Project Title: Online Retail Sales Database Design
Tools Used: MySQL, MySQL Workbench, ER Modeling

1. Introduction

The Online Retail Sales Database project focuses on designing and implementing a structured, normalized backend database for an e-commerce platform. The aim is to store and manage information related to customers, products, orders, and payments efficiently. This database supports core business operations such as order processing, inventory tracking, and customer management.

2. Abstract

This project includes the development of a complete SQL schema for an online shopping system. The schema is built using normalization principles to ensure data consistency and integrity. Using MySQL Workbench, tables were designed, foreign key relationships were defined, and constraints were applied. The project also includes sample data creation and query development for reporting and analytics.

3. Tools & Technologies Used

- **MySQL 8.0** – Database server
- **MySQL Workbench** – ERD design, forward engineering
- **SQL (DDL & DML)** – Table creation, constraints, data insertion
- **UTF8MB4 Encoding** – Ensures full Unicode support

4. Steps Involved in Building the Project

4.1 ER Diagram Design

Using MySQL Workbench, key entities were identified and modeled:

- **Customers**
- **Products**
- **Orders**
- **Payments**
- **Payment Mode**

Relationships were established:

- A customer can place multiple orders.
- An order can include multiple products.
- Each order has a corresponding payment record.

4.2 Database & Schema Creation

A schema named **e_commerce** was created with UTF8MB4 character encoding. Workbench forward engineering generated the SQL DDL script automatically.

4.3 Table Creation & Constraints

Created following major tables:

Customers Table

Stores customer information with a unique email constraint and default join date.

Products Table

Maintains product details including pricing and inventory count.

Orders Table

Links customers and products using foreign keys.
Includes order date and total amount fields.

Payments Table

Stores payment details linked to specific orders.

Payment_mode Table

Defines various available payment methods.

All tables use **InnoDB** engine and implement **referential integrity** via foreign keys.

4.4 Auto-Increment Handling

Primary keys such as `customer_id`, `product_id`, and `order_id` use AUTO_INCREMENT for unique identity management.

4.5 Indexing & Optimization

Indexes were added to:

- `customer_id`
- `product_id`
- `order_id`

This improves query performance for joins and reports.

4.6 Sample Data Population

Sample customers, products, orders, and payments were inserted to simulate real-world retail transactions.

4.7 SQL Query Development

Various analytical and operational queries were created, such as:

- Customer purchase history
- Total sales summary
- Payment reports
- Low-stock product checks
- Order-wise revenue calculations

5. Conclusion

The Online Retail Sales Database successfully demonstrates the design and implementation of a normalized relational schema for an e-commerce platform. The project includes all components required for real-world retail data management—customer records, product catalog, order processing, and payment tracking. The structured schema, constraints, and indexing ensure data accuracy, scalability, and reliability. This project provides a solid foundation for backend development and can be extended into a full-stack e-commerce system.