

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

*Advanced Database SYSTEMS project*

*TITLE: e-commerce*

**Background**  
Introduction (Background)

E-commerce has dramatically transformed the global marketplace, revolutionizing the way businesses operate and consumers interact with products and services. By leveraging digital platforms, businesses can now reach a broader audience, break geographical barriers, and provide 24/7 accessibility. Customers, in turn, benefit from the convenience of shopping from their homes, access to diverse product options, and simplified purchasing processes. This paradigm shift has not only enhanced customer experiences but also fostered competitive innovation among businesses.

However, with the increasing complexity and scale of E-commerce operations, managing data efficiently has become a critical challenge. Key data management aspects include tracking user information, product inventories, orders, and customer interactions, which are essential for ensuring seamless functionality. A well-designed database architecture is the backbone of any successful E-commerce platform. It supports scalable data storage, quick retrieval, robust transaction handling, and consistent data integrity, which are vital for both operational and strategic decision-making.

The provided E-commerce project focuses on addressing these challenges by introducing a comprehensive database design built on relational principles and Entity-Relationship (ER) modeling. This design integrates key entities such as users, products, categories, orders, and shopping carts, along with their relationships. The system's capabilities encompass user management, product categorization, order placement, and the tracking of customer shopping behaviors. Each feature is underpinned by efficient query processing and data handling mechanisms to ensure high performance and reliability.

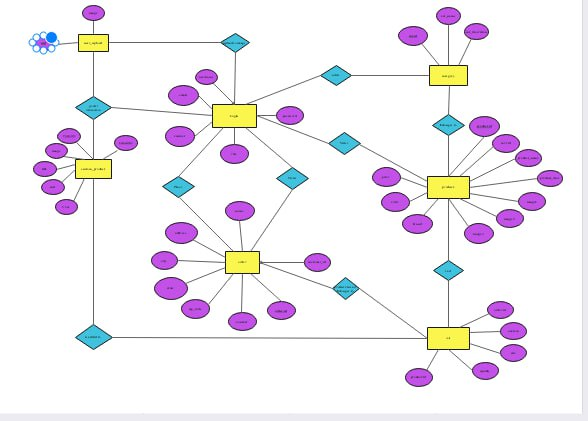
Additionally, the inclusion of vertical, horizontal, and mixed fragmentation techniques within the database structure enhances its adaptability and performance in distributed environments. These techniques allow data to be partitioned logically or physically based on usage patterns, further optimizing query efficiency and minimizing system latency.

In summary, this project serves as a practical demonstration of how an E-commerce platform can achieve operational efficiency and scalability through robust database architecture. It highlights the importance of thoughtful system design in meeting the dynamic demands of the digital marketplace while ensuring an exceptional experience for users and businesses alike.

**Project Title : E-Commerce**

**Step** 1**. Entity-Relationship (ER) Diagram**:

Our E-Commerce ER diagram provided here is , it typically includes entities (tables), relationships between entities, and attributes associated with each entity.



**Entities and Attributes:**

**1. UserUpload:** Stores user uploads, primarily images.

**Attributes:** UserID (Primary Key), Image

**2. CustomerProducts:** Links products with customers and users who uploaded the product-related data.

**Attributes:**  ProductID (Primary Key), Title, Image, Cost

UserID (Foreign Key, references UserUpload.UserID)

CustomerID (Foreign Key, references Login.UserID)

**3. Login:** Manages user authentication and basic user details.

**Attributes:** UserID (Primary Key), Username, Email, Contact, Role, Password

**4. Order:** Stores information about customer orders.

**Attributes:**

OrderID (Primary Key), Name, Address, City, State, ZipCode, Contact

CustomerID (Foreign Key, references Login.UserID)

1. **Category:** Categorizes products into groups.

**Attributes:** CatID (Primary Key)**,** CatName**,** CatDescription

1. **Product :** Stores detailed information about products.

**Attributes:** ProductID (Primary Key)**,** Price**,** Color**,** Brand

CatID (Foreign Key, references Category.CatID), ProductName, ProductDesc

Image1, Image2, Image3

1. **Cart :** Represents items added to a customer's cart.

