End-to-End ETL Pipeline Using Microsoft Fabric Lakehouse, Dataflows, and PowerBI

Introduction

In this project, I designed and implemented a complete, automated ETL (Extract, Transform, Load) pipeline.

The goal was to create a robust system for handling monthly sales data, ensuring data quality, eliminating duplicates, and refreshing Power BI reports without manual intervention.

I used Microsoft Fabric Lakehouse, Dataflows, Power BI Desktop and Service, and Pipelines to achieve this.

This document outlines the challenges faced, solutions implemented, workflow architecture, and detailed steps of the ETL process.

Challenges Faced

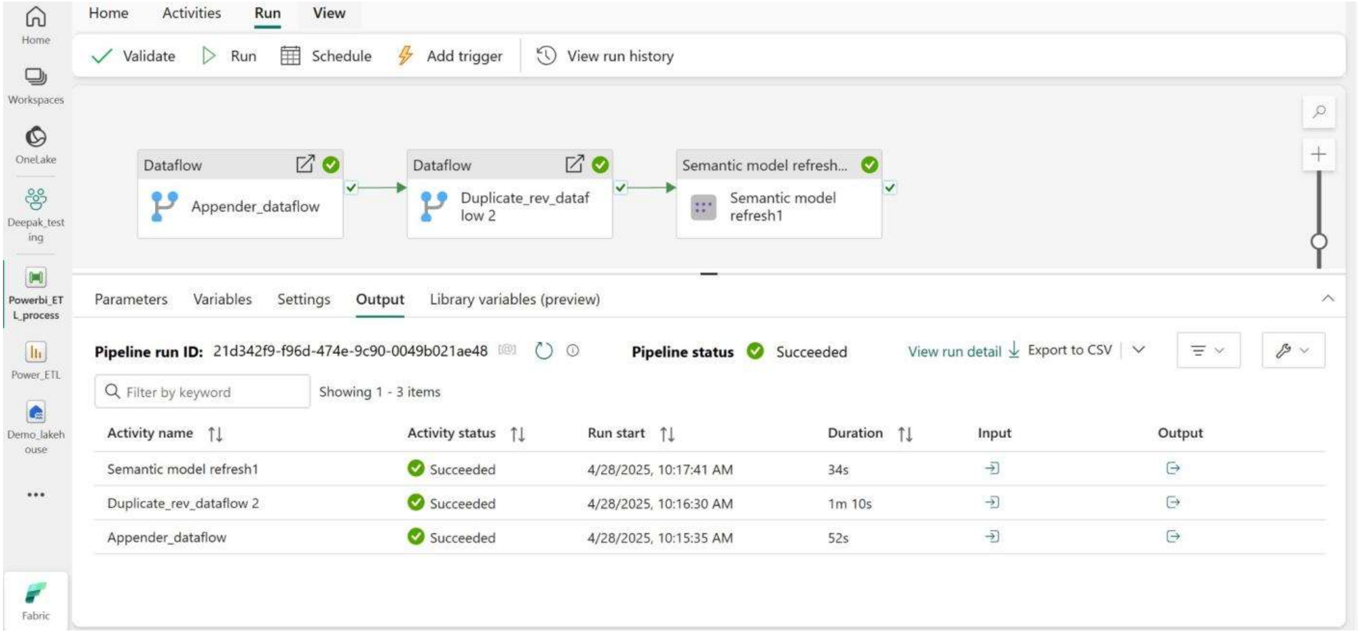
- Data Silos: Monthly sales files were stored separately, making aggregation difficult.
- Manual Data Refresh: Each month, new data needed manual update and validation in Power BI.
- Duplicate Records: Repeated entries from different CSV files led to data inconsistency.
- Scalability Issues: The system had to handle growing sales data over time.
- Automation Need: Reducing human error and building an automated, reliable process was critical.

Solution Overview

The solution was to build a modular, automated, and scalable ETL system:

- Store monthly sales data files in a Lakehouse under a structured folder.
- Create two Dataflows:
  - Dataflow 1 (Appender): Append all CSV files into a single dataset.
  - Dataflow 2 (Duplicate Removal): Remove duplicate records from the appended data.
- Load cleaned data into Power BI Desktop for report creation.
- Publish the report to Power BI Service.
- Create a Fabric Pipeline to automate the refresh of: Dataflow 1 -> Dataflow 2 -> Power BI Semantic Model.
- Thus, every new file added to Lakehouse is automatically picked up and reflected in the Power BI dashboard.

