

PROJECT REPORT ON

ONLINE RETAIL SALES DATABASE DESIGN

Introduction

This project focuses on designing and implementing a SQL-based database for an Online Retail Sales . The system supports online products, order placement, payment tracking, and customer management. The goal is to create a scalable, normalized, and query-efficient relational database.

Abstract

- Develop a relational database schema for an online retail platform.
- Ensure data consistency, integrity, and security through proper constraints.
- Enable efficient querying for sales analysis, order tracking, and customer behavior.
- Support business processes like inventory management, payment processing, and customer service.

Tools used: MySQL , dbdiagram.io

Requirements Analysis

After analyzing the typical workflow of an e-commerce system, the following data requirements were identified:

- Customer registration and management
- Product catalog with categories and stock
- order processing
- Payment tracking
- Shipping and delivery information

Entity-Relationship (ER) Design

The ER diagram includes the following main entities:

- **Customer** (CustomerID, Name, Email, Address, Phone)
- **Product** (ProductID, Name, Description, Price, CategoryID, Stock)
- **Orders** (OrderID, CustomerID, OrderDate, Status, TotalAmount)
- **Payment** (PaymentID, OrderID, PaymentMethod, PaymentDate, Amount)
- **Shipping** (ShippingID, OrderID, ShippingAddress, ShippingDate, DeliveryStatus)

Relationships:

- A **Customer** can place multiple **Orders**.
- An **Order** can contain multiple **Products** through **OrderDetails**.
- Each **Order** is linked to one **Payment** and one **Shipping** record.

Database Implementation (SQL)

The database was implemented using **MySQL**. Tables were created with appropriate primary and foreign keys, data types, and constraints:

```
CREATE TABLE customers (  
    C_id INT PRIMARY KEY,  
    CUST_NAME VARCHAR(20) UNIQUE,  
    CUST_MAIL VARCHAR(20),  
    CUST_NUM VARCHAR(10) CHECK (CHAR_LENGTH(CUST_NUM) = 10),  
    CUST_LOC VARCHAR(15));  
  
CREATE TABLE `Orders` (  
    `Order_id` INTEGER PRIMARY KEY,  
    `C_id` INTEGER,  
    `order_date` DATE,  
    `Order_total` INTEGER,  
    `order_status` VARCHAR(10),  
    `P_id` INTEGER,  
    `Order_Quantity` INTEGER,  
    FOREIGN KEY (`C_id`) REFERENCES customers(`C_id`)  
    FOREIGN KEY (`P_id`) REFERENCES products(`P_id`));
```

Normalization

The database was normalized to **Third Normal Form (3NF)**:

- Redundant data was minimized.
- Each non-key attribute depends only on the primary key.

Monthly sales:

```
SELECT MONTH(order_date) AS Month, SUM(Order_total) AS Revenue
```

```
FROM Orders
```

```
GROUP BY MONTH(order_date);
```



The screenshot shows a SQL query editor with the following query:

```
235  
236 * SELECT MONTH(order_date) AS Month, SUM(Order_total) AS Revenue  
237 FROM Orders  
238 GROUP BY MONTH(order_date);
```

Below the query, the 'Result Grid' is displayed with the following data:

Month	Revenue
8	8530

Conclusion

The SQL database designed for the Online Retail Sales System successfully captures all essential business processes. It provides efficient data retrieval for customers, orders, inventory, and payment management.