**Data Engineering Project 1**

**Real-time Data Processing**

* **Harshini V**

**PROJECT OVERVIEW:**

This project aims to create a real-time data processing solution for analysing streaming datasets using Azure Data Factory and Azure Databricks. The solution involves ingesting streaming data into Azure through Azure Data Factory and processing it using Azure Databricks with Spark Streaming. The processed data is then analysed in near-real time to derive actionable insights. This solution demonstrates the capabilities of Azure services for building scalable and efficient real-time data pipelines. Data transformations and analytics are performed using Databricks notebooks, while Azure Data Factory orchestrates the entire workflow.

**DATA OVERVIEW:**

The **Online Retail dataset** contains transactional data from a UK-based e-commerce store spanning the years 2009 to 2011. This dataset provides valuable insights into online purchasing behavior, capturing key details such as invoice numbers, product descriptions, customer IDs, transaction dates, quantities, and prices. The data primarily focuses on business-to-business (B2B) transactions, often involving bulk purchases, making it an excellent resource for analyzing large-scale retail operations. With a two-year time frame, the dataset offers a robust foundation for identifying trends and patterns over time.

Key fields in the dataset include:

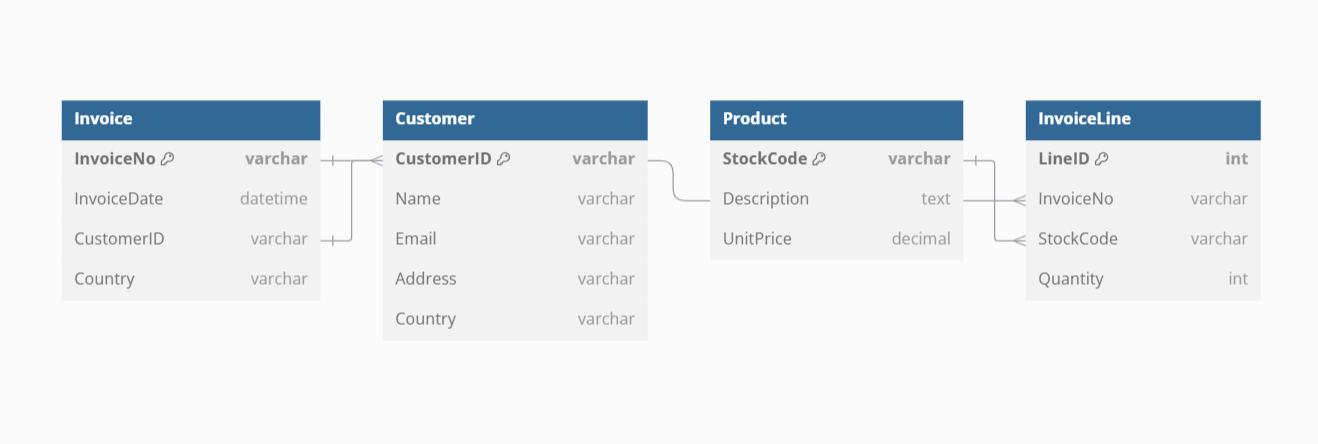
1. **InvoiceNo** - a unique identifier for each transaction
2. **StockCode** - representing each product
3. **Description** - which provides textual details about the products
4. **Quantity** - denoting the number of items purchased in a transaction
5. **InvoiceDate** - the timestamp of the transaction
6. **UnitPrice** - indicating the price per unit of the product
7. **CustomerID** - which identifies the purchasing customer
8. **Country** - representing the customer's location.

Given its comprehensive structure, this dataset is ideal for simulating a real-time streaming pipeline, offering opportunities to analyze customer purchasing behavior, product sales trends, and geographical demand patterns in near-real-time.

**ARCHITECTURE DIAGRAM:**

**ER Diagram:**

The structure of the normalized database is shown in the following ER Diagram.

****

**HOW IT WORKS:**

**Source Data Files:**

We are referring to a dataset from a UK-based e-commerce store spanning the years 2009 to 2011 (from Kaggle).The data primarily focuses on business-to-business (B2B) transactions.

**File Name File Type**

Online Retail Dataset 2009-10 CSV

Online Retail Dataset 2010-11 CSV

**EXECUTION OVERVIEW:**

* Azure Data Factory (ADF) orchestrates the execution and monitoring of Azure Databricks notebooks. The data pipeline begins with importing datasets from Kaggle into Azure Data Lake Storage Gen2 (ADLS), where raw data is stored in the **Bronze zone** (landing zone).
* Data from the Bronze zone is processed using Azure Databricks notebooks. In the first notebook, the raw data is cleaned and merged into a unified table, with the output stored in Delta tables within the **Silver zone** (standardization zone).
* The Silver zone data is further transformed using Azure Databricks PySpark and Spark SQL in the second notebook. The cleaned and structured data is joined, aggregated, and prepared for analytical and visualization purposes. Finally, the transformed data is loaded into the **Gold zone** (analytical zone).

