



CHANDIGARH
UNIVERSITY
Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

CASE STUDY REPORT ON PARTICULAR CASE STUDY

Program Name: BCA

Subject Name/Code: Database Management System
(23CAT-251)

Submitted by:

Name: Keshav Sharma

UID: 23BCA10688

Section: BCA 4(A)

Submitted to:

Name: Arvinder Sir

Designation: Faculty

ABSTRACT

- Introduction:
-

- Technique:
-

- System Configuration:
-

- INPUT:
-

- ER DIAGRAM:
-

- TABLE RELATION:
-

- TABULAR FORMAT:
-

- TABLE CREATION:
-

- SQL QUERIES WITH OUTPUT (at least 10 to 15):
-

- SUMMARY:
-

- CONCLUSION:

Introduction

In the digital age, e-commerce has revolutionized how people shop, offering convenience, variety, and accessibility at their fingertips. The **E-Commerce Website Database Management System (DBMS) Project** aims to model and manage the backend data infrastructure for an online shopping platform. This project provides a comprehensive and scalable relational database solution that supports the operations of a modern e-commerce business.

The database is designed to handle various components essential to e-commerce platforms, including **user registrations, product listings, order processing, payment transactions, and customer reviews**. Each of these components is represented as a separate table with well-defined relationships, enabling efficient data retrieval, insertion, updating, and deletion using SQL.

This project I particularly focused on providing **real-world DBMS concepts** such as:

- **Entity-Relationship modeling**
- **Data normalization**
- **Primary and foreign key constraints**
- **JOIN operations**
- **Aggregate functions**
- **Data integrity enforcement**

The system supports **multiple users (customers and admins)**, each capable of performing operations like browsing products, placing orders, submitting reviews, and making payments.

Key Modules:

- **Users:** Stores customer and admin data.
- **Products:** Manages product details like name, price, stock, and category.
- **Categories:** Groups products into meaningful sections.
- **Orders:** Handles orders placed by customers.
- **Order Details:** Tracks specific items and quantities in each order.
- **Payments:** Records payment details and methods.
- **Reviews:** Allows customers to leave feedback on products.

Educational Objectives:

- Apply ER modeling to design relational databases.
- Implement complex SQL queries for various use cases.
- Understand real-world use of normalization and data integrity.

This project not only showcases **theoretical understanding of DBMS** concepts but also prepares students for **real-world applications in web and software development**, especially in domains like **online retail, inventory management, and digital payments**.

Technique

The **E-Commerce Website Database Management System (DBMS) Project** is built using industry-standard relational database techniques. The goal was to design a normalized, efficient, and scalable backend database to support typical e-commerce operations such as user management, product listings, order processing, payment tracking, and product reviews. The following database techniques and concepts were employed:

- ◆ **1. Relational Database Design**

A relational model was chosen for the project due to its structured nature and wide usage in web applications. Each real-world entity (like **Users**, **Orders**, **Products**, etc.) is represented as a **separate table**.

- ◆ **2. Entity-Relationship (ER) Modeling**

An ER diagram was created to visualize and design the relationships among the entities. This step helped in identifying:

- **Entities:** Users, Products, Orders, etc.
- **Attributes:** e.g., `username`, `email`, `price`, `order_date`, etc.
- **Relationships:** such as “places” between Users and Orders, or “contains” between Orders and Products.

- ◆ **3. SQL for Data Definition and Manipulation**

- **DDL** commands were used to create tables and define keys and constraints.
- **DML** was used to insert, update, and query data.

- ◆ **4. Normalization**

The schema was normalized up to the **Third Normal Form (3NF)** to reduce redundancy and improve data integrity. For example:

- Product categories were stored in a separate **Categories** table instead of repeating them in the **Products** table.
- Orders and their detailed items were split into **Orders** and **OrderDetails**.

- ◆ **5. Constraints and Integrity Rules**

The database uses various constraints to ensure consistency and correctness:

- **Primary keys** uniquely identify records.
- **Foreign keys** enforce referential integrity between related tables.

- ◆ **6. Query Optimization Practices**

Efficient indexing (like **AUTO_INCREMENT** on primary keys) and avoiding unnecessary redundancy were considered during design to ensure quick data retrieval and scalability.

◆ 7. Sample Data and Testing

To simulate a realistic e-commerce platform:

- At least **50 users, several orders, reviews, and payments** were inserted.
- Sample queries tested the relationships, constraints, and functionalities.
- These queries included **selection, joins, grouping, insertion, and deletion operations**.



System Configuration

The **E-Commerce Website Database Management System (DBMS)** requires specific hardware, software, and network configurations to operate efficiently and support a scalable ecommerce platform. Below is a detailed description of the system configuration, including the environment setup, software tools, and system requirements used to develop and run the project.

Hardware Requirements

To support database management, web operations, and handling large data efficiently, the system should meet the following hardware specifications:

- **Processor:** Minimum Intel i3 or equivalent with 2.0 GHz or higher for fast query execution.
- **RAM:** At least **4 GB** (recommended 8 GB) to ensure smooth database handling and multi-tasking.
- **Hard Disk:** Minimum of **500 GB** of storage, preferably SSD (Solid State Drive), for faster read/write operations.
- **Network:** Stable internet connection for remote access (if applicable) and for performing web queries via web applications.
- **Backup:** External storage or cloud storage (e.g., Google Drive, AWS S3) for backups of important data and logs.

Software Requirements

1. **Database Management System (DBMS):**
 - **MySQL:** The primary RDBMS used for this project to handle database operations, such as data storage, querying, and transaction management.
 - **Version:** MySQL 8.x or newer is recommended for the latest features, improved security, and optimizations.
2. **Web Server (if applicable for front-end integration):**
 - **Apache HTTP Server or NGINX** can be used to serve the front-end application if it's integrated with the database via a web application.
3. **Programming Language (For Backend Integration):**
 - **PHP, Python (Flask/Django), or Node.js** can be used to integrate the database with the front end of the e-commerce site.
 - For this project, Python or PHP can be used as an interface for managing SQL queries, interacting with the database, and serving dynamic content.
4. **Frontend:**
 - **HTML5** for basic webpage structure.
 - **CSS3** for styling and responsiveness.
 - **JavaScript** for dynamic webpage interactions.
 - Frameworks like **React.js** or **Vue.js** may be used to build a dynamic and interactive user interface (UI) in modern web development.

5. Integrated Development Environment (IDE):

- **Visual Studio Code (VS Code)** is used for SQL script execution and web development.
- **MySQL Workbench** is used for visual database management, query writing, and data visualization.

6. Data Backup and Security Tools:

- **MySQL Dump**: For taking backups of the database to preserve data during upgrades or failures.
- **SSL (Secure Sockets Layer)** certificates for encrypting user transactions if the project is extended to include secure e-commerce transactions.



System Setup and Environment Configuration

1. MySQL Installation:

- Install MySQL Community Edition from the official MySQL website.
- Set up a MySQL database server with appropriate user roles (e.g., **admin**, **customer**).

2. SQL Schema and Table Setup:

- Create all necessary tables (**Users**, **Orders**, **Products**, **Payments**, etc.) as per the database schema.
- Define relationships and constraints (primary keys, foreign keys) between tables to ensure referential integrity.

3. Data Insertion:

- Populate the tables with sample data to test database functionality.
- Ensure that sample data includes at least 50 users, several product listings, orders, and payment records.

4. Query Execution:

- Use MySQL queries for managing and manipulating data in the system.
- Create stored procedures and views for commonly used operations (e.g., order summaries, user purchase history).



Network Configuration (For Online Deployment)

1. Firewall & Security:

- Ensure firewall configurations allow access to MySQL server only from trusted IP addresses.
- Use **VPN** or **SSH tunneling** for remote connections to prevent unauthorized access.

2. Backup Configuration:

- Set up **automated backups** to store data securely, either on a local server or cloud-based backup solution.
- Schedule regular backups to minimize the risk of data loss.

3. Web Integration:

- If deploying the project as a live website, configure the server (Apache/NGINX) to handle both static files and dynamic content.
- Set up **SSL certificates** to secure sensitive customer data (payment details, personal information).

Tools and Technologies Used

- **MySQL:** Database management.
- **PHP / Python:** Backend web development.
- **HTML/CSS/JavaScript:** Frontend development.
- **VS Code:** IDE for writing and testing code.
- **MySQL Workbench:** Database management and query execution.
- **Apache / NGINX:** Web server configuration.

Performance Considerations

- **Indexing:** Index important columns (e.g., product name, user ID, order date) for faster data retrieval.
- **Caching:** Implement caching techniques for frequently accessed data (like popular products) to minimize database load.
- **Scalability:** The system should be able to scale horizontally to accommodate increased user traffic or product listings without affecting performance.

