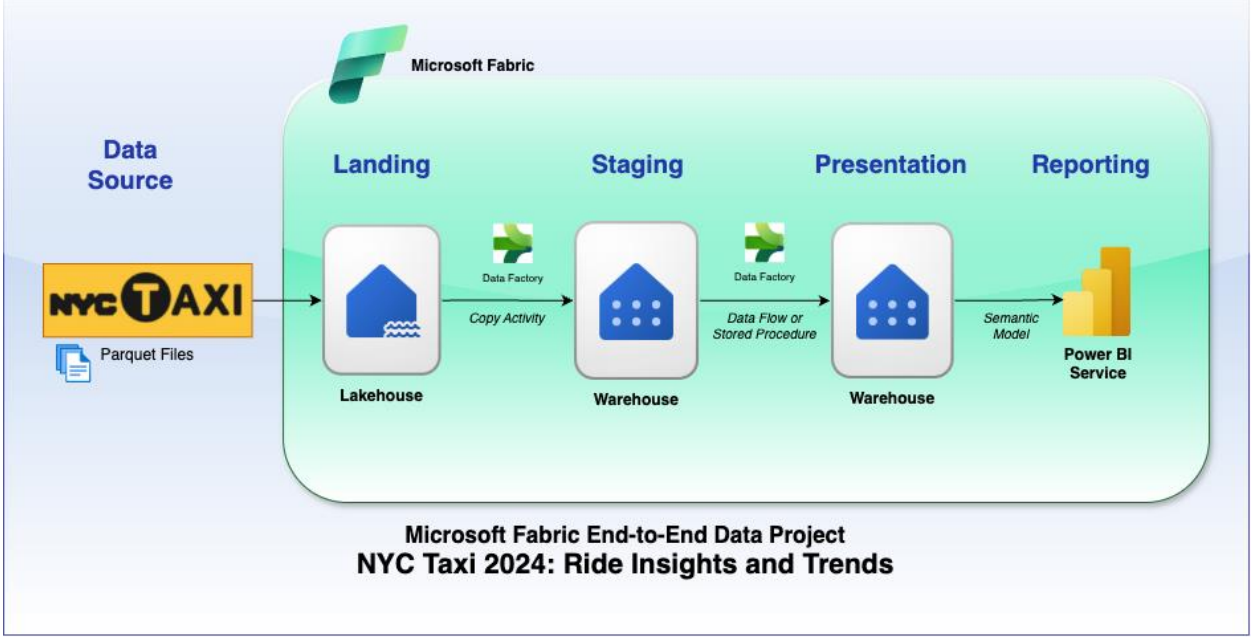
Microsoft Fabric End-to-End Data Engineering Project:
NYC Taxi 2024: Ride Insights and Trends

by Randy A. Velasco
Certified Data Engineer

Project Overview



This project leverages **Microsoft Fabric** to build a scalable, automated end-to-end data pipeline for analyzing 2024 NYC Taxi Trip data. The solution ingests raw data, stages it for transformation, prepares it for analysis in a Power BI dashboard, and includes metadata-driven automation for continuous data processing.

🔗 Data Sources

- **Taxi Trip Data (Parquet)**
  - Format: Parquet
    Frequency: Monthly (Jan - Dec 2024)
    Content: Trip records including pickup/drop-off times, passenger count, trip distance, fare, etc.
- **Taxi Zone Lookup Table (CSV)**
  - Contains mapping of location IDs to taxi zones, boroughs, and service zones.

