

Carmen municipal college

A.Y. 2024-2025

**Shopease**

CC105 – database management system

Midterm project

Lapuag kier

Cirunay lhara

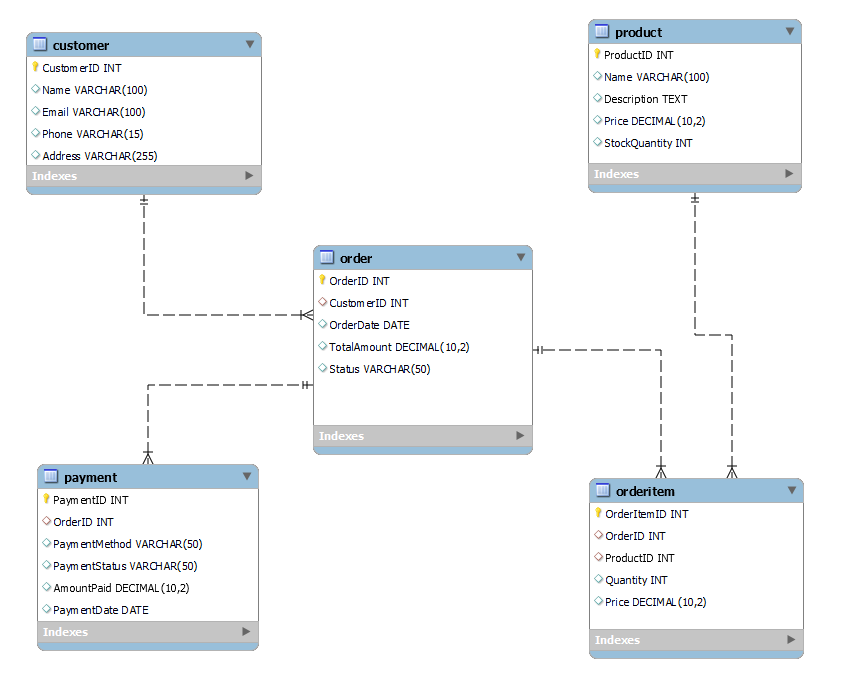
I. Introduction

**ShopEase** is a comprehensive platform designed to manage the operations of an e-commerce store. The primary goal of the system is to handle customer information, product catalog, orders, and payments efficiently, ensuring a smooth shopping experience for both the customers and the store administrators.

The target users of **ShopEase** are online shoppers, administrators, and business owners who need a reliable system to manage product listings, track customer orders, and process payments. The system allows users to browse products, place orders, and make payments securely. Administrators can update product details, monitor order statuses, and manage customer information.

The database supports the creation and management of multiple product categories, customer profiles, and order histories. Users can filter their searches by product type, price range, and availability, while administrators can easily update or delete data when necessary, ensuring data accuracy and efficient storage management.

1. Entity Relationship Diagram



**Table Name: Customer**

This table contains the personal information of the customer.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| CustomerID | INT | Unique identifier of the customer. |
| Name | VARCHAR(100) | Name of the customer. |
| Email | VARCHAR(100) | Email address of the customer. |
| Phone | VARCHAR(15) | Phone number of the customer. |
| Address | VARCHAR(255) | Address of the customer. |

**Table Name: Product**

This table contains the details of the products available in the store.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| ProductID | INT | Unique identifier of the product. |
| Name | VARCHAR(100) | Name of the product. |
| Description | TEXT | Description of the product. |
| Price | DECIMAL(10, 2) | Price of the product. |
| StockQuantity | INT | Quantity of the product available in stock. |

**Table Name: Order**

This table contains the details of the orders placed by the customers.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| OrderID | INT | Unique identifier of the order. |
| CustomerID | INT | The customer who placed the order. |
| OrderDate | DATE | Date when the order was placed. |
| TotalAmount | DECIMAL(10, 2) | Total cost of the order. |
| Status | VARCHAR(50) | Status of the order (e.g., "Shipped"). |
| StockQuantity | INT | Quantity of the product available in stock. |

**Table Name: OrderItem**

This table contains the details of the items included in an order.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| OrderItemID | INT | Unique identifier for each item in the order. |
| OrderID | INT | The order that the item belongs to. |
| ProductID | INT | The product being purchased. |
| Quantity | INT | Quantity of the product in the order. |
| Price | DECIMAL(10, 2) | Price of the product at the time of the order. |
| StockQuantity | INT | Quantity of the product available in stock. |

