SQL Analytics for E-Commerce Database: Insights and Performance Optimization

Aravind Panchanathan
Graduate Student
Information, Science and Technology Department
George Mason University
Email: apanchan@gmu.edu

***Abstract*—This paper presents a structured SQL analytics approach applied to an E-Commerce Database. The database was designed following industry best practices, enabling data-driven insights for business intelligence. The analysis includes key performance indicators such as sales trends, customer behavior, product performance, and user engagement analytics. SQL queries ranging from basic to advanced analytics are applied to gain actionable insights. The paper also discusses the efficiency and optimization of queries. Finally, the dataset was generated with the assistance of ChatGPT, which streamlined the validation process.**

***Index Terms*—E-Commerce, SQL Analytics, Database Optimization, Data-Driven Insights, ER Diagram**

I. Introduction

SQL analytics plays a crucial role in extracting meaningful insights from structured databases. This paper presents an analysis of a well-structured **E-Commerce Database**, focusing on key business areas such as **user behavior, order management, revenue trends, and customer retention**. The queries are classified into three levels: **Easy, Intermediate, and Hard**, covering fundamental to complex data retrieval techniques.

II. Database Structure and Schema

The E-Commerce database consists of multiple interrelated tables optimized for high performance. The schema follows **3rd Normal Form (3NF)** to minimize redundancy while maintaining referential integrity. The key tables include:

- **Users:** Stores customer details and roles.
- **Products:** Includes product catalog, categories, and brands.
- **Orders:** Manages transaction details and purchased items.
- **Payments:** Tracks payment status and methods.
- **Logistics:** Handles shipments and delivery tracking.
- **Promotions:** Manages discounts and coupon redemptions.
- **Reviews and Analytics:** Captures customer reviews, search behavior, and engagement data.

III. Entity-Relationship Diagram

Figure 1 illustrates the ER diagram of the database, highlighting the relationships among the entities.

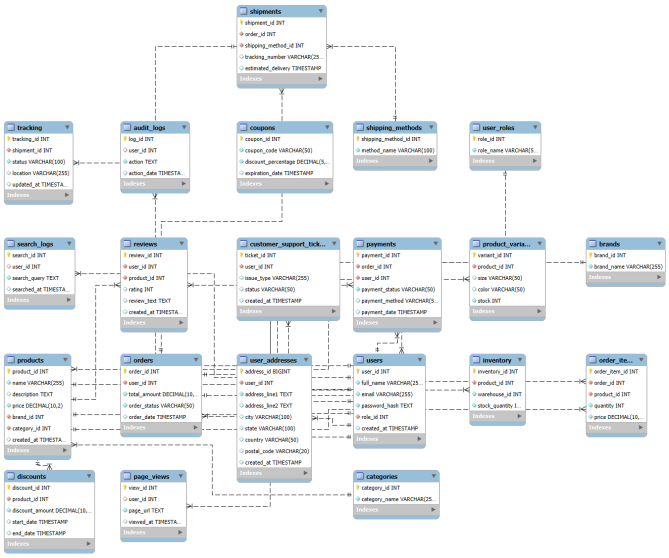

Fig. 1. Entity-Relationship (ER) Diagram of the E-Commerce Database.

IV. SQL Analysis and Queries

The database analysis is structured into three categories: **Basic (Easy), Intermediate, and Advanced (Hard)** SQL queries. The following sections explain the objectives and implementation of these queries.

A. Easy Queries

These queries focus on retrieving basic insights such as user count, total sales, and most ordered products.

- **Total Orders per User:** Counts how many orders each user has placed.
- **Total Revenue per Product:** Calculates the revenue generated for each product.
- **Most Expensive Product:** Identifies the highest-priced product in the catalog.
- **Most Recent Reviews:** Retrieves the latest customer feedback.

B. Intermediate Queries

Intermediate queries involve **ranking functions, conditional aggregations, and subqueries** to gain deeper insights into business performance.

- **Identify the Most Popular Product:** Determines which product has the highest number of orders.
- **Customer Retention Analysis:** Finds users who have placed repeat orders.
- **Revenue Contribution by Brand:** Aggregates total revenue by product brand.
- **Payment Method Diversity:** Identifies users who have used multiple payment methods.

C. Hard Queries

Advanced queries involve **window functions, common table expressions (CTEs), recursive queries, and trend analysis**.

- **High-Value Customers:** Ranks users based on total spending using window functions.
- **Churn Rate Analysis:** Identifies customers who have not placed an order in the last six months.
- **Most Discounted Products:** Determines the products with the highest discount value.
- **Seasonal Order Trends:** Extracts peak sales periods by month.

V. Optimization and Performance Improvements

To ensure optimal database performance, the following optimization strategies were applied:

- **Indexing** on frequently queried columns such as `user_id`, `product_id`, and `order_date`.
- **Partitioning** of order data by date range to optimize retrieval speed.
- **Foreign Key Constraints** to enforce referential integrity.
- **Materialized Views** for pre-aggregating heavy computations.
- **Query Execution Plan Analysis** to detect and improve slow-performing queries.

VI. Conclusion and Future Work

The SQL analytics applied to the **E-Commerce Database** provides actionable insights into **customer behavior, sales trends, product performance, and logistics efficiency**. Future work will focus on **real-time analytics integration, machine learning-based recommendations, and further query optimization for large-scale datasets**.

Acknowledgment

We acknowledge the assistance of **ChatGPT** in generating realistic datasets, which significantly streamlined the database validation and testing process.