🧱 Architecture Layers

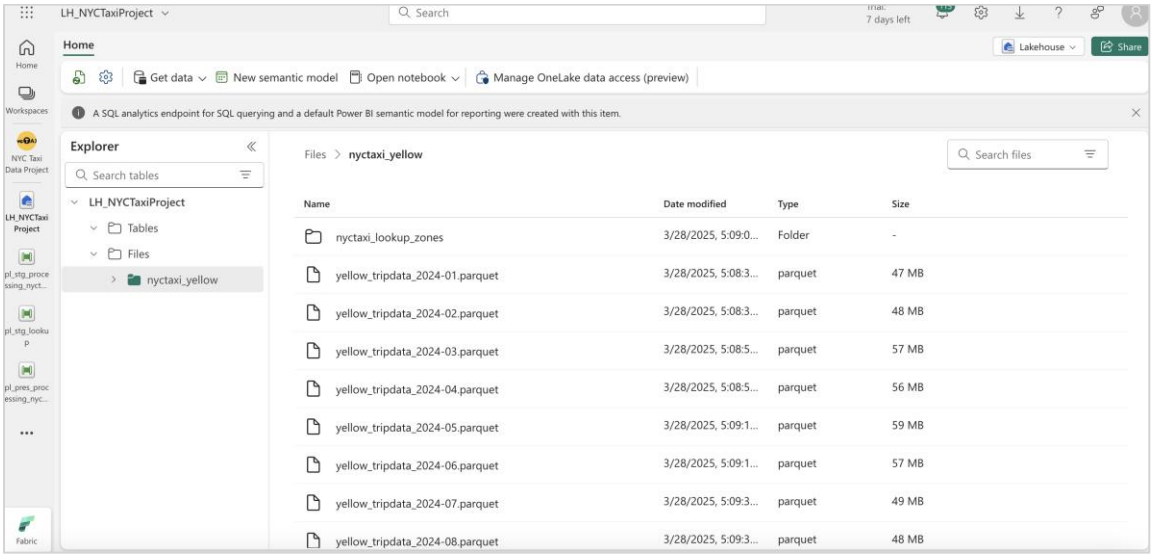
• Landing Layer: Lakehouse

Data is uploaded to the **Lakehouse landing zone** in Microsoft Fabric.

Files stored:

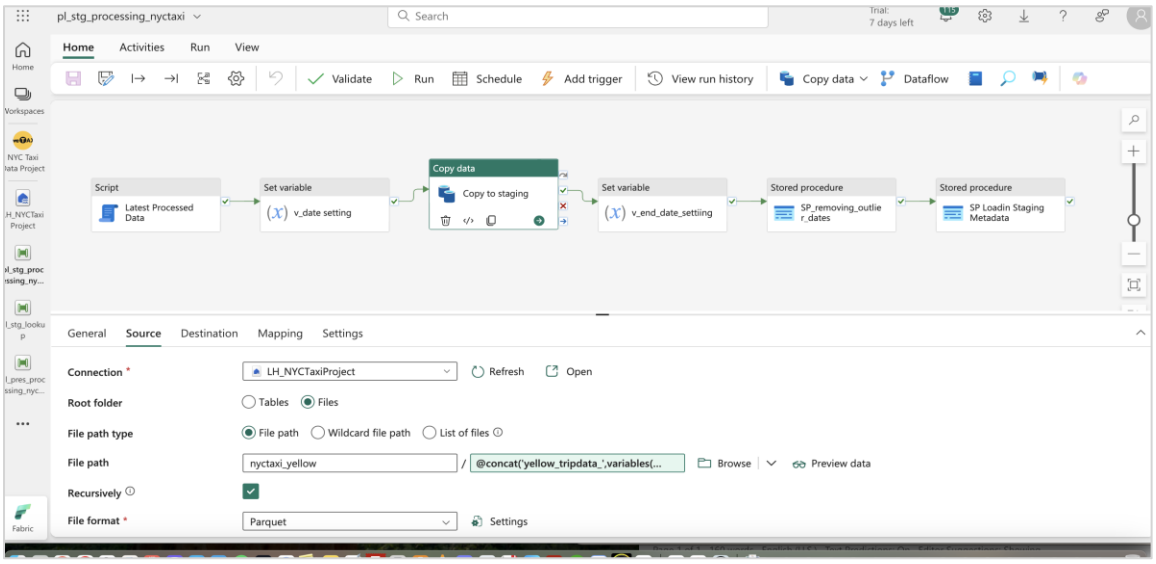
- *yellow_tripdata_2024-01.parquet* to *yellow_tripdata_2024-12.parquet*
- *nyctaxi_lookup_zones.csv*

Purpose: Centralized, raw data storage for structured processing downstream.

