

SQL Internship Project Report

Introduction

The goal of this project is to design and implement a relational database system for an online retail sales platform. It focuses on building a structured backend using SQL to manage critical business operations such as customer management, product inventory, order tracking, and payment processing. The project simulates a simplified version of a real-world e-commerce system and emphasizes clean schema design, normalization, and analytical querying.

Abstract

In modern e-commerce platforms, managing data integrity, sales insights, and inventory efficiency is crucial. This project delivers a fully normalized SQL database in **Third Normal Form (3NF)** using **MySQL Workbench**. It includes the creation of entities like Customers, Products, Orders, Order_Items, and Payments. The database supports transaction tracking, stock updates via triggers, and sales reporting through views. Key queries were written to fetch daily sales, best-selling products, and customer order histories. The final database is scalable, optimized, and easily adaptable for future e-commerce expansion.

Tools Used

- **MySQL Workbench** – For writing SQL scripts and managing the database
 - **dbdiagram.io** – For creating the Entity-Relationship Diagram (ER)
 - **GitHub** – For version control and showcasing the project
-

Steps Involved in Building the Project

1. **Requirements Analysis**
Identified core entities: Customers, Products, Orders, Order_Items (many-to-many), and Payments.
2. **ER Diagram Design**
Designed and exported a normalized ERD using dbdiagram.io. All relationships were mapped with appropriate keys.
3. **Schema Development (DDL)**
Created tables using SQL with necessary constraints (PK, FK, data types, defaults).
4. **Sample Data Insertion (DML)**
Inserted realistic sample data for all tables to test functionality and run queries.
5. **Query Writing**
Wrote SQL queries to:

- Track customer order history
- Generate daily sales summaries
- Identify best-selling products

6. **Views Creation**

Developed reusable views like:

- Daily_Sales_View
- Best_Selling_Products

7. **Trigger Implementation**

Created a trigger to automatically reduce product stock when a new order item is inserted.

8. **Sales Reporting**

Exported query results to Excel and visualized daily revenue and order counts via charts.

9. **Testing and Validation**

Verified data integrity with test_cases.sql and ensured accurate query results.

Conclusion

This project demonstrates the practical application of SQL in building a reliable and scalable retail database system. Through normalization, entity design, query optimization, and automation (via triggers and views), the system supports essential e-commerce operations efficiently. The project also reinforces skills in ER modeling, relational integrity, and analytical data handling. It is a strong portfolio piece that showcases real-world SQL database design and implementation.

Note:

All project assets — including .sql files, ER diagram, test cases, and Excel reports — are available in the GitHub repository for review and future development.