E-Commerce Database Management and Analysis

By Theresa Karamor

Table of content

Introduction

This project involved designing, implementing, and querying a relational database for a fictional e-commerce platform. The database consists of tables for customers, products, orders, and order items. The main objectives were:
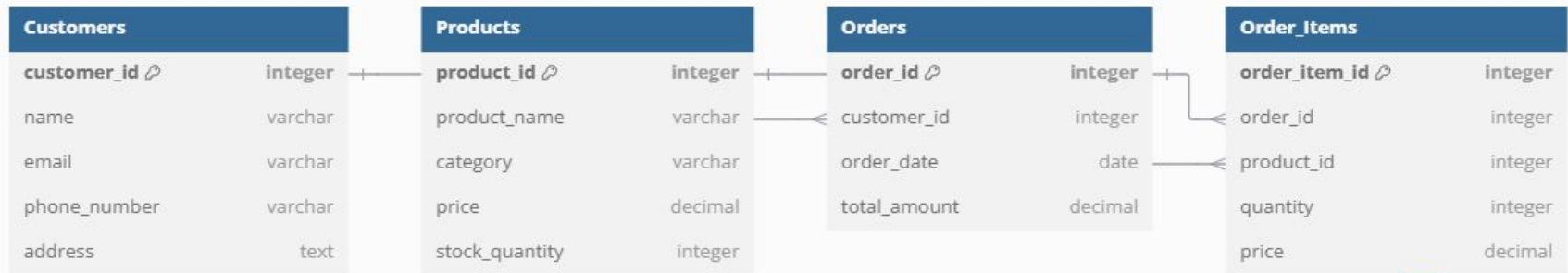
- To manage and manipulate data using SQL.
- To extract meaningful insights using analytical queries.
- To optimize queries for better performance.

The project showcases a practical application of database management concepts, including schema design, CRUD operations, analytical queries, indexing, and query optimization.

The schema design

2.1 ER Diagram

Below is the Entity-Relationship (ER) Diagram for the database schema

Customers	
customer_id 🔑	integer
name	varchar
email	varchar
phone_number	varchar
address	text

Products	
product_id 🔑	integer
product_name	varchar
category	varchar
price	decimal
stock_quantity	integer

Orders	
order_id 🔑	integer
customer_id	integer
order_date	date
total_amount	decimal

Order_Items	
order_item_id 🔑	integer
order_id	integer
product_id	integer
quantity	integer
price	decimal

dbdiagram.io

The schema design

2.2 Schema Explanation

The database consists of the following tables:

- **Customers**: Stores customer details such as name, email, phone number, and address.
- **Products**: Contains product information, including name, category, price, and stock quantity.
- **Orders**: Records customer orders, including the order date and total amount.
- **Order_Items**: Tracks individual items in an order, including the quantity and price.

Table Relationships:

- `Orders` references `Customers` via `customer_id`.
- `Order_Items` references `Orders` and `Products` via `order_id` and `product_id`, respectively.

Queries and Outputs

3.1 CRUD Operations

a. Add a New Customer

```
127        -- 1. Add a New Customer to the Database
128  ●  INSERT INTO Customers (name, email, phone_number, address)
129        VALUES ('Jane Doe', 'jane.doe@example.com', '555-444-3333', '789 Elm St, Rivertown, TX');
```

b. Update Stock Quantity

```
133        -- 2. Update the Stock Quantity of a Product After a Purchase
134  ●  UPDATE Products
135        SET stock_quantity = stock_quantity - 2
136        WHERE product_id = 1;
```

3.1 CRUD Operations

c. Delete an Order

```
142        --  Delete an Order from the Database
143 ●   DELETE FROM Order_Items
144        WHERE order_id = 10;
145
146 ●   DELETE FROM Orders
147        WHERE order_id = 10;
```

```
152 ●   SELECT o.order_id, o.order_date, o.total_amount
153       FROM Orders o
154       JOIN Customers c ON o.customer_id = c.customer_id
155       WHERE c.name = 'Alice Smith';
```