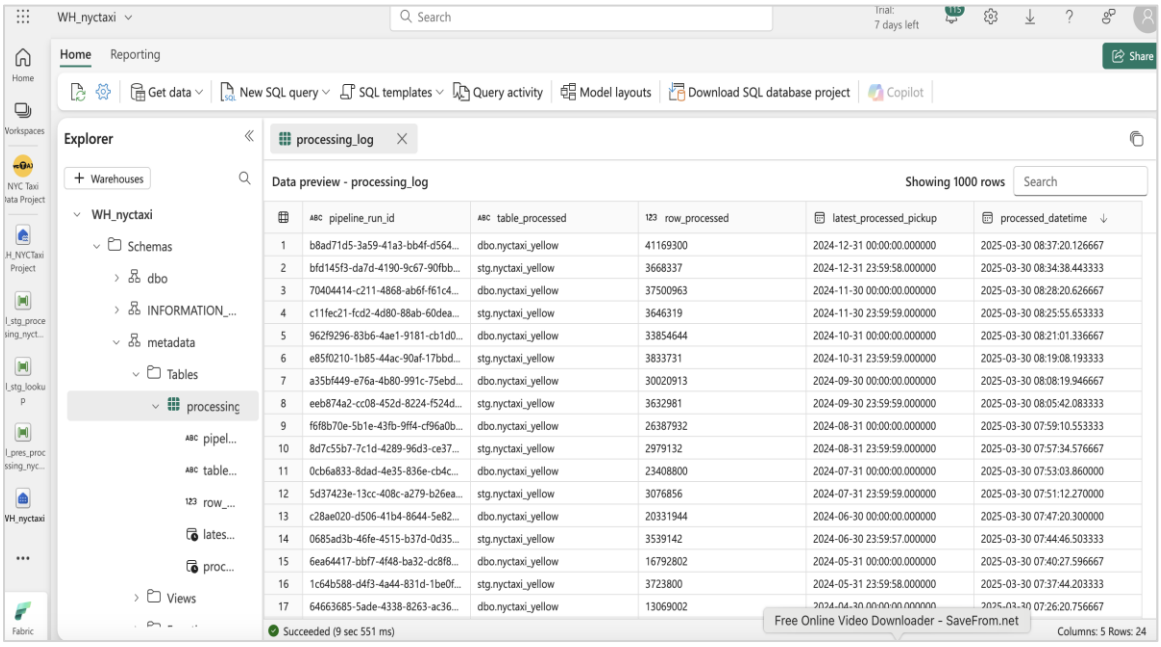


2. Staging Layer: Data Warehouse

- **Pipeline: Load to Staging**
    - **Copy Data Activity**: Loads individual monthly parquet files into the staging table.
    - **Stored Procedure Activity**: Filters rows based on the file's corresponding month using dynamic variables (@start_date, @end_date).
    - **Variable Configuration**: Dynamically sets processing window using metadata table reference.

