Step 1: Create the Database

CREATE DATABASE OnlineTicketBooking;

--

-- Step 2: Switch to the created database

USE OnlineTicketBooking;

-- Step 3: Create Tables

-- Create Users Table

CREATE TABLE Users (

user\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

phone\_number VARCHAR(15)

);

-- Create Events Table

CREATE TABLE Events (

event\_id INT PRIMARY KEY AUTO\_INCREMENT,

event\_name VARCHAR(100),

event\_date DATE,

event\_time TIME,

location VARCHAR(100),

total\_seats INT

);

-- Create Bookings Table

CREATE TABLE Bookings (

booking\_id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT,

event\_id INT,

number\_of\_tickets INT,

booking\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES Users(user\_id),

FOREIGN KEY (event\_id) REFERENCES Events(event\_id)

);

-- Step 4: Insert Sample Data

-- Insert Users Data

INSERT INTO Users (name, email, phone\_number) VALUES

('John Doe', 'johndoe@example.com', '1234567890'),

('Jane Smith', 'janesmith@example.com', '9876543210');

-- Insert Events Data

INSERT INTO Events (event\_name, event\_date, event\_time, location, total\_seats) VALUES

('Rock Concert', '2024-12-15', '18:00:00', 'Stadium A', 100),

('Movie Screening', '2024-12-16', '20:00:00', 'Cinema Hall B', 150);

-- Insert Bookings Data

INSERT INTO Bookings (user\_id, event\_id, number\_of\_tickets) VALUES

(1, 1, 2), -- John Doe books 2 tickets for Rock Concert

(2, 2, 3); -- Jane Smith books 3 tickets for Movie Screening

-- Step 5: Queries for Operations

-- 1. Check Available Seats for Event 1 (Rock Concert)

SELECT total\_seats - COALESCE(SUM(number\_of\_tickets), 0) AS available\_seats

FROM Bookings

WHERE event\_id = 1;

-- 2. Book New Ticket for User 1 (2 tickets for Rock Concert)

-- Check if seats are available first (assuming there's enough availability)

-- Book 2 tickets for User 1 for the Rock Concert (Event ID = 1)

INSERT INTO Bookings (user\_id, event\_id, number\_of\_tickets)

VALUES (1, 1, 2);

-- 3. View All Bookings for User 1

SELECT b.booking\_id, e.event\_name, e.event\_date, e.event\_time, b.number\_of\_tickets, b.booking\_date

FROM Bookings b

JOIN Events e ON b.event\_id = e.event\_id

WHERE b.user\_id = 1;

-- 4. Cancel Booking with booking\_id = 1

DELETE FROM Bookings WHERE booking\_id = 1;

-- 5. View All Bookings for User 1 after Canceling the Booking

SELECT b.booking\_id, e.event\_name, e.event\_date, e.event\_time, b.number\_of\_tickets, b.booking\_date

FROM Bookings b

JOIN Events e ON b.event\_id = e.event\_id

WHERE b.user\_id = 1;

-- 6. Check Available Seats for Event 1 (Rock Concert) After Booking

SELECT total\_seats - COALESCE(SUM(number\_of\_tickets), 0) AS available\_seats

FROM Bookings

WHERE event\_id = 1;