--Create Database

CREATE DATABASE OSMS;

--Use Database

USE OSMS;

-- Create Tables

CREATE TABLE Customers (

CustomerID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE

);

CREATE TABLE Categories (

CategoryID INT AUTO\_INCREMENT PRIMARY KEY,

CategoryName VARCHAR(100) NOT NULL

);

CREATE TABLE Products (

ProductID INT AUTO\_INCREMENT PRIMARY KEY,

ProductName VARCHAR(100) NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

CategoryID INT,

FOREIGN KEY (CategoryID) REFERENCES Categories (CategoryID) ON DELETE CASCADE

);

CREATE TABLE Orders (

OrderID INT AUTO\_INCREMENT PRIMARY KEY,

OrderDate DATE NOT NULL,

CustomerID INT,

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

);

CREATE TABLE OrderDetails (

OrderDetailID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT NOT NULL,

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

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

);

CREATE TABLE Suppliers (

SupplierID INT AUTO\_INCREMENT PRIMARY KEY,

SupplierName VARCHAR(100) NOT NULL

);

CREATE TABLE ProductSuppliers (

ProductID INT,

SupplierID INT,

PRIMARY KEY (ProductID, SupplierID),

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

FOREIGN KEY (SupplierID) REFERENCES Suppliers (SupplierID) ON DELETE CASCADE

);

CREATE TABLE Shippers (

ShipperID INT AUTO\_INCREMENT PRIMARY KEY,

ShipperName VARCHAR(100) NOT NULL

);

CREATE TABLE Shipments (

ShipmentID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT,

ShipperID INT,

ShipmentDate DATE,

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

FOREIGN KEY (ShipperID) REFERENCES Shippers (ShipperID) ON DELETE CASCADE

);

CREATE TABLE Reviews (

ReviewID INT AUTO\_INCREMENT PRIMARY KEY,

ProductID INT,

CustomerID INT,

Rating INT CHECK (Rating BETWEEN 1 AND 5),

Comment TEXT,

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

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

);

CREATE TABLE PaymentMethods (

PaymentMethodID INT AUTO\_INCREMENT PRIMARY KEY,

MethodName VARCHAR(50) NOT NULL

);

CREATE TABLE Payments (

PaymentID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT,

PaymentMethodID INT,

Amount DECIMAL(10, 2) NOT NULL,

PaymentDate DATE NOT NULL,

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

FOREIGN KEY (PaymentMethodID) REFERENCES PaymentMethods (PaymentMethodID) ON DELETE CASCADE

);

CREATE TABLE Discounts (

DiscountID INT AUTO\_INCREMENT PRIMARY KEY,

DiscountCode VARCHAR(20) NOT NULL,

DiscountAmount DECIMAL(5, 2) NOT NULL

);

CREATE TABLE OrderDiscounts (

OrderID INT,

DiscountID INT,

PRIMARY KEY (OrderID, DiscountID),

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

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

);

CREATE TABLE Carts (

CartID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT,

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

);

CREATE TABLE CartItems (

CartItemID INT AUTO\_INCREMENT PRIMARY KEY,

CartID INT,

ProductID INT,

Quantity INT NOT NULL,

FOREIGN KEY (CartID) REFERENCES Carts (CartID) ON DELETE CASCADE,

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

);

CREATE TABLE Addresses (

AddressID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT,

AddressLine VARCHAR(255) NOT NULL,

City VARCHAR(100) NOT NULL,

State VARCHAR(100) NOT NULL,

ZipCode VARCHAR(10) NOT NULL,

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

);

CREATE TABLE Wishlists (

WishlistID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT,

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

);

CREATE TABLE WishlistItems (

WishlistItemID INT AUTO\_INCREMENT PRIMARY KEY,

WishlistID INT,

ProductID INT,

FOREIGN KEY (WishlistID) REFERENCES Wishlists (WishlistID) ON DELETE CASCADE,

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

);

CREATE TABLE Returns (

ReturnID INT AUTO\_INCREMENT PRIMARY KEY,

OrderID INT,

ProductID INT,

ReturnDate DATE NOT NULL,

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

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

);

-- Insert Sample Data

INSERT INTO Customers (FirstName, LastName, Email)

VALUES ('Jerin', 'Achol', 'achol@gmail.com');

INSERT INTO Categories (CategoryName)

VALUES ('Electronics');

INSERT INTO Products (ProductName, Price, CategoryID)

VALUES ('Laptop', 999.99, 1);

INSERT INTO Orders (OrderDate, CustomerID)

VALUES ('2023-10-01', 1);

