# Online Retail Sales Database Design

**Domain:** Database Management Systems (DBMS)

**Tools:** MySQL / PostgreSQL, dbdiagram.io, MySQL Workbench

## 1. INTRODUCTION

The 'Online Retail Sales Database Design' project focuses on creating a well-structured and normalized database schema for an e-commerce platform. The aim is to efficiently manage data for online sales operations, including customer information, product inventory, orders, and payment records. A normalized database reduces redundancy, improves consistency, and ensures faster query execution. This database can serve as the foundation for a complete retail management system or analytics dashboard.

## 2. ABSTRACT

In the digital commerce era, data handling plays a vital role in ensuring business success. This project demonstrates the complete lifecycle of designing a relational database for an online retail system. It begins by identifying entities such as Customers, Products, Orders, and Payments. Relationships are mapped through an Entity–Relationship (ER) Diagram. The schema is then normalized up to Third Normal Form (3NF) to ensure data integrity. After normalization, SQL DDL scripts are written to define tables and constraints. Finally, sample data is inserted, and SQL JOIN queries and views are developed to extract insights such as total sales, customer spending, and category-wise revenue.

## 3. TOOLS USED

| Database | MySQL / PostgreSQL |
|---|---|
| Diagram Tool | dbdiagram.io |
| Data Generator | Mockaroo or manual CSV creation |
| Query Editor | MySQL Workbench / pgAdmin |
| Text Editor | VS Code / Notepad++ |

## 4. STEPS INVOLVED IN BUILDING THE PROJECT

### *4.1 Entity Identification*

The first step is to identify all major entities that exist in the online retail environment. The core entities are: **Customers** (who place orders), **Products** (that are sold), **Orders** (that record transactions), **OrderDetails** (which list individual product quantities per order), and **Payments** (that record payment information).

### *4.2 ER Diagram Design*

Using dbdiagram.io, an Entity–Relationship diagram is created to visualize how entities relate. The relationships include: one Customer can have multiple Orders, one Order can include multiple Products through OrderDetails, and one Order is linked to one Payment. This diagram serves as the blueprint for table creation.

### *4.3 Normalization*

Normalization ensures the database is efficient and free from anomalies. 1NF removes repeating columns, 2NF removes partial dependencies, and 3NF ensures every non-key attribute depends only on the primary key. This eliminates redundancy and ensures reliable relationships between tables.

### *4.4 Schema Creation (DDL)*

SQL Data Definition Language (DDL) scripts are written to create the database tables with appropriate primary keys, foreign keys, and constraints to maintain referential integrity. Each table is defined clearly as shown below:

**Example:**
CREATE TABLE Customers (CustomerID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(100), Email VARCHAR(100) UNIQUE, City VARCHAR(50));

### *4.5 Sample Data Insertion*

To simulate real business operations, sample data (10–15 records per table) was inserted manually or using Mockaroo. This helps test the database structure and verify the accuracy of queries.

### *4.6 Querying and Reporting*

The database supports analytical queries and sales reporting. For example:
- Total Sales by Product: SUM(Quantity * Price)
- Customer Spending Summary: Aggregated payments by CustomerID

Views can be created to simplify repetitive queries and assist in dashboard visualization.

## 5. CONCLUSION

The Online Retail Sales Database Design project demonstrates strong understanding of database normalization, schema design, and SQL operations. It forms a solid base for e-commerce systems by managing customer and order data efficiently. This structured approach ensures scalability, accuracy, and optimal data retrieval, which are crucial for real-time applications like online shopping platforms.

## 6. DELIVERABLES

1. ER Diagram (from dbdiagram.io)

2. SQL Schema (DDL Script)

3. Sample Data (INSERT statements)

4. Query Report (JOINs, Views, and Aggregates)

5. Final Project Report (this PDF)