PROJECT 1: Online Retail Sales Database Design Report

Project Title: Online Retail Sales Database Design
Prepared by: SHEETAL
Date: August 23, 2025

---

1. Introduction

This report presents a structured relational database project designed to manage and analyse sales data for an Ecommerce Database. Using MySQL Workbench, the database was developed to store and retrieve data related to products, customers, orders, and payments. The aim was to create an efficient reporting system that provides insights into business performance and supports informed decision-making.

---

2. Abstract

The project implements a normalized database with strong referential integrity. Key entities were identified and structured through 3NF normalization, followed by SQL DDL implementation. The database was populated with sample data to simulate real-world scenarios. Various JOIN queries and views were created to analyse product sales, order statuses, customer activity, and payment methods.

---

3. Tools Used

- **MySQL Workbench**:
  Used for designing the ER diagram, writing DDL scripts, executing SQL queries, and validating data integrity and relationships.

---

4. Steps Involved in Building the Project

Step 1: Identify Key Entities

- Products: Details like product name, category, price, quantity
- Customers: Personal and contact details, registration info
- Orders: Status, date, and total amount
- Payments: Amount, method, and payment status

Step 2: ER Diagram Design

Designed using dbdiagram.io to establish one-to-many relationships:

- One product → Many customers

- One customer → Many orders
- One order → One payment

Step 3: Normalize Schema to 3NF

- Removing partial and transitive dependencies
- Creating foreign keys for relationships
- Ensuring atomicity and uniqueness

Step 4: Create Tables Using DDL

Tables in an Ecommerce database are created using Data Definition Language (DDL) commands, specifically the CREATE TABLE statement. This statement defines the structure of tables, including its name, the names of its columns, their respective data types, and constraints such as Primary Key, NOT Null, Unique, Check, Default and Foreign Key.

DDL Commands: CREATE DATABASE, CREATE TABLE, CREATE VIEW, ALTER, DROP, TRUNCATE and RENAME.

Step 5: Populate Tables with Sample Data

- Inserted 10+ records into each table using **INSERT** statements.
- Simulated real-world transactions, including pending, returned, and refunded orders.

Step 6: Write JOIN Queries and Views

Create a sales report using a view that shows complete sales information by creating a view called Sales Report. JOIN queries are for joining two or more tables together. Some JOIN queries are INNER, LEFT, RIGHT, FULL.

Other query highlights:

- Identify high-value sales: ORDER BY price DESC LIMIT 1
- Cross-referencing order and payment status
- LEFT JOIN and RIGHT JOIN to include unmatched data
- Aggregate views to calculate total sales per product or per customer.

---

5. Conclusion

The project effectively showcases the power of relational databases in managing complex data. With proper normalization, DDL scripting, and the use of JOINs and views, the system supports meaningful analysis. The solution is scalable, maintainable, and can be extended to support real-time dashboards or analytics platforms.