Statement of Work (SoW) for Retail Sales Data Processing & Analysis

By Aaisha Modak

Intern

Table of Contents

1. Introduction

In today's data-driven sales landscape, organizations generate vast volumes of data from customer interactions, transactions, sales channels, and operational workflows. Effectively leveraging this data can significantly enhance decision-making, reduce operational expenses, and improve sales performance. However, this data is often unstructured, scattered across multiple systems, and difficult to interpret in its raw form.

This project focuses on building a data processing and analytics platform using Azure Databricks, PySpark, MySQL, and Power BI, structured according to the Medallion Architecture methodology. The Medallion Architecture offers a layered approach to data refinement—starting with raw data (Bronze), moving to cleaned and enriched data (Silver), and culminating in aggregated, business-ready insights (Gold).

The primary goal was to automate the ingestion, transformation, and visualization of sales data, enabling stakeholders to monitor key performance indicators (KPIs) such as revenue trends, sales conversion rates, and product performance. The entire solution was developed over a span of 30 hours using scalable, cloud-native tools, with the results visualized through an interactive Power BI dashboard.

1.1 Basic Concepts

- **ETL (Extract, Transform, Load):** A process that involves extracting data from source systems, transforming it as needed, and loading it into a destination for further analysis or reporting.
- **Data Pipeline:** An automated sequence of processes that manages data movement and transformation across different systems seamlessly.
- **Parquet:** A columnar storage file format designed for efficient data retrieval and optimized for big data processing workloads.
- **Azure:** Microsoft's cloud platform offering a wide range of computing, networking, and storage services.
- **Databricks:** A unified analytics platform built for cloud environments that enables scalable data processing and machine learning.
- **Azure Data Factory (ADF):** A cloud-based data integration service used for building, managing, and automating data pipelines.
- **Power BI:** Microsoft's business intelligence tool that helps users visualize data, share insights, and create interactive dashboards and reports.
- **PySpark:** The Python API for Apache Spark, enabling large-scale distributed data processing and analytics using Python.
- **MySQL:** An open-source relational database management system used for structured data storage and retrieval.
- **Azure Blob Storage:** A Microsoft service designed for storing large volumes of unstructured data like logs, images, and documents.

- **Azure Data Lake Storage Gen2:** A scalable and high-performance storage solution from Microsoft that supports big data analytics with both structured and unstructured data.
- **Automation in Azure Data Factory (ADF):** Refers to automating ETL workflows through triggers, scheduling, and pipeline orchestration to ensure smooth, hands-off data processing across various sources and destinations.

1.2 Medallion Architecture

Medallion Architecture is a data engineering pattern that organizes data processing into logical layers to improve quality and usability at each step. The model typically consists of three layers:

- **Bronze Layer:** Holds raw, unprocessed data directly ingested from source systems.
- **Silver Layer:** Contains data that has been cleaned, filtered, and transformed from the bronze layer.
- **Gold Layer:** Stores curated, high-quality data optimized for analytics and business decision-making.

Data flows between these layers through pipelines, improving in structure and quality at each stage.

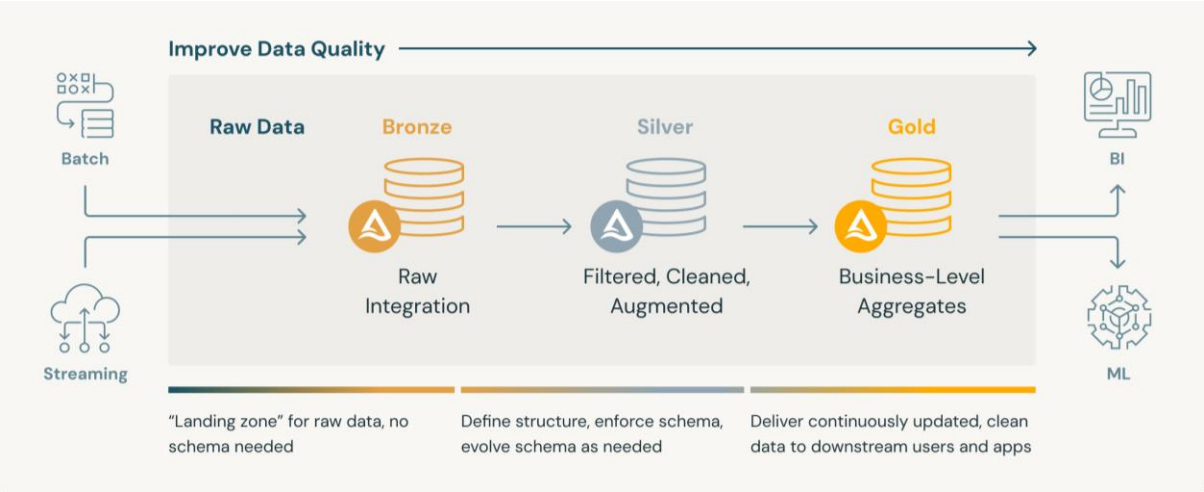


Figure 1. Medallion architecture

2. Architecture Overview

This project adopts the Medallion Architecture framework, as previously outlined, which structures the data pipeline into three core layers: **Bronze**, **Silver**, and **Gold**. Each of these layers signifies a different stage of data refinement, facilitating a scalable, maintainable, and auditable data flow.

