contains PySpark code, MySQL scripts, and Power BI .pbix file.

https://github.com/Akash05111998/Project_Bizmetric.git

🔁 Automation & Best Practices

- Pipeline is fully automated from raw data ingestion to dashboard refresh.
- Consistent naming conventions across files and tables.
- Modular code with reusable functions.
- Data audit logs maintained at each transformation stage.
- Easy to scale and integrate with more data sources.