Airline Reservation System – SQL Project Report

📌 Project Overview

The **Airport Reservation System** is a relational database project developed using SQL to manage airline operations, including flights, bookings, passengers, and airports. This system provides a scalable solution for handling data related to air travel, and facilitates efficient data queries, booking management, and flight tracking.

🧩 Database Structure

🔷 Database Name: airline_reservation

🔶 Tables Included:

1. **airport** – Stores details about airports (name, city, country)
2. **airplane** – Contains aircraft models and seat capacity
3. **flight** – Records each flight with times, source & destination
4. **passenger** – Holds passenger contact and identity details
5. **booking** – Maintains ticket bookings, seat numbers, and status

🔄 Entity Relationships

- Each **flight** references:
    - One **departure airport**
    - One **arrival airport**
    - One **airplane**
- Each **booking** references:
    - One **passenger**
    - One **flight**

These relationships are enforced using **foreign key constraints**, ensuring referential integrity across the system.

📥 Sample Data Inserted

Airports:

- John F. Kennedy International (USA)
- Heathrow (UK)
- Haneda (Japan)

Airplanes:

- Boeing 737 (180 seats)
- Airbus A320 (150 seats)

Flights:

- AA101: From JFK to Heathrow
- BA202: From Heathrow to Haneda

Passengers:

- John Doe
- Jane Smith

Bookings:

- John: Flight AA101, Seat 12A
- Jane: Flight BA202, Seat 14B

🧠 Key SQL Functionalities Implemented

1. Flight Listings with Airport Names

Shows all flights with departure/arrival airport names.

2. Booking Overview

Displays all bookings along with passenger name and flight number.

3. Bookings Count per Flight

Gives number of passengers booked for each flight.

4. Passenger List on a Flight

Fetches passengers assigned to a specific flight ID.

5. Available Seats Calculation

Determines unbooked seats by subtracting bookings from total seats.

6. Flights Between Dates

Lists flights scheduled between given date ranges.

7. Departures per Airport

Shows how many flights depart from each airport.

8. Booking Status Summary

Gives count of bookings grouped by status (e.g., confirmed).

9. Flights Without Bookings

Helps identify underutilized or unbooked flights.

10. Frequent Flyers

Highlights passengers with more than one booking.

11. Full Booking Details with Airport Info

Comprehensive summary including airport and seat info.

12. Flight Capacity Utilization

Calculates percentage of seats booked per flight.

📊 **Example Query – Booking Summary**

```
SELECT
   p.first_name,
   p.last_name,
   f.flight_number,
   b.seat_number,
   b.status
FROM booking b
JOIN passenger p ON b.passenger_id = p.passenger_id
JOIN flight f ON b.flight_id = f.flight_id;
```

✅ **Key Features**

- **Normalized design** for optimized data storage.
- **Realistic constraints and relationships** (e.g., foreign keys).
- **Comprehensive sample queries** for analytics and insights.
- **Date and seat-based logic** to model real-time reservations.

🚀 Potential Enhancements

- Add **payment and billing** module.
- Include **user roles** (admin, staff, customer).
- Integrate with **real-time flight APIs** for live scheduling.
- Build a **front-end interface** using Flask, React, or Java.

📂 Use Case Applications

- Academic SQL projects
- Database system demos
- Backend for flight booking systems
- Learning normalization and query optimization
