-- Delete the database if it exists

DROP DATABASE IF EXISTS OnlineStore;

-- Create a new database

CREATE DATABASE OnlineStore;

-- Use the new database

USE OnlineStore;

-- Create Customers table

CREATE TABLE IF NOT EXISTS Customers (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100) UNIQUE,

PhoneNumber VARCHAR(15),

Address TEXT,

DateOfBirth DATE,

Status ENUM('Active', 'Inactive', 'Banned') DEFAULT 'Active',

CreatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

) ENGINE=InnoDB;

-- Create Users table

CREATE TABLE IF NOT EXISTS Users (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

Username VARCHAR(50) UNIQUE NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

PasswordHash VARCHAR(255) NOT NULL,

Role ENUM('Admin', 'Manager', 'Employee') NOT NULL DEFAULT 'Employee',

CreatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB;

-- Create Products table

CREATE TABLE IF NOT EXISTS Products (

ProductID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10,2),

Description TEXT,

StockLevel INT,

ProductImage VARCHAR(255),

SKU VARCHAR(100) UNIQUE,

Status ENUM('Available', 'Out of Stock', 'Discontinued') DEFAULT 'Available',

CreatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

) ENGINE=InnoDB;

-- Create Categories table

CREATE TABLE IF NOT EXISTS Categories (

CategoryID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(50) UNIQUE

) ENGINE=InnoDB;

-- Add foreign key to Products for Categories

ALTER TABLE Products ADD CategoryID INT;

ALTER TABLE Products ADD FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID) ON DELETE CASCADE;

-- Create Orders table

CREATE TABLE IF NOT EXISTS Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT,

OrderDate DATE,

TotalPrice DECIMAL(10,2),

Shoping\_Adress TEXT, -- العمود المفقود تم إضافته هنا

PreferredPaymentMethod ENUM('Credit Card', 'PayPal', 'Debit Card', 'Cash') DEFAULT 'Cash',

ShippingStatus VARCHAR(50),

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

) ENGINE=InnoDB;

-- Create Payments table

CREATE TABLE IF NOT EXISTS Payments (

PaymentID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

PaymentMethod VARCHAR(50),

Amount DECIMAL(10,2),

TransactionDate DATE,

PaymentStatus VARCHAR(50),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE

) ENGINE=InnoDB;

-- Create Inventory table

CREATE TABLE IF NOT EXISTS Inventory (

InventoryID INT PRIMARY KEY AUTO\_INCREMENT,

ProductID INT,

QuantityInStock INT,

RestockDate DATE,

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

) ENGINE=InnoDB;

-- Create Shipping table

CREATE TABLE IF NOT EXISTS Shipping (

ShippingID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

ShippingAddress TEXT,

TrackingNumber VARCHAR(100),

ExpectedDeliveryDate DATE,

ShippingStatus ENUM('Pending', 'In Transit', 'Delivered') DEFAULT 'Pending',

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE

) ENGINE=InnoDB;

-- Create OrderProducts table

CREATE TABLE IF NOT EXISTS OrderProducts (

OrderProductID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

ProductID INT,

Quantity INT,

UnitPrice DECIMAL(10,2),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

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

) ENGINE=InnoDB;

-- Create Discounts table

CREATE TABLE IF NOT EXISTS Discounts (

DiscountID INT PRIMARY KEY AUTO\_INCREMENT,

Code VARCHAR(50) UNIQUE,

Description VARCHAR(255),

DiscountPercentage DECIMAL(5,2),

StartDate DATE,

EndDate DATE

) ENGINE=InnoDB;

-- Create OrderDiscounts table

CREATE TABLE IF NOT EXISTS OrderDiscounts (

OrderDiscountID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

DiscountID INT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

FOREIGN KEY (DiscountID) REFERENCES Discounts(DiscountID) ON DELETE CASCADE

) ENGINE=InnoDB;

-- Create ActivityLogs table

CREATE TABLE IF NOT EXISTS ActivityLogs (

LogID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Activity VARCHAR(255),

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

) ENGINE=InnoDB;