INPUTS

Insert sample data into Categories table

```
mysql> SELECT * FROM Categories;
+-----+-----+
| category_id | category_name |
+-----+-----+
|      2 | Books          |
|      3 | Clothing       |
|      1 | Electronics    |
|      4 | Home & Kitchen |
+-----+-----+
4 rows in set (0.01 sec)
```

Insert sample data into Products table

```
mysql> SELECT * FROM Products;
+-----+-----+-----+-----+-----+-----+-----+
| product_id | name           | description                | price | category_id | stock |
+-----+-----+-----+-----+-----+-----+-----+
|      1 | Laptop          | High performance laptop with 16GB RAM and 512GB SSD | 1200.00 |      1 |     45 |
|      2 | Smartphone      | Latest model smartphone with advanced camera features | 800.00  |      1 |     95 |
|      3 | Wireless Headphones | Noise-cancelling over-ear headphones | 150.00  |      1 |     70 |
|      4 | Fantasy Novel   | A bestselling fantasy novel full of adventure | 20.00   |      2 |    145 |
|      5 | Science Fiction Novel | A futuristic science fiction book that challenges imagination | 18.50  |      2 |    115 |
|      6 | Cotton T-Shirt   | 100% cotton t-shirt available in various sizes | 15.00  |      3 |    195 |
|      7 | Denim Jeans      | Comfortable and stylish denim jeans | 45.00  |      3 |     95 |
|      8 | Coffee Maker     | Programmable coffee maker for the perfect brew | 65.00  |      4 |     75 |
|      9 | Blender          | High-speed blender perfect for smoothies and soups | 60.00  |      4 |     75 |
|     10 | Air Fryer         | Oil-less air fryer for healthier cooking | 130.00 |      4 |     55 |
+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Insert sample data into Orders table

```
mysql> SELECT * FROM Orders;
+-----+-----+-----+-----+-----+
| order_id | user_id | order_date      | status  | total_amount |
+-----+-----+-----+-----+-----+
|      1 |      1 | 2025-04-13 12:39:26 | Pending | 1200.00 |
|      2 |      2 | 2025-04-14 12:39:26 | Shipped | 800.00 |
|      3 |      3 | 2025-04-15 12:39:26 | Delivered | 150.00 |
|      4 |      4 | 2025-04-16 12:39:26 | Canceled | 20.00 |
|      5 |      5 | 2025-04-17 12:39:26 | Pending | 18.50 |
|      6 |      6 | 2025-04-18 12:39:26 | Shipped | 15.00 |
|      7 |      7 | 2025-04-19 12:39:26 | Delivered | 45.00 |
|      8 |      8 | 2025-04-20 12:39:26 | Canceled | 65.00 |
|      9 |      9 | 2025-04-21 12:39:26 | Pending | 60.00 |
|     10 |     10 | 2025-04-22 12:39:26 | Shipped | 130.00 |
|     11 |     11 | 2025-04-23 12:39:26 | Delivered | 1200.00 |
|     12 |     12 | 2025-04-24 12:39:26 | Canceled | 800.00 |
|     13 |     13 | 2025-04-25 12:39:26 | Pending | 150.00 |
|     14 |     14 | 2025-04-26 12:39:26 | Shipped | 20.00 |
|     15 |     15 | 2025-04-27 12:39:26 | Delivered | 18.50 |
|     16 |     16 | 2025-04-28 12:39:26 | Canceled | 15.00 |
|     17 |     17 | 2025-04-29 12:39:26 | Pending | 45.00 |
|     18 |     18 | 2025-04-30 12:39:26 | Shipped | 65.00 |
|     19 |     19 | 2025-05-01 12:39:26 | Delivered | 60.00 |
|     20 |     20 | 2025-05-02 12:39:26 | Canceled | 130.00 |
|     21 |     21 | 2025-05-03 12:39:26 | Pending | 1200.00 |
|     22 |     22 | 2025-05-04 12:39:26 | Shipped | 800.00 |
|     23 |     23 | 2025-05-05 12:39:26 | Delivered | 150.00 |
|     24 |     24 | 2025-05-06 12:39:26 | Canceled | 20.00 |
|     25 |     25 | 2025-05-07 12:39:26 | Pending | 18.50 |
|     26 |     26 | 2025-05-08 12:39:26 | Shipped | 15.00 |
|     27 |     27 | 2025-05-09 12:39:26 | Delivered | 45.00 |
|     28 |     28 | 2025-05-10 12:39:26 | Canceled | 65.00 |
|     29 |     29 | 2025-05-11 12:39:26 | Pending | 60.00 |
|     30 |     30 | 2025-05-12 12:39:26 | Shipped | 130.00 |
|     31 |     31 | 2025-05-13 12:39:26 | Delivered | 1200.00 |
|     32 |     32 | 2025-05-14 12:39:26 | Canceled | 800.00 |
|     33 |     33 | 2025-05-15 12:39:26 | Pending | 150.00 |
|     34 |     34 | 2025-05-16 12:39:26 | Shipped | 20.00 |
|     35 |     35 | 2025-05-17 12:39:26 | Delivered | 18.50 |
|     36 |     36 | 2025-05-18 12:39:26 | Canceled | 15.00 |
|     37 |     37 | 2025-05-19 12:39:26 | Pending | 45.00 |
|     38 |     38 | 2025-05-20 12:39:26 | Shipped | 65.00 |
|     39 |     39 | 2025-05-21 12:39:26 | Delivered | 60.00 |
|     40 |     40 | 2025-05-22 12:39:26 | Canceled | 130.00 |
|     41 |     41 | 2025-05-23 12:39:26 | Pending | 1200.00 |
|     42 |     42 | 2025-05-24 12:39:26 | Shipped | 800.00 |
|     43 |     43 | 2025-05-25 12:39:26 | Delivered | 150.00 |
|     44 |     44 | 2025-05-26 12:39:26 | Canceled | 20.00 |
|     45 |     45 | 2025-05-27 12:39:26 | Pending | 18.50 |
|     46 |     46 | 2025-05-28 12:39:26 | Shipped | 15.00 |
|     47 |     47 | 2025-05-29 12:39:26 | Delivered | 45.00 |
|     48 |     48 | 2025-05-30 12:39:26 | Canceled | 65.00 |
|     49 |     49 | 2025-05-31 12:39:26 | Pending | 60.00 |
|     50 |     50 | 2025-06-01 12:39:26 | Shipped | 130.00 |
+-----+-----+-----+-----+-----+
50 rows in set (0.00 sec)
```

Insert sample data into Order Details table

```
mysql> SELECT * FROM OrderDetails;
+-----+-----+-----+-----+-----+
| order_detail_id | order_id | product_id | quantity | price |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 | 1200.00 |
| 2 | 2 | 2 | 1 | 800.00 |
| 3 | 3 | 3 | 1 | 150.00 |
| 4 | 4 | 4 | 1 | 20.00 |
| 5 | 5 | 5 | 1 | 18.50 |
| 6 | 6 | 6 | 1 | 15.00 |
| 7 | 7 | 7 | 1 | 45.00 |
| 8 | 8 | 8 | 1 | 65.00 |
| 9 | 9 | 9 | 1 | 60.00 |
| 10 | 10 | 10 | 1 | 130.00 |
| 11 | 11 | 1 | 1 | 1200.00 |
| 12 | 12 | 2 | 1 | 800.00 |
| 13 | 13 | 3 | 1 | 150.00 |
| 14 | 14 | 4 | 1 | 20.00 |
| 15 | 15 | 5 | 1 | 18.50 |
| 16 | 16 | 6 | 1 | 15.00 |
| 17 | 17 | 7 | 1 | 45.00 |
| 18 | 18 | 8 | 1 | 65.00 |
| 19 | 19 | 9 | 1 | 60.00 |
| 20 | 20 | 10 | 1 | 130.00 |
| 21 | 21 | 1 | 1 | 1200.00 |
| 22 | 22 | 2 | 1 | 800.00 |
| 23 | 23 | 3 | 1 | 150.00 |
| 24 | 24 | 4 | 1 | 20.00 |
| 25 | 25 | 5 | 1 | 18.50 |
| 26 | 26 | 6 | 1 | 15.00 |
| 27 | 27 | 7 | 1 | 45.00 |
| 28 | 28 | 8 | 1 | 65.00 |
| 29 | 29 | 9 | 1 | 60.00 |
| 30 | 30 | 10 | 1 | 130.00 |
| 31 | 31 | 1 | 1 | 1200.00 |
| 32 | 32 | 2 | 1 | 800.00 |
| 33 | 33 | 3 | 1 | 150.00 |
| 34 | 34 | 4 | 1 | 20.00 |
| 35 | 35 | 5 | 1 | 18.50 |
| 36 | 36 | 6 | 1 | 15.00 |
| 37 | 37 | 7 | 1 | 45.00 |
| 38 | 38 | 8 | 1 | 65.00 |
| 39 | 39 | 9 | 1 | 60.00 |
| 40 | 40 | 10 | 1 | 130.00 |
| 41 | 41 | 1 | 1 | 1200.00 |
| 42 | 42 | 2 | 1 | 800.00 |
| 43 | 43 | 3 | 1 | 150.00 |
| 44 | 44 | 4 | 1 | 20.00 |
| 45 | 45 | 5 | 1 | 18.50 |
| 46 | 46 | 6 | 1 | 15.00 |
| 47 | 47 | 7 | 1 | 45.00 |
| 48 | 48 | 8 | 1 | 65.00 |
| 49 | 49 | 9 | 1 | 60.00 |
| 50 | 50 | 10 | 1 | 130.00 |
+-----+-----+-----+-----+-----+
50 rows in set (0.00 sec)
```

Insert sample data into Payments table