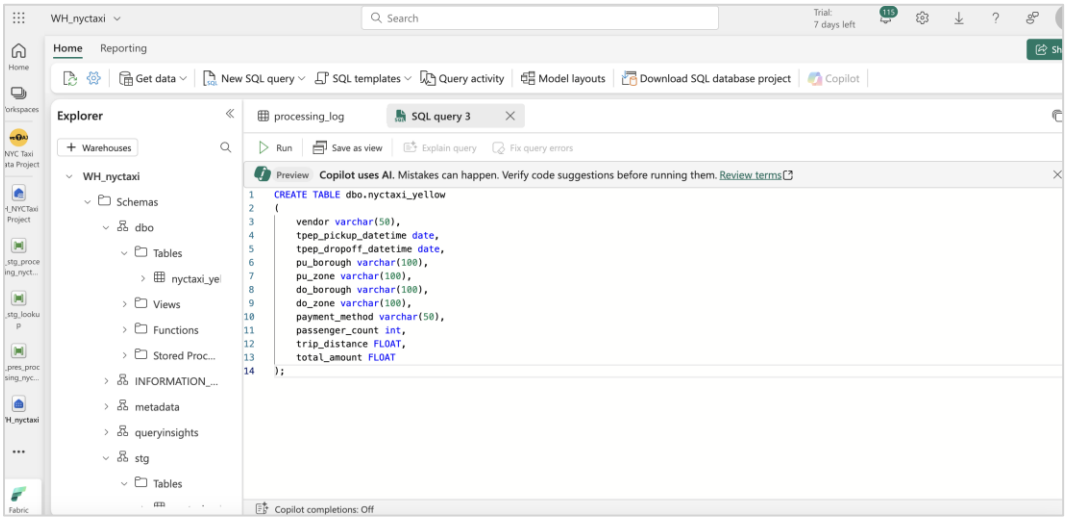


- **Metadata Table (metadata.processing_log)**:
    - Columns: *pipeline_run_id, table_processed, rows_processed, processed_datetime, latest_processed_datetime*.
    - Used to track processed files and determine the next file to load.

