1. Create Table

-- Table for Authors

CREATE TABLE Authors (

AuthorID INT PRIMARY KEY AUTO\_INCREMENT,

AuthorName VARCHAR(100) NOT NULL

);

-- Table for Books

CREATE TABLE Books (

BookID INT PRIMARY KEY AUTO\_INCREMENT,

Title VARCHAR(200) NOT NULL,

AuthorID INT NOT NULL,

CategoryID INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID),

FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)

);

-- Table for Categories

CREATE TABLE Categories (

CategoryID INT PRIMARY KEY AUTO\_INCREMENT,

CategoryName VARCHAR(100) NOT NULL

);

-- Table for Customers

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerName VARCHAR(100) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL

);

-- Table for Orders

CREATE TABLE Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT NOT NULL,

OrderDate DATE NOT NULL,

ShippingMethodID INT NOT NULL,

TrackingNumber VARCHAR(50),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (ShippingMethodID) REFERENCES ShippingMethods(ShippingMethodID)

);

-- Table for Order Details

CREATE TABLE OrderDetails (

OrderDetailID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT NOT NULL,

BookID INT NOT NULL,

Quantity INT NOT NULL,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (BookID) REFERENCES Books(BookID)

);

-- Table for Reviews

CREATE TABLE Reviews (

ReviewID INT PRIMARY KEY AUTO\_INCREMENT,

BookID INT NOT NULL,

CustomerID INT NOT NULL,

ReviewText TEXT,

Rating INT CHECK (Rating BETWEEN 1 AND 5),

FOREIGN KEY (BookID) REFERENCES Books(BookID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

-- Table for Shipping Methods

CREATE TABLE ShippingMethods (

ShippingMethodID INT PRIMARY KEY AUTO\_INCREMENT,

MethodName VARCHAR(100) NOT NULL,

Cost DECIMAL(10, 2) NOT NULL

);

1. Insert Sample Data

-- Insert Authors

INSERT INTO Authors (AuthorName) VALUES

('J.K. Rowling'),

('George R.R. Martin'),

('J.R.R. Tolkien');

-- Insert Categories

INSERT INTO Categories (CategoryName) VALUES

('Fantasy'),

('Science Fiction'),

('Mystery');

-- Insert Books

INSERT INTO Books (Title, AuthorID, CategoryID, Price) VALUES

('Harry Potter and the Sorcerer\'s Stone', 1, 1, 19.99),

('A Game of Thrones', 2, 1, 25.99),

('The Hobbit', 3, 1, 15.99);

-- Insert Customers

INSERT INTO Customers (CustomerName, Email) VALUES

('Alice Johnson', 'alice@example.com'),

('Bob Smith', 'bob@example.com');

-- Insert Shipping Methods

INSERT INTO ShippingMethods (MethodName, Cost) VALUES

('Standard Shipping', 5.99),

('Express Shipping', 15.99);

-- Insert Orders

INSERT INTO Orders (CustomerID, OrderDate, ShippingMethodID, TrackingNumber) VALUES

(1, '2024-11-20', 1, 'TRACK12345'),

(2, '2024-11-21', 2, 'TRACK67890');

-- Insert Order Details

INSERT INTO OrderDetails (OrderID, BookID, Quantity) VALUES

(1, 1, 2),

(2, 3, 1);

-- Insert Reviews

INSERT INTO Reviews (BookID, CustomerID, ReviewText, Rating) VALUES

(1, 1, 'Amazing book!', 5),

(3, 2, 'Loved it!', 4);

1. SQL Queries

Retrieve all books by specific author

-- Replace 'AuthorName' with the actual name of the author

SELECT Books.Title

FROM Books

JOIN Authors ON Books.AuthorID = Authors.AuthorID

WHERE Authors.AuthorName = 'J.K. Rowling';

Get order history for Customer

-- Replace 'CustomerName' with the actual name of the customer

SELECT Orders.OrderID, Orders.OrderDate, Books.Title, OrderDetails.Quantity

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Books ON OrderDetails.BookID = Books.BookID

WHERE Customers.CustomerName = 'Alice Johnson';

Find Reviews for a book

-- Replace 'BookTitle' with the actual title of the book

SELECT Reviews.ReviewText, Reviews.Rating, Customers.CustomerName

FROM Reviews

JOIN Books ON Reviews.BookID = Books.BookID

JOIN Customers ON Reviews.CustomerID = Customers.CustomerID

WHERE Books.Title = 'Harry Potter and the Sorcerer\'s Stone';

Retrieve list of books with category names

SELECT Books.Title, Categories.CategoryName

FROM Books

JOIN Categories ON Books.CategoryID = Categories.CategoryID;

Get all Orders with their shipping method and tracking number

SELECT Orders.OrderID, Customers.CustomerName, ShippingMethods.MethodName, Orders.TrackingNumber

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

JOIN ShippingMethods ON Orders.ShippingMethodID = ShippingMethods.ShippingMethodID;

Update Shipping cost

-- Replace 'ShippingMethodID' and 'NewCost' with the actual values

UPDATE ShippingMethods

SET Cost = 7.99

WHERE ShippingMethodID = 1;

Delete records of authors with no books

DELETE FROM Authors

WHERE AuthorID NOT IN (SELECT DISTINCT AuthorID FROM Books);