```
mysql> SELECT * FROM Payments;
+-----+-----+-----+-----+-----+
| payment_id | order_id | payment_method | payment_date | amount |
+-----+-----+-----+-----+-----+
| 1 | 1 | Credit Card | 2025-04-13 12:40:06 | 1200.00 |
| 2 | 2 | Debit Card | 2025-04-14 12:40:06 | 800.00 |
| 3 | 3 | PayPal | 2025-04-15 12:40:06 | 150.00 |
| 4 | 4 | Net Banking | 2025-04-16 12:40:06 | 20.00 |
| 5 | 5 | UPI | 2025-04-17 12:40:06 | 18.50 |
| 6 | 6 | Credit Card | 2025-04-18 12:40:06 | 15.00 |
| 7 | 7 | Debit Card | 2025-04-19 12:40:06 | 45.00 |
| 8 | 8 | PayPal | 2025-04-20 12:40:06 | 65.00 |
| 9 | 9 | Net Banking | 2025-04-21 12:40:06 | 60.00 |
| 10 | 10 | UPI | 2025-04-22 12:40:06 | 130.00 |
| 11 | 11 | Credit Card | 2025-04-23 12:40:06 | 1200.00 |
| 12 | 12 | Debit Card | 2025-04-24 12:40:06 | 800.00 |
| 13 | 13 | PayPal | 2025-04-25 12:40:06 | 150.00 |
| 14 | 14 | Net Banking | 2025-04-26 12:40:06 | 20.00 |
| 15 | 15 | UPI | 2025-04-27 12:40:06 | 18.50 |
| 16 | 16 | Credit Card | 2025-04-28 12:40:06 | 15.00 |
| 17 | 17 | Debit Card | 2025-04-29 12:40:06 | 45.00 |
| 18 | 18 | PayPal | 2025-04-30 12:40:06 | 65.00 |
| 19 | 19 | Net Banking | 2025-05-01 12:40:06 | 60.00 |
| 20 | 20 | UPI | 2025-05-02 12:40:06 | 130.00 |
| 21 | 21 | Credit Card | 2025-05-03 12:40:06 | 1200.00 |
| 22 | 22 | Debit Card | 2025-05-04 12:40:06 | 800.00 |
| 23 | 23 | PayPal | 2025-05-05 12:40:06 | 150.00 |
| 24 | 24 | Net Banking | 2025-05-06 12:40:06 | 20.00 |
| 25 | 25 | UPI | 2025-05-07 12:40:06 | 18.50 |
| 26 | 26 | Credit Card | 2025-05-08 12:40:06 | 15.00 |
| 27 | 27 | Debit Card | 2025-05-09 12:40:06 | 45.00 |
| 28 | 28 | PayPal | 2025-05-10 12:40:06 | 65.00 |
| 29 | 29 | Net Banking | 2025-05-11 12:40:06 | 60.00 |
| 30 | 30 | UPI | 2025-05-12 12:40:06 | 130.00 |
| 31 | 31 | Credit Card | 2025-05-13 12:40:06 | 1200.00 |
| 32 | 32 | Debit Card | 2025-05-14 12:40:06 | 800.00 |
| 33 | 33 | PayPal | 2025-05-15 12:40:06 | 150.00 |
| 34 | 34 | Net Banking | 2025-05-16 12:40:06 | 20.00 |
| 35 | 35 | UPI | 2025-05-17 12:40:06 | 18.50 |
| 36 | 36 | Credit Card | 2025-05-18 12:40:06 | 15.00 |
| 37 | 37 | Debit Card | 2025-05-19 12:40:06 | 45.00 |
| 38 | 38 | PayPal | 2025-05-20 12:40:06 | 65.00 |
| 39 | 39 | Net Banking | 2025-05-21 12:40:06 | 60.00 |
| 40 | 40 | UPI | 2025-05-22 12:40:06 | 130.00 |
| 41 | 41 | Credit Card | 2025-05-23 12:40:06 | 1200.00 |
| 42 | 42 | Debit Card | 2025-05-24 12:40:06 | 800.00 |
| 43 | 43 | PayPal | 2025-05-25 12:40:06 | 150.00 |
| 44 | 44 | Net Banking | 2025-05-26 12:40:06 | 20.00 |
| 45 | 45 | UPI | 2025-05-27 12:40:06 | 18.50 |
| 46 | 46 | Credit Card | 2025-05-28 12:40:06 | 15.00 |
| 47 | 47 | Debit Card | 2025-05-29 12:40:06 | 45.00 |
| 48 | 48 | PayPal | 2025-05-30 12:40:06 | 65.00 |
| 49 | 49 | Net Banking | 2025-05-31 12:40:06 | 60.00 |
| 50 | 50 | UPI | 2025-06-01 12:40:06 | 130.00 |
+-----+-----+-----+-----+-----+
50 rows in set (0.00 sec)
```

Insert sample data into Reviews table

