**Online Retail Sales Database Design Project Report**

**Date:** July 25, 2025

**Student Name: Umme Haani**

**Internship Program:**Sql internship at ELEVATE LABS

**1. Introduction**

This report details the design and implementation of a relational database schema for an online retail sales platform. In today's e-commerce landscape, a robust and well-structured database is fundamental for managing critical business operations, including product inventory, customer information, order processing, and payment tracking. This project aimed to create a normalized SQL database that efficiently stores and retrieves this data, providing a foundational backend for an online store application. The design emphasizes data integrity, scalability, and ease of querying for business intelligence and reporting.

**2. Abstract**

This project successfully designed and implemented a normalized SQL database for an online retail sales system. Utilizing Entity-Relationship (ER) modeling, the database schema was structured into five key entities: Customers, Products, Orders, Order\_Items, and Payments, ensuring adherence to 3rd Normal Form (3NF) principles to minimize data redundancy and enhance integrity. Data Definition Language (DDL) scripts were developed to create the tables with appropriate constraints, and Data Manipulation Language (DML) scripts populated these tables with realistic sample data. Comprehensive SQL queries, including JOIN operations and a database view, were crafted to generate essential sales reports and derive key business insights, such as average order value and the identification of high-value customers. The entire project, including the database file, SQL scripts, and ER diagram, was version-controlled and published on GitHub, demonstrating proficiency in modern development practices.

**3. Tools Used**

The following tools were instrumental in the development and deployment of this database project:

* **DB Browser for SQLite:**
  + **Purpose:** A lightweight, open-source, and visual tool used for creating, designing, and managing SQLite database files. It provided an intuitive graphical user interface (GUI) for executing SQL queries, browsing data, and verifying the database structure.
  + **Role in Project:** Used to create the retail\_sales.db file, run the DDL and DML scripts, and execute various reporting queries.
* **dbdiagram.io:**
  + **Purpose:** An online, web-based tool for quickly sketching and visualizing Entity-Relationship (ER) diagrams from a simple SQL-like text syntax.
  + **Role in Project:** Utilized to design and visually represent the database schema, illustrating the entities and their relationships, which is a crucial step in database design.
* **Git:**
  + **Purpose:** A distributed version control system (VCS) that tracks changes in source code and other files during software development. It enables collaboration, maintains a complete history of modifications, and allows for easy rollback to previous versions.
  + **Role in Project:** Used locally to manage the project files (.sql scripts, .db file, README.md, ER diagram) by tracking changes and preparing them for sharing.
* **GitHub:**
  + **Purpose:** A web-based platform that provides hosting for Git repositories. It's widely used for version control, collaboration, and showcasing projects publicly.
  + **Role in Project:** Served as the central remote repository for storing and sharing the entire database project, making it accessible for review and demonstration.
* **Text Editor (e.g., Notepad, VS Code):**
  + **Purpose:** A basic software application for writing and editing plain text files.
  + **Role in Project:** Used to write and save the .sql scripts (schema.sql, sample\_data.sql, queries\_reports.sql) and the README.md file.

**4. Steps Involved in Building the Project**

The project followed a structured approach, encompassing design, implementation, data population, and reporting:

1. **Entity-Relationship (ER) Modeling:**
   * Identified core entities for an online retail system: Customers, Products, Orders, Order\_Items, and Payments.
   * Defined attributes (columns) for each entity and established primary keys.
   * Determined relationships between entities (e.g., one-to-many between Customers and Orders).
   * Utilized dbdiagram.io to visually represent the ER diagram, ensuring a clear and normalized schema (adhering to 3NF).
2. **Database Schema Definition (DDL):**
   * Wrote Data Definition Language (DDL) scripts in schema.sql to create the five tables identified in the ER model.
   * Each table definition included appropriate data types (INTEGER, VARCHAR, DECIMAL, DATETIME), NOT NULL constraints for essential fields, PRIMARY KEY definitions, and FOREIGN KEY constraints to enforce referential integrity between related tables.
3. **Database Creation and Schema Application:**
   * Used DB Browser for SQLite to create a new SQLite database file named retail\_sales.db.
   * Executed the schema.sql script within DB Browser's SQL editor to apply the table definitions and create the database structure.
4. **Sample Data Population (DML):**
   * Developed Data Manipulation Language (DML) scripts in sample\_data.sql to insert realistic sample records into each of the created tables.
   * This step populated the database with example customer details, product information, order records, order item specifics, and payment transactions, making the database functional for testing and reporting.
5. **Reporting and Business Insights (SQL Queries & Views):**
   * Authored a comprehensive set of SQL queries in queries\_reports.sql to extract meaningful information from the populated database. These queries involved:
     + **JOIN operations:** To combine data from multiple related tables (e.g., linking customers to their orders, or orders to the products they contain).
     + **Aggregate functions:** Such as SUM() and COUNT() for calculating totals and counts.
     + **Filtering and Sorting:** Using WHERE and ORDER BY clauses to refine results.
   * **Simple Innovation:** Implemented specific queries to derive key business insights:
     + **Average Order Value:** Calculated the overall average amount spent per order, a crucial metric for understanding customer spending.
     + **High-Value Customers (Potential 'Prime' Candidates):** Identified customers whose total spending significantly exceeded the calculated average order value. This demonstrates how data analysis can inform strategies for identifying and targeting customers for loyalty programs or exclusive offers, similar to Amazon Prime.
   * Created a DailySalesSummary SQL VIEW to provide a reusable summary of daily sales and order counts.
6. **Version Control and Deployment (Git & GitHub):**
   * Initialized a local Git repository within the project folder.
   * Added all project files (retail\_sales.db, schema.sql, sample\_data.sql, queries\_reports.sql, retail\_er\_diagram.png, README.md) to Git's staging area.
   * Committed the changes, creating a version history of the project.
   * Created a new public repository on GitHub.
   * Linked the local Git repository to the remote GitHub repository.
   * Pushed all project files from the local machine to GitHub, making the project publicly accessible and showcasing version control best practices.

**5. Conclusion**

This project successfully delivered a fully functional and normalized SQL database for an online retail sales platform, demonstrating proficiency in database design principles, SQL DDL/DML, and data querying for business reporting. The implementation of a structured schema, populated with sample data, allows for efficient management of core e-commerce operations. Furthermore, the inclusion of analytical queries to determine the average order value and identify high-value customers highlights the ability to extract actionable business intelligence from raw data, a critical skill in data-driven environments. The entire development process, from initial design to final deployment on GitHub, showcases a practical understanding of industry-standard tools and version control, providing a solid foundation for future database and data analysis endeavors.