

Project Report on Airline Booking System

Introduction

What is the system?

The **Airline Booking System** is a database management system designed to manage airline ticket reservations, passenger details, flight schedules, and seat availability. It helps airlines streamline the booking process by storing and retrieving essential data efficiently.

Who will be the users?

The system will have different types of users:

- **Passengers:** Users who book and manage their flight tickets.
- **Airline Staff:** Employees who update flight schedules, seat availability, and passenger details.
- **Administrators:** Manage the entire system, ensuring data security and proper functionality.

Future Scope

- Integration with **AI** for price prediction based on demand.
- Implementation of **real-time seat availability updates**.
- Adding a **loyalty program** for frequent travelers.

- Enhanced security features like **biometric authentication** for passenger verification.

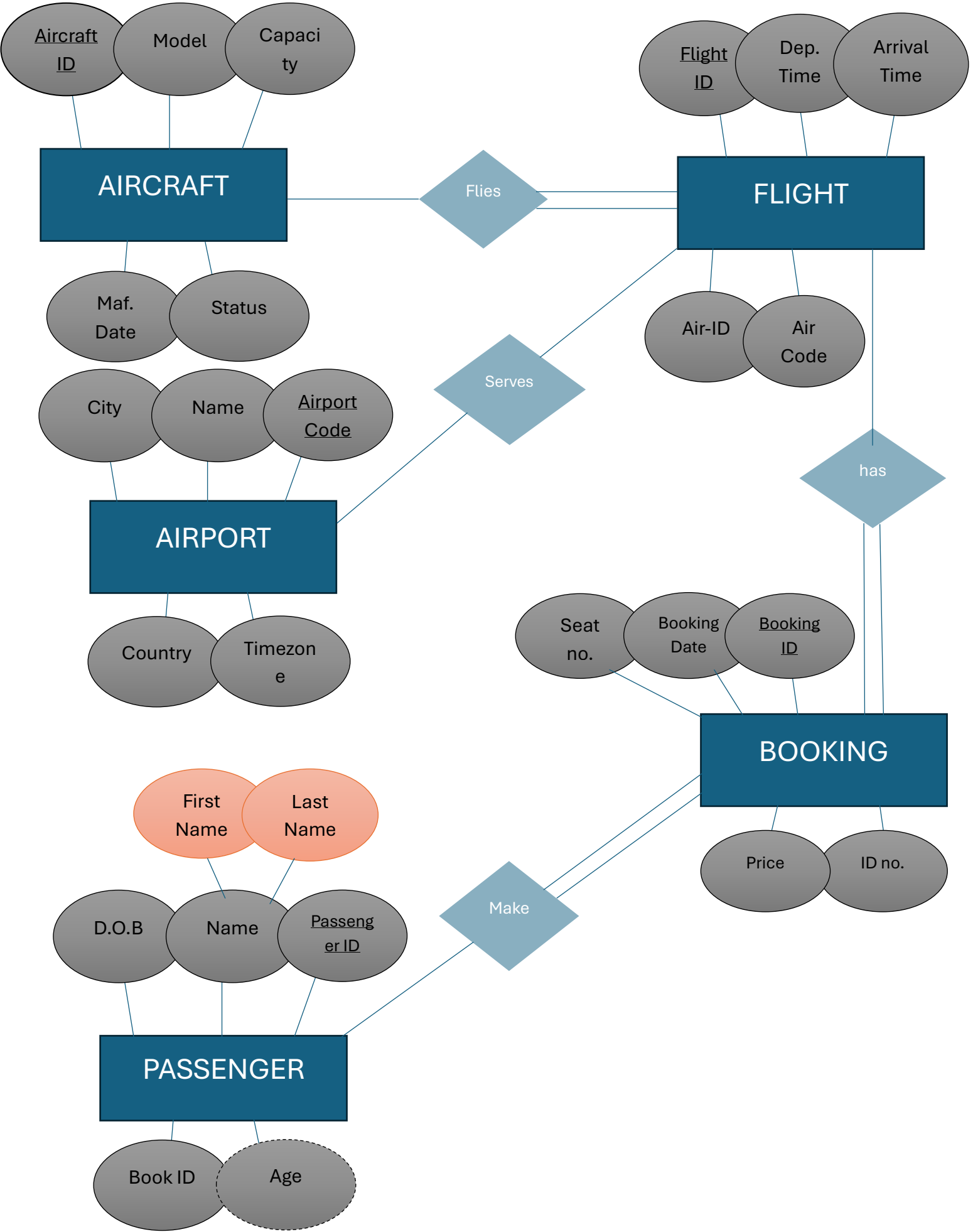
Objective

The **main objective** of the Airline Booking System is to simplify the process of flight reservations, passenger management, and ticket booking. The system ensures smooth and efficient handling of flight schedules and seat availability.

Key Objectives:

1. **Booking Management** – Allow users to book, modify, and cancel tickets easily.
2. **Ticket Handling** – Generate and store ticket details, including passenger information.
3. **Buyer & Passenger Database** – Maintain a structured database for passengers, storing details like name, age, contact information, and travel history.
4. **Flight Scheduling** – Manage flight timings, routes, and seat availability.
5. **Payment Processing** – Integrate secure payment gateways for hassle-free transactions.
6. **Data Security & Integrity** – Ensure proper access control and data protection to prevent unauthorized changes.

ER DIAGRAM



RELATIONAL SCHEMA