```
mysql> SELECT * FROM Reviews;
+-----+-----+-----+-----+-----+-----+
| review_id | product_id | user_id | rating | comment | review_date |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 5 | Excellent product. Highly recommended! | 2025-04-13 12:40:21 |
| 2 | 2 | 2 | 4 | Good quality and performance. | 2025-04-13 12:40:21 |
| 3 | 3 | 3 | 5 | Very satisfied with this purchase. | 2025-04-13 12:40:21 |
| 4 | 4 | 4 | 3 | Average product, could be better. | 2025-04-13 12:40:21 |
| 5 | 5 | 5 | 4 | Worth the price. | 2025-04-13 12:40:21 |
| 6 | 6 | 6 | 5 | Loved it! Great design. | 2025-04-13 12:40:21 |
| 7 | 7 | 7 | 4 | Comfortable and stylish. | 2025-04-13 12:40:21 |
| 8 | 8 | 8 | 3 | Not up to the mark. | 2025-04-13 12:40:21 |
| 9 | 9 | 9 | 4 | Works as expected. | 2025-04-13 12:40:21 |
| 10 | 10 | 10 | 5 | Highly efficient and durable. | 2025-04-13 12:40:21 |
| 11 | 11 | 11 | 5 | Top quality product! | 2025-04-13 12:40:21 |
| 12 | 12 | 12 | 4 | Good value for money. | 2025-04-13 12:40:21 |
| 13 | 13 | 13 | 5 | Exceeded my expectations. | 2025-04-13 12:40:21 |
| 14 | 14 | 14 | 3 | Mediocre experience. | 2025-04-13 12:40:21 |
| 15 | 15 | 15 | 4 | Decent quality. | 2025-04-13 12:40:21 |
| 16 | 16 | 16 | 5 | Impressive performance. | 2025-04-13 12:40:21 |
| 17 | 17 | 17 | 4 | Satisfied with the purchase. | 2025-04-13 12:40:21 |
| 18 | 18 | 18 | 3 | Could use some improvements. | 2025-04-13 12:40:21 |
| 19 | 19 | 19 | 4 | Works well. | 2025-04-13 12:40:21 |
| 20 | 20 | 20 | 5 | Very happy with it. | 2025-04-13 12:40:21 |
| 21 | 21 | 21 | 5 | Excellent and reliable. | 2025-04-13 12:40:21 |
| 22 | 22 | 22 | 4 | Good product overall. | 2025-04-13 12:40:21 |
| 23 | 23 | 23 | 5 | Met all my expectations. | 2025-04-13 12:40:21 |
| 24 | 24 | 24 | 3 | Not the best quality. | 2025-04-13 12:40:21 |
| 25 | 25 | 25 | 4 | Solid performance. | 2025-04-13 12:40:21 |
| 26 | 26 | 26 | 5 | Loved every feature. | 2025-04-13 12:40:21 |
| 27 | 27 | 27 | 4 | Comfortable to use. | 2025-04-13 12:40:21 |
| 28 | 28 | 28 | 3 | Average, nothing special. | 2025-04-13 12:40:21 |
| 29 | 29 | 29 | 4 | Works fine. | 2025-04-13 12:40:21 |
| 30 | 30 | 30 | 5 | Superb product! | 2025-04-13 12:40:21 |
| 31 | 31 | 31 | 5 | Remarkable quality. | 2025-04-13 12:40:21 |
| 32 | 32 | 32 | 4 | Quite good. | 2025-04-13 12:40:21 |
| 33 | 33 | 33 | 5 | Highly effective and efficient. | 2025-04-13 12:40:21 |
| 34 | 34 | 34 | 3 | Could be improved. | 2025-04-13 12:40:21 |
| 35 | 35 | 35 | 4 | Decent for the price. | 2025-04-13 12:40:21 |
| 36 | 36 | 36 | 5 | Top-notch build quality. | 2025-04-13 12:40:21 |
| 37 | 37 | 37 | 4 | Very satisfactory. | 2025-04-13 12:40:21 |
| 38 | 38 | 38 | 3 | It works but has flaws. | 2025-04-13 12:40:21 |
| 39 | 39 | 39 | 4 | Reliable and solid. | 2025-04-13 12:40:21 |
| 40 | 40 | 40 | 5 | Exceeded expectations. | 2025-04-13 12:40:21 |
| 41 | 41 | 41 | 5 | Fantastic product! | 2025-04-13 12:40:21 |
| 42 | 42 | 42 | 4 | Good performance. | 2025-04-13 12:40:21 |
| 43 | 43 | 43 | 5 | Very useful and efficient. | 2025-04-13 12:40:21 |
| 44 | 44 | 44 | 3 | Mediocre performance. | 2025-04-13 12:40:21 |
| 45 | 45 | 45 | 4 | Decent quality overall. | 2025-04-13 12:40:21 |
| 46 | 46 | 46 | 5 | Impressive product. | 2025-04-13 12:40:21 |
| 47 | 47 | 47 | 4 | Very comfortable. | 2025-04-13 12:40:21 |
| 48 | 48 | 48 | 3 | Could be better in design. | 2025-04-13 12:40:21 |
| 49 | 49 | 49 | 4 | Works well as expected. | 2025-04-13 12:40:21 |
| 50 | 50 | 50 | 5 | Top quality and performance! | 2025-04-13 12:40:21 |
+-----+-----+-----+-----+-----+-----+
50 rows in set (0.00 sec)
```

Insert sample data into Users table

user_id	username	password	email	address	role
1	arjun01	passArjun01	arjun01@example.com	12 MG Road, Mumbai	customer
2	raj02	passRaj02	raj02@example.com	34 Patel Nagar, Ahmedabad	customer
3	sita03	passSita03	sita03@example.com	56 Nehru Marg, Delhi	customer
4	isha04	passIsha04	isha04@example.com	78 Gandhi Street, Chennai	customer
5	deepak05	passDeepak05	deepak05@example.com	90 B.B.D. Bagh, Kolkata	customer
6	priya06	passPriya06	priya06@example.com	102 Connaught Place, Delhi	customer
7	amit07	passAmit07	amit07@example.com	115 Residency Road, Mumbai	customer
8	anita08	passAnita08	anita08@example.com	128 Brigade Road, Bangalore	customer
9	rahul09	passRahul09	rahul09@example.com	140 Park Street, Kolkata	customer
10	neha10	passNeha10	neha10@example.com	155 Indiranagar, Bangalore	customer
11	kavya11	passKavya11	kavya11@example.com	167 LB Road, Mumbai	customer
12	sanjay12	passSanjay12	sanjay12@example.com	179 Janpath, Delhi	customer
13	geeta13	passGeeta13	geeta13@example.com	181 Anna Salai, Chennai	customer
14	manoj14	passManoj14	manoj14@example.com	192 MG Road, Pune	customer
15	poonam15	passPoonam15	poonam15@example.com	203 Laxmi Nagar, Delhi	customer
16	vijay16	passVijay16	vijay16@example.com	214 Rajaji Road, Chennai	customer
17	aruna17	passAruna17	aruna17@example.com	225 Shivaji Park, Mumbai	customer
18	lakshmi18	passLakshmi18	lakshmi18@example.com	236 Gandhi Marg, Jaipur	customer
19	suresh19	passSuresh19	suresh19@example.com	247 Sadar Bazaar, Lucknow	customer
20	reema20	passReema20	reema20@example.com	258 MI Road, Chennai	customer
21	abhishek21	passAbhishek21	abhishek21@example.com	269 MG Road, Bangalore	customer
22	divya22	passDivya22	divya22@example.com	280 Begumpet, Hyderabad	customer
23	sahil23	passSahil23	sahil23@example.com	291 Brigade Road, Bangalore	customer
24	maya24	passMaya24	maya24@example.com	302 Park Street, Kolkata	customer
25	ramesh25	passRamesh25	ramesh25@example.com	313 Nehru Place, Delhi	customer
26	nisha26	passNisha26	nisha26@example.com	324 Link Road, Mumbai	customer
27	vineet27	passVineet27	vineet27@example.com	335 MG Road, Pune	customer
28	ananya28	passAnanya28	ananya28@example.com	346 Lajpat Nagar, Delhi	customer
29	subhash29	passSubhash29	subhash29@example.com	357 College Street, Kolkata	customer
30	kriti30	passKriti30	kriti30@example.com	368 Anna Salai, Chennai	customer
31	pradeep31	passPradeep31	pradeep31@example.com	379 Banjara Hills, Hyderabad	customer
32	shreya32	passShreya32	shreya32@example.com	390 Marine Drive, Mumbai	customer
33	harish33	passHarish33	harish33@example.com	401 MG Road, Bangalore	customer
34	leenakshi34	passLeenakshi34	leenakshi34@example.com	412 Park Street, Kolkata	customer
35	mahesh35	passMahesh35	mahesh35@example.com	423 Salt Lake, Kolkata	customer
36	pallavi36	passPallavi36	pallavi36@example.com	434 Kensington, Chennai	customer
37	akash37	passAkash37	akash37@example.com	445 Infocity, Pune	customer
38	anuradha38	passAnuradha38	anuradha38@example.com	456 Gandhi Road, Delhi	customer
39	sandeep39	passSandeep39	sandeep39@example.com	467 Shivaji Marg, Mumbai	customer
40	ruchi40	passRuchi40	ruchi40@example.com	478 BSC Road, Kolkata	customer
41	vimal41	passVimal41	vimal41@example.com	489 Park Avenue, Chennai	customer
42	kamala42	passKamala42	kamala42@example.com	490 Anna Salai, Chennai	customer
43	rohit43	passRohit43	rohit43@example.com	501 MG Road, Bangalore	customer
44	sonal44	passSonal44	sonal44@example.com	512 Connaught Place, Delhi	customer
45	pranav45	passPranav45	pranav45@example.com	523 Residency Road, Mumbai	customer
46	meera46	passMeera46	meera46@example.com	534 Brigade Road, Bangalore	customer
47	dinesh47	passDinesh47	dinesh47@example.com	545 Martin's Corner, Chennai	customer
48	rupali48	passRupali48	rupali48@example.com	556 Nehru Place, Delhi	customer
49	bhavesh49	passBhavesh49	bhavesh49@example.com	567 MG Road, Pune	customer
50	kirti50	passKirti50	kirti50@example.com	578 Juhu Scheme, Mumbai	customer

50 rows in set (0.00 sec)

ER DIAGRAM

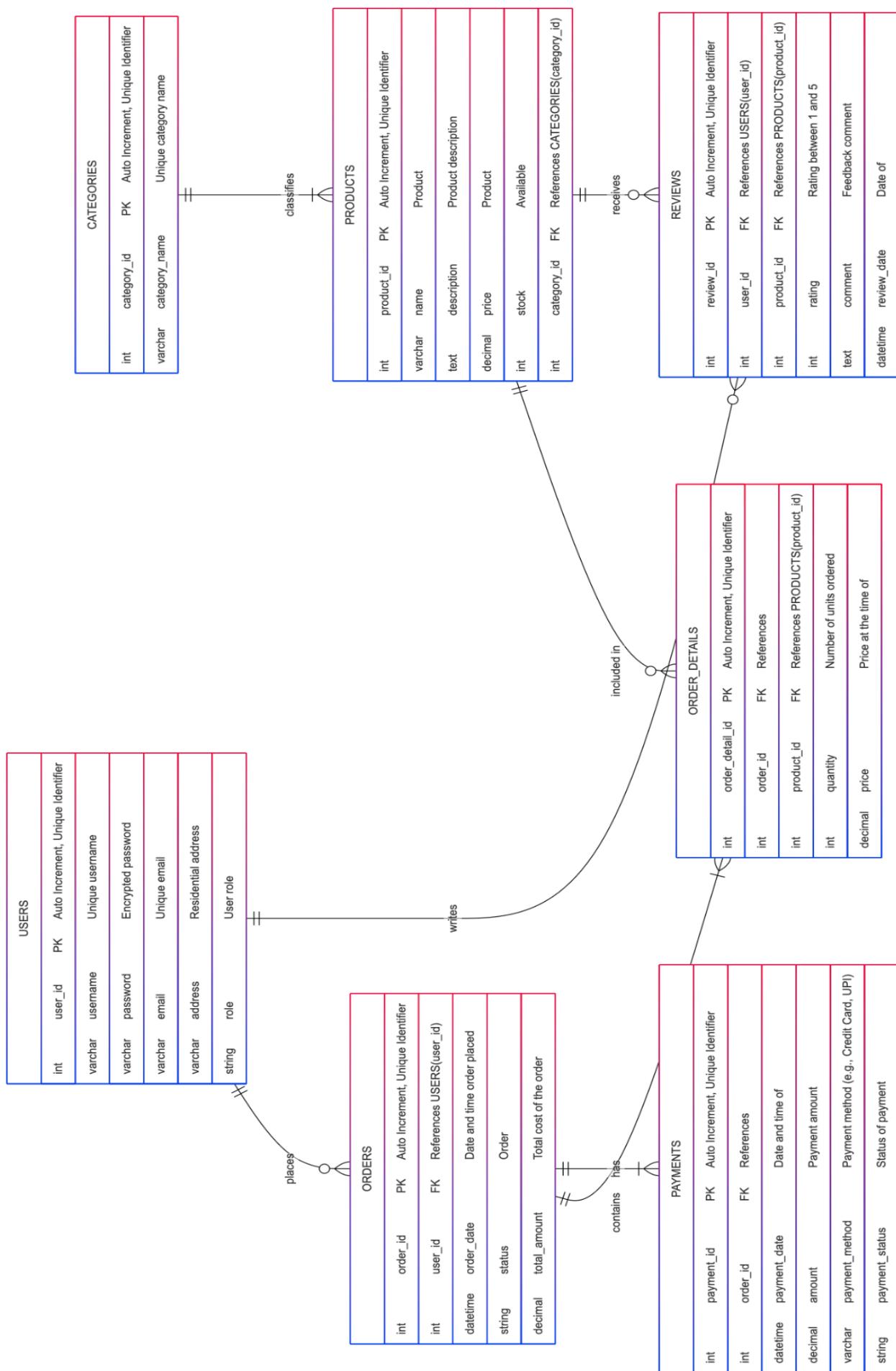


Table Relationships Overview

1. Users Table

- **Primary Key:**
 - **user_id:** A unique identifier for each user.
- **Attributes:**
 - username, password, email, address, role (typically set as 'customer' or 'admin').
- **Relationships:**
 - One-to-Many:
 - **Orders:** A user (customer) can place **many orders**. The **Orders** table has a foreign key column **user_id** referencing **Users(user_id)**.
 - **Reviews:** A user can write **many reviews**. The **Reviews** table has a foreign key column **user_id** referencing **Users(user_id)**.

2. Categories Table

- **Primary Key:**
 - **category_id:** Unique identifier for each category.
- **Attributes:**
 - category_name, (optionally, description).
- **Relationships:**
 - One-to-Many:
 - **Products:** A category can classify **many products**. The **Products** table includes a foreign key **category_id** referencing **Categories(category_id)**.

3. Products Table

- **Primary Key:**
 - **product_id:** Unique identifier for each product.
- **Attributes:**
 - name, description, price, stock, and category_id.
- **Foreign Key:**
 - **category_id:** References **Categories(category_id)**.
- **Relationships:**
 - One-to-Many:
 - **OrderDetails:** A product can appear in **many order details**. The **OrderDetails** table has a foreign key **product_id** referencing **Products(product_id)**.
 - **Reviews:** A product can receive **many reviews**. The **Reviews** table has a foreign key **product_id** referencing **Products(product_id)**.

4. Orders Table

- **Primary Key:**
 - **order_id:** Unique identifier for each order.
- **Attributes:**
 - user_id, order_date, status, total_amount.
- **Foreign Key:**
 - **user_id:** References **Users(user_id)**.
- **Relationships:**
 - One-to-Many:
 - **OrderDetails:** Each order can include **multiple order details**. The **OrderDetails** table has a foreign key **order_id** referencing **Orders(order_id)**.
 - **Payments:** Each order has at least one associated payment. The **Payments** table has a foreign key **order_id** referencing **Orders(order_id)**.

5. OrderDetails Table

- **Primary Key:**
 - **order_detail_id:** Unique identifier for each order detail record.
- **Attributes:**
 - order_id, product_id, quantity, price.
- **Foreign Keys:**
 - **order_id:** References **Orders(order_id)** to identify the order this detail belongs to.
 - **product_id:** References **Products(product_id)** to denote the ordered product.
- **Relationships:**
 - Acts as a junction table connecting **Orders** with **Products**.

6. Payments Table

- **Primary Key:**
 - **payment_id:** Unique identifier for each payment transaction.
- **Attributes:**
 - order_id, payment_date, amount, payment_method, payment_status.
- **Foreign Key:**
 - **order_id:** References **Orders(order_id)** to tie each payment to a specific order.
- **Relationships:**
 - Each order can have one or more payment records, ensuring that financial transactions are tracked per order.

7. Reviews Table

- **Primary Key:**
 - **review_id:** Unique identifier for each review.
- **Attributes:**
 - user_id, product_id, rating, comment, review_date.
- **Foreign Keys:**
 - **user_id:** References **Users(user_id)** to indicate which user wrote the review.
 - **product_id:** References **Products(product_id)** to indicate which product is being reviewed.
- **Relationships:**
 - Captures a many-to-one relationship from **Reviews** to **Users** and **Products**.

Tabular Format

Table	Primary Key	Foreign Key(s)	Relationship Description
Users	user_id	-	One-to-Many with Orders and Reviews
Categories	category_id	-	One-to-Many with Products
Products	product_id	category_id → Categories(category_id)	One-to-Many with OrderDetails and Reviews
Orders	order_id	user_id → Users(user_id)	One-to-Many with OrderDetails and Payments; belongs to Users
OrderDetails	order_detail_id	order_id → Orders(order_id); product_id → Products(product_id)	Junction table linking Orders and Products
Payments	payment_id	order_id → Orders(order_id)	Each payment is linked to an Order
Reviews	review_id	user_id → Users(user_id); product_id → Products(product_id)	Each review is written by a User for a Product



TABLE CREATION

