

ONLINE RETAIL SALES DATABASE DESIGN

1. Introduction

With the rapid growth of e-commerce platforms, managing large volumes of customer, product, and order data efficiently has become critical. An Online Retail Sales Database plays a key role in organizing transactional data, ensuring data integrity, and enabling meaningful business analysis.

This project focuses on designing and implementing a **normalized relational database** for an online retail system using SQL. The system stores information related to customers, products, orders, and order items. The database is designed to support common retail operations such as order tracking, sales analysis, and inventory management.

2. Abstract

The Online Retail Sales Database Design project aims to build a structured and scalable SQL-based database for an e-commerce environment. The project includes creating an **Entity Relationship (ER) Diagram**, implementing **Data Definition Language (DDL)** scripts, inserting sample data using **DML**, and writing **JOIN queries** for analytical reporting.

The database is normalized to reduce redundancy and maintain data consistency. Sample datasets are used to demonstrate how customer orders and product sales can be tracked efficiently. This project helps in understanding real-world database design concepts and strengthens core SQL skills required for interviews and industry applications.

3. Tools Used

- **MySQL** – Database creation and query execution
 - **VS Code** – Writing and managing SQL scripts
 - **dbdiagram.io** – ER Diagram design
 - **GitHub** – Version control and project hosting
-

4. Steps Involved in Building the Project

4.1 Requirement Analysis

The following key entities were identified:

- Customers
- Products
- Orders
- Order Items

Each entity represents a real-world component of an online retail system.

4.2 ER Diagram Design

An ER diagram was designed using [dbdiagram.io](#) to visually represent:

- One-to-many relationship between Customers and Orders
- One-to-many relationship between Orders and Order Items
- One-to-many relationship between Products and Order Items

This helped in clearly understanding relationships before implementing the database.

4.3 Database Schema Creation (DDL)

SQL CREATE TABLE statements were used to:

- Define primary keys for unique identification
- Establish foreign key relationships
- Ensure referential integrity between tables

The database was created in a structured and normalized form.

4.4 Data Insertion (DML)

Sample records were inserted into each table to simulate real-world data such as:

- Customer details
- Product listings
- Orders placed by customers
- Products included in each order

This data was used for testing and query execution.

4.5 Query Implementation

SQL queries were written using JOIN, GROUP BY, and aggregate functions to:

- Retrieve customer order details
- Analyze product sales performance
- Generate summarized sales reports

These queries demonstrate practical usage of SQL in business scenarios.

5. Sample Queries Used

- Fetch customer-wise order details
- Calculate total quantity sold for each product
- Analyze sales using JOIN operations

These queries help in understanding how relational data can be combined and analyzed.

6. Conclusion

The Online Retail Sales Database Design project successfully demonstrates the complete lifecycle of database development — from requirement analysis to ER modeling, schema creation, data insertion, and query execution. The project provides hands-on experience with core SQL concepts such as primary keys, foreign keys, joins, and aggregation.

This project is simple yet powerful, making it easy to explain during interviews. It closely aligns with real-world e-commerce applications and strengthens foundational database skills required for entry-level roles and internships.