In this implementation, the Medallion Architecture is applied within **Azure Data Lake Storage Gen2 (ADLS Gen2)**. However, as the original source data was in **CSV format**, and Parquet format is preferred for efficient processing and storage, the CSV files were first uploaded to a dedicated **Azure Blob Storage**. Using **PySpark in an Azure Databricks notebook**, the data was then transformed into Parquet format and loaded into the **Bronze layer** of the ADLS container to initiate the refinement process.

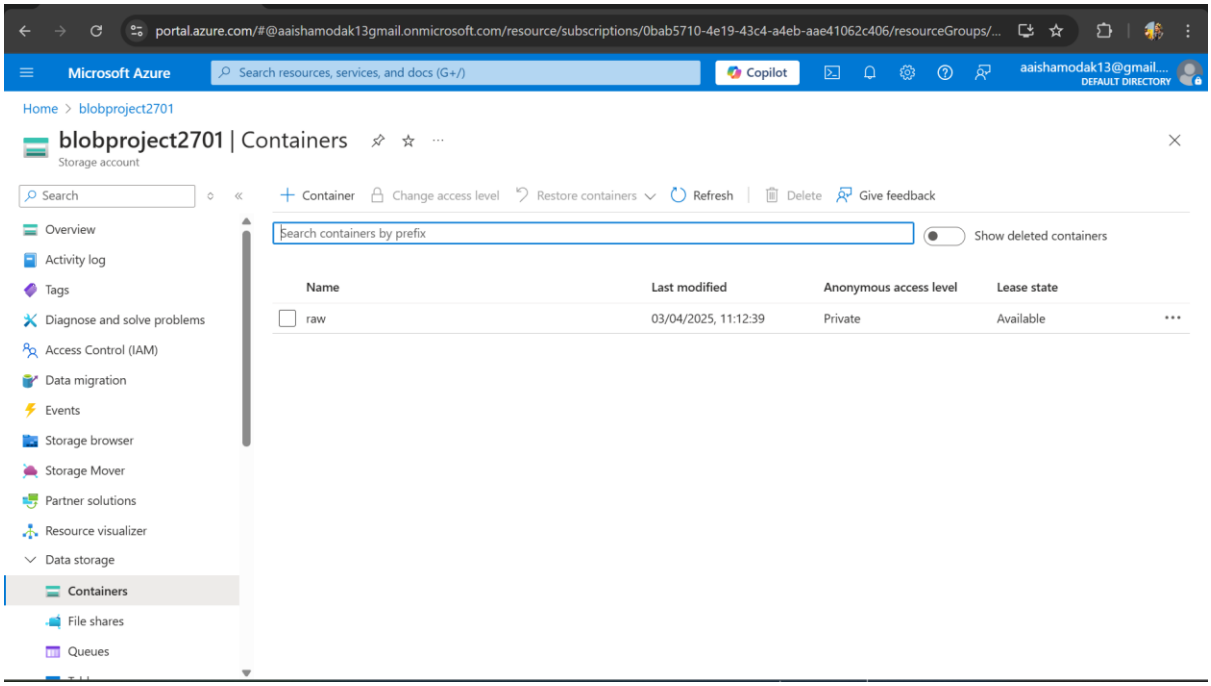


Figure 2. Source data Azure blob storage

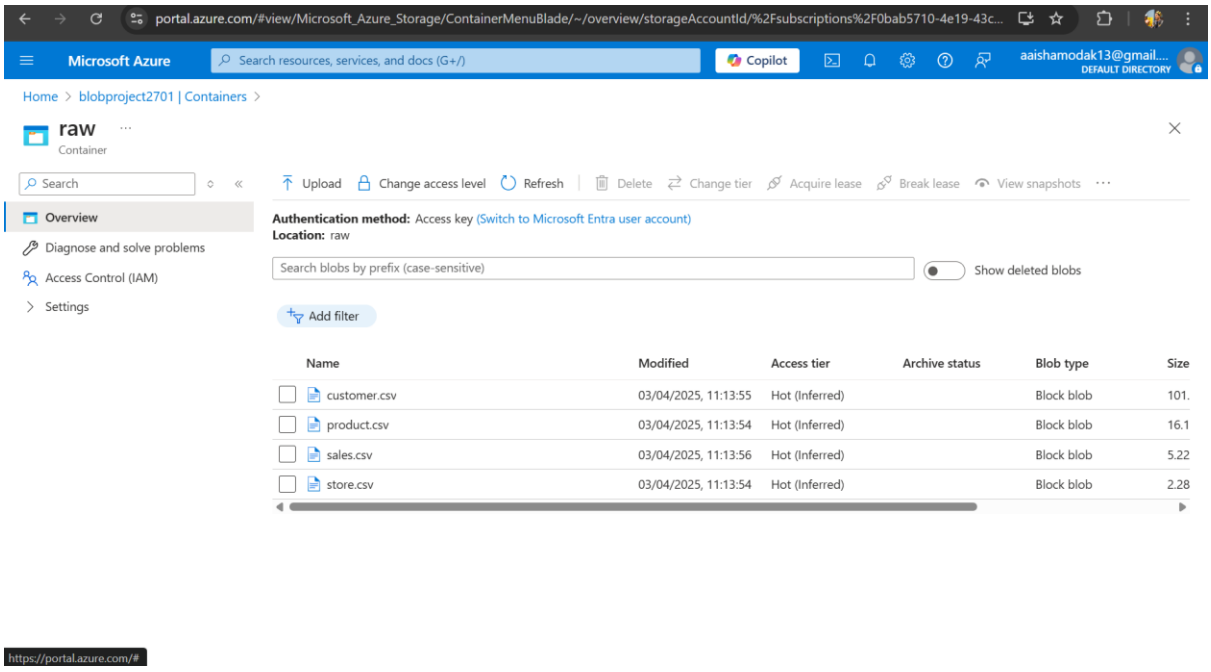


Figure 3. raw container with source csv files

Bronze Layer:

The **Bronze layer** holds the **raw, unprocessed sales data** ingested directly from the source systems. Although the format has been converted to **Parquet** for optimized performance and storage, the data remains unchanged in terms of structure and content—no specific data types or transformations are applied at this stage.

This layer serves as the **foundation** for all further data refinement and transformation processes. It often includes **metadata**, such as the timestamp of data ingestion, original file names, or the name of the data source. The Bronze layer can comprise multiple tables, each capturing different aspects or stages of the incoming raw sales data.