**Attributes:** OrderID (Primary Key)**,** CustomerID (Foreign Key, references Login.UserID)**,** Size**,** Quantity

ProductID (Foreign Key, references Product.ProductID)

**Entity Relationship  
  
1. User Upload ↔️ Custom Product**

**1. Relationship:** Uploads

**2. Description:** A user uploads a custom product. Each User Upload (represented by uid) can be associated with multiple Custom Products.

**2. Login ↔️ Category**

**1. Relationship:** Views

**2. Description:** Users (represented by the Login entity) can view multiple categories of products. Similarly, a single category can be viewed by multiple users.

**3. Category ↔️ Product**

**1. Relationship:** Adds

**2. Description:** Each Category can have multiple Products, and each Product belongs to a specific category**.**

**4. Login ↔️ Order**

**1. Relationship:** Places

**2. Description:** Users (via Login) can place multiple Orders, but each order is placed by one user.

**5. Order ↔️ Cart**

**1. Relationship**: Is Added To

**2. Description:** An Order can have multiple Carts associated with it, each containing different products, sizes, or quantities.

**6. Product ↔️ Cart**

**1. Relationship:** Belongs To

**2. Description:** A Product can belong to multiple Carts, and a cart can contain multiple products.

**7. Product ↔️ Category**

**1. Relationship**: Belongs To

**2. Description:** A Product belongs to a single Category, but a Category can include many products.

**8. Cart ↔️ Order**

**1. Relationship:** Is Contained In

**2. Description:** A Cart is linked to a specific Order and represents the collection of products within that order.  
**9. Login ↔️ Category**

**1. Relationship Name**: Views

**2. Description**: Users represented by the Login entity can view multiple Categories. Similarly, each Category can be viewed by many users.

**10. Login ↔️ User Upload**

**1. Relationship Name:** Uploads Image

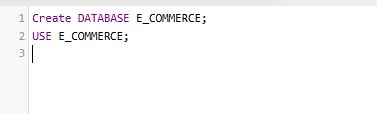
**2. Description**: A user logged into the system can upload images of products via the User Upload entity. Each upload is linked to a specific user.

**11. Login ↔️ Custom Product**

**1. Relationship Name:** Manages/Owns

**2. Description**: A user logged into the system can manage or own multiple Custom Products. Each product is associated with a specific user.

**CODES  
   
 PART I Query processing**  
 **Creating a table and inserting into the table for the classes :  
  
 1 . Create DATABASE E\_COMMERCE;**

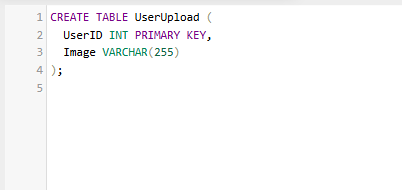
** USE E\_COMMERCE;**

**2. CREATE TABLE UserUpload (**

**UserID INT PRIMARY KEY,**

**Image VARCHAR(255)**

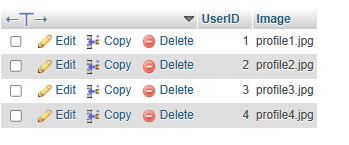
**);**



**INSERT INTO** **UserUpload** (UserID, Image) VALUES (1, 'profile1.jpg');

**INSERT INTO** **UserUpload** (UserID, Image) VALUES (2, 'profile2.jpg');

**INSERT INTO** **UserUpload** (UserID, Image) VALUES (3, 'profile3.jpg');

 **INSERT INTO** **UserUpload** (UserID, Image) VALUES (4, 'profile4.jpg');

**3. CREATE TABLE Login (**

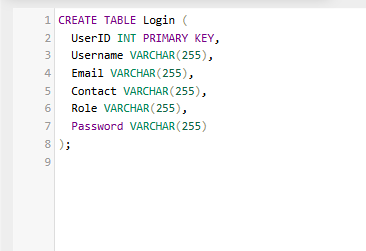
**UserID INT PRIMARY KEY,**

**Username VARCHAR(255),**

**Email VARCHAR(255),**

**Contact VARCHAR(255),**

**Role VARCHAR(255),**

 **Password VARCHAR(255));**  
  
   
  