Architecture Diagram



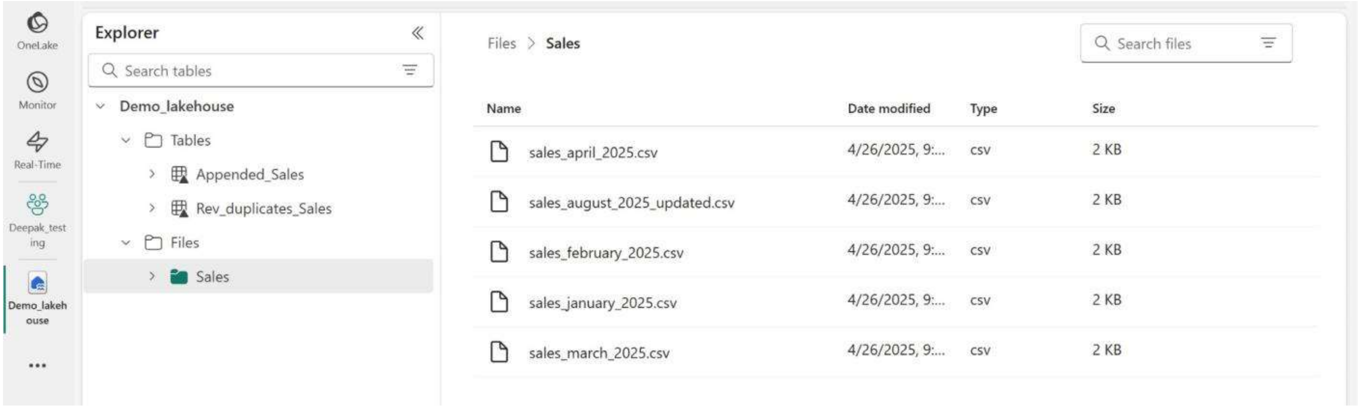
Pipeline run ID: 21d342f9-f96d-474e-9c90-0049b021ae48

Pipeline status ✓ Succeeded

Showing 1 - 3 items

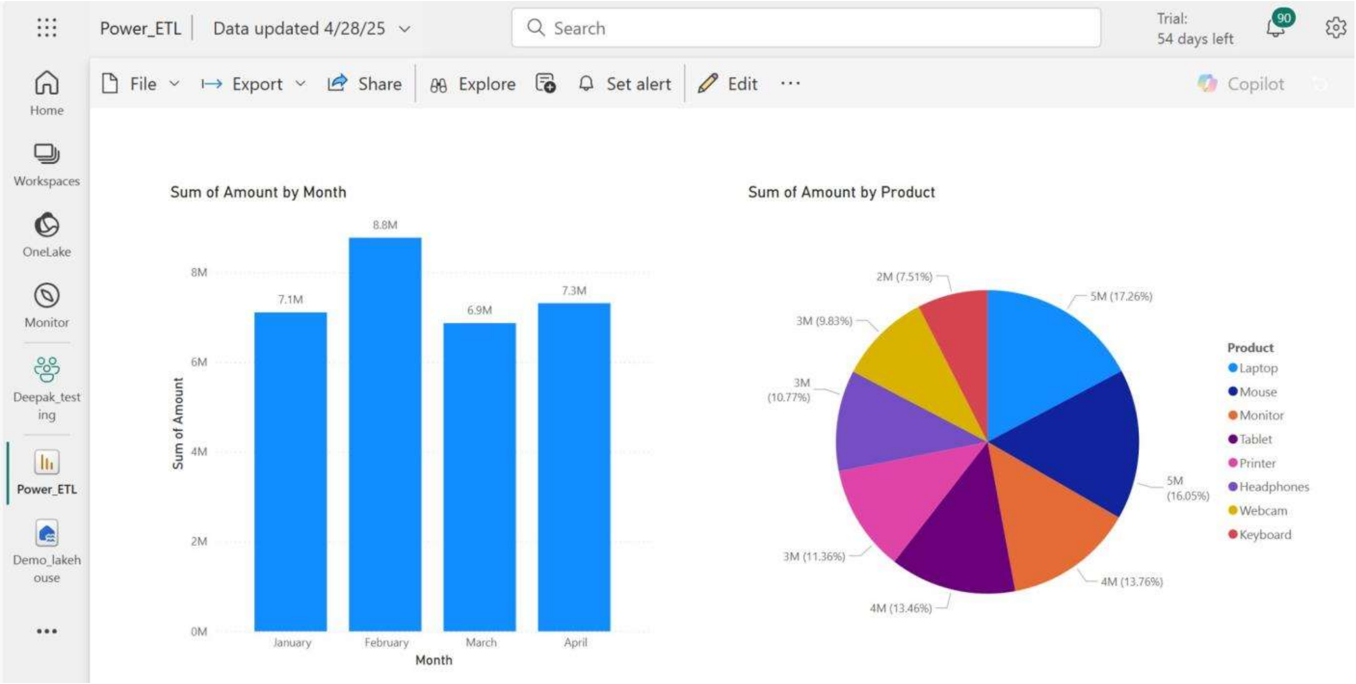
Activity name	Activity status	Run start	Duration	Input	Output
Semantic model refresh1	✓ Succeeded	4/28/2025, 10:17:41 AM	34s	⇥	↦
Duplicate_rev_dataflow 2	✓ Succeeded	4/28/2025, 10:16:30 AM	1m 10s	⇥	↦
Appender_dataflow	✓ Succeeded	4/28/2025, 10:15:35 AM	52s	⇥	↦

