Database Design for Online Clothing Store Management Application

Entities and Their Properties

1. Customer

Stores customer details, including personal and contact information.

- ID: Unique identifier for each customer (Primary Key).
- Name: The customer’s full name (VARCHAR 100).
- Email: Customer's email (VARCHAR 100, unique).
- Password: Hashed password for authentication (VARCHAR 255).
- Address: Customer's shipping address (VARCHAR 255).
- Phone: Contact number (VARCHAR 15).

2. Product

Stores details about products available in the store.

- ID: Unique identifier for each product (Primary Key).
- Category: The product category (e.g., Men, Women, Accessories) (VARCHAR 100).
- Name: Product name (VARCHAR 100).
- Description: Detailed description of the product (TEXT).
- Price: Product price (DECIMAL 10,2) to support two decimal places.
- Stock: Number of units available in the inventory (INT).

3. Order

Contains customer orders and their current status.

- ID: Unique identifier for each order (Primary Key).
- CustomerID: Links the order to a customer (Foreign Key from Customer).
- TotalAmount: Total amount of the order (DECIMAL 10,2).
- StartDate: The date and time when the order was placed (DATETIME).
- Status: Current order status (ENUM – Pending, Shipping, Delivered).

4. OrderItem

Represents the items purchased in each order.

- ID: Unique identifier for each order item (Primary Key).
- OrderID: Links the item to an order (Foreign Key from Order).
- ProductID: Links the item to a product (Foreign Key from Product).
- Quantity: Number of units purchased (INT).
- Price: Price per unit of the product at the time of purchase (DECIMAL 10,2).

5. Invoice

Holds billing information for each order.

- ID: Unique identifier for each invoice (Primary Key).
- OrderID: Links the invoice to an order (Foreign Key from Order).
- InvoiceDate: The date when the invoice was generated (DATETIME).
- TotalAmount: The total amount billed on the invoice (DECIMAL 10,2).

6. Payment

Tracks payments made for invoices.

- ID: Unique identifier for each payment (Primary Key).
- Amount: Amount paid (DECIMAL 10,2).
- PaymentDate: Date of the payment (DATETIME).
- InvoiceID: Links the payment to an invoice (Foreign Key from Invoice).
- PaymentMethod: The method used for payment (ENUM – COD, VISA).

Adjustments Made

1. Naming Conventions

Used PascalCase for consistency in table names (e.g., OrderItem instead of Orderitem), making it more readable and conventional.

2. Data Types
    - Price and TotalAmount: Adjusted to DECIMAL(10, 2) to ensure accuracy when handling monetary values.
    - Email: Set as VARCHAR(100) and made unique to prevent duplicate customer records.
    - Password: Used VARCHAR(255) to store hashed passwords for security.
    - Date and Time Fields: Set as DATETIME for precise tracking of when orders, invoices, and payments are processed.

3. Constraints
    - Foreign Keys: Enforced relationships between Order, OrderItem, Invoice, and Payment to ensure referential integrity.
    - Cascading Deletes: Implemented cascading deletes for related records (e.g., deleting an order also deletes related order items and invoices) to maintain data consistency.
    - ENUM Fields: For Status in Order and PaymentMethod in Payment, ENUM types were used to restrict the values to valid choices, improving data accuracy.

MySQL Queries

Create Tables

```
-- Table: Customer
CREATE TABLE Customer (
ID INT AUTO_INCREMENT PRIMARY KEY,
Name VARCHAR(100) NOT NULL,
Email VARCHAR(100) NOT NULL UNIQUE,
Password VARCHAR(255) NOT NULL,
Address VARCHAR(255) NOT NULL,
Phone VARCHAR(15) NOT NULL
);

-- Table: Product
CREATE TABLE Product (
ID INT AUTO_INCREMENT PRIMARY KEY,
Category VARCHAR(100) NOT NULL,
Name VARCHAR(100) NOT NULL,
Description TEXT,
Price DECIMAL(10, 2) NOT NULL,
Stock INT NOT NULL
);

-- Table: Order
CREATE TABLE `Order` (
ID INT AUTO_INCREMENT PRIMARY KEY,
CustomerID INT NOT NULL,
TotalAmount DECIMAL(10, 2) NOT NULL,
StartDate DATETIME NOT NULL,
Status ENUM('Pending', 'Shipping', 'Delivered') DEFAULT 'Pending',
FOREIGN KEY (CustomerID) REFERENCES Customer(ID) ON DELETE
CASCADE
);
```

```
-- Table: OrderItem
CREATE TABLE OrderItem (
ID INT AUTO_INCREMENT PRIMARY KEY,
OrderID INT NOT NULL,
ProductID INT NOT NULL,
Quantity INT NOT NULL,
Price DECIMAL(10, 2) NOT NULL,
FOREIGN KEY (OrderID) REFERENCES `Order` (ID) ON DELETE CASCADE,
FOREIGN KEY (ProductID) REFERENCES Product(ID)
);

-- Table: Invoice
CREATE TABLE Invoice (
ID INT AUTO_INCREMENT PRIMARY KEY,
OrderID INT NOT NULL,
InvoiceDate DATETIME NOT NULL,
TotalAmount DECIMAL(10, 2) NOT NULL,
FOREIGN KEY (OrderID) REFERENCES `Order` (ID) ON DELETE CASCADE
);

-- Table: Payment
CREATE TABLE Payment (
ID INT AUTO_INCREMENT PRIMARY KEY,
Amount DECIMAL(10, 2) NOT NULL,
PaymentDate DATETIME NOT NULL,
InvoiceID INT NOT NULL,
PaymentMethod ENUM('COD', 'VISA') NOT NULL,
FOREIGN KEY (InvoiceID) REFERENCES Invoice(ID) ON DELETE CASCADE
);
```

Configure Database

Type these commands in terminal

1. mysql -u [username] -p -e "CREATE DATABASE clothing_store_db;"
2. mysql -u root -p clothing_store_db < clothing_store_db_backup.sql

~~mysqldump -u root -p clothing_store_db > clothing_store_db_backup.sql~~