  
  
  
  
   
**INSERT INTO Login** (UserID, Username, Email, Contact, Role, Password)

**VALUES** (1, 'Naol', 'naol@gmail.com', '123-456-7890', 'admin', 'password123');

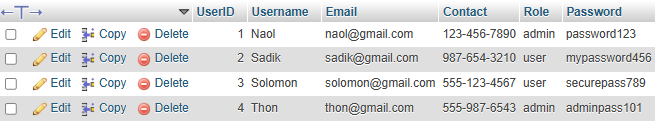
**INSERT INTO Login** (UserID, Username, Email, Contact, Role, Password)

**VALUES** (2, 'Sadik', 'sadik@gmail.com', '987-654-3210', 'user', 'mypassword456');

**INSERT INTO** **Login** (UserID, Username, Email, Contact, Role, Password)

**VALUES** (3, 'Solomon', 'solomon@gmail.com', '555-123-4567', 'user', 'securepass789');

**INSERT INTO Login** (UserID, Username, Email, Contact, Role, Password)

**VALUES** (4, 'Thon', 'thon@gmail.com', '555-987-6543', 'admin', 'adminpass101');

**4. CREATE TABLE CustomerProducts (**

**ProductID INT PRIMARY KEY,**

**Title VARCHAR(255),**

**Image VARCHAR(255),**

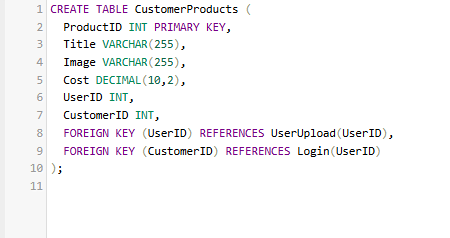
**Cost DECIMAL(10,2),**

**UserID INT,**

**CustomerID INT,**

**FOREIGN KEY (UserID) REFERENCES UserUpload(UserID),**

**FOREIGN KEY (CustomerID) REFERENCES Login(UserID)**

  **);**

**INSERT INTO** **CustomerProducts** (ProductID, Title, Image, Cost, UserID, CustomerID)

**VALUES** (1, 'Wireless Headphones', 'http://example.com/headphones.jpg', 129.99, 1, 1);

**INSERT INTO** **CustomerProducts** (ProductID, Title, Image, Cost, UserID, CustomerID)

**VALUES**  (2, 'Bluetooth Speaker', 'http://example.com/speaker.jpg', 49.99, 2, 2);

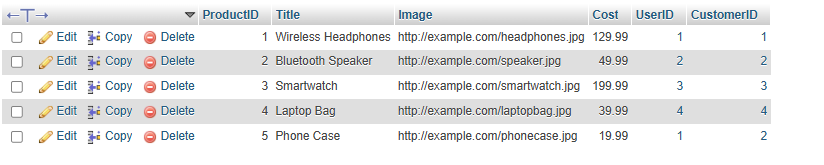
**INSERT INTO** **CustomerProducts** (ProductID, Title, Image, Cost, UserID, CustomerID)

**VALUES** (3, 'Smartwatch', 'http://example.com/smartwatch.jpg', 199.99, 3, 3);

**INSERT INTO** **CustomerProducts (**ProductID, Title, Image, Cost, UserID, CustomerID)

**VALUES** (4, 'Laptop Bag', 'http://example.com/laptopbag.jpg', 39.99, 4, 4);

**INSERT INTO** **CustomerProducts** (ProductID, Title, Image, Cost, UserID, CustomerID)

 VALUES (5, 'Phone Case', 'http://example.com/phonecase.jpg', 19.99, 1, 2);