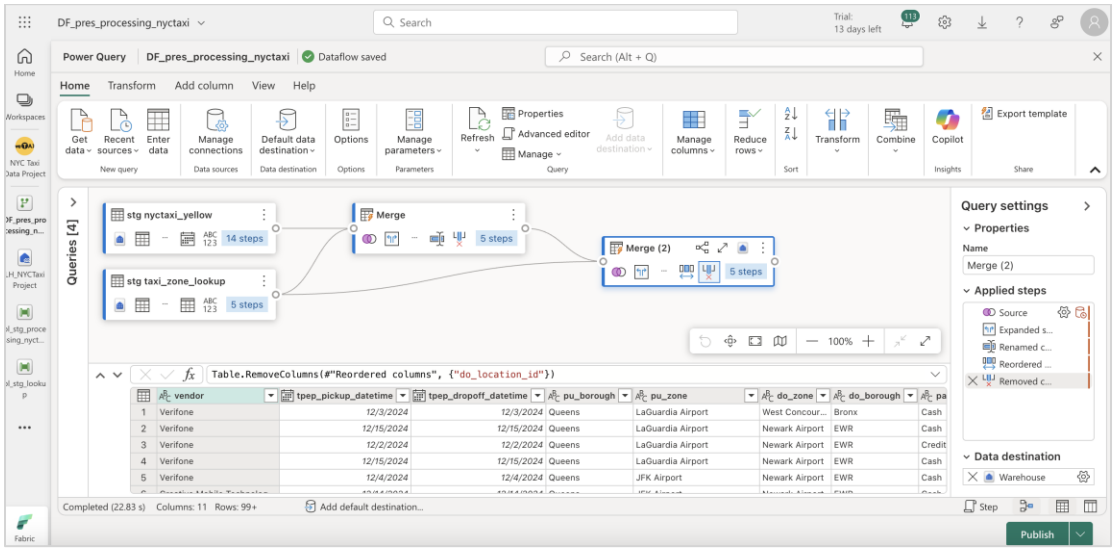


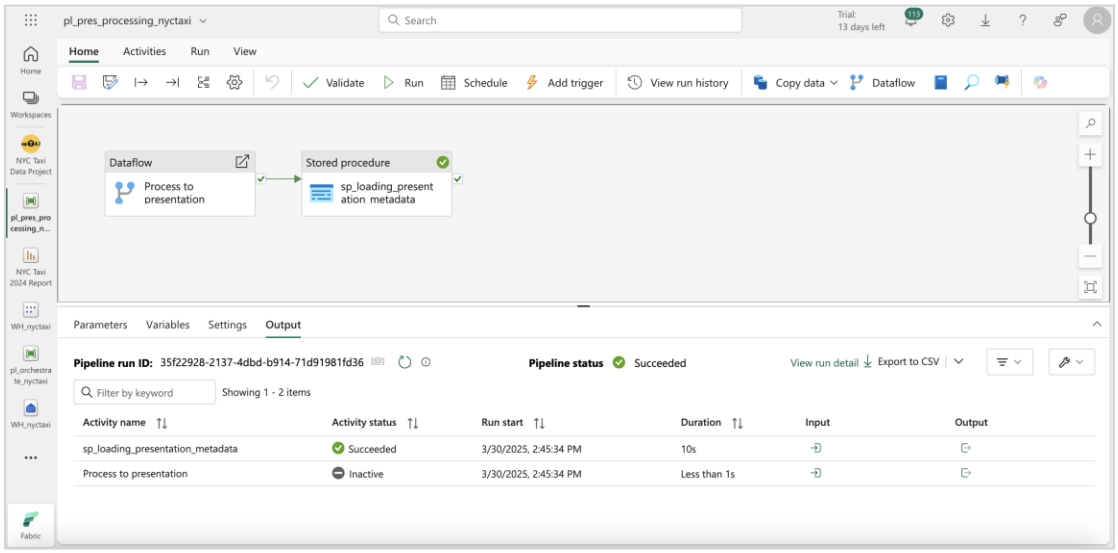
3. Presentation Layer: Data Warehouse

- **Table Initialization**:
    - A blank **presentation table** is created to house enriched and final-formatted data.



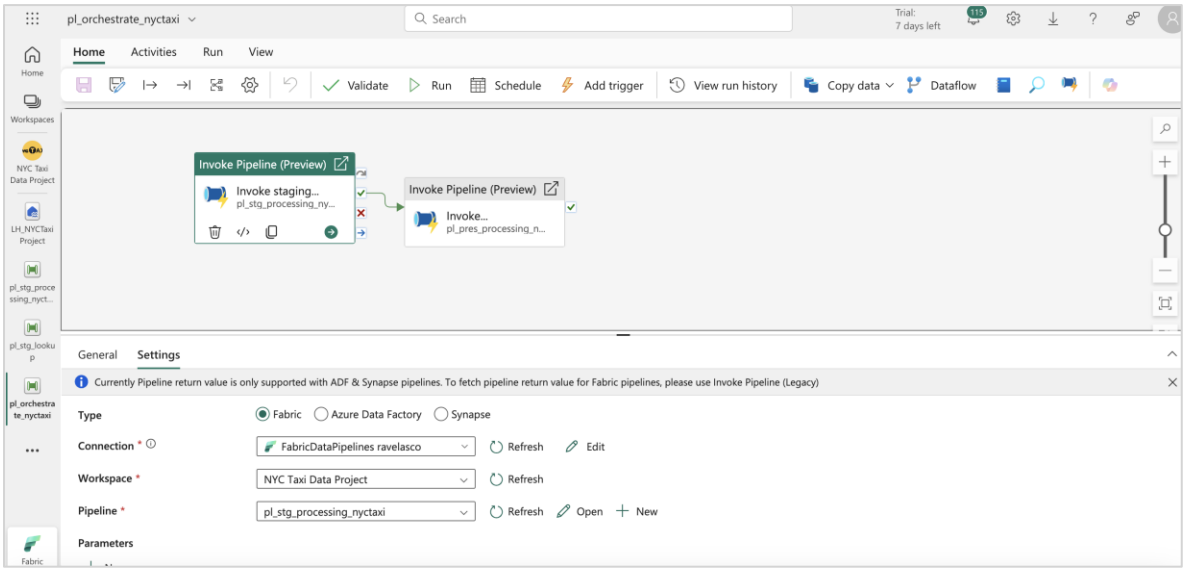
- **Pipeline: Load to Presentation**
  - **Dataflow Activity**:
    - Reads from staging table
    - Performs transformations:
      - Removes unwanted columns
      - Adds new columns *(e.g., vendor, payment_method)*
      - Joins with **Taxi Zone Lookup Table** (converted to table in staging)
    - Merges output into presentation table for analysis.





