# Retail Sales Dashboard – Pipeline Automation using Azure DevOps

## Introduction

The Retail Sales Dashboard project focuses on automating the process of analyzing, transforming, and reporting sales performance data through Azure DevOps. This project integrates data engineering and DevOps principles to enable weekly automated analysis of retail sales data, reducing manual workload and ensuring data-driven decision-making.

## Objective

The objective of this project is to design and implement an automated CI/CD pipeline in Azure DevOps to process, analyze, and report retail sales data. The pipeline executes weekly, generates sales insights, identifies the lowest-performing stores, and publishes structured reports for business review.

## Tools and Technologies Used

• Azure DevOps – for pipeline automation, work item tracking, and CI/CD integration  
• MySQL and MongoDB – for structured and unstructured retail data storage  
• Python (Pandas, NumPy) – for sales data cleaning, preprocessing, and transformation  
• PySpark – for store-level analytics and ETL processing  
• Azure Databricks – for running large-scale data processing jobs

## Implementation Overview

The project was executed as part of the capstone activities under the HexaDevOps environment. It involved creating structured datasets, developing analytical scripts, performing ETL operations, and automating the analysis process using Azure DevOps pipelines.

### 1. Data Preparation and Storage

This phase involved creating and managing databases in MySQL and MongoDB for retail products, stores, and sales. Data integrity and consistency were ensured through validation and relationship mapping between tables.

### 2. Python Analysis and Cleaning

Python scripts were developed using Pandas to clean and preprocess the sales data. Missing values, duplicates, and datatype inconsistencies were handled, preparing the dataset for further analysis. The cleaned data was exported to CSV format for integration with downstream processes.

### 3. PySpark and ETL Transformation

PySpark was used to perform store-wise sales analysis and compute aggregated metrics such as total revenue, profit, and product performance. An ETL job was created in Azure Databricks to automate data extraction, transformation, and loading.

### 4. Azure DevOps Pipeline Automation

A YAML-based pipeline was created in Azure DevOps to automate weekly sales analysis. The pipeline triggers a Python script that processes sales data, exports insights as CSV files, and identifies the top five lowest-performing stores. The workflow ensures consistency and reliability in weekly performance reports.

### 5. Reporting and Alerts

The pipeline includes an automated step that logs or emails the top five lowest-performing stores. This ensures that management teams receive timely alerts for corrective actions. The final CSV reports are stored as Azure pipeline artifacts and can be integrated with visualization tools like Power BI for dashboard creation.

## Work Items in Azure DevOps

All project activities were tracked in Azure DevOps using a hierarchical work item structure under a single Epic titled 'Retail Sales Dashboard Automation'. Each major activity was categorized as a Feature, followed by specific User Stories and detailed Tasks. This ensured clear progress tracking, accountability, and documentation of the entire development workflow.

The work items were categorized as follows:  
• Epic – Retail Sales Dashboard Automation  
• Features – Data Preparation, Analysis, ETL, Pipeline Automation, and Reporting  
• User Stories – Defined under each feature to specify business logic and functionality  
• Tasks – Action items that contributed to completing each user story

## Screenshots from Azure DevOps

Below are the screenshots representing the Azure DevOps board and pipeline automation work items for the Retail Sales Dashboard project:

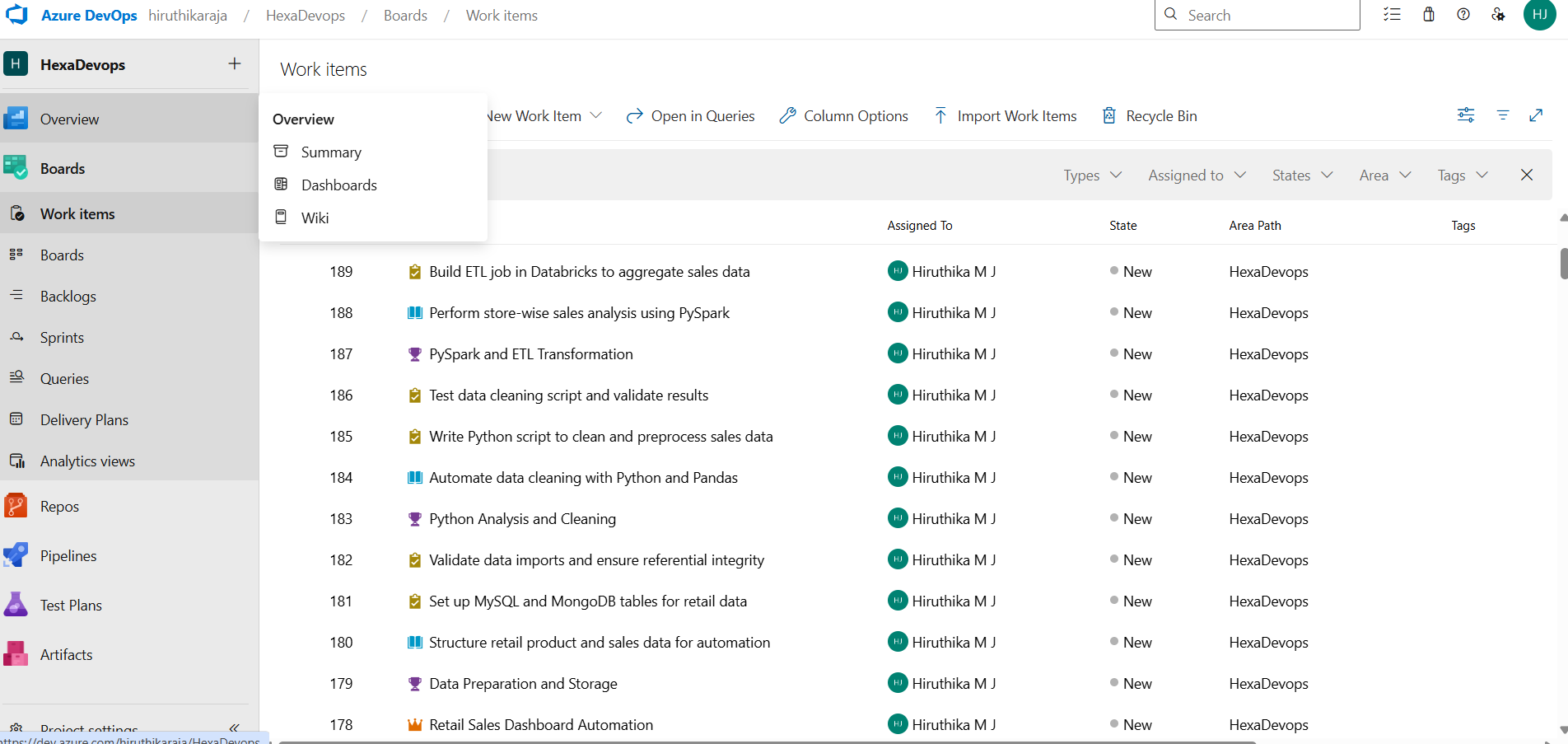


Figure 1: Azure DevOps Work Items for Retail Sales Dashboard Automation

This screenshot shows all created work items under the HexaDevOps project, categorized into Epics, Features, User Stories, and Tasks.