**5. CREATE TABLE OrderTable (**

**OrderID INT PRIMARY KEY,**

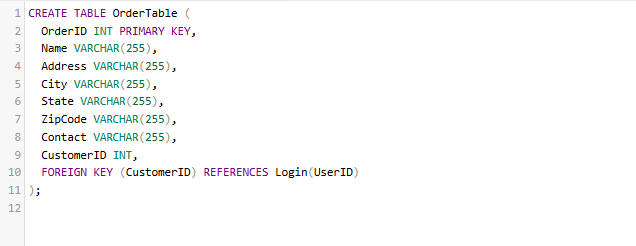
**Name VARCHAR(255),**

**Address VARCHAR(255),**

**City VARCHAR(255),**

**State VARCHAR(255),**

**ZipCode VARCHAR(255),**

** Contact VARCHAR(255),**

**CustomerID INT,**

**FOREIGN KEY (CustomerID) REFERENCES Login(UserID)**

**);**

**INSERT INTO** **OrderTable** (OrderID, Name, Address, City, State, ZipCode, Contact, CustomerID)

**VALUES** (1, 'Yohannes Tesfaye', 'Bahir Dar, Gojjam', 'Bahir Dar', 'Amhara', '6000', '0911-123-456', 1);

**INSERT INTO** **OrderTable** (OrderID, Name, Address, City, State, ZipCode, Contact, CustomerID)

**VALUES** (2, 'Marta Alemu', 'Mekelle, Tigray', 'Mekelle', 'Tigray', '7000', '0912-654-321', 2);

**INSERT INTO** **OrderTable** (OrderID, Name, Address, City, State, ZipCode, Contact, CustomerID)

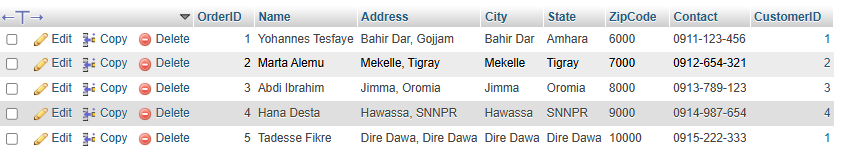
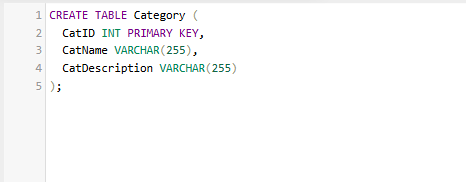
**VALUES** (3, 'Abdi Ibrahim', 'Jimma, Oromia', 'Jimma', 'Oromia', '8000', '0913-789-123', 3);

**INSERT INTO** **OrderTable** (OrderID, Name, Address, City, State, ZipCode, Contact, CustomerID)

**VALUES** (4, 'Hana Desta', 'Hawassa, SNNPR', 'Hawassa', 'SNNPR', '9000', '0914-987-654', 4);

**INSERT INTO** **OrderTable** (OrderID, Name, Address, City, State, ZipCode, Contact, CustomerID)

**VALUES** (5, 'Tadesse Fikre', 'Dire Dawa, Dire Dawa', 'Dire Dawa', 'Dire Dawa', '10000', '0915-222-333', 1);

  
  
  
   
  
  
  
  
  
 **6 . CREATE TABLE Category (**

**CatID INT PRIMARY KEY,**

**CatName VARCHAR(255),**

**CatDescription VARCHAR(255) );**  
  
 **INSERT INTO** **Category** (CatID, CatName, CatDescription)

**VALUES** (1, 'Electronics', 'Various electronic gadgets and accessories');

**INSERT INTO** **Category** (CatID, CatName, CatDescription)

**VALUES** (2, 'Clothing', 'Apparel for men, women, and children');

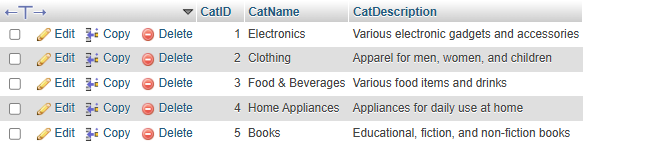
**INSERT INTO** **Category** (CatID, CatName, CatDescription)