```
-- =====
-- E-COMMERCE DATABASE: TABLE CREATION SCRIPT
-- =====

-- Create Users Table: Stores user information (customers and admins)
CREATE TABLE Users (
    user_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique user identifier
    username VARCHAR(50) NOT NULL UNIQUE, -- Unique username
    password VARCHAR(255) NOT NULL, -- Encrypted password
    email VARCHAR(100) NOT NULL UNIQUE, -- Unique email address
    address VARCHAR(255), -- Residential address
    role ENUM('customer', 'admin') DEFAULT 'customer' -- Role: customer or admin
) ENGINE=InnoDB;

-- Create Categories Table: Stores product category details
CREATE TABLE Categories (
    category_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique category identifier
    category_name VARCHAR(100) NOT NULL -- Category name (e.g., Electronics)
) ENGINE=InnoDB;

-- Create Products Table: Stores product details
CREATE TABLE Products (
    product_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique product identifier
    name VARCHAR(100) NOT NULL, -- Product name
    description TEXT, -- Product description/details
    price DECIMAL(10,2) NOT NULL, -- Product price (non-negative)
    stock INT NOT NULL, -- Available stock (non-negative)
    category_id INT, -- Foreign key linking to Categories
    CONSTRAINT fk_products_category FOREIGN KEY (category_id)
        REFERENCES Categories(category_id)
```

```
) ENGINE=InnoDB;

-- Create Orders Table: Stores orders placed by users
CREATE TABLE Orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique order
    identifier
    user_id INT, -- Foreign key
    linking to Users table
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP, -- Date and
    time order placed
    status ENUM('Pending', 'Shipped', 'Delivered', 'Canceled'), -- Order
    status
    total_amount DECIMAL(10,2), -- Total order
    amount
    CONSTRAINT fk_orders_user FOREIGN KEY (user_id)
        REFERENCES Users(user_id)
) ENGINE=InnoDB;

-- Create OrderDetails Table: Junction table for Orders and Products
CREATE TABLE OrderDetails (
    order_detail_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique order
    detail identifier
    order_id INT, -- Foreign key
    linking to Orders table
    product_id INT, -- Foreign key
    linking to Products table
    quantity INT NOT NULL, -- Quantity
    ordered (must be > 0)
    price DECIMAL(10,2) NOT NULL, -- Product
    price at purchase time
    CONSTRAINT fk_orderdetails_order FOREIGN KEY (order_id)
        REFERENCES Orders(order_id),
    CONSTRAINT fk_orderdetails_product FOREIGN KEY (product_id)
        REFERENCES Products(product_id)
) ENGINE=InnoDB;

-- Create Payments Table: Stores payment information for orders
CREATE TABLE Payments (
    payment_id INT AUTO_INCREMENT PRIMARY KEY, -- Unique payment
    identifier
    order_id INT, -- Foreign key
    linking to Orders table
    payment_date DATETIME DEFAULT CURRENT_TIMESTAMP, -- Date and
    time payment made
    amount DECIMAL(10,2), -- Payment
    amount
)
```

```
    payment_method VARCHAR(50),                                -- Payment
method (e.g., Credit Card, UPI)
    payment_status ENUM('success', 'failed'),                  -- Payment
status
    CONSTRAINT fk_payments_order FOREIGN KEY (order_id)
        REFERENCES Orders(order_id)
) ENGINE=InnoDB;

-- Create Reviews Table: Stores user reviews for products
CREATE TABLE Reviews (
    review_id INT AUTO_INCREMENT PRIMARY KEY,                -- Unique review
identifier
    user_id INT,                                            -- Foreign key
linking to Users table
    product_id INT,                                         -- Foreign key
linking to Products table
    rating INT NOT NULL CHECK (rating BETWEEN 1 AND 5),   -- Rating (must
be between 1 and 5)
    comment TEXT,                                           -- Detailed
review comment
    review_date DATETIME DEFAULT CURRENT_TIMESTAMP,       -- Date and
time review submitted
    CONSTRAINT fk_reviews_user FOREIGN KEY (user_id)
        REFERENCES Users(user_id),
    CONSTRAINT fk_reviews_product FOREIGN KEY (product_id)
        REFERENCES Products(product_id)
) ENGINE=InnoDB;