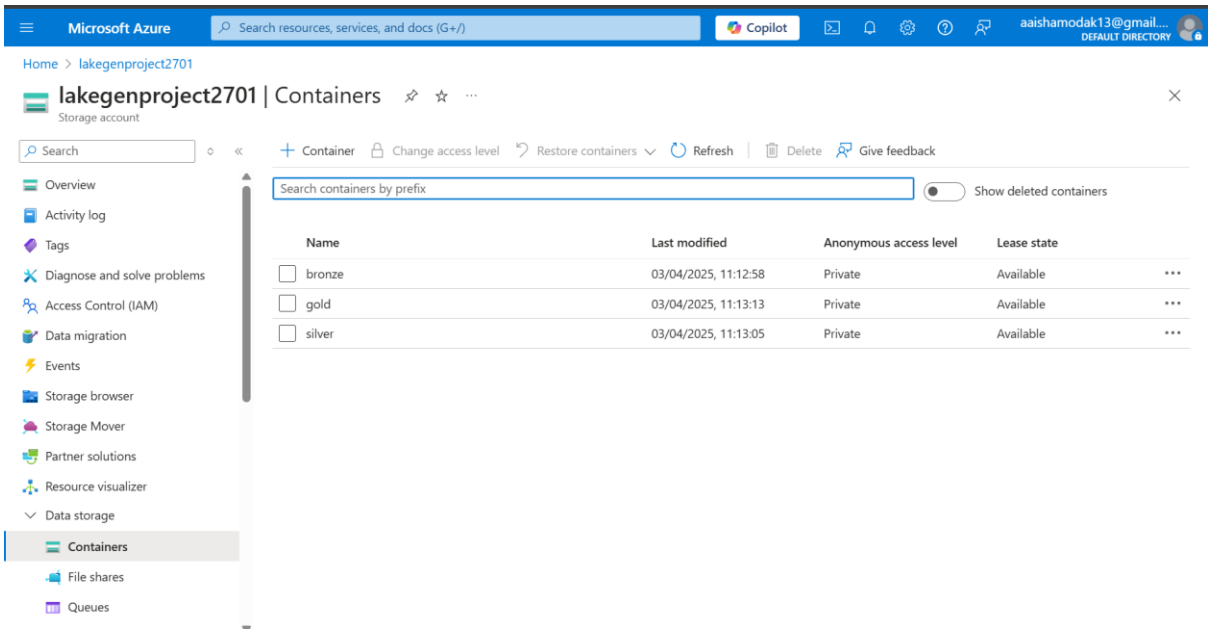


Figure 4. ADLS Gen2 Storage

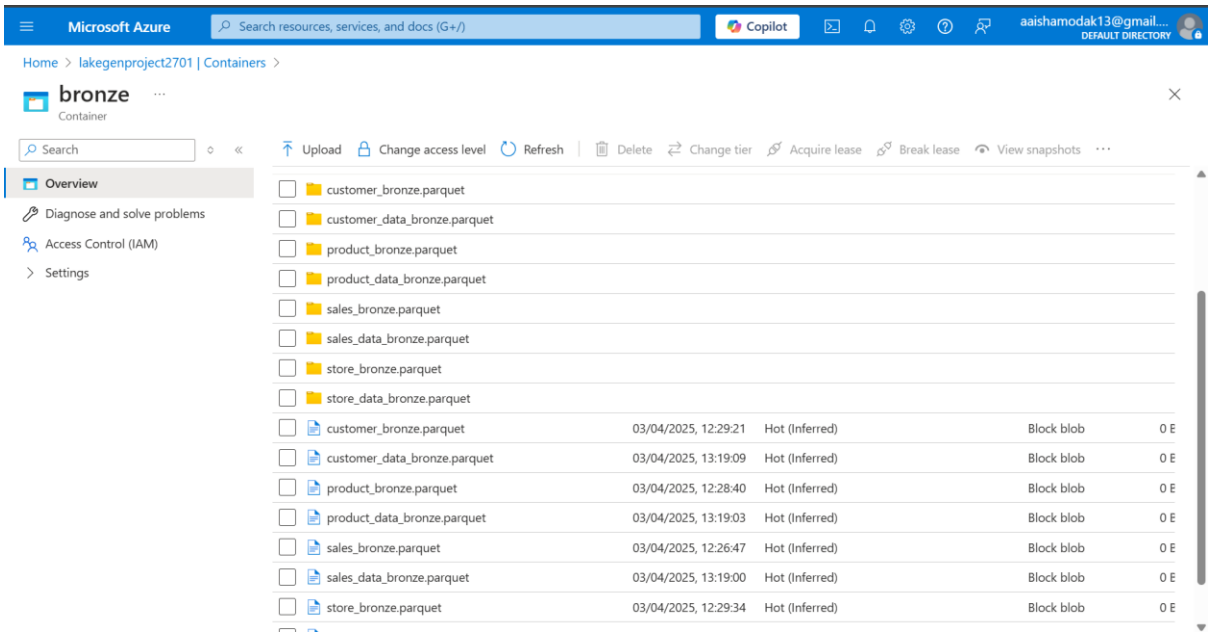


Figure 5. ADLS bronze container

Silver Layer:

The **Silver layer** contains **cleaned, standardized, and structured sales data**, refined using **PySpark in Databricks**. This stage involves key data transformation tasks such as **removing duplicates**, **correcting**

inaccurate or inconsistent entries, **assigning appropriate data types**, and **organizing the data into well-defined columns**.

The processed data in the Silver layer is maintained in **two formats**:

1. As a **Parquet file**, stored in the Silver container within **Azure Data Lake Storage Gen2 (ADLS)**.
2. As a **SQL table**, stored in the **MySQL database** under silver_db, with the table named sales_data_silver.

