NED UNIVERSITY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND IT

FYDS (with Specialization in Data Science)

CT-472 Data Warehousing & Business Intelligence

PROJECT REPORT

Unified E-commerce Analytics: A Data Warehouse for Integrated Sales and Product Insights

SUBMITTED BY:

Ebaad Khan	**DT-22045**
Ezaan Khan	**DT-22046**
Syed Ahmed Ali	**DT-22301**
Muhammad Khuzaima Hassan	**DT-22302**

Submitted to: Dr. Umer Farooq

Contents

Objective

The primary objective of this project is to strategically design, develop, and deploy a high-performance data warehouse system. This system is engineered to break down data silos by integrating disparate sales, customer, and product data from various transactional sources. By centralizing this information into a unified and optimized Star Schema model, the warehouse will serve as the single source of truth for the organization, enabling efficient, complex analytical reporting, powering interactive visualizations through Power BI, and serving as a foundation for advanced machine learning applications.

Project Overview

This project addresses the critical business need for integrated analytics. In a typical e-commerce environment, customer data, product metadata, and sales transactions often reside in separate, non-communicating systems (OLTP databases, CSV files, third-party spreadsheets). This project focuses on building a robust Sales Data Warehouse to consolidate these assets.

The core of the project is a comprehensive ETL (Extract, Transform, Load) pipeline that extracts raw data, applies rigorous cleaning and business logic transformations, and loads it into a structured data warehouse. This clean, reliable data foundation enables both descriptive analytics (via Power BI dashboards) and advanced analytics (via direct ML model application). The ultimate goal is to empower business leaders with actionable insights into sales performance, customer behavior, and profitability trends.

1. Data Warehouse Architecture

Architecture Type: Hybrid Data Warehouse Architecture

This project implements a **Hybrid Architecture**, which strategically combines the strengths of both on-premise and cloud-based systems. This model allows transactional source systems (OLTP) to remain secure and performant within a local environment, while the analytical workload (ETL, Data Warehouse storage, and BI) is offloaded to a scalable, flexible, and cost-efficient cloud platform. This approach provides optimal performance for both operations and analytics.

Layers of the System

The architecture is logically segmented into four distinct layers:

- **Source Layer:** This foundational layer consists of the heterogeneous source systems where data is generated. This includes the transactional database (OLTP), sales data from flat files (CSV), and rest APIs.
- **Staging Layer:** A crucial intermediary database layer that acts as a buffer between source systems and the data warehouse. This temporary storage area is where the primary data transformation (the "T" in ETL) occurs. Data is profiled, cleansed, deduplicated, standardized, and conformed before being loaded into the final warehouse.

- **Data Warehouse Layer (Core):** This is the permanent, structured repository for historical and analytical data, built on a **PostgreSQL** database. The data here is modeled into a **Star Schema**, which is optimized for high-speed, read-heavy analytical queries (OLAP).
- **Presentation Layer:** This is the user-facing BI and Analytics layer where data is consumed. It includes **Power BI** for interactive dashboards.

2. Star Schema Design

The Star Schema was chosen as the foundational model for its simplicity, ease of understanding for business users, and high-performance query capabilities. It consists of one central fact table surrounded by multiple descriptive dimension tables.