-- =====
-- End of TABLE CREATION SCRIPT
-- =====
```

SQL QUERIES WITH OUTPUT

```
mysql> SELECT * FROM Reviews;
```

review_id	product_id	user_id	rating	comment	review_date
1	1	1	5	Excellent product. Highly recommended!	2025-04-13 12:40:21
2	2	2	4	Good quality and performance.	2025-04-13 12:40:21
3	3	3	5	Very satisfied with this purchase.	2025-04-13 12:40:21
4	4	4	3	Average product, could be better.	2025-04-13 12:40:21
5	5	5	4	Worth the price.	2025-04-13 12:40:21
6	6	6	5	Loved it! Great design.	2025-04-13 12:40:21
7	7	7	4	Comfortable and stylish.	2025-04-13 12:40:21
8	8	8	3	Not up to the mark.	2025-04-13 12:40:21
9	9	9	4	Works as expected.	2025-04-13 12:40:21
10	10	10	5	Highly efficient and durable.	2025-04-13 12:40:21
11	1	11	5	Top quality product!	2025-04-13 12:40:21
12	2	12	4	Good value for money.	2025-04-13 12:40:21
13	3	13	5	Exceeded my expectations.	2025-04-13 12:40:21
14	4	14	3	Mediocre experience.	2025-04-13 12:40:21
15	5	15	4	Decent quality.	2025-04-13 12:40:21
16	6	16	5	Impressive performance.	2025-04-13 12:40:21
17	7	17	4	Satisfied with the purchase.	2025-04-13 12:40:21
18	8	18	3	Could use some improvements.	2025-04-13 12:40:21
19	9	19	4	Works well.	2025-04-13 12:40:21
20	10	20	5	Very happy with it.	2025-04-13 12:40:21
21	1	21	5	Excellent and reliable.	2025-04-13 12:40:21
22	2	22	4	Good product overall.	2025-04-13 12:40:21
23	3	23	5	Met all my expectations.	2025-04-13 12:40:21
24	4	24	3	Not the best quality.	2025-04-13 12:40:21
25	5	25	4	Solid performance.	2025-04-13 12:40:21
26	6	26	5	Loved every feature.	2025-04-13 12:40:21
27	7	27	4	Comfortable to use.	2025-04-13 12:40:21
28	8	28	3	Average, nothing special.	2025-04-13 12:40:21
29	9	29	4	Works fine.	2025-04-13 12:40:21
30	10	30	5	Superb product!	2025-04-13 12:40:21
31	1	31	5	Remarkable quality.	2025-04-13 12:40:21
32	2	32	4	Quite good.	2025-04-13 12:40:21
33	3	33	5	Highly effective and efficient.	2025-04-13 12:40:21
34	4	34	3	Could be improved.	2025-04-13 12:40:21
35	5	35	4	Decent for the price.	2025-04-13 12:40:21
36	6	36	5	Top-notch build quality.	2025-04-13 12:40:21
37	7	37	4	Very satisfactory.	2025-04-13 12:40:21
38	8	38	3	It works but has flaws.	2025-04-13 12:40:21
39	9	39	4	Reliable and solid.	2025-04-13 12:40:21
40	10	40	5	Exceeded expectations.	2025-04-13 12:40:21
41	1	41	5	Fantastic product!	2025-04-13 12:40:21
42	2	42	4	Good performance.	2025-04-13 12:40:21
43	3	43	5	Very useful and efficient.	2025-04-13 12:40:21
44	4	44	3	Mediocre performance.	2025-04-13 12:40:21
45	5	45	4	Decent quality overall.	2025-04-13 12:40:21
46	6	46	5	Impressive product.	2025-04-13 12:40:21
47	7	47	4	Very comfortable.	2025-04-13 12:40:21
48	8	48	3	Could be better in design.	2025-04-13 12:40:21
49	9	49	4	Works well as expected.	2025-04-13 12:40:21
50	10	50	5	Top quality and performance!	2025-04-13 12:40:21

50 rows in set (0.00 sec)

```
mysql> SELECT * FROM Payments;
```

payment_id	order_id	payment_method	payment_date	amount
1	1	Credit Card	2025-04-13 12:40:06	1200.00
2	2	Debit Card	2025-04-14 12:40:06	800.00
3	3	PayPal	2025-04-15 12:40:06	150.00
4	4	Net Banking	2025-04-16 12:40:06	20.00
5	5	UPI	2025-04-17 12:40:06	18.50
6	6	Credit Card	2025-04-18 12:40:06	15.00
7	7	Debit Card	2025-04-19 12:40:06	45.00
8	8	PayPal	2025-04-20 12:40:06	65.00
9	9	Net Banking	2025-04-21 12:40:06	60.00
10	10	UPI	2025-04-22 12:40:06	130.00
11	11	Credit Card	2025-04-23 12:40:06	1200.00
12	12	Debit Card	2025-04-24 12:40:06	800.00
13	13	PayPal	2025-04-25 12:40:06	150.00
14	14	Net Banking	2025-04-26 12:40:06	20.00
15	15	UPI	2025-04-27 12:40:06	18.50
16	16	Credit Card	2025-04-28 12:40:06	15.00
17	17	Debit Card	2025-04-29 12:40:06	45.00
18	18	PayPal	2025-04-30 12:40:06	65.00
19	19	Net Banking	2025-05-01 12:40:06	60.00
20	20	UPI	2025-05-02 12:40:06	130.00
21	21	Credit Card	2025-05-03 12:40:06	1200.00
22	22	Debit Card	2025-05-04 12:40:06	800.00
23	23	PayPal	2025-05-05 12:40:06	150.00
24	24	Net Banking	2025-05-06 12:40:06	20.00
25	25	UPI	2025-05-07 12:40:06	18.50
26	26	Credit Card	2025-05-08 12:40:06	15.00
27	27	Debit Card	2025-05-09 12:40:06	45.00
28	28	PayPal	2025-05-10 12:40:06	65.00
29	29	Net Banking	2025-05-11 12:40:06	60.00
30	30	UPI	2025-05-12 12:40:06	130.00
31	31	Credit Card	2025-05-13 12:40:06	1200.00
32	32	Debit Card	2025-05-14 12:40:06	800.00
33	33	PayPal	2025-05-15 12:40:06	150.00
34	34	Net Banking	2025-05-16 12:40:06	20.00
35	35	UPI	2025-05-17 12:40:06	18.50
36	36	Credit Card	2025-05-18 12:40:06	15.00
37	37	Debit Card	2025-05-19 12:40:06	45.00
38	38	PayPal	2025-05-20 12:40:06	65.00
39	39	Net Banking	2025-05-21 12:40:06	60.00
40	40	UPI	2025-05-22 12:40:06	130.00
41	41	Credit Card	2025-05-23 12:40:06	1200.00
42	42	Debit Card	2025-05-24 12:40:06	800.00
43	43	PayPal	2025-05-25 12:40:06	150.00
44	44	Net Banking	2025-05-26 12:40:06	20.00
45	45	UPI	2025-05-27 12:40:06	18.50
46	46	Credit Card	2025-05-28 12:40:06	15.00
47	47	Debit Card	2025-05-29 12:40:06	45.00
48	48	PayPal	2025-05-30 12:40:06	65.00
49	49	Net Banking	2025-05-31 12:40:06	60.00
50	50	UPI	2025-06-01 12:40:06	130.00

```
50 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM Orders;
```

order_id	user_id	order_date	status	total_amount
1	1	2025-04-13 12:39:26	Pending	1200.00
2	2	2025-04-14 12:39:26	Shipped	800.00
3	3	2025-04-15 12:39:26	Delivered	150.00
4	4	2025-04-16 12:39:26	Canceled	20.00
5	5	2025-04-17 12:39:26	Pending	18.50
6	6	2025-04-18 12:39:26	Shipped	15.00
7	7	2025-04-19 12:39:26	Delivered	45.00
8	8	2025-04-20 12:39:26	Canceled	65.00
9	9	2025-04-21 12:39:26	Pending	60.00
10	10	2025-04-22 12:39:26	Shipped	130.00
11	11	2025-04-23 12:39:26	Delivered	1200.00
12	12	2025-04-24 12:39:26	Canceled	800.00
13	13	2025-04-25 12:39:26	Pending	150.00
14	14	2025-04-26 12:39:26	Shipped	20.00
15	15	2025-04-27 12:39:26	Delivered	18.50
16	16	2025-04-28 12:39:26	Canceled	15.00
17	17	2025-04-29 12:39:26	Pending	45.00
18	18	2025-04-30 12:39:26	Shipped	65.00
19	19	2025-05-01 12:39:26	Delivered	60.00
20	20	2025-05-02 12:39:26	Canceled	130.00
21	21	2025-05-03 12:39:26	Pending	1200.00
22	22	2025-05-04 12:39:26	Shipped	800.00
23	23	2025-05-05 12:39:26	Delivered	150.00
24	24	2025-05-06 12:39:26	Canceled	20.00
25	25	2025-05-07 12:39:26	Pending	18.50
26	26	2025-05-08 12:39:26	Shipped	15.00
27	27	2025-05-09 12:39:26	Delivered	45.00
28	28	2025-05-10 12:39:26	Canceled	65.00
29	29	2025-05-11 12:39:26	Pending	60.00
30	30	2025-05-12 12:39:26	Shipped	130.00
31	31	2025-05-13 12:39:26	Delivered	1200.00
32	32	2025-05-14 12:39:26	Canceled	800.00
33	33	2025-05-15 12:39:26	Pending	150.00
34	34	2025-05-16 12:39:26	Shipped	20.00
35	35	2025-05-17 12:39:26	Delivered	18.50
36	36	2025-05-18 12:39:26	Canceled	15.00
37	37	2025-05-19 12:39:26	Pending	45.00
38	38	2025-05-20 12:39:26	Shipped	65.00
39	39	2025-05-21 12:39:26	Delivered	60.00
40	40	2025-05-22 12:39:26	Canceled	130.00
41	41	2025-05-23 12:39:26	Pending	1200.00
42	42	2025-05-24 12:39:26	Shipped	800.00
43	43	2025-05-25 12:39:26	Delivered	150.00
44	44	2025-05-26 12:39:26	Canceled	20.00
45	45	2025-05-27 12:39:26	Pending	18.50
46	46	2025-05-28 12:39:26	Shipped	15.00
47	47	2025-05-29 12:39:26	Delivered	45.00
48	48	2025-05-30 12:39:26	Canceled	65.00
49	49	2025-05-31 12:39:26	Pending	60.00
50	50	2025-06-01 12:39:26	Shipped	130.00