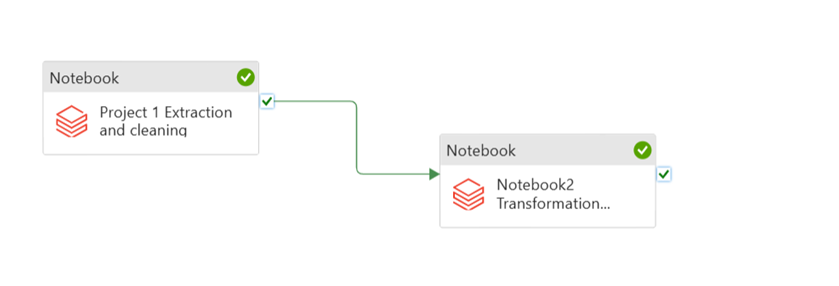
**ETL PIPELINE:**

ETL flow comprises two parts:

* Ingestion: Process data from **Bronze zone** to **Silver zone**
* Transformation: Process data from **Silver zone** to **Gold zone**

In the pipeline, data stored in CSV format is read using Apache Spark in the first Azure Databricks notebook. Minimal transformations, such as dropping unnecessary columns, renaming headers, applying schemas, and adding audit columns (`ingestion\_date` and `file\_source`), along with the parameterized `file\_date`, are applied. The cleaned and merged data is saved as a Delta table. This notebook serves as the extraction and cleaning step.

The second Azure Databricks notebook processes the cleaned Delta table to transform it into the final dimensional model tables in Delta format. Transformations performed include dropping duplicates, joining tables using `join`, and aggregating data with window functions.

The ADF pipeline is designed to execute Notebook 2 only if Notebook 1 completes successfully to ensure a sequential and error-free workflow. ADF is scheduled to run every Wednesday at 11.20 AM and is configured to skip execution if there is no relevant data (e.g., no new files for the specified `file\_date`). The `file\_date` parameter is passed dynamically as part of a tumbling window trigger for managing the pipeline execution. 

**AZURE RESOURCES USED FOR THIS PROJECT:**

* Azure Data Lake Storage
* Azure Data Factory
* Azure Databricks

**PROJECT REQUIREMENTS:**

The requirements for this project are broken down into six different parts which are

**1. Data Ingestion Requirements**

* Ingest all 2 files into Azure data factory.
* Ingested data must have the same schema applied.
* Ingested data must be stored in csv format..
* We must be able to analyze the ingested data via Spark SQL and PySpark.

**2. Data cleaning and Transformation Requirements**

* Join the key information required for reporting to create a new table.
* Remove the null values and format the column names.
* We must be able to analyze the transformed data via Spark SQL and PySpark.
* Cleaned and Transformed data must be stored in a new delta table.
* Transformation logic must be able to handle the incremental load.

**3. Data Analysis Requirements**

* Create Databricks dashboards..
* Find the Dominant drivers.
* Find the Dominant Teams.
* Visualize the Outputs.

**4. Pipeline Requirements:**

* Scheduled to run every Wednesday at 11.20 AM.
* Create ADF pipeline
* Add the activities to the pipeline
* Publish the pipeline and schedule it.

**5. Scheduling Requirements**

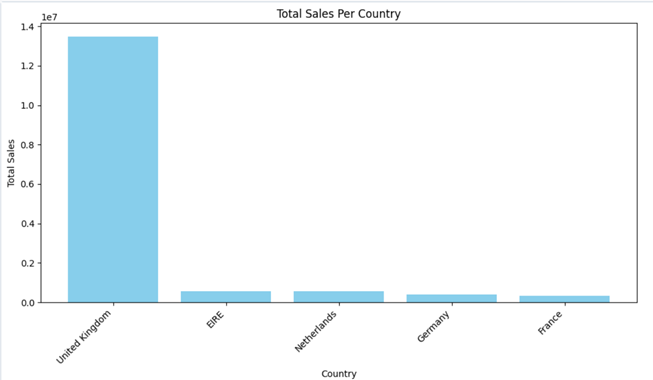
* Scheduled to run every Wednesday at 11.20 AM.
* Ability to monitor pipelines.
* Ability to rerun failed pipelines.
* Ability to set up alerts on failures

**6. Other Non-Functional Requirements**

* Ability to delete individual records
* Ability to see history and have backups
* Ability to roll back to a previous version

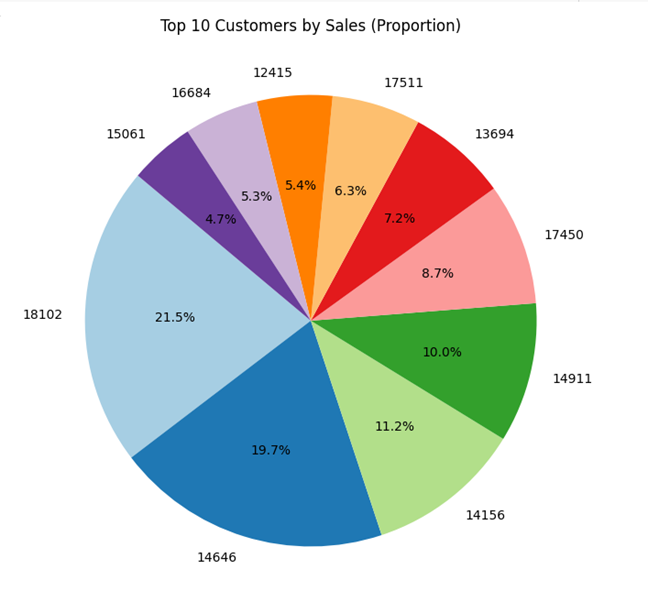
**ANALYSIS RESULT:**

**Total Sales Per Country:**



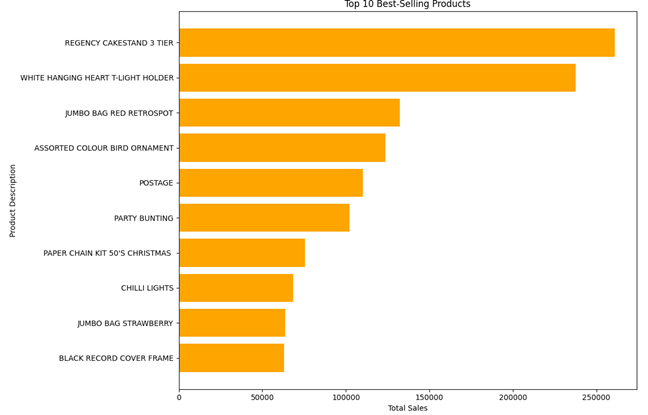
This bar chart illustrates the total sales achieved in each country. It provides a comparative analysis of sales performance across different regions, showcasing the countries with the highest and lowest sales.

**Top 10 Customers by Sales:**



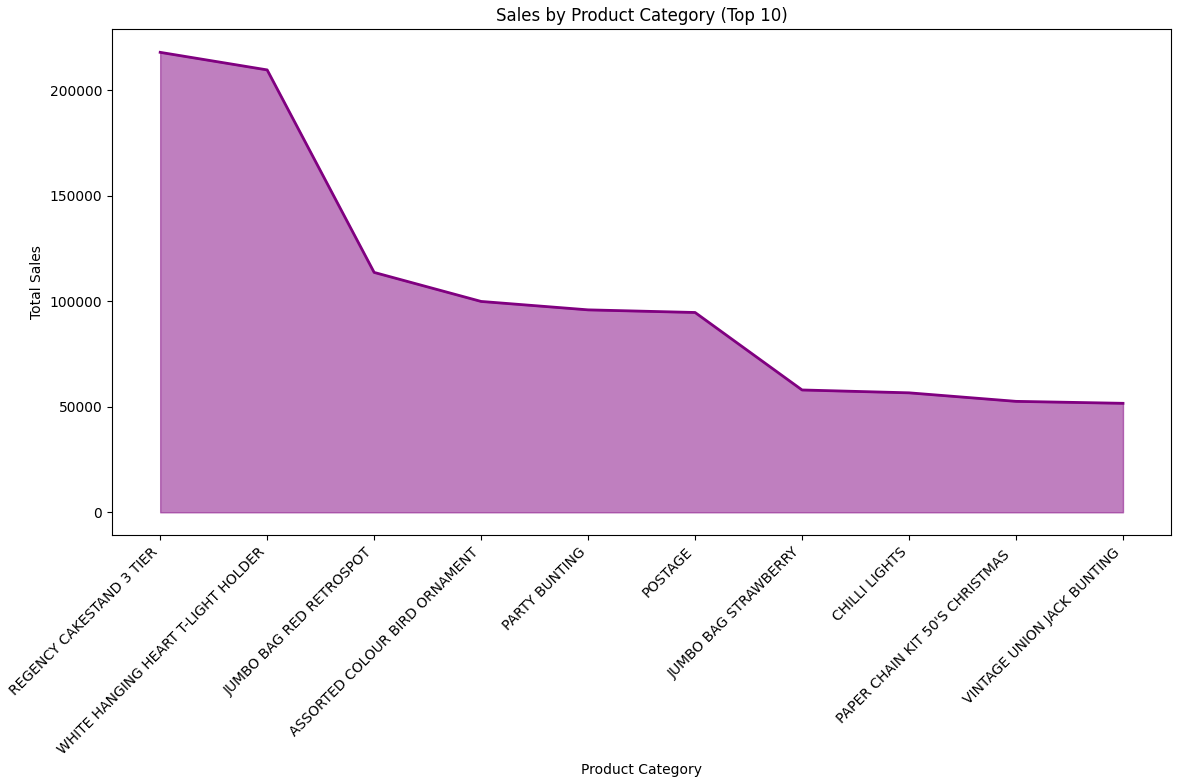
The pie chart represents the proportion of total sales contributed by the top 10 customers. It visually depicts the dominance of individual customers in the overall sales figures. Here we can see the total sales contributed by the top 10 customers by their customer IDs.

**Top 10 Best-Selling Products:**



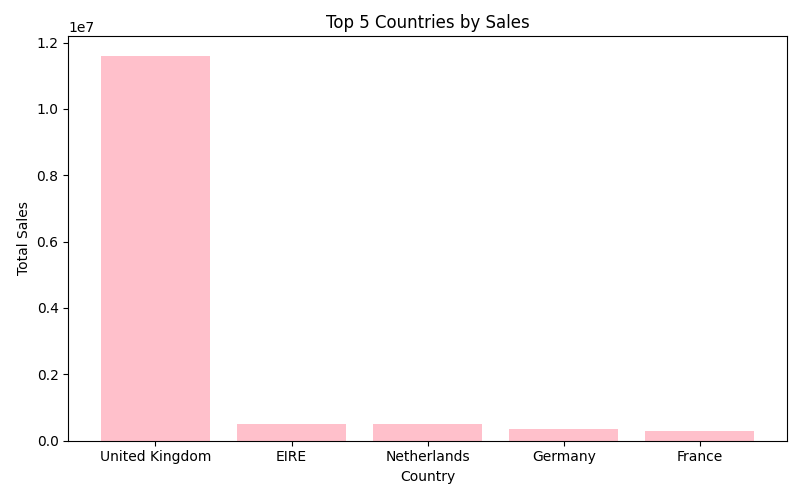
This horizontal bar chart highlights the top 10 best-selling products based on their total sales. The inverted y-axis enhances readability, making it easy to identify the most popular products.

**Sales by Product Category:**



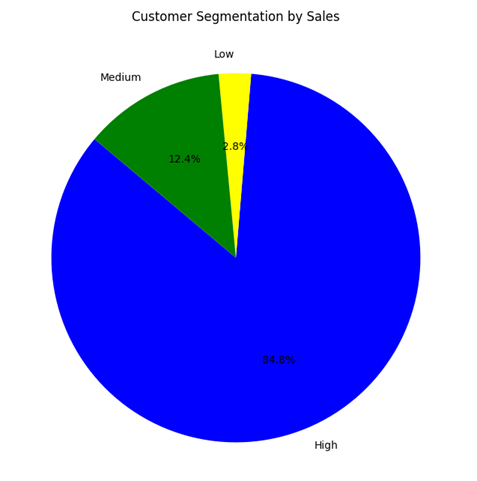
This area chart represents the top 10 product categories by total sales, highlighting their contribution visually. The shaded area and line emphasize the differences in sales among the categories.

**Top 5 Countries by Sales:**



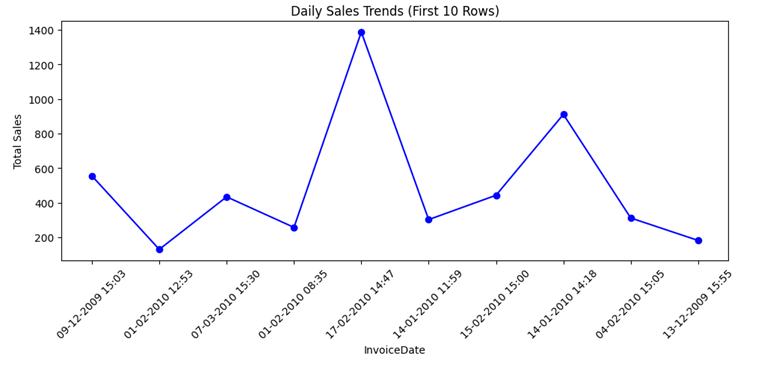
This bar chart showcases the top five countries with the highest total sales. It allows quick identification of the most lucrative markets for the business.

**Customer Segmentation by Sales:**



The pie chart categorizes customers into different sales segments: Low, Medium, and High. This segmentation helps understand the distribution of customers based on their purchase value.

**Daily Sales Trends:**



The line chart displays sales trends over the first 10 days of data. It reveals patterns in sales activity, helping to identify peak and low-performance days.