Azure Car Sales ETL Pipeline – Project Documentation

Author: Mohamed Alasmawi

Date: 20/1/2025

Version: 2.3

Contents

1. Introduction

Purpose of the Project

The goal of this project is to develop a scalable, cloud-based ETL pipeline for car sales data using Microsoft Azure. The process involves ingesting, cleaning, transforming, and storing raw data for analysis and visualization. Key services used include Azure Blob Storage, Azure SQL Database (hosted on an Azure VM), Azure Data Factory (ADF), and Power BI for visualization.

Objectives

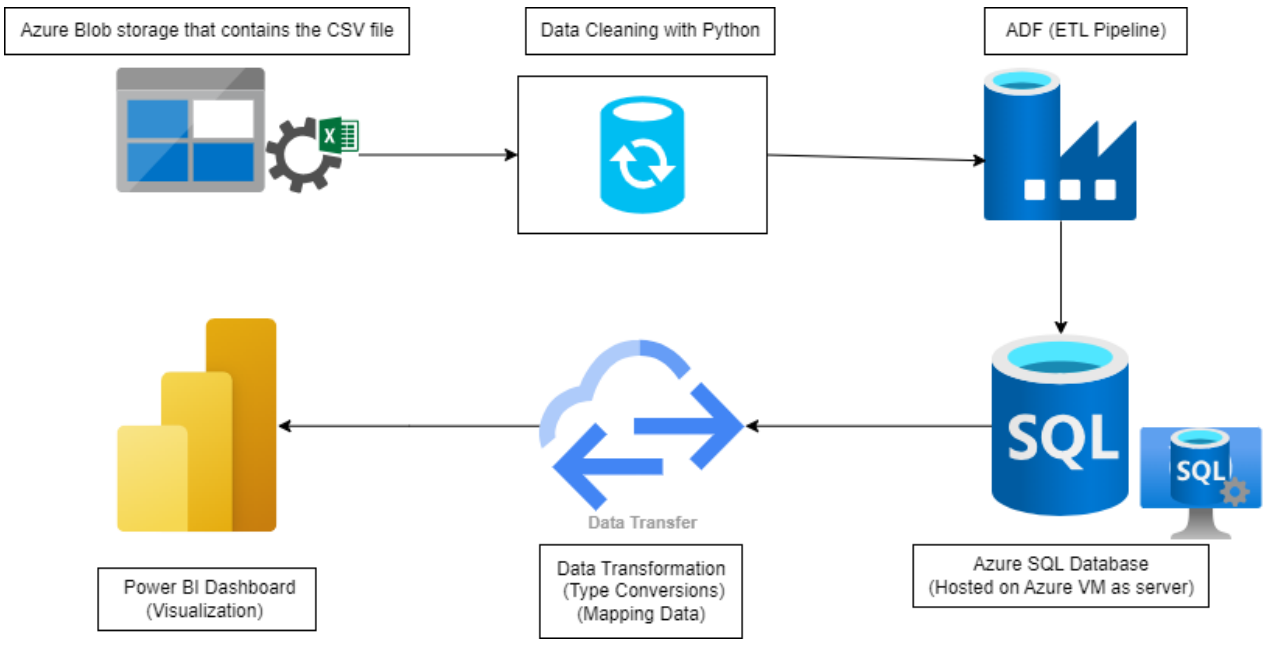
- Build an automated ETL pipeline with Azure Data Factory
- Store structured data in Azure SQL Database (hosted on VM)
- Visualize insights using Power BI
- Ensure data security and governance

2. System Architecture

Pipeline Overview

- Raw car sales data is stored in Azure Blob Storage.
- Data is cleaned and preprocessed using Python and Pandas.
- Cleaned data is loaded into an Azure SQL Database (hosted on Azure VM).
- Azure Data Factory automates ETL workflows.
- Power BI provides interactive dashboards for visualization.

Architecture Diagram



3. Data Preprocessing & Partitioning

Raw Data Format

The dataset contains columns such as:
Car_id, Date, Customer Name, Gender, Annual Income, Dealer_Name, Company, Model, Engine, Transmission, Color, Price ($), Dealer_No, Body Style, Phone, Dealer_Reg

Data Cleaning Steps

Using Python and Pandas:

- Handled missing values and corrected data types.
- Removed duplicate entries.
- Partitioned the dataset by key attributes such as Dealer_Name, Company, and Year.
- Standardized all attributes.

The cleaned dataset was saved as separate sheets in an Excel file for structured storage.