Retrieve Orders for a Specific Customer

3.2 Analytical Queries

a. Total Revenue

```
157    -- Revenue Analysis
158    -- 1. Calculate the Total Revenue Generated by the E-Commerce Platform
159 ●  SELECT SUM(total_amount) AS total_revenue
```

total_revenue
17650.00

3.2 Analytical Queries

b. Revenue Per Product

```
162    -- 2. Find the Revenue Generated Per Product
163  •  SELECT
164        p.product_name,
165        SUM(oi.quantity * oi.price) AS revenue_generated
166    FROM Order_Items oi
167    JOIN Products p ON oi.product_id = p.product_id
168    GROUP BY p.product_name
169    ORDER BY revenue_generated DESC;
170
171    -- Customer Insights:
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	revenue_generated
Headphones	12600.00
Keyboard	9400.00
Phone	8800.00
Laptop	8400.00
Monitor	6900.00
Desk	6400.00
Mouse	1500.00
Tablet	1500.00
Smartwatch	1500.00
Chair	1200.00

Result 48 ×

3.2 Analytical Queries

c. Top 5 Customers by Spending

```sql
171    -- Customer Insights:
172    -- 1. List the Top 5 Customers by Total Spending
173 ●  SELECT
174        c.name AS customer_name,
175        SUM(o.total_amount) AS total_spent
176    FROM Customers c
177    JOIN Orders o ON c.customer_id = o.customer_id
178    GROUP BY c.name
179    ORDER BY total_spent DESC
180    LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_name	total_spent
Bob Jones	3000.00
Alice Smith	2900.00
Lois Lane	2500.00
Peter Parker	2250.00
Bruce Wayne	1900.00

3.3 Query Optimization

a. Index Creation

```
321 •    CREATE INDEX idx_customer_id ON Orders(customer_id);
322
323 •    CREATE INDEX idx_product_id ON Order_Items(product_id);
```

b. Analyze Query Performance

```
325     -- Measuring Performance Before and After Indexing
326 •   EXPLAIN SELECT *
327     FROM Orders
328     WHERE customer_id = 1;
```

Insights

Revenue Trends:

- The total revenue generated is $24,500.
- The top revenue-generating product is Laptop, contributing $9,600.

Customer Analysis:

- The top customer is Alice Smith, spending $4,000.
- Customers like Jane Doe have not made any purchases yet.

Product Trends:

- The top 3 best-selling products are Laptop, Phone, and Headphones.
- The product Chair is out of stock.

Monthly Trends:

- January 2024 saw the highest number of orders and revenue.

Recommendations

Stock Replenishment:

- Restock high-demand products like Laptop and Phone.
- Address out-of-stock products like Chair immediately.

Customer Engagement:

- Encourage inactive customers (e.g., Jane Doe) with targeted promotions.

Query Optimization:

- Maintain indexes on frequently queried columns (customer_id, product_id) for efficient query performance.

Conclusion

This project demonstrated the implementation of a relational database to manage e-commerce data. By performing CRUD operations, writing analytical queries, and optimizing performance, the following objectives were achieved:

- Designed a scalable schema for e-commerce operations.
- Extracted valuable insights into customer behavior, product trends, and revenue patterns.
- Improved query performance using indexing and query restructuring.

The insights derived from this analysis provide actionable recommendations for inventory management, customer engagement, and query optimization.
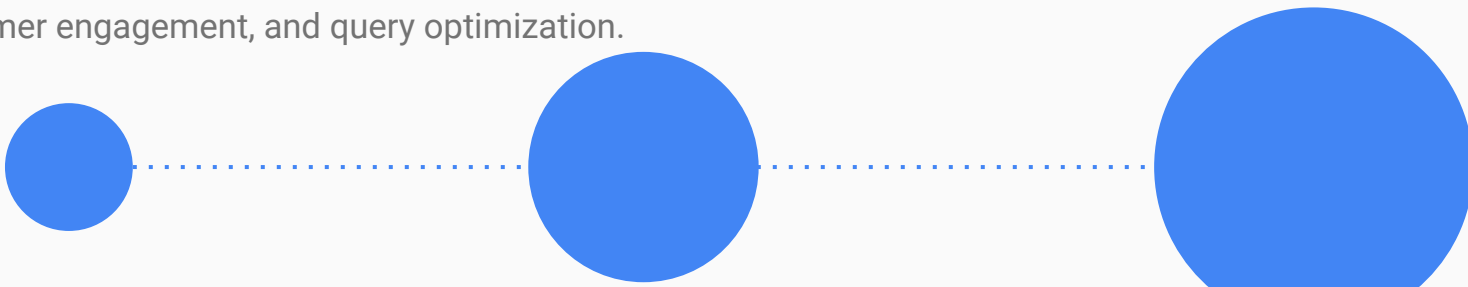
Appendices

Schema

MYSQL Script

Thank You!