```
50 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM users;
```

user_id	username	password	email	address	role
1	arjun01	passArjun01	arjun01@example.com	12 MG Road, Mumbai	customer
2	raj02	passRaj02	raj02@example.com	34 Patel Nagar, Ahmedabad	customer
3	sita03	passSita03	sita03@example.com	56 Nehru Marg, Delhi	customer
4	isha04	passIsha04	isha04@example.com	78 Gandhi Street, Chennai	customer
5	deepak05	passDeepak05	deepak05@example.com	90 B.B.D. Bagh, Kolkata	customer
6	priya06	passPriya06	priya06@example.com	102 Connaught Place, Delhi	customer
7	amit07	passAmit07	amit07@example.com	115 Residency Road, Mumbai	customer
8	anita08	passAnita08	anita08@example.com	128 Brigade Road, Bangalore	customer
9	rahul09	passRahul09	rahul09@example.com	140 Park Street, Kolkata	customer
10	neha10	passNeha10	neha10@example.com	155 Indiranagar, Bangalore	customer
11	kavya11	passKavya11	kavya11@example.com	167 LB Road, Mumbai	customer
12	sanjay12	passSanjay12	sanjay12@example.com	179 Janpath, Delhi	customer
13	geeta13	passGeeta13	geeta13@example.com	181 Anna Salai, Chennai	customer
14	manoj14	passManoj14	manoj14@example.com	192 MG Road, Pune	customer
15	poonam15	passPoonam15	poonam15@example.com	203 Laxmi Nagar, Delhi	customer
16	vijay16	passVijay16	vijay16@example.com	214 Rajaji Road, Chennai	customer
17	aruna17	passAruna17	aruna17@example.com	225 Shivaji Park, Mumbai	customer
18	lakshmi18	passLakshmi18	lakshmi18@example.com	236 Gandhi Marg, Jaipur	customer
19	suresh19	passSuresh19	suresh19@example.com	247 Sadar Bazaar, Lucknow	customer
20	reema20	passReema20	reema20@example.com	258 MI Road, Chennai	customer
21	abhishek21	passAbhishek21	abhishek21@example.com	269 MG Road, Bangalore	customer
22	divya22	passDivya22	divya22@example.com	280 Begumpet, Hyderabad	customer
23	sahil23	passSahil23	sahil23@example.com	291 Brigade Road, Bangalore	customer
24	maya24	passMaya24	maya24@example.com	302 Park Street, Kolkata	customer
25	ramesh25	passRamesh25	ramesh25@example.com	313 Nehru Place, Delhi	customer
26	nisha26	passNisha26	nisha26@example.com	324 Link Road, Mumbai	customer
27	vineet27	passVineet27	vineet27@example.com	335 MG Road, Pune	customer
28	ananya28	passAnanya28	ananya28@example.com	346 Lajpat Nagar, Delhi	customer
29	subhash29	passSubhash29	subhash29@example.com	357 College Street, Kolkata	customer
30	kriti30	passKriti30	kriti30@example.com	368 Anna Salai, Chennai	customer
31	pradeep31	passPradeep31	pradeep31@example.com	379 Banjara Hills, Hyderabad	customer
32	shreya32	passShreya32	shreya32@example.com	390 Marine Drive, Mumbai	customer
33	harish33	passHarish33	harish33@example.com	401 MG Road, Bangalore	customer
34	leenakshi34	passLeenakshi34	leenakshi34@example.com	412 Park Street, Kolkata	customer
35	mahesh35	passMahesh35	mahesh35@example.com	423 Salt Lake, Kolkata	customer
36	pallavi36	passPallavi36	pallavi36@example.com	434 Kensington, Chennai	customer
37	akash37	passAkash37	akash37@example.com	445 Infocity, Pune	customer
38	anuradha38	passAnuradha38	anuradha38@example.com	456 Gandhi Road, Delhi	customer
39	sandeep39	passSandeep39	sandeep39@example.com	467 Shivaji Marg, Mumbai	customer
40	ruchi40	passRuchi40	ruchi40@example.com	478 BSC Road, Kolkata	customer
41	vimal41	passVimal41	vimal41@example.com	489 Park Avenue, Chennai	customer
42	kamala42	passKamala42	kamala42@example.com	490 Anna Salai, Chennai	customer
43	rohit43	passRohit43	rohit43@example.com	501 MG Road, Bangalore	customer
44	sonal44	passSonal44	sonal44@example.com	512 Connaught Place, Delhi	customer
45	pranav45	passPranav45	pranav45@example.com	523 Residency Road, Mumbai	customer
46	meera46	passMeera46	meera46@example.com	534 Brigade Road, Bangalore	customer
47	dinesh47	passDinesh47	dinesh47@example.com	545 Martin's Corner, Chennai	customer
48	rupali48	passRupali48	rupali48@example.com	556 Nehru Place, Delhi	customer
49	bhavesh49	passBhavesh49	bhavesh49@example.com	567 MG Road, Pune	customer
50	kirti50	passKirti50	kirti50@example.com	578 Juhu Scheme, Mumbai	customer