-- Create Reviews table

CREATE TABLE IF NOT EXISTS Reviews (

ReviewID INT PRIMARY KEY AUTO\_INCREMENT,

ProductID INT,

CustomerID INT,

Rating INT CHECK (Rating BETWEEN 1 AND 5),

Comment TEXT,

CreatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE,

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

) ENGINE=InnoDB;

-- Insert initial data into Customers table

INSERT INTO Customers (FirstName, LastName, Email, PhoneNumber, Address, DateOfBirth) VALUES

('Mazen', 'Sahoyne', 'Mazen.sahoyne@example.com', '1234567890', '123 Elm Street', '1990-01-01'),

('Hassan', 'Karakra', 'Hassan.Karakra@example.com', '9876543210', '456 Oak Avenue', '1992-05-15'),

('Ayham', 'Alzer', 'Ayham.Alzer@example.com', '5555555555', '789 Pine Road', '1985-07-20');

-- Insert initial data into Users table

INSERT INTO Users (Username, Email, PasswordHash, Role) VALUES

('admin', 'admin@example.com', SHA2('admin123', 256), 'Admin'),

('manager1', 'manager1@example.com', SHA2('manager123', 256), 'Manager'),

('employee1', 'employee1@example.com', SHA2('employee123', 256), 'Employee');

-- Insert initial data into Products table

INSERT INTO Products (Name, Category, Price, Description, StockLevel, ProductImage, SKU) VALUES

('Laptop', 'Electronics', 1200.99, 'High performance laptop', 50, 'laptop.jpg', 'SKU001'),

('Headphones', 'Accessories', 99.99, 'Noise-cancelling headphones', 200, 'headphones.jpg', 'SKU002'),

('Smartphone', 'Electronics', 799.49, 'Latest model smartphone', 30, 'smartphone.jpg', 'SKU003');

-- Insert initial data into Categories table

INSERT INTO Categories (Name) VALUES ('Electronics'), ('Accessories'), ('Home Appliances');

-- Insert initial data into Orders table

INSERT INTO Orders (CustomerID, OrderDate, TotalPrice, Shoping\_Adress, ShippingStatus) VALUES

(1, '2024-01-10', 1299.99, '123 Elm Street', 'Pending'),

(2, '2024-01-12', 99.99, '456 Oak Avenue', 'Shipped'),

(3, '2024-01-15', 799.49, '789 Pine Road', 'Delivered');

-- Insert initial data into Payments table

INSERT INTO Payments (OrderID, PaymentMethod, Amount, TransactionDate, PaymentStatus) VALUES

(1, 'Credit Card', 1299.99, '2024-01-10', 'Completed'),

(2, 'PayPal', 99.99, '2024-01-12', 'Completed'),

(3, 'Debit Card', 799.49, '2024-01-15', 'Completed');

-- Insert initial data into Inventory table

INSERT INTO Inventory (ProductID, QuantityInStock, RestockDate) VALUES

(1, 50, '2024-02-01'),

(2, 200, '2024-01-20'),

(3, 30, '2024-03-05');

-- Insert initial data into Shipping table

INSERT INTO Shipping (OrderID, ShippingAddress, TrackingNumber, ExpectedDeliveryDate, ShippingStatus) VALUES

(1, '123 Elm Street', 'TRACK123', '2024-01-17', 'Pending'),

(2, '456 Oak Avenue', 'TRACK456', '2024-01-20', 'In Transit'),

(3, '789 Pine Road', 'TRACK789', '2024-01-25', 'Delivered');

-- Insert initial data into OrderProducts table

INSERT INTO OrderProducts (OrderID, ProductID, Quantity, UnitPrice) VALUES

(1, 1, 1, 1200.99),

(2, 2, 1, 99.99),

(3, 3, 1, 799.49);

-- Insert initial data into Discounts table

INSERT INTO Discounts (Code, Description, DiscountPercentage, StartDate, EndDate) VALUES

('DISC10', '10% off on all items', 10.00, '2024-01-01', '2024-12-31');

-- Insert initial data into OrderDiscounts table

INSERT INTO OrderDiscounts (OrderID, DiscountID) VALUES

(1, 1);