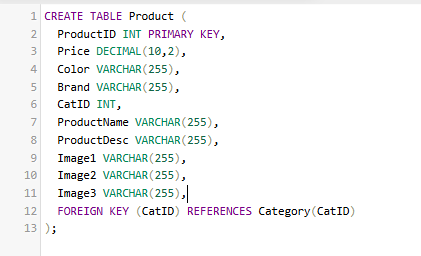
**VALUES** (3, 'Food & Beverages', 'Various food items and drinks');

**INSERT INTO Category** (CatID, CatName, CatDescription)

**VALUES** (4, 'Home Appliances', 'Appliances for daily use at home');

**INSERT INTO** **Category** (CatID, CatName, CatDescription)

**VALUES** (5, 'Books', 'Educational, fiction, and non-fiction books');**7 . CREATE TABLE Product (**

** ProductID INT PRIMARY KEY,**

**Price DECIMAL(10,2),**

**Color VARCHAR(255),**

**Brand VARCHAR(255),**

**CatID INT,**

**ProductName VARCHAR(255),**

**ProductDesc VARCHAR(255),**

**Image1 VARCHAR(255),**

**Image2 VARCHAR(255),**

**Image3 VARCHAR(255),**

**FOREIGN KEY (CatID) REFERENCES Category(CatID**)

**);**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**INSERT INTO** **Product** (ProductID, Price, Color, Brand, CatID, ProductName, ProductDesc, Image1, Image2, Image3)

**VALUES** (1, 129.99, 'Black', 'Sony', 1, 'Wireless Headphones', 'High-quality wireless headphones with noise cancellation', 'http://example.com/headphones1.jpg', 'http://example.com/headphones2.jpg', 'http://example.com/headphones3.jpg');

**INSERT INTO** **Product** (ProductID, Price, Color, Brand, CatID, ProductName, ProductDesc, Image1, Image2, Image3)

**VALUES** (2, 49.99, 'Blue', 'JBL', 1, 'Bluetooth Speaker', 'Portable Bluetooth speaker with high bass', 'http://example.com/speaker1.jpg', 'http://example.com/speaker2.jpg', 'http://example.com/speaker3.jpg');

**INSERT INTO** **Product** (ProductID, Price, Color, Brand, CatID, ProductName, ProductDesc, Image1, Image2, Image3)

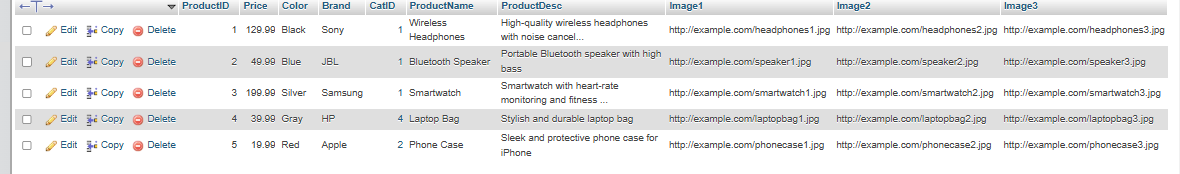
**VALUES** (3, 199.99, 'Silver', 'Samsung', 1, 'Smartwatch', 'Smartwatch with heart-rate monitoring and fitness tracking', 'http://example.com/smartwatch1.jpg', 'http://example.com/smartwatch2.jpg', 'http://example.com/smartwatch3.jpg');

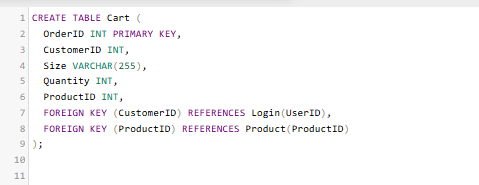
**INSERT INTO** **Product** (ProductID, Price, Color, Brand, CatID, ProductName, ProductDesc, Image1, Image2, Image3)

**VALUES** (4, 39.99, 'Gray', 'HP', 4, 'Laptop Bag', 'Stylish and durable laptop bag', 'http://example.com/laptopbag1.jpg', 'http://example.com/laptopbag2.jpg', 'http://example.com/laptopbag3.jpg');