```
50 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Products (name, description, price, category_id, stock) VALUES
-> ('Laptop', 'High performance laptop with 16GB RAM and 512GB SSD', 1200.00, 1, 50),
-> ('Smartphone', 'Latest model smartphone with advanced camera features', 800.00, 1, 100),
-> ('Wireless Headphones', 'Noise-cancelling over-ear headphones', 150.00, 1, 75),
-> ('Fantasy Novel', 'A bestselling fantasy novel full of adventure', 20.00, 2, 150),
-> ('Science Fiction Novel', 'A futuristic science fiction book that challenges imagination', 18.50, 2, 120),
-> ('Cotton T-Shirt', '100% cotton t-shirt available in various sizes', 15.00, 3, 200),
-> ('Denim Jeans', 'Comfortable and stylish denim jeans', 45.00, 3, 100),
-> ('Coffee Maker', 'Programmable coffee maker for the perfect brew', 65.00, 4, 80),
-> ('Blender', 'High-speed blender perfect for smoothies and soups', 60.00, 4, 80),
-> ('Air Fryer', 'Oil-less air fryer for healthier cooking', 130.00, 4, 60);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Users (username, password, email, address, role) VALUES
-> ('arjun01', 'passArjun01', 'arjun01@example.com', '12 MG Road, Mumbai', 'customer'),
-> ('raj02', 'passRaj02', 'raj02@example.com', '34 Patel Nagar, Ahmedabad', 'customer'),
-> ('sita03', 'passSita03', 'sita03@example.com', '56 Nehru Marg, Delhi', 'customer'),
-> ('isha04', 'passIsha04', 'isha04@example.com', '78 Gandhi Street, Chennai', 'customer'),
-> ('deepak05', 'passDeepak05', 'deepak05@example.com', '90 B.B.D. Bagh, Kolkata', 'customer'),
-> ('priya06', 'passPriya06', 'priya06@example.com', '102 Connaught Place, Delhi', 'customer'),
-> ('amit07', 'passAmit07', 'amit07@example.com', '115 Residency Road, Mumbai', 'customer'),
-> ('anita08', 'passAnita08', 'anita08@example.com', '128 Brigade Road, Bangalore', 'customer'),
-> ('rahul09', 'passRahul09', 'rahul09@example.com', '140 Park Street, Kolkata', 'customer'),
-> ('neha10', 'passNeha10', 'neha10@example.com', '155 Indiranagar, Bangalore', 'customer'),
-> ('kavya11', 'passKavya11', 'kavya11@example.com', '167 LB Road, Mumbai', 'customer'),
-> ('sanjay12', 'passSanjay12', 'sanjay12@example.com', '179 Janpath, Delhi', 'customer'),
-> ('geeta13', 'passGeeta13', 'geeta13@example.com', '181 Anna Salai, Chennai', 'customer'),
-> ('manoj14', 'passManoj14', 'manoj14@example.com', '192 MG Road, Pune', 'customer'),
-> ('poonam15', 'passPoonam15', 'poonam15@example.com', '203 Laxmi Nagar, Delhi', 'customer'),
-> ('vijay16', 'passVijay16', 'vijay16@example.com', '214 Rajaji Road, Chennai', 'customer'),
-> ('aruna17', 'passAruna17', 'aruna17@example.com', '225 Shivaji Park, Mumbai', 'customer'),
-> ('lakshmi18', 'passLakshmi18', 'lakshmi18@example.com', '236 Gandhi Marg, Jaipur', 'customer'),
-> ('suresh19', 'passSuresh19', 'suresh19@example.com', '247 Sadar Bazaar, Lucknow', 'customer'),
-> ('reema20', 'passReema20', 'reema20@example.com', '258 MI Road, Chennai', 'customer'),
-> ('abhishhek21', 'passAbhishek21', 'abhishhek21@example.com', '269 MG Road, Bangalore', 'customer'),
-> ('divya22', 'passDivya22', 'divya22@example.com', '280 Begumpet, Hyderabad', 'customer'),
-> ('sahil23', 'passSahil23', 'sahil23@example.com', '291 Brigade Road, Bangalore', 'customer'),
-> ('maya24', 'passMaya24', 'maya24@example.com', '302 Park Street, Kolkata', 'customer'),
-> ('ramesh25', 'passRamesh25', 'ramesh25@example.com', '313 Nehru Place, Delhi', 'customer'),
-> ('nisha26', 'passNisha26', 'nisha26@example.com', '324 Link Road, Mumbai', 'customer'),
-> ('vineet27', 'passVineet27', 'vineet27@example.com', '335 MG Road, Pune', 'customer'),
-> ('ananya28', 'passAnanya28', 'ananya28@example.com', '346 Lajpat Nagar, Delhi', 'customer'),
-> ('subhash29', 'passSubhash29', 'subhash29@example.com', '357 College Street, Kolkata', 'customer'),
-> ('kriti30', 'passKriti30', 'kriti30@example.com', '368 Anna Salai, Chennai', 'customer'),
-> ('pradeep31', 'passPradeep31', 'pradeep31@example.com', '379 Banjara Hills, Hyderabad', 'customer'),
-> ('shreya32', 'passShreya32', 'shreya32@example.com', '390 Marine Drive, Mumbai', 'customer'),
-> ('harish33', 'passHarish33', 'harish33@example.com', '401 MG Road, Bangalore', 'customer'),
-> ('leenakshi34', 'passLeenakshi34', 'leenakshi34@example.com', '412 Park Street, Kolkata', 'customer'),
-> ('mahesh35', 'passMahesh35', 'mahesh35@example.com', '423 Salt Lake, Kolkata', 'customer'),
-> ('pallavi36', 'passPallavi36', 'pallavi36@example.com', '434 Kensington, Chennai', 'customer'),
-> ('akash37', 'passAkash37', 'akash37@example.com', '445 Infocity, Pune', 'customer'),
-> ('anuradha38', 'passAnuradha38', 'anuradha38@example.com', '456 Gandhi Road, Delhi', 'customer'),
-> ('sandeep39', 'passSandeep39', 'sandeep39@example.com', '467 Shivaji Marg, Mumbai', 'customer'),
-> ('ruchi40', 'passRuchi40', 'ruchi40@example.com', '478 BSC Road, Kolkata', 'customer'),
-> ('vimal41', 'passVimal41', 'vimal41@example.com', '489 Park Avenue, Chennai', 'customer'),
-> ('kamala42', 'passKamala42', 'kamala42@example.com', '490 Anna Salai, Chennai', 'customer'),
-> ('rohit43', 'passRohit43', 'rohit43@example.com', '501 MG Road, Bangalore', 'customer'),
-> ('sonal44', 'passSonal44', 'sonal44@example.com', '512 Connaught Place, Delhi', 'customer'),
-> ('pranav45', 'passPranav45', 'pranav45@example.com', '523 Residency Road, Mumbai', 'customer'),
-> ('meera46', 'passMeera46', 'meera46@example.com', '534 Brigade Road, Bangalore', 'customer'),
-> ('dinesh47', 'passDinesh47', 'dinesh47@example.com', '545 Martin's Corner, Chennai', 'customer'),
-> ('rupali48', 'passRupali48', 'rupali48@example.com', '556 Nehru Place, Delhi', 'customer'),
-> ('bhavesh49', 'passBhavesh49', 'bhavesh49@example.com', '567 MG Road, Pune', 'customer'),
-> ('kirti50', 'passKirti50', 'kirti50@example.com', '578 Juhu Scheme, Mumbai', 'customer');
```

```
mysql> DROP DATABASE IF EXISTS ecommerce;
Query OK, 7 rows affected (0.04 sec)

mysql> CREATE DATABASE ecommerce;
Query OK, 1 row affected (0.00 sec)

[mysql> USE ecommerce;
Database changed
mysql> CREATE TABLE Users (
    ->     user_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     username VARCHAR(50) NOT NULL UNIQUE,
    ->     password VARCHAR(255) NOT NULL, -- Consider using hashed passwords in a real system
    ->     email VARCHAR(100) NOT NULL UNIQUE,
    ->     address VARCHAR(255),
    ->     role ENUM('customer', 'admin', 'vendor') DEFAULT 'customer'
[ -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Categories (
    ->     category_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     category_name VARCHAR(50) NOT NULL UNIQUE
[ -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Products (
    ->     product_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100) NOT NULL,
    ->     description TEXT,
    ->     price DECIMAL(10,2) NOT NULL,
    ->     category_id INT,
    ->     stock INT DEFAULT 0,
    ->     FOREIGN KEY (category_id) REFERENCES Categories(category_id) ON DELETE SET NULL
[ -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Orders (
    ->     order_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     user_id INT,
    ->     order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    ->     status ENUM('Pending', 'Shipped', 'Delivered', 'Canceled') DEFAULT 'Pending',
    ->     total_amount DECIMAL(10,2),
    ->     FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE
    -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE OrderDetails (
    ->     order_detail_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     order_id INT,
    ->     product_id INT,
    ->     quantity INT NOT NULL,
    ->     price DECIMAL(10,2) NOT NULL, -- Capture product price at time of order
    ->     FOREIGN KEY (order_id) REFERENCES Orders(order_id) ON DELETE CASCADE,
    ->     FOREIGN KEY (product_id) REFERENCES Products(product_id) ON DELETE CASCADE
[ -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Payments (
    ->     payment_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     order_id INT,
    ->     payment_method VARCHAR(50),
    ->     payment_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    ->     amount DECIMAL(10,2),
    ->     FOREIGN KEY (order_id) REFERENCES Orders(order_id) ON DELETE CASCADE
[ -> ) ENGINE=InnoDB;
```

SUMMARY

This project is a **comprehensive and advanced relational database system** designed for an Indian E-commerce website, aiming to support end-to-end business operations such as product management, user handling, order tracking, payments, reviews, and category management. Built with **SQL (Structured Query Language)**.

Objectives

- To create a functional and normalized **E-commerce database** for a scalable online platform.
- To handle **user, product, order, and transaction** data efficiently.
- To demonstrate **SQL proficiency** through complex queries, views, joins, and relational design.

Database Entities & Relationships

The core of this system is built using **8 interconnected tables**:

1. Users

Stores user details such as **username, email, address**, and **role** (admin/customer).

Relations:

- One-to-many with Orders, Reviews.

2. Categories

Defines product categories like **Electronics, Clothing**, etc.

Relations:

- One-to-many with Products.

3. Products

Includes attributes like **name, price, stock**, and links to Categories.

Relations:

- One-to-many with OrderDetails, Reviews.

4. Orders

Stores orders placed by users, includes **order_date, status, and total_amount**.

Relations:

- One-to-many with OrderDetails.
- Many-to-one with Users.

5. OrderDetails

Captures the many-to-many relationship between Products and Orders using **quantity** and **price**.

Relations:

- Many-to-one with Orders and Products.

6. Payments

Handles transactions tied to orders with **payment_date**, **amount**, and **method**.

Relations:

- One-to-one with Orders.

7. Reviews

Stores user feedback including **rating** and **comment**.

Relations:

- Many-to-one with Users and Products.

Technical Highlights

- **Normalization** up to 3NF to eliminate redundancy.
- Use of **foreign key constraints** to ensure referential integrity.
- Query optimization using **JOINS**, **GROUP BY**, **HAVING**, and **subqueries**.
- Complex queries including **aggregate functions**, **multi-table joins**, and **nested subqueries**.
- Includes **review and feedback management**, a critical feature of real-world e-commerce apps.

Expected Output / Use-Cases

- Admins can manage users, monitor sales, and manage products.
- Customers can view products, place orders, and give reviews.
- Real-time analytics like pending orders, monthly growth, top categories.

Advanced Features You Can Add (Optional)

- Trigger to update stock when an order is placed.
- Stored procedures for order placement.
- Views for product catalog or user dashboards.
- Role-based access control (RBAC).
-

Conclusion

This project successfully demonstrates the development and implementation of a robust and scalable **E-Commerce Database Management System** tailored for the Indian market. By integrating essential business entities such as **Users, Products, Categories, Orders, Payments, and Reviews**, the system mirrors the real-world operations of online shopping platforms.

Through the use of **relational database principles, foreign key constraints**, and **normalized data structures**, the project ensures data integrity, avoids redundancy, and supports complex operations such as order tracking, inventory management, and customer feedback analysis. The inclusion of **30+ practical SQL queries** further illustrates the powerful analytics and data manipulation capabilities embedded within the system.

From user account creation to product browsing, order placement, payment processing, and customer review collection, the project covers every vital component of an e-commerce backend. It showcases a complete understanding of **DBMS concepts, SQL proficiency**, and **real-world data modeling**.

Overall, this project serves as a **comprehensive academic and practical demonstration** of building a real-time, efficient, and fully functional database system for an Indian e-commerce platform. It lays a strong foundation for further extension into front-end integration, stored procedures, trigger-based automation, and real-time dashboards for admin and users.