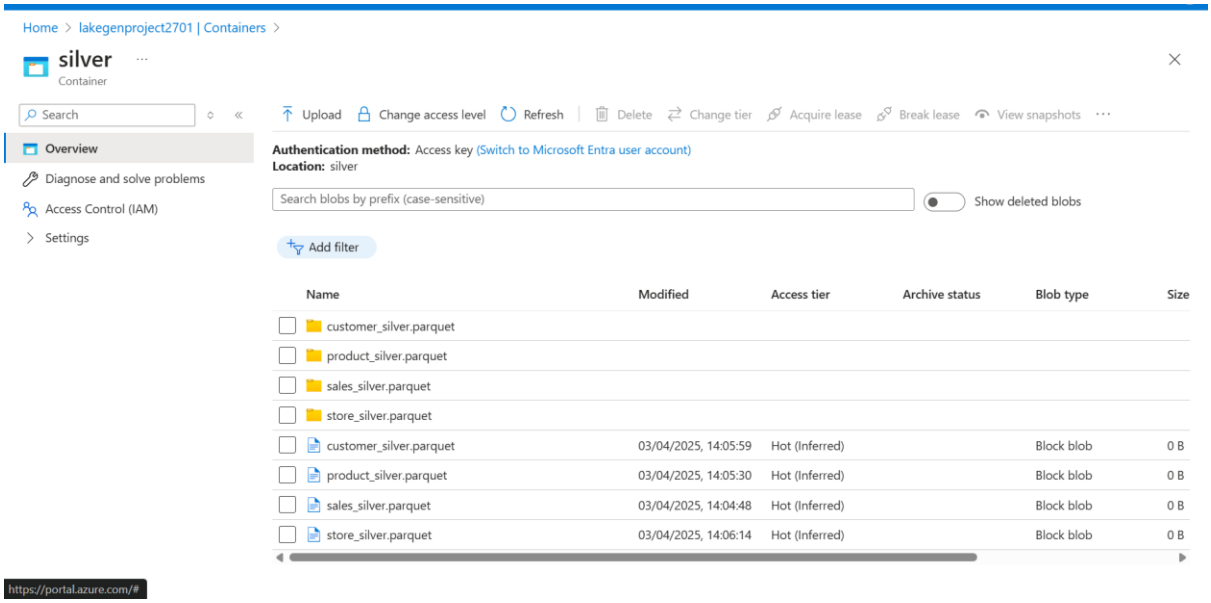


Figure 6. Silver container

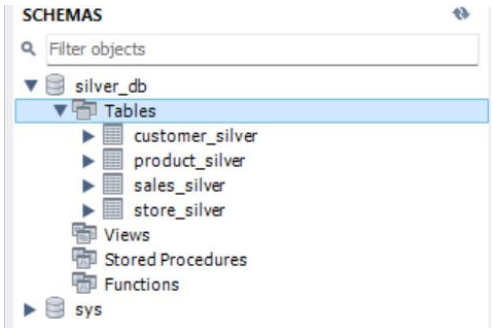


Figure 7: Silver_db in MySQL

Gold Layer:

The **Gold layer** contains the **final aggregated sales dataset** stored in **MySQL**, specifically designed to support **Power BI reports** and facilitate the computation of **key business performance metrics**.

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

- **Sales Region Analysis**

    - Total sales per region
    - Average revenue per region
    - Sales trend by region