Data Storage Setup (Lakehouse Structure)



Name	Date modified	Type	Size
sales_april_2025.csv	4/26/2025, 9:...	csv	2 KB
sales_august_2025_updated.csv	4/26/2025, 9:...	csv	2 KB
sales_february_2025.csv	4/26/2025, 9:...	csv	2 KB
sales_january_2025.csv	4/26/2025, 9:...	csv	2 KB
sales_march_2025.csv	4/26/2025, 9:...	csv	2 KB

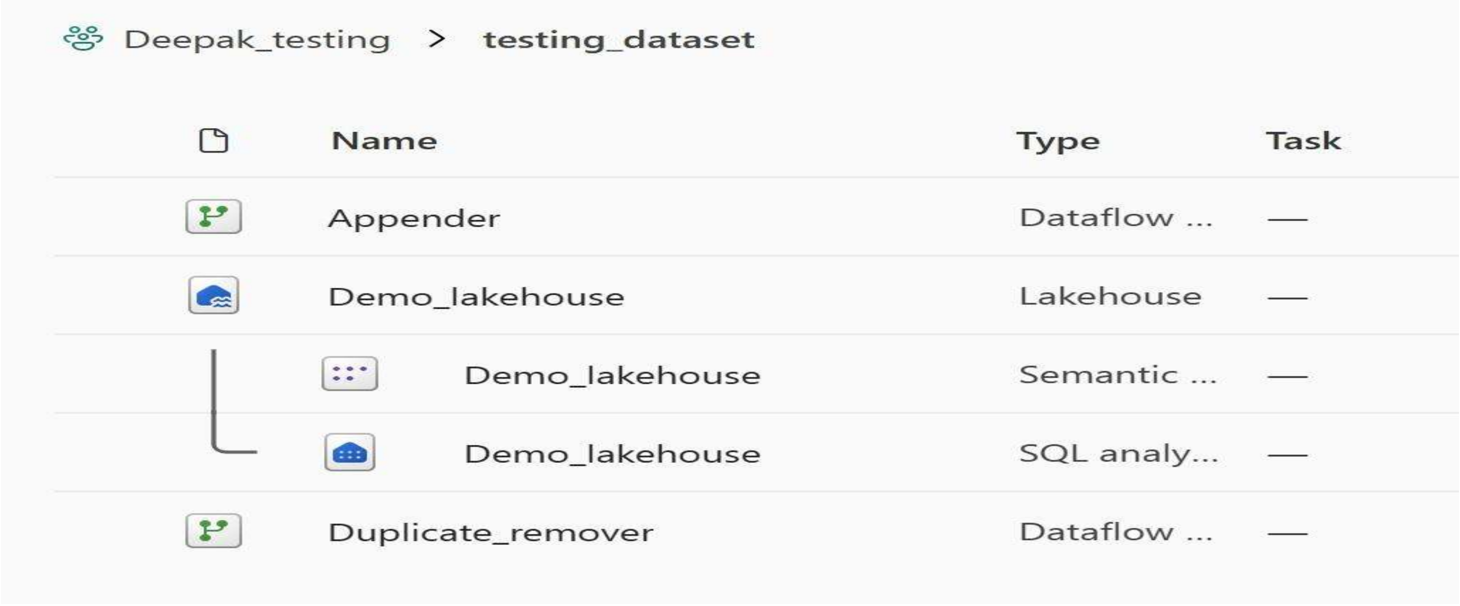
ETL Process Deep Dive



Power BI Integration

- Connected Power BI Desktop directly to Dataflow 2 (cleaned data).
- Built interactive visualizations and KPIs on top of the clean dataset.
- Published the Power BI Report to Power BI Service workspace for online access and sharing.

Pipeline Setup for Automation



Deepak_testing > testing_dataset

	Name	Type	Task
	Appender	Dataflow ...	—
	Demo_lakehouse	Lakehouse	—
	Demo_lakehouse	Semantic ...	—
	Demo_lakehouse	SQL analy...	—
	Duplicate_remover	Dataflow ...	—

Conclusion

Through the implementation of this end-to-end ETL pipeline, I successfully:

- Automated the ingestion, transformation, and loading of monthly sales data.
- Eliminated duplicate records ensuring data reliability.
- Achieved seamless, automatic refresh of Power BI reports.
- Built a highly scalable and modular architecture.
- Improved efficiency, minimized human intervention, and enhanced overall data quality.

This project showcases strong skills in Microsoft Fabric, Data Engineering, ETL Design, and Power BI development