Airline Reservation System

Abstract:

The Airline Reservation System is a relational database project developed using MySQL that streamlines the process of managing flight schedules, customer data, seat availability, and booking transactions. The system is designed with normalization principles to ensure data integrity and efficient querying. It supports core functionalities such as seat booking, cancellation, and real-time seat availability updates through SQL triggers. Additionally, it features a dynamic view that summarizes booking information for reporting purposes. This project demonstrates how SQL-based automation and structured database design can build a scalable foundation for real-world reservation systems.

1. Project Title:
Airline Reservation System using SQL (MySQL Workbench)

2. Objective:
To design and implement a relational database system that efficiently manages flights, customers, seat availability, and booking operations. The system includes automation through triggers and offers reporting functionality via SQL views.

3. Tools Used:

- DBMS: MySQL
- Development Tool: MySQL Workbench
- Language: SQL

4. Database Design:
Entities and Tables:

- **Flights**: FlightID, Airline, Source, Destination, DepartureTime, ArrivalTime
- **Customers**: CustomerID, Name, Email, Phone
- **Seats**: SeatID, FlightID, SeatNumber, Class (Economy/Business/First), IsBooked
- **Bookings**: BookingID, FlightID, CustomerID, SeatID, BookingDate, Status (Booked/Cancelled)

Normalization:

- 1NF: Each field contains atomic values with no repeating groups.
- 2NF: Non-key attributes depend on the whole primary key.
- 3NF: All attributes are only dependent on the primary key.

5. Constraints:

- Primary and foreign key constraints for referential integrity
- Unique constraint on Customer Email

- Default values for BookingDate (current timestamp), IsBooked (false), and Status (Booked)

6. Sample Data Inserted:

- 2 Flights: Delhi-Mumbai and Hyderabad-Chennai
- 2 Customers: Ravi Kumar, Anjali Singh
- 6 Seats distributed between two flights
- 1 Booking, automatically updating seat availability

7. Triggers:

- **After Booking Insert Trigger**: Automatically marks the seat as booked after a new booking.
- **After Booking Cancel Trigger**: Updates the seat to unbooked when a booking is cancelled.

8. Booking Summary View:
This view joins data from multiple tables to give a user-friendly summary of bookings:

```
CREATE VIEW BookingSummary AS

SELECT b.BookingID, c.Name AS CustomerName, f.Airline, f.Source, f.Destination,

    s.SeatNumber, s.Class, b.BookingDate, b.Status

FROM Bookings b

JOIN Customers c ON b.CustomerID = c.CustomerID

JOIN Flights f ON b.FlightID = f.FlightID

JOIN Seats s ON b.SeatID = s.SeatID;
```

9. Sample Queries:

- **Search Flights:**

```
SELECT * FROM Flights WHERE Source = 'Delhi' AND Destination = 'Mumbai';
```

- **Available Seats:**

```
SELECT SeatNumber FROM Seats WHERE FlightID = 1 AND IsBooked = FALSE;
```

- **Booking Summary:**

```
SELECT * FROM BookingSummary;
```

10. Conclusion:
The SQL-based Airline Reservation System provides a solid backend solution for managing flight schedules, seat bookings, and customer information. By using normalized data structures, enforcing constraints, and automating updates through triggers, the system ensures accuracy and efficiency. This project serves as a scalable base for further development into a complete reservation application.