- **Product Performance**
    - Total sales per product
    - Average revenue per product
    - Top-selling and underperforming products
- **Salesperson Performance**
    - Total sales per salesperson
    - Average revenue per salesperson
    - Sales conversion rate and efficiency

These aggregated metrics are then visualized in **Power BI**, enabling business users to derive insights, track performance, and make informed strategic decisions.

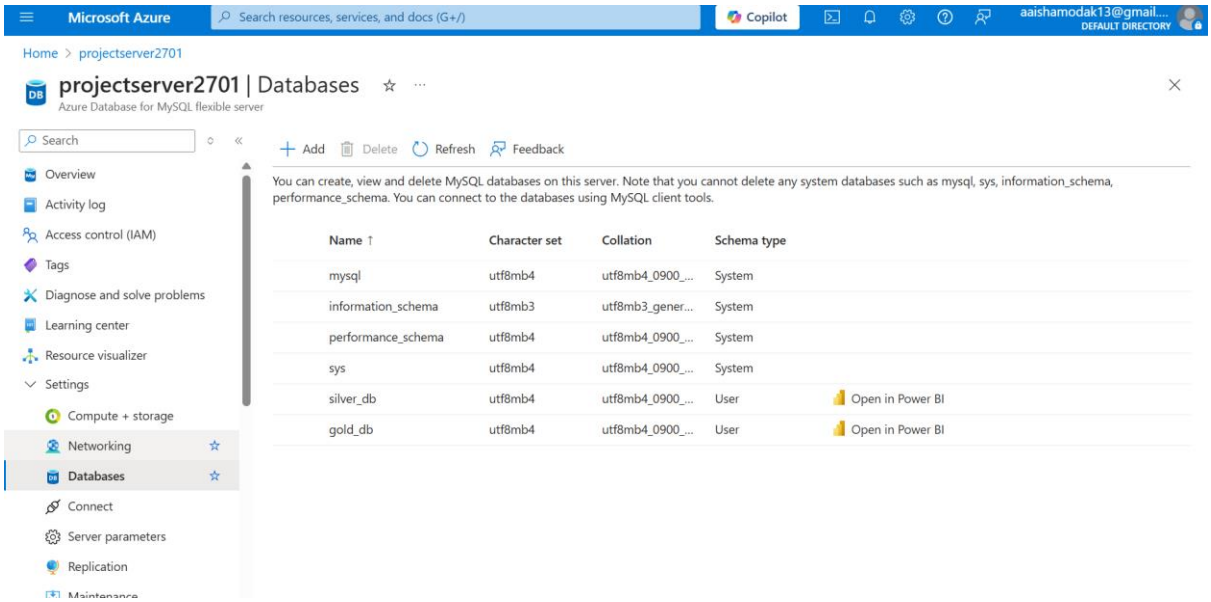
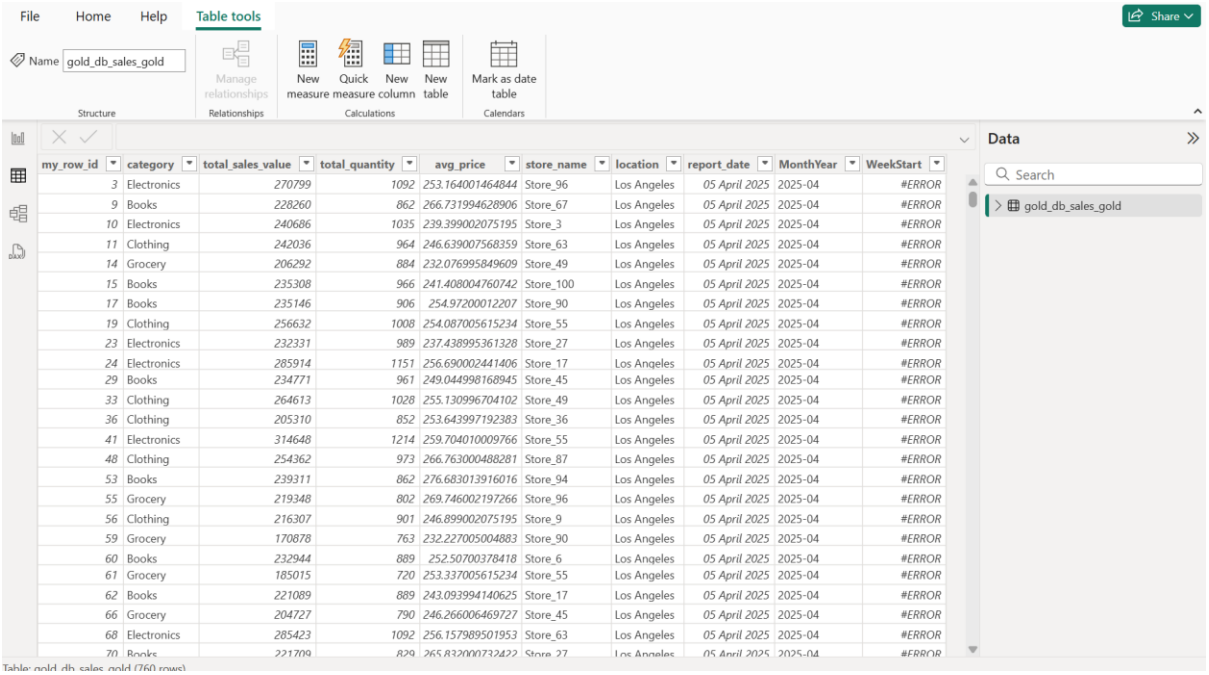