Uploading Process

- The cleaned dataset was uploaded to Azure Blob Storage for further processing.

- Note: you can find the python file for the cleaning in the github.

4. Database Design & Relationships

Database Schema (Azure SQL Database hosted on Azure VM)

- **Tables**
  - **Customers**: Customer_ID, Name, Gender, Annual_Income
  - **Dealers**: Dealer_ID, Dealer_Name, Dealer_Reg
  - **Cars**: Car_ID, Model, Engine, Transmission, Price, Body_Style, Color
  - **Sales**: Sale_ID, Date, Customer_ID, Dealer_ID, Car_ID, Price
- **Relationships**
  - **Customers → Sales** (1-to-Many)
  - **Dealers → Sales** (1-to-Many)
  - **Cars → Sales** (1-to-Many)

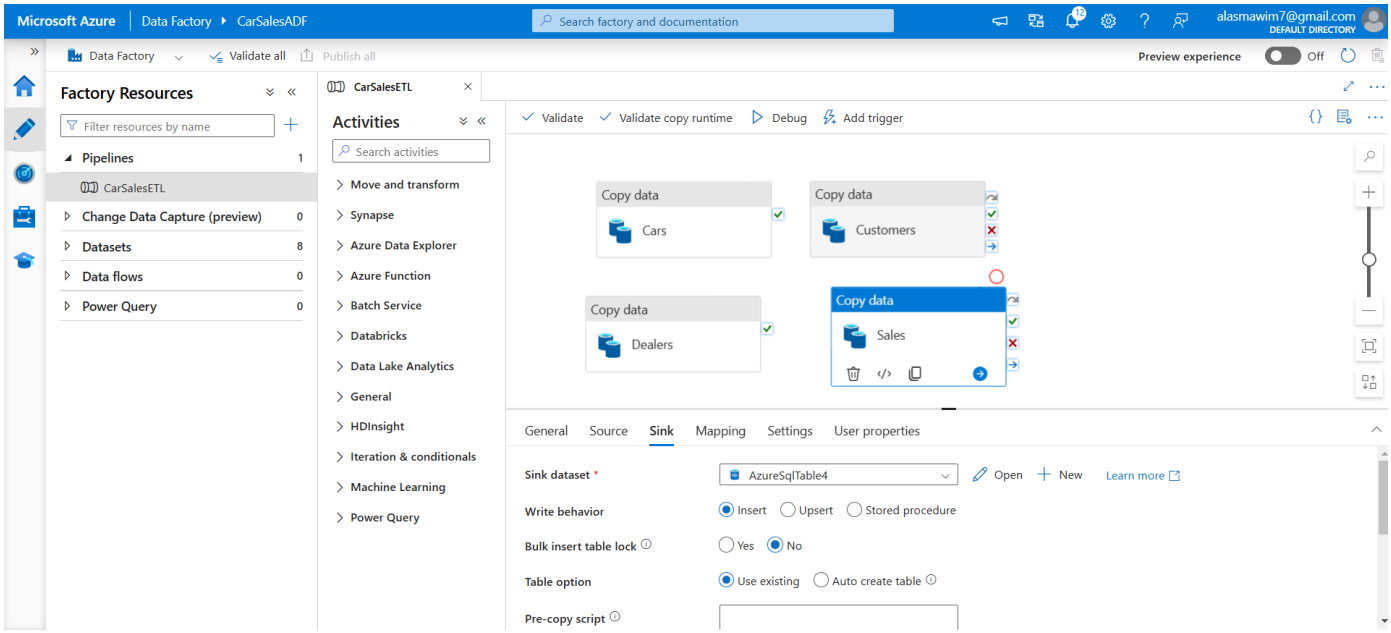
Foreign keys were defined in Azure SQL Database to ensure referential integrity.

Note: you can find the SQL file in the GitHub to create the database in Azure data studio

5. ETL Pipeline Using Azure Data Factory

Steps to Create the ETL Workflow

- **Create Azure Data Factory**
- **Set Up Linked Services**
  - Connect to Azure Blob Storage (for raw data)
  - Connect to Azure SQL Database (on VM)
- **Create Data Pipelines**
  - Extract data from Azure Blob Storage (raw CSV/Excel files)
  - Transform the data with mapping and type conversions
  - Load the transformed data into Azure SQL Database as structured tables
- **Define Data Transformations**
  - Applied mapping and data flows for transforming the raw data to fit the target schema.
- **Schedule & Automate Pipelines**
  - Automated the pipeline execution to run on scheduled intervals.
- **Pipeline Screenshot**



6. Data Server (Azure VM)

- **Why Azure VM for Hosting SQL Database?**
  Azure Virtual Machines are used to provide a scalable environment for hosting the SQL Server instance. This allows for efficient querying, storage, and processing of the structured data.