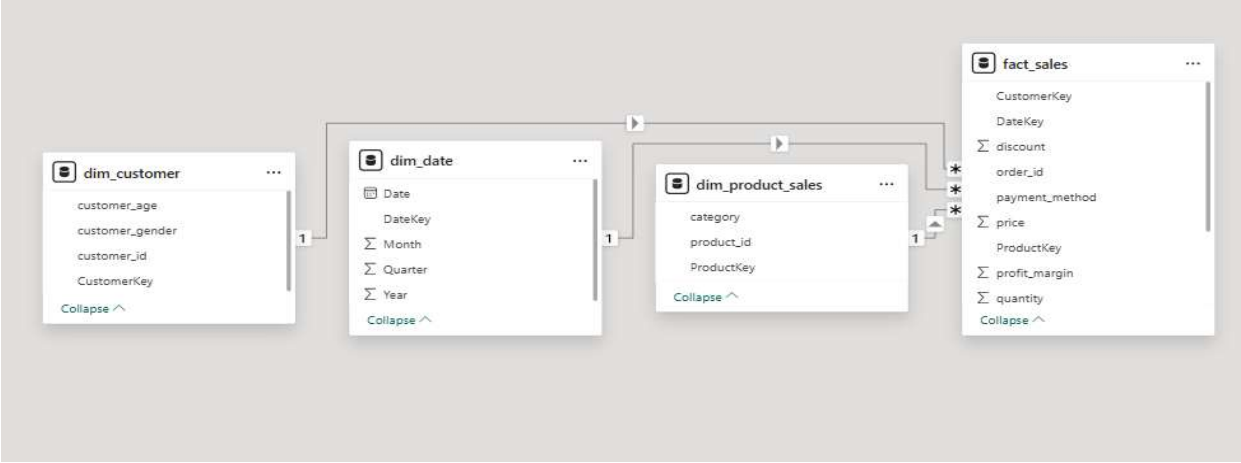
Fact Table:

- **fact_sales:** This is the central table and represents the core business *event*—a single sales transaction or line item. It contains:
  - **Quantitative Measures:** Additive and semi-additive metrics (the "facts") like price, discount, profit_margin, and quantity.
  - **Foreign Keys:** A composite primary key made up of foreign keys (CustomerKey, DateKey, ProductKey) that link to the surrounding dimension tables.
  - **Degenerate Dimensions:** Attributes from the source transaction, like order_id or payment_method, that are useful for analysis but don't warrant their own dimension table.

Dimension Tables:

These tables provide the descriptive context (the "who, what, when, and where") for the facts.

- **dim_customer:** This table stores all descriptive attributes related to customers. Each customer is assigned a unique, system-generated **surrogate key** (CustomerKey). Attributes include customer_age, customer_gender, and customer_id (as a business key).
- **dim_date:** A pre-populated, comprehensive date dimension critical for time-series analysis. It provides consistent temporal attributes for slicing data, including DateKey (PK), Date, Month, Quarter, Year, DayOfWeek, and IsHoliday.
- **dim_product_sales:** This table contains all metadata about the products sold. It uses a ProductKey (PK) to store details like product_id (business key), category, sub_category, and brand.



3. ETL Process Implementation

A robust ETL pipeline, developed using Colab, manages the flow of data from source to presentation.

- **Extract:** Data is extracted from the various source systems (OLTP DB, CSVs) using Python libraries (like Pandas for files and SQLAlchemy for the database). This process is designed to be incremental, pulling only new or updated records after the initial historical load.
- **Transform:** This is the most complex stage, occurring in the staging layer. Key transformations include:
  - **Data Cleansing:** Handling NULL or missing values, correcting erroneous data, and standardizing text (e.g., "Credit Card" vs. "CC").
  - **Deduplication:** Identifying and removing duplicate customer or product records.
  - **Business Rule Application:** Calculating new, derived metrics. For example, profit_margin was calculated using (price - unit_cost) / price.
  - **Key Generation:** Creating and looking up **surrogate keys** (e.g., CustomerKey, ProductKey, DateKey) to link the fact table to the dimensions.
- **Load:** Once transformed and validated, the clean data is loaded into the final PostgreSQL Star Schema. The dimension tables are loaded first, followed by the fact_sales table to ensure referential integrity.

4. Database Implementation

- **Platform: PostgreSQL hosted on Aiven (Cloud)** was selected as the database platform. Aiven provides a fully managed, scalable, and reliable cloud database service, which handles backups, security, and maintenance.
- **Tools: pgAdmin** was used as the primary database management and development tool for writing DDL (Data Definition Language) to create the schema, running ad-hoc queries for validation, and managing user permissions.
- **Primary Keys:** Each dimension table (dim_customer, dim_date, dim_product_sales) has a single, system-generated integer primary key (surrogate key) for fast and simple joins.
- **Foreign Keys:** The fact_sales table implements foreign key constraints that reference the primary keys of the dimension tables. This enforces **referential integrity** at the database level, preventing "orphan" fact records.