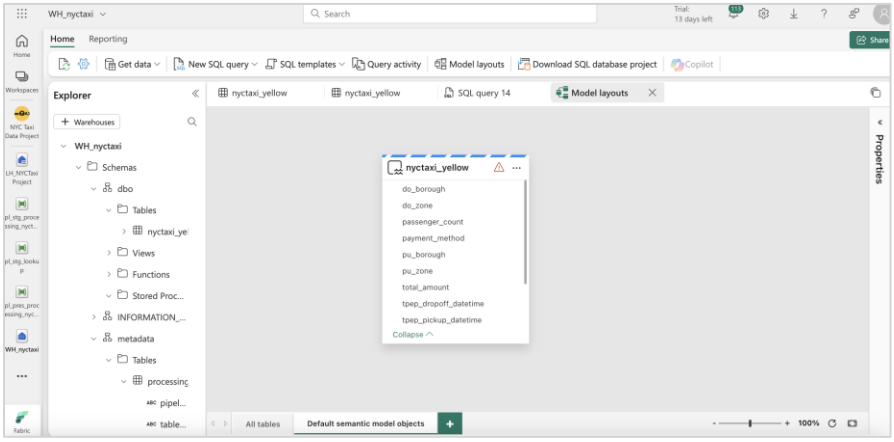
4. Pipeline Orchestration

- A **master pipeline** orchestrates:
  - The **staging pipeline**
  - The **presentation pipeline**