Figure 9. Downloading (.pbids) file of the database from Azure Databases for MySQL flexible servers



my_row_id	category	total_sales_value	total_quantity	avg_price	store_name	location	report_date	MonthYear	WeekStart
3	Electronics	270799	1092	253.164001464844	Store_96	Los Angeles	05 April 2025	2025-04	#ERROR
9	Books	228260	862	266.731994628906	Store_67	Los Angeles	05 April 2025	2025-04	#ERROR
10	Electronics	240686	1035	239.399002075195	Store_3	Los Angeles	05 April 2025	2025-04	#ERROR
11	Clothing	242036	964	246.639007568359	Store_63	Los Angeles	05 April 2025	2025-04	#ERROR
14	Grocery	206292	884	232.076995849609	Store_49	Los Angeles	05 April 2025	2025-04	#ERROR
15	Books	235308	966	241.408004760742	Store_100	Los Angeles	05 April 2025	2025-04	#ERROR
17	Books	235146	906	254.97200012207	Store_90	Los Angeles	05 April 2025	2025-04	#ERROR
19	Clothing	256632	1008	254.087005615234	Store_55	Los Angeles	05 April 2025	2025-04	#ERROR
23	Electronics	232331	989	237.438995361328	Store_27	Los Angeles	05 April 2025	2025-04	#ERROR
24	Electronics	285914	1151	256.690002441406	Store_17	Los Angeles	05 April 2025	2025-04	#ERROR
29	Books	234771	961	249.044998168945	Store_45	Los Angeles	05 April 2025	2025-04	#ERROR
33	Clothing	264613	1028	255.130996704102	Store_49	Los Angeles	05 April 2025	2025-04	#ERROR
36	Clothing	205310	852	253.643997192383	Store_36	Los Angeles	05 April 2025	2025-04	#ERROR
41	Electronics	314648	1214	259.704010009766	Store_55	Los Angeles	05 April 2025	2025-04	#ERROR
48	Clothing	254362	973	266.763000488281	Store_87	Los Angeles	05 April 2025	2025-04	#ERROR
53	Books	239311	862	276.683013916016	Store_94	Los Angeles	05 April 2025	2025-04	#ERROR
55	Grocery	219348	802	269.746002197266	Store_96	Los Angeles	05 April 2025	2025-04	#ERROR
56	Clothing	216307	901	246.899002075195	Store_9	Los Angeles	05 April 2025	2025-04	#ERROR
59	Grocery	170878	763	232.227005004883	Store_90	Los Angeles	05 April 2025	2025-04	#ERROR
60	Books	232944	889	252.50700378418	Store_6	Los Angeles	05 April 2025	2025-04	#ERROR
61	Grocery	185015	720	253.337005615234	Store_55	Los Angeles	05 April 2025	2025-04	#ERROR
62	Books	221089	889	243.093994140625	Store_17	Los Angeles	05 April 2025	2025-04	#ERROR
66	Grocery	204727	790	246.266006469727	Store_45	Los Angeles	05 April 2025	2025-04	#ERROR
68	Electronics	285423	1092	256.157989501953	Store_63	Los Angeles	05 April 2025	2025-04	#ERROR
70	Books	221709	829	265.832000732422	Store_27	Los Angeles	05 April 2025	2025-04	#ERROR