**INSERT INTO** **Product** (ProductID, Price, Color, Brand, CatID, ProductName, ProductDesc, Image1, Image2, Image3)

**VALUES** (5, 19.99, 'Red', 'Apple', 2, 'Phone Case', 'Sleek and protective phone case for iPhone', 'http://example.com/phonecase1.jpg', 'http://example.com/phonecase2.jpg', 'http://example.com/phonecase3.jpg');

  
  
  
  
  
  
  
  
  
  
**8. CREATE TABLE Cart (**

** OrderID INT PRIMARY KEY,**

**CustomerID INT,**

**Size VARCHAR(255),**

**Quantity INT,**

**ProductID INT,**

**FOREIGN KEY (CustomerID) REFERENCES Login(UserID),**

**FOREIGN KEY (ProductID) REFERENCES Product(ProductID)**

**);**

  
  
**INSERT INTO** **Cart** (OrderID, CustomerID, Size, Quantity, ProductID)

**VALUES** (1, 1, 'Medium', 2, 1);

**INSERT INTO** **Cart** (OrderID, CustomerID, Size, Quantity, ProductID)

**VALUES** (2, 2, 'Large', 1, 2);

**INSERT INTO** Cart (OrderID, CustomerID, Size, Quantity, ProductID)

**VALUES** (3, 3, 'Small', 3, 3);

**INSERT INTO** **Cart** (OrderID, CustomerID, Size, Quantity, ProductID)

**VALUES** (4, 4, 'Large', 1, 4);

**INSERT INTO** **Cart** (OrderID, CustomerID, Size, Quantity, ProductID)

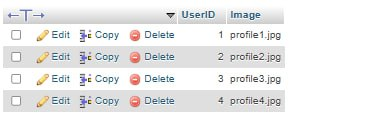
**VALUES** (5, 1, 'Small', 2, 5);  
  
Conclusion

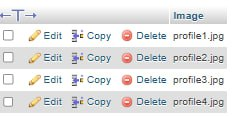
**Vertical Fragmentation**

Vertical fragmentation involves dividing a table by selecting subsets of columns (attributes). For instance:

**UserUpload Table:**

**Fragment 1**: { UserID, Image}



****

**Horizontal Fragmentation**

Horizontal fragmentation involves dividing a table into subsets of rows based on conditions. For example:

**UserUpload Table:**

**Fragment 1:** UserID <= 2

**Fragment 2**: UserID > 2



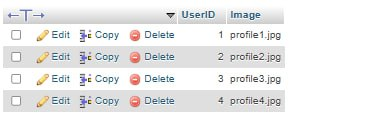
**Mixed Fragmentation**

Mixed fragmentation combines horizontal and vertical fragmentation. For instance:

**UserUpload Table:**

**Fragment 1:** UserID <= 2 with {UserID, Image}

**Fragment 2:** UserID > 2 with {UserID, Image}



**Appendix**

**ER Diagram**

The ER diagram provides a graphical representation of the relationships between key entities:

**UserUpload:** Manages user-uploaded images.

**Login:** Handles user authentication and basic details.

**CustomerProducts:** Connects products with their respective users and customers.

**OrderTable:** Stores details about orders placed by customers.

**Category:** Categorizes products into specific groups.

**Product:** Maintains detailed information about available products.

**Cart:** Represents items added to a customer's cart.

**Database Schema and Queries**

The database schema includes structured tables with primary and foreign key constraints. Queries for data insertion and retrieval are provided, ensuring proper system functionality. Examples include:

Creating and populating tables like UserUpload, Login, CustomerProducts, OrderTable, Category, Product, and Cart.

Efficient data relationships through foreign keys, such as linking CustomerProducts to UserUpload and Login.

**Conclusions**

The E-commerce database design presented in this project serves as a comprehensive solution for managing a variety of business processes. By integrating key components such as users, products, orders, and categories, it ensures streamlined data flow and operational efficiency. The application of relational database principles, coupled with careful entity and relationship modeling, underscores the importance of structured data management in E-commerce systems. The inclusion of vertical, horizontal, and mixed fragmentation enhances data accessibility and performance in distributed environments. This project lays a strong foundation for scalable and user-friendly E-commerce platforms, meeting the dynamic demands of modern online markets.