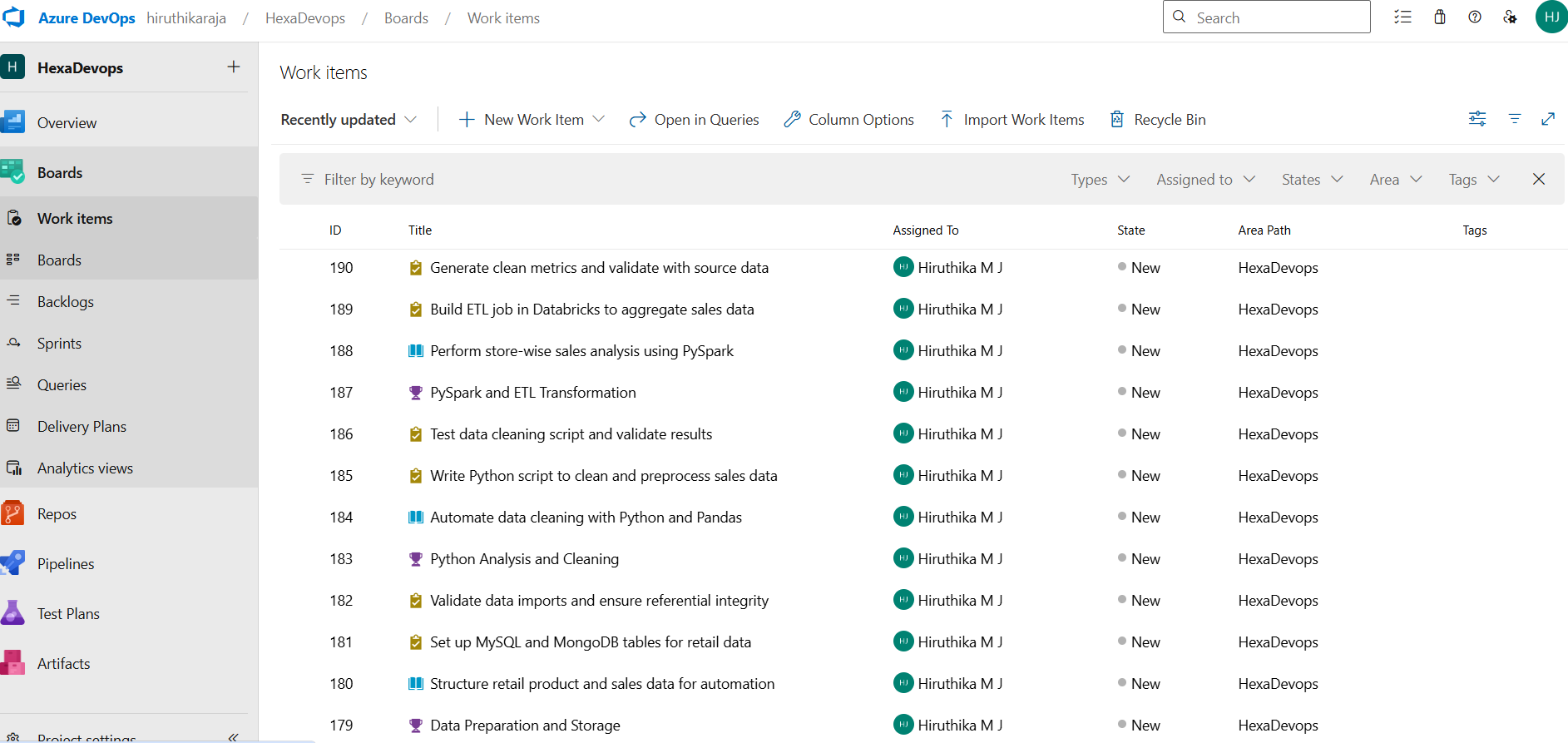


Figure 2: Azure DevOps Work Items for Pipeline Automation

This screenshot displays the work items specifically related to pipeline automation, including YAML configuration and reporting tasks.

## Outcome

The project successfully established an automated workflow for retail sales analysis. With the Azure DevOps pipeline, weekly reports are generated without manual intervention, providing consistent insights into store performance. The system ensures accurate data tracking, effective resource utilization, and enhanced decision-making for sales optimization.

## Conclusion

The Retail Sales Dashboard project demonstrates the effective use of Azure DevOps for data-driven automation in retail analytics. By integrating Python, PySpark, Databricks, and DevOps tools, the project achieved full lifecycle automation — from data preparation to reporting. This implementation reinforces the value of CI/CD principles in data engineering and business intelligence solutions.