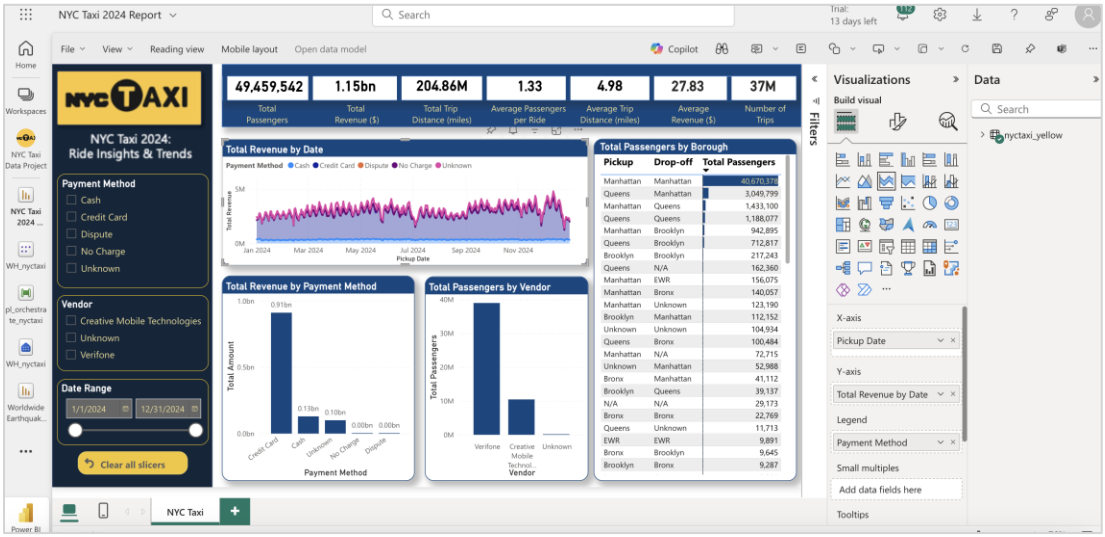


5. Power BI Dashboard

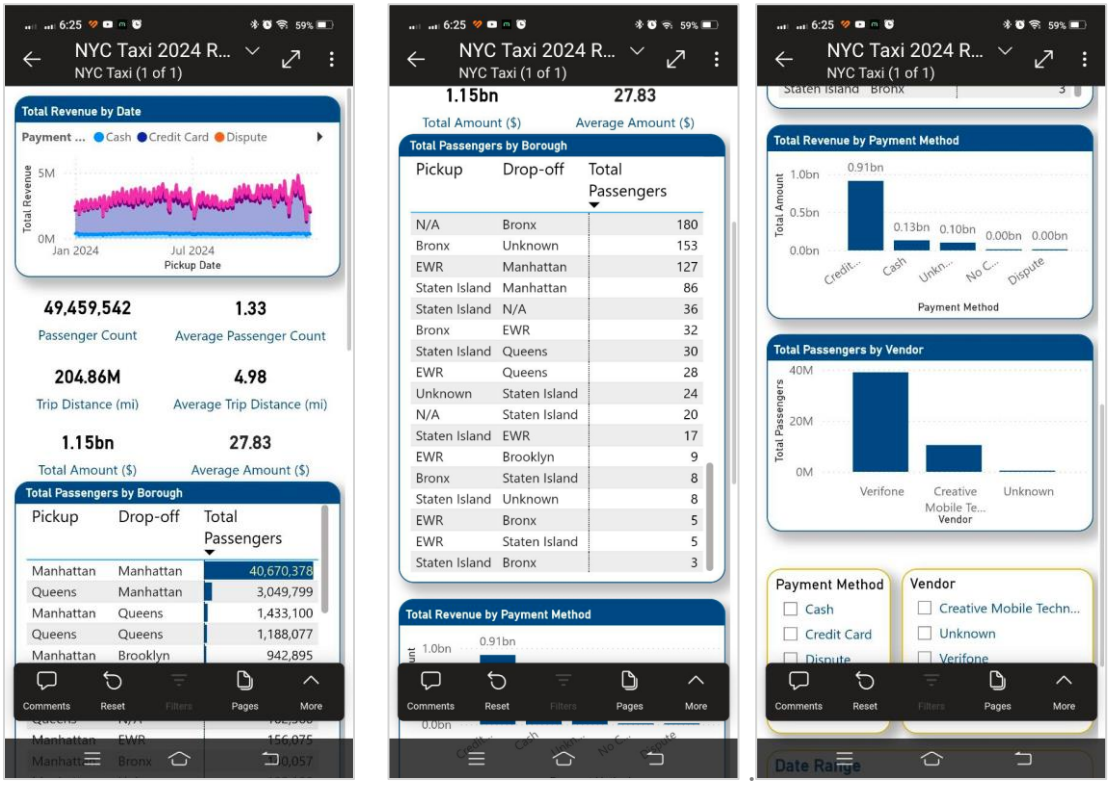
- Built on the **default semantic model** from the presentation table.



- Metrics visualized:
  - 📊 **Revenue Trends** (monthly/yearly)
  - 🚕 **Total and Average Passengers**
  - 💰 **Total and Average Revenue**
  - 🛣️ **Trip Distance Totals & Averages**



Power BI Report Desktop Version



Power BI Report Mobile Version

⚙ Tools and Skills Utilized in Creating This Project

This project leveraged a modern data architecture built on Microsoft Fabric, utilizing a diverse set of tools and skills to ensure efficient, scalable, and automated data processing from ingestion to visualization.

To store and manage raw and curated data, I implemented a Lakehouse architecture within Microsoft Fabric, providing the flexibility of both structured and unstructured data storage. For more structured reporting and querying needs, I also utilized Microsoft Fabric Warehouse, enabling robust data warehousing capabilities.