**Table Name: Payment**

This table contains the details of the payments made for orders.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| PaymentID | INT | Unique identifier for each payment. |
| OrderID | INT | The order that was paid for. |
| PaymentMethod | VARCHAR(50) | Method of payment (e.g., "Credit Card"). |
| PaymentStatus | VARCHAR(50) | Payment status (e.g., "Completed"). |
| AmountPaid | DECIMAL(10, 2) | The amount paid for the order. |

1. **Functionalities**

**Creating Tables:**

This code shows the creation of the tables in the ShopEase database.

CREATE TABLE Customer (

CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100) UNIQUE,

Phone VARCHAR(15),

Address VARCHAR(255)

) ENGINE=InnoDB;

CREATE TABLE Product (

ProductID INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(100),

Description TEXT,

Price DECIMAL(10, 2),

StockQuantity INT

) ENGINE=InnoDB;

CREATE TABLE `Order` (

OrderID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

TotalAmount DECIMAL(10, 2),

Status VARCHAR(50),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID) ON DELETE CASCADE

) ENGINE=InnoDB;

CREATE TABLE OrderItem (

OrderItemID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT,

Price DECIMAL(10, 2),

FOREIGN KEY (OrderID) REFERENCES `Order`(OrderID) ON DELETE CASCADE,

FOREIGN KEY (ProductID) REFERENCES Product(ProductID) ON DELETE CASCADE

) ENGINE=InnoDB;

CREATE TABLE Payment (

PaymentID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT,

PaymentMethod VARCHAR(50),

PaymentStatus VARCHAR(50),

AmountPaid DECIMAL(10, 2),

PaymentDate DATE,

FOREIGN KEY (OrderID) REFERENCES `Order`(OrderID) ON DELETE CASCADE

) ENGINE=InnoDB;

**Inserting Customer Information:**

This code represents the insertion of customer information into the Customer table.

INSERT INTO Customer (Name, Email, Phone, Address)

VALUES

('John Doe', 'john.doe@example.com', '1234567890', '123 Main St, Springfield'),

('Jane Smith', 'jane.smith@example.com', '0987654321', '456 Oak St, Springfield');

**Inserting Product Information:**

This code represents the insertion of product information into the Product table.

INSERT INTO Product (Name, Description, Price, StockQuantity)

VALUES

('Laptop', '15-inch laptop with 8GB RAM', 999.99, 50),

('Smartphone', '5-inch smartphone with 128GB storage', 499.99, 100);

**Inserting Orders:**

This code represents the insertion of orders placed by customers into the Order table.

INSERT INTO `Order` (CustomerID, OrderDate, TotalAmount, Status)

VALUES

(1, '2025-03-12', 1499.98, 'Shipped'),

(2, '2025-03-12', 499.99, 'Processing');

**Inserting Order Items:**

This code represents the insertion of order items for specific orders into the OrderItem table.

INSERT INTO OrderItem (OrderID, ProductID, Quantity, Price)

VALUES

(1, 1, 1, 999.99),

(1, 2, 1, 499.99),

(2, 2, 1, 499.99);

**Inserting Payments:**

This code represents the insertion of payment information for orders into the Payment table.

INSERT INTO Payment (OrderID, PaymentMethod, PaymentStatus, AmountPaid, PaymentDate)

VALUES

(1, 'Credit Card', 'Completed', 1499.98, '2025-03-12'),

(2, 'PayPal', 'Pending', 499.99, '2025-03-12');

**Displaying Information:**

This code represents different queries to display filtered information from the tables.

SELECT \* FROM Customer;

SELECT \* FROM Product WHERE Price > 500;

SELECT category, COUNT(ProductID)

FROM Product

GROUP BY category;

SELECT OrderID, TotalAmount, Status

FROM `Order`

WHERE TotalAmount > 1000

ORDER BY OrderID

SELECT \* FROM OrderItem WHERE OrderID = 1;

SELECT \* FROM Payment WHERE PaymentStatus = 'Completed' AND AmountPaid > 1000;