Table: gold_db_sales_gold (760 rows)

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

Solution Implementation

The project was built using the **Medallion Architecture** to ensure scalable and automated processing of structured sales data. The solution was orchestrated using Azure tools and big data technologies.

Data Ingestion – Bronze Layer

The raw data files were provided in **CSV format**, representing various aspects of a retail sales environment, such as sales.csv, product.csv, customer.csv, and store.csv.

These files were ingested using **PySpark in Azure Databricks** and stored in **Azure Data Lake Storage (ADLS)** Bronze layer from the raw zone in Azure Blob Storage.

During ingestion:

- The schema was inferred automatically by PySpark.
- Additional metadata fields such as ingestion_date and source_file_name were added to maintain traceability.
- Data was converted and stored in **Parquet format** to enhance storage efficiency and query performance.

This ingestion layer forms the **foundation of reliable and auditable data processing** in the pipeline.

4.2 Data Processing – Silver Layer

The **Silver layer** is focused on producing **cleaned and enriched sales data** that is analysis-ready. The operations performed include:

- **Null Handling**: Records missing critical information such as product_id, customer_id, or store_id were removed.
- **Data Type Casting**: Fields like quantity, and total_amount were cast to appropriate formats.
- **Join Operations**:
    - sales was joined with product, customer, and store to enrich transactions with contextual data such as product category, customer demographics, and store location.

The enriched data was written to **Parquet files in the ADLS Silver container**, and also pushed to a **MySQL table (sales_data_silver)** via JDBC to support BI tools.

4.3 Aggregation and Business Logic – Gold Layer

In the **Gold layer**, business-level **aggregated insights** were created to support executive reporting and dashboards. These aggregations were created through **SQL queries on the Silver-layer MySQL table**.

The resulting metrics were stored in a **fact table sales_gold within the gold_db schema** in MySQL, ready for consumption by Power BI.

Aggregated metrics include:

- Total sales revenue per product, store, and region.
- Average basket size (quantity per sale).
- Top-selling products by revenue and quantity.
- Monthly sales trends.
- Customer retention and frequency.

3.4 Dashboarding with Power BI

The final step was building a dynamic **Power BI dashboard** connected to the **Gold-layer MySQL table**. The dashboard delivers actionable business intelligence through real-time KPIs and interactive visuals.

KPIs:

1. Total Sales Revenue (Sum of total_amount from gold_sales)

2. Average Order Value (Total Sales / Number of Transactions)

3. Top 5 Best-Selling Products (Based on total_quantity from gold_sales)

4. Sales Trend Over Time (Monthly/Weekly sales visualization)

Visualizations:

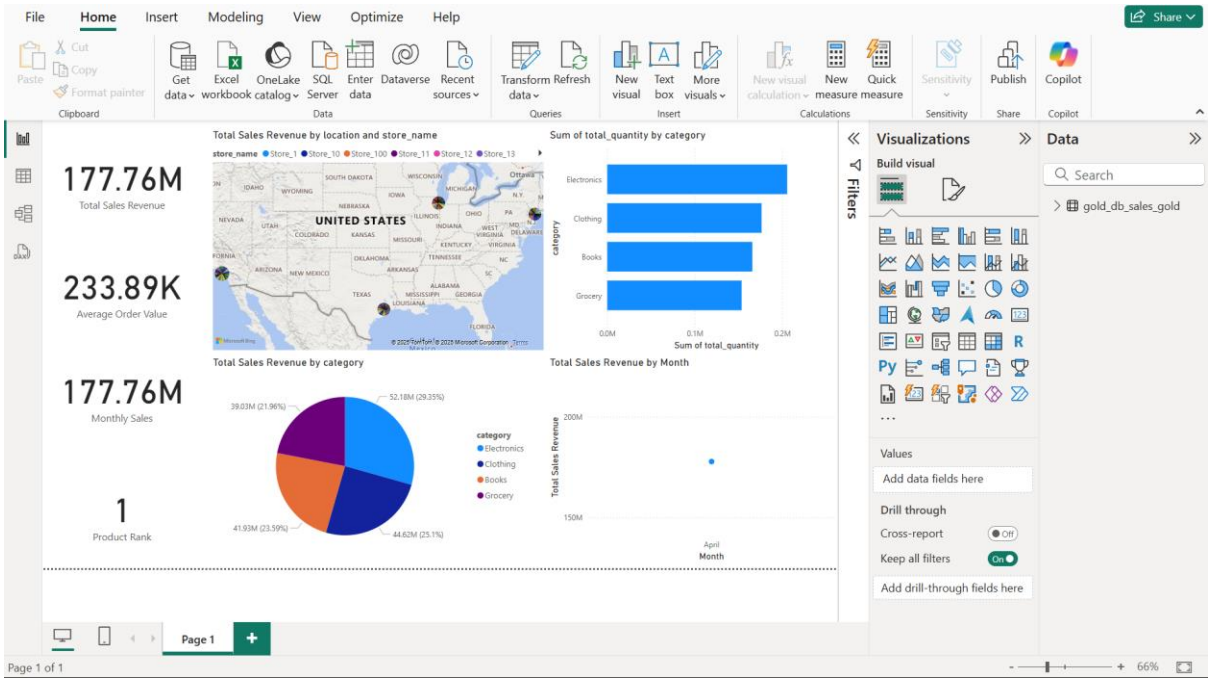
1. Sales Trend Chart – Line chart showing sales over time.

2. Top 5 Best-Selling Products – Bar chart of top-selling products.

3. Revenue by Store Location – Map visualization to show revenue distribution across stores.

4. Category-wise Sales Distribution – Pie chart to show product category-wise revenue split.

The Power BI dataset is configured for **automated refresh**, ensuring that the dashboard reflects the most recent Gold-layer data from MySQL.



4. 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 ensured collaboration, reproducibility, and code backup throughout the project lifecycle.

Repository Name: retail_sales_data
GitHub Link: https://github.com/aaishamodak/retail_sales_data

---

5. Challenges & Solutions

During the implementation, several challenges arose while working with big data technologies, cloud services, and dashboard integrations. These challenges became valuable learning opportunities, enhancing my ability to troubleshoot and problem-solve effectively.

Challenges Encountered:

- Finding an appropriate Azure region that supports MySQL Flexible Server.

- Establishing a connection between MySQL Flexible Server and MySQL Workbench for querying.
- Errors due to missing libraries while writing data to MySQL from Databricks.
- MySQL connectivity issues with Power BI due to driver-related errors.

Solutions Implemented:

- Referred to the latest Azure documentation and community blogs to identify supported regions for MySQL Flexible Server.
- Resolved connection errors by correctly configuring access and reviewing documentation.
- Installed the required Python libraries directly into the Databricks cluster to resolve dependency issues.
- Downloaded and configured the necessary .NET components to ensure seamless MySQL integration with Power BI.

---

6. Learnings

This internship project marked a significant milestone in my professional journey, offering hands-on exposure to real-world data and industry-grade tools. As a fresher, the project introduced me to best practices in data engineering and cloud integration.

Key Takeaways:

- Understood the benefits of Medallion Architecture (Bronze, Silver, Gold) for structuring and managing data lifecycle stages.
- Gained practical experience in designing and managing scalable data pipelines using PySpark and Azure Databricks.
- Learned to handle schema evolution, perform complex joins, and implement business logic through SQL.
- Developed and deployed an interactive Power BI dashboard for business insights.
- Improved technical documentation skills and appreciation for structured, modular workflows.

---

7. Conclusion

This project successfully delivered a comprehensive, cloud-native data pipeline and analytics solution for a sales and logistics scenario. By integrating Azure Databricks, Data Lake Storage, MySQL, Power BI, and Azure Data Factory, I built an end-to-end ETL solution based on the Medallion Architecture.

8.1 Achievements

- Ingested raw logistics data from CSV files into a structured data lake.
- Developed PySpark pipelines for data ingestion, cleaning, and transformation.

- Implemented SQL-based aggregation logic to compute business-relevant KPIs.
- Designed and published a Power BI dashboard connected to the Gold layer for real-time insights.
- Automated the entire data flow using Azure Data Factory for scalability and maintainability.

8.2 My Takeaways

- Learned to work with modern data engineering tools and frameworks.
- Gained exposure to real-world datasets and overcame practical implementation challenges.
- Developed a solid understanding of backend architecture and frontend data visualization.
- Improved communication and project documentation skills, which are critical in real-world settings.

This experience has laid a strong foundation for my career in data engineering and analytics, instilling both confidence and a passion for building data-driven solutions.