-- Insert initial data into ActivityLogs table

INSERT INTO ActivityLogs (UserID, Activity) VALUES

(1, 'Logged in as Admin'),

(2, 'Updated product inventory'),

(3, 'Processed an order');

-- Create Notifications table

CREATE TABLE IF NOT EXISTS Notifications (

NotificationID INT PRIMARY KEY AUTO\_INCREMENT,

Title VARCHAR(255) NOT NULL,

Message TEXT NOT NULL,

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP, -- الحقل المفقود

Status ENUM('Unread', 'Read') DEFAULT 'Unread',

CreatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

) ENGINE=InnoDB;

-- Insert initial data into Notifications table

INSERT INTO Notifications (Title, Message, Status) VALUES

('System Update', 'The system will undergo maintenance on 2024-01-15.', 'Unread'),

('New Feature', 'A new feature has been added to the product management module.', 'Read');

CREATE TABLE IF NOT EXISTS OrderProducts (

OrderProductID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

ProductID INT,

Quantity INT,

UnitPrice DECIMAL(10,2),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

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

) ENGINE=InnoDB;

CREATE INDEX idx\_order\_products ON OrderProducts(ProductID, OrderID);

CREATE INDEX idx\_orders ON Orders(CustomerID, OrderDate);

-- تعديل الجدول الحالي لإضافة العمود إن لم يكن موجوداً

ALTER TABLE Users ADD COLUMN Permissions TEXT DEFAULT NULL;

-- تعديل الجدول الحالي إن لم يكن يحتوي

CREATE TABLE IF NOT EXISTS ActivityLogs (

LogID INT AUTO\_INCREMENT PRIMARY KEY,

UserID INT NOT NULL,

Activity VARCHAR(255) NOT NULL,

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

SELECT

DATE(OrderDate) AS ReportDate,

SUM(TotalPrice) AS TotalSales,

COUNT(OrderID) AS TotalOrders

FROM Orders

GROUP BY ReportDate

ORDER BY ReportDate DESC;

SELECT

Products.Name AS ProductName,

SUM(OrderProducts.Quantity) AS TotalSold

FROM OrderProducts

JOIN Products ON OrderProducts.ProductID = Products.ProductID

GROUP BY Products.Name

ORDER BY TotalSold DESC

LIMIT 10;

SELECT

CONCAT(Customers.FirstName, ' ', Customers.LastName) AS CustomerName,

COUNT(Orders.OrderID) AS TotalOrders,

SUM(Orders.TotalPrice) AS TotalSpent

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

GROUP BY Customers.CustomerID

ORDER BY TotalSpent DESC

LIMIT 10;

ALTER TABLE Products ADD LowStockThreshold INT DEFAULT 10;

SELECT

ProductID, Name, StockLevel

FROM Products

WHERE StockLevel <= LowStockThreshold;

ALTER TABLE Orders

ADD COLUMN OrderStatus ENUM('Prepared', 'Shipped', 'Completed', 'Returned') DEFAULT 'Prepared',

ADD COLUMN TrackingNumber VARCHAR(50),

ADD COLUMN ReturnReason TEXT;

CREATE TABLE Returns (

ReturnID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT NOT NULL,

ProductID INT NOT NULL,

ReturnReason TEXT,

ReturnDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

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

);

CREATE TABLE Orders (

OrderID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT NOT NULL,

OrderDate DATE,

ShippingWeight DOUBLE,

ShippingDistance DOUBLE,

ShippingCost DOUBLE,

ShippingCompany VARCHAR(100),

OrderStatus VARCHAR(50) DEFAULT 'Pending',

TrackingNumber VARCHAR(50),

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

);

CREATE TABLE Orders (

OrderID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT NOT NULL,

OrderDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

ShippingWeight DOUBLE,

ShippingDistance DOUBLE,

ShippingCost DOUBLE,

ShippingCompany VARCHAR(100),

OrderStatus VARCHAR(50) DEFAULT 'Pending',

TrackingNumber VARCHAR(50),

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

);

CREATE TABLE ShippingCompanies (

CompanyID INT AUTO\_INCREMENT PRIMARY KEY,

CompanyName VARCHAR(100),

Region VARCHAR(50)

);