INSERT INTO OrderDetails (OrderID, ProductID, Quantity)

VALUES (1, 1, 1);

INSERT INTO Suppliers (SupplierName)

VALUES ('Tech Supplier');

INSERT INTO ProductSuppliers (ProductID, SupplierID)

VALUES (1, 1);

INSERT INTO Shippers (ShipperName)

VALUES ('Fast Shipping');

INSERT INTO Shipments (OrderID, ShipperID, ShipmentDate)

VALUES (1, 1, '2023-10-02');

INSERT INTO Reviews (ProductID, CustomerID, Rating, Comment)

VALUES (1, 1, 5, 'Great product!');

INSERT INTO PaymentMethods (MethodName)

VALUES ('Credit Card');

INSERT INTO Payments (OrderID, PaymentMethodID, Amount, PaymentDate)

VALUES (1, 1, 999.99, '2023-10-01');

INSERT INTO Discounts (DiscountCode, DiscountAmount)

VALUES ('SAVE10', 10.00);

INSERT INTO OrderDiscounts (OrderID, DiscountID)

VALUES (1, 1);

INSERT INTO Carts (CustomerID)

VALUES (1);

INSERT INTO CartItems (CartID, ProductID, Quantity)

VALUES (1, 1, 1);

INSERT INTO Addresses (CustomerID, AddressLine, City, State, ZipCode)

VALUES (1, '123 Main St', 'Anytown', 'Anystate', '12345');

INSERT INTO Wishlists (CustomerID)

VALUES (1);

INSERT INTO WishlistItems (WishlistID, ProductID)

VALUES (1, 1);

INSERT INTO Returns (OrderID, ProductID, ReturnDate)

VALUES (1, 1, '2023-10-05');

-- SQL Operators and Nested SQL

SELECT \* FROM Products

WHERE Price BETWEEN 500 AND 1500 AND ProductID IN (

SELECT ProductID FROM OrderDetails

WHERE Quantity > 0

);

SELECT \* FROM Customers

WHERE NOT EXISTS (

SELECT \* FROM Orders

WHERE Orders.CustomerID = Customers.CustomerID AND OrderDate > '2023-01-01'

);

-- Having Clause and Aggregate Functions

SELECT CategoryID, COUNT(ProductID) AS ProductCount

FROM Products

GROUP BY CategoryID

HAVING COUNT(ProductID) > 0;

SELECT SupplierID, AVG(Price) AS AveragePrice

FROM ProductSuppliers

JOIN Products ON ProductSuppliers.ProductID = Products.ProductID

GROUP BY SupplierID

HAVING AVG(Price) > 100;

-- Join and Views

CREATE VIEW CustomerOrders AS

SELECT Customers.FirstName, Customers.LastName, Orders.OrderID, Orders.OrderDate

FROM Customers

JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

CREATE VIEW ProductReviews AS

SELECT Products.ProductName, Reviews.Rating, Reviews.Comment

FROM Products

JOIN Reviews ON Products.ProductID = Reviews.ProductID;

-- Procedures

DELIMITER / /

CREATE PROCEDURE GetCustomerOrders (IN custID INT)

BEGIN

SELECT \* FROM Orders

WHERE CustomerID = custID;

END / /

DELIMITER;

DELIMITER / /

CREATE PROCEDURE GetProductDetails (IN prodID INT)

BEGIN

SELECT \* FROM Products

WHERE ProductID = prodID;

END / /

DELIMITER;

-- Triggers

DELIMITER / /

CREATE TRIGGER BeforeOrderInsert

BEFORE INSERT ON Orders

FOR EACH ROW

BEGIN

DECLARE

msg VARCHAR(255);

IF NEW.OrderDate < CURDATE () THEN

SET msg = 'Order date cannot be in the past.';

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = msg;

END IF;

END / /

DELIMITER;

DELIMITER / /

CREATE TRIGGER AfterOrderInsert

AFTER INSERT ON Orders

FOR EACH ROW

BEGIN

INSERT INTO Shipments (OrderID, ShipperID, ShipmentDate)

VALUES (NEW.OrderID, 1, CURDATE ());

END / /

DELIMITER;

-- Transactions

START TRANSACTION;

SAVEPOINT BeforeDiscount;

UPDATE Orders

SET OrderDate = '2023-10-03'

WHERE OrderID = 1;

ROLLBACK TO BeforeDiscount;

COMMIT;

START TRANSACTION;

SAVEPOINT BeforePayment;

UPDATE Payments

SET Amount = 989.99

WHERE PaymentID = 1;

ROLLBACK TO BeforePayment;

COMMIT;