5. Power BI Dashboard

- **Integration:** Power BI was connected directly to the Aiven PostgreSQL database using the native PostgreSQL connector. An **Import** data model was used to load a snapshot of the data warehouse into Power BI's in-memory engine, providing maximum query performance.
- **Dashboard Highlights:** The dashboard was designed to provide a "top-down" analytical experience, starting with high-level KPIs and allowing users to drill down into details.
  - **KPIs:** At-a-glance metrics for Total Revenue, Average Profit Margin %, Total Orders, Unique Products Sold, and Total Customers.
  - **Visuals:**
    - **Monthly Revenue Trend (Line Chart):** To identify seasonality and growth patterns.
    - **Revenue by Category (Pie Chart):** To see which product categories drive revenue.
    - **Regional Sales (Bar Chart):** To pinpoint top-performing geographical markets.

- **Orders by Payment Method (Column Chart):** To understand customer payment preferences.
- **Discount vs Revenue (Combo Chart):** To analyze the impact of promotional strategies.
- **Detailed Transaction Table:** A drill-through page for granular analysis.
- **Relationships in Power BI:** The Star Schema's 1-to-many relationships were defined in the Power BI data model, connecting the "one" side (dimensions, e.g., dim_customer.CustomerKey) to the "many" side (fact table, e.g., fact_sales.CustomerKey).

6. Data Flow Diagram

The end-to-end flow of data follows this logical path:

1. **Sources:** Data originates in **Source Files (CSV), Rest API and OLTP Databases**.
2. **Extraction:** The ETL process performs **Data Extraction** and moves the raw data into the Staging Layer.
3. **Transformation:** In the staging area, **Data Transformation (Cleansing, Business Rules, Key Generation)** is applied.
4. **Load:** The clean, structured data is loaded into the **PostgreSQL Data Warehouse (Star Schema)**.
5. **Connection: Power BI Data Connection** (in Import mode) queries the warehouse for BI.
6. **Presentation:** Data is visualized in Power BI as **KPI Cards + Charts + Reports**.
7. **Value:** The process concludes by delivering **Actionable Business Insights** to end-users.

7. Key Insights and Analytics

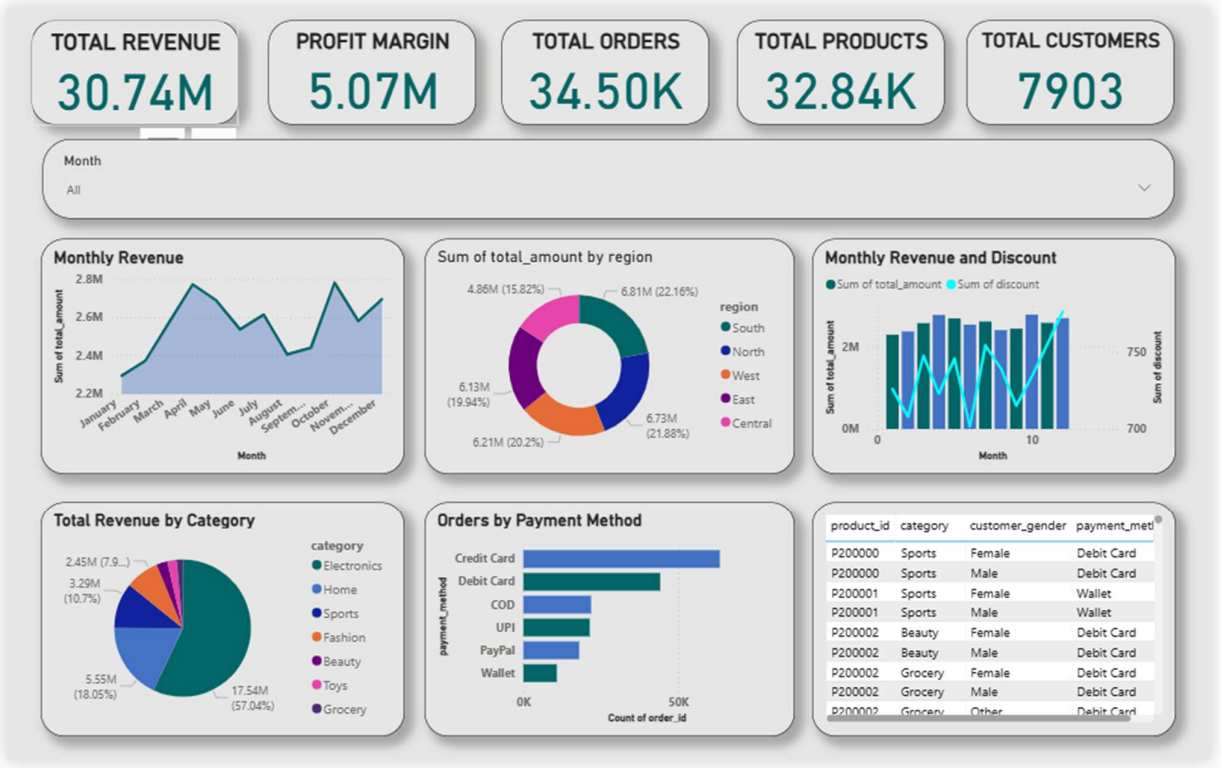
This section details the insights derived from both the Power BI dashboard and advanced ML modeling.