The ETL pipelines were built using Microsoft Fabric Data Pipelines, which facilitated seamless data movement across systems. Within these pipelines, Dataflows were used to perform data transformation, ensuring clean, structured, and enriched data ready for consumption.

We employed T-SQL stored procedures for advanced scripting requirements, allowing for efficient querying and data manipulation directly within the warehouse environment. For managing complex workflows, pipeline orchestration with variables was implemented, enabling conditional logic and dynamic execution paths throughout the data process.

A custom metadata table was created to handle metadata management, with dynamic logic incorporated to drive scalable and reusable pipeline behaviors. This helped to minimize hardcoding and ensured flexibility as data sources and structures evolved.

For data visualization, Power BI was integrated directly with the semantic model, providing rich, interactive dashboards and reports that reflect real-time data insights. This allowed stakeholders to derive meaningful conclusions and make data-driven decisions.

Lastly, the project emphasized automation and scalability, with dynamic logic built in to support monthly file processing. This ensured that the system could adapt to changing data loads and growing datasets without the need for constant manual intervention.

Overall, the combination of these tools and skills created a powerful, end-to-end data solution that is robust, scalable, and aligned with modern data platform best practices.

💡 How This Project Enhanced My Skills as a Data Engineer

Working on this project has significantly sharpened and expanded my skill set as a Data Engineer, particularly in designing and building modern, scalable data solutions using Microsoft Fabric.

Throughout the project, I gained hands-on experience with **Lakehouse architecture**, allowing me to effectively manage both structured and semi-structured data within a unified platform. This gave me a deeper understanding of how to balance flexibility with performance in data storage strategies.

I also deepened my proficiency in **data warehousing** through the use of **Microsoft Fabric Warehouse**, optimizing query performance and storage efficiency for large-scale data reporting.

Building **ETL pipelines** using **Microsoft Fabric Data Pipelines** taught me how to efficiently orchestrate and automate data movement, transformation, and integration processes. I implemented **Dataflows** for seamless transformation logic, while also enhancing my scripting skills through **T-SQL stored procedures**, solving complex business logic scenarios with precision and clarity.

One of the key skills I developed was **pipeline orchestration with variables**, which allowed for dynamic and intelligent workflow management. This required a solid understanding of logic design and helped me build systems that are resilient, adaptive, and easy to maintain.

By designing a **custom metadata management layer**, I introduced a reusable, scalable approach to data ingestion and transformation. This not only minimized redundancy but also made the overall architecture more future-proof—a critical skill for any Data Engineer working in agile environments.

I also got to work closely with **Power BI**, connecting my backend data model to create impactful dashboards. This full-cycle experience—from data ingestion to visualization—gave me a holistic perspective on how data engineering directly supports business insights and decision-making.

What truly set this project apart was the opportunity to build **automated and scalable systems**, such as dynamic monthly file processing logic. This taught me how to design solutions that minimize manual intervention and gracefully handle growing data volumes—skills that are essential for enterprise-level data platforms.

About Me

I am **Randy A. Velasco**, a seasoned Professional Electronics Engineer with a strong foundation in data governance, data engineering, data analysis, and database management. I am a certified Data Engineer and Data Analyst. I am also certified in Python, SQL, Data Literacy, Microsoft Azure AI Fundamentals (AI-900) and Microsoft Azure Fundamentals (AZ-900).

By working on end-to-end projects like this, I mimic real-world workflows, such as designing pipelines, ensuring data quality, and presenting insights. It solidifies my understanding of tools, technologies and techniques, such as ETL processes, data pipelines, and visualization tools like Power BI. I gained experience handling issues like missing data, data cleaning, and integrating multiple sources into cohesive datasets. Tackling real-world scenarios helps you bridge the gap between theory and practice, which is invaluable in mastering concepts like SQL queries, cloud technologies, and Python data manipulation.