7. Data Visualization in Power BI

Connecting Power BI to Azure SQL Database

- Power BI was connected to the Azure SQL Database hosted on Azure VM.

Reports Created

- **Sales Performance**: Reports showing performance by dealer, brand, and region.
- **Customer Segmentation**: Insights based on income levels and customer demographics.

Note: you can find the PowerBI file in the GitHub to view the visuals

8. Security & Data Governance

Security Measures in Azure VM

- Azure Firewall was used to restrict unauthorized access to the VM.
- SSL Encryption ensured secure database connections.
- Role-Based Access Control (RBAC) provided access management for various users.

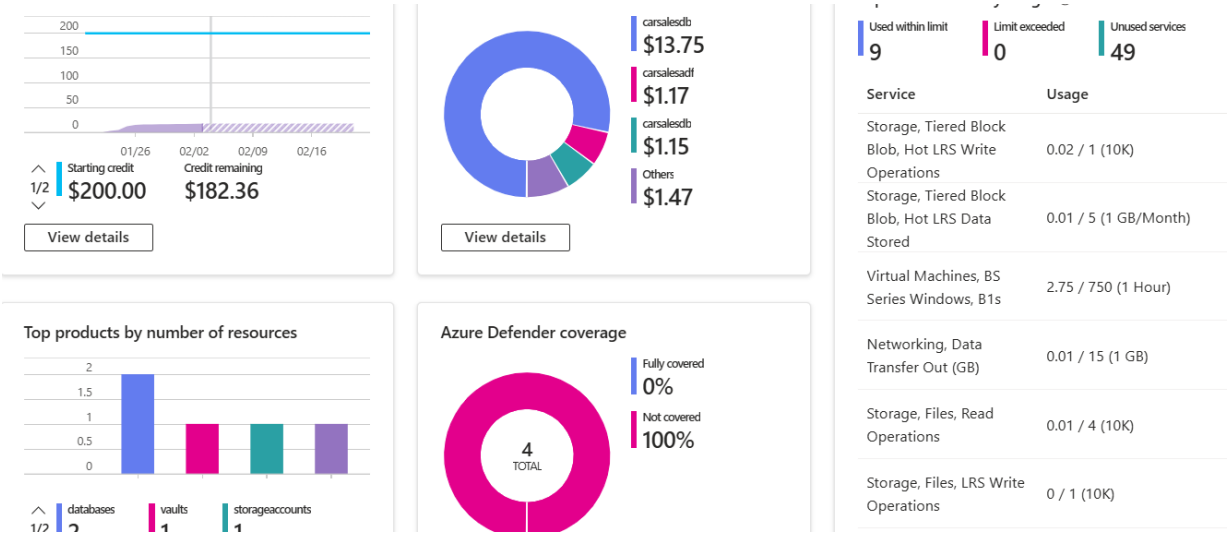
Data Compliance

- Ensured GDPR compliance for customer data privacy.
- Set up audit logs to track changes and access to the data.

9. Deployment & Cost Optimization

Cost Optimization Strategies

- Chose a cost-efficient Azure VM size (B-series or D-series).
- Enabled Auto-pause for the SQL database during non-peak hours.
- Used Azure Reserved Instances to reduce long-term costs.
- Optimized SQL queries to reduce compute costs.



Deployment Steps

- Deployed Azure Blob Storage.
- Set up and automated ETL pipelines using Azure Data Factory.
- Hosted the SQL Database on Azure VM.
- Integrated Power BI for reporting and dashboards.

10. Challenges & Future Enhancements

Challenges Faced

- Handling large datasets and ensuring fast query performance.
- Time taken for initial data processing and loading.

Future Enhancements

- **Machine Learning Integration**:
  - Enhance predictive capabilities using Azure Machine Learning.
  - Implement more advanced models for sales forecasting and customer insights.
- **Real-Time Data Integration**:
  - Introduce real-time data streaming using Azure Event Hub for more immediate insights.
- **Enhanced Reporting**:
  - Expand Power BI dashboards to include more detailed car performance analytics.

11. References

- Microsoft Azure Documentation (https://learn.microsoft.com/en-us/azure/?product=popular)
- Power BI Official Guides (https://learn.microsoft.com/en-us/power-bi/)
- Pandas Documentation for Data Processing(https://pandas.pydata.org/)