Dashboard Insights (Descriptive Analytics)

The integrated dashboard immediately revealed several actionable insights:

- **Top Category:** Electronics emerged as the dominant category, **contributing over 57% of total revenue**.
- **Seasonal Trends:** A clear revenue peak was identified **during the May–August period**, allowing for data-driven planning for peak-season stock levels.
- **Payment Insights: Credit Card and Cash on Delivery (COD)** were found to be the most preferred payment methods.
- **Customer Profile:** The data showed a higher purchase frequency among the **younger demographic (18–35)**.
- **Profit Trend:** Analysis showed that **profit margin correlates positively with the average discount strategy**, suggesting that promotions are effective at driving volume without excessively eroding profitability.

POWER BI DASHBOARD:



Advanced Analytics: Customer Segmentation (RFM)

To move beyond descriptive analytics, a machine learning model was applied directly to the data warehouse to segment customers using the RFM (Recency, Frequency, Monetary) model.

Methodology:

1. **Data Retrieval:** A SQL query was executed against the warehouse, joining FactSales and DimDate to calculate **Recency** (days since last purchase), **Frequency** (total unique orders), and **MonetaryValue** (total amount spent) for each customer.
2. **Preprocessing:** The data was log-transformed (using np.log1p) to handle the high skew common in sales data and then scaled using StandardScaler to prepare it for clustering.
3. **Clustering:** The **K-Means** algorithm (with k=3) was applied to the scaled RFM data to group customers into three distinct segments.

Results and Interpretation:

The model produced the following segments, characterized by their average RFM values:

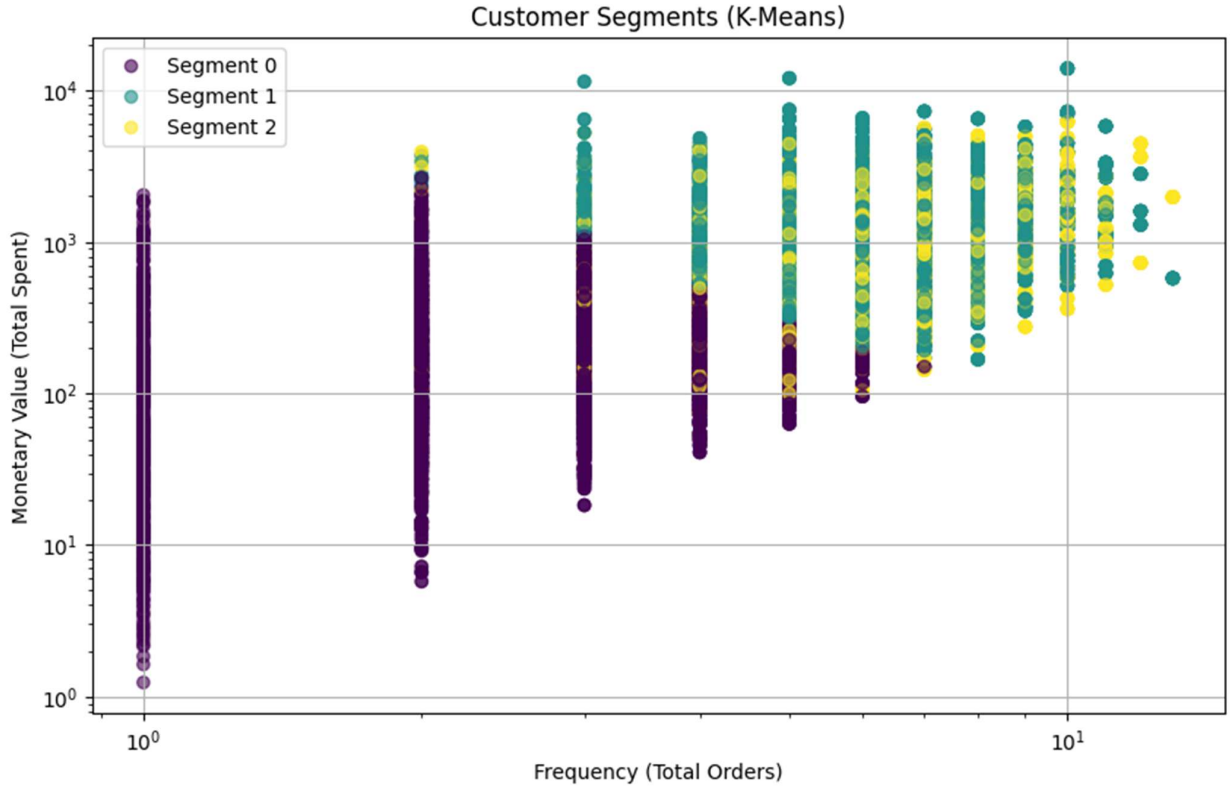
Segment	Recency (Avg Days Ago)	Frequency (Avg Orders)	MonetaryValue (Avg Spend)	Proposed Name
0	204.86	3.23	$286.43	**At-Risk / Lapsed**
1	140.79	6.09	$1,237.77	**Loyal Customers**
2	19.35	6.33	$1,042.35	**Champions / Recent VIPs**

Interpretation:

- **Segment 0 (At-Risk):** These customers have the highest recency (haven't shopped in ~205 days) and the lowest frequency and spend. They are likely lapsed and require targeted "win-back" campaigns.
- **Segment 1 (Loyal Customers):** This group spends the most ($1,238) and shops frequently (6 orders). However, their recency is high (141 days). They are high-value but need re-engagement to prevent them from lapsing.
- **Segment 2 (Champions):** This is the most valuable group. They have shopped very recently (19 days), shop frequently (6.3 orders), and spend significantly ($1,042). These customers should be targeted with loyalty rewards and VIP treatment.

Visualization:

The plot below visualizes the segments based on their Frequency and Monetary value, confirming the high value of Segments 1 and 2.



8. Tools & Technologies Used

Component	Tool/Platform	Purpose
Data Storage	**PostgreSQL (Aiven)**	Cloud-based, scalable relational database serving as the core data warehouse.
Data Modeling & Viz	**Power BI**	Business analytics service for building the data model and designing interactive dashboards.
ETL Pipeline	**Python (Pandas, SQLAlchemy)**	Custom scripts for extracting, transforming, and loading data into the warehouse.
Data Science & ML	**Python (Scikit-learn, Pandas)**	RFM customer segmentation using K-Means clustering.
DB Management	**pgAdmin**	Administration and management tool for the PostgreSQL database.
Diagramming	**Power BI Model View**	Used for designing, visualizing, and documenting the Star Schema.

9. Results & Conclusion

The project successfully designed and implemented a complete, end-to-end data warehouse lifecycle, from raw data ETL to both BI visualization and advanced ML analytics. It transitioned the organization from fragmented, manual reporting to a centralized, automated, and dynamic analytics platform. The system successfully integrates data, ensures high quality through ETL, and provides accessible insights via Power BI and direct model application.

This data warehouse system is a strategic business asset that enables real-time, data-driven decision-making. It improves data accessibility and reliability and provides a proven foundation for advanced analytics, as demonstrated by the successful RFM segmentation. This integrated system not only answers "what happened" (via BI) but also "who are our customers" (via ML), paving the way for further predictive modeling, such as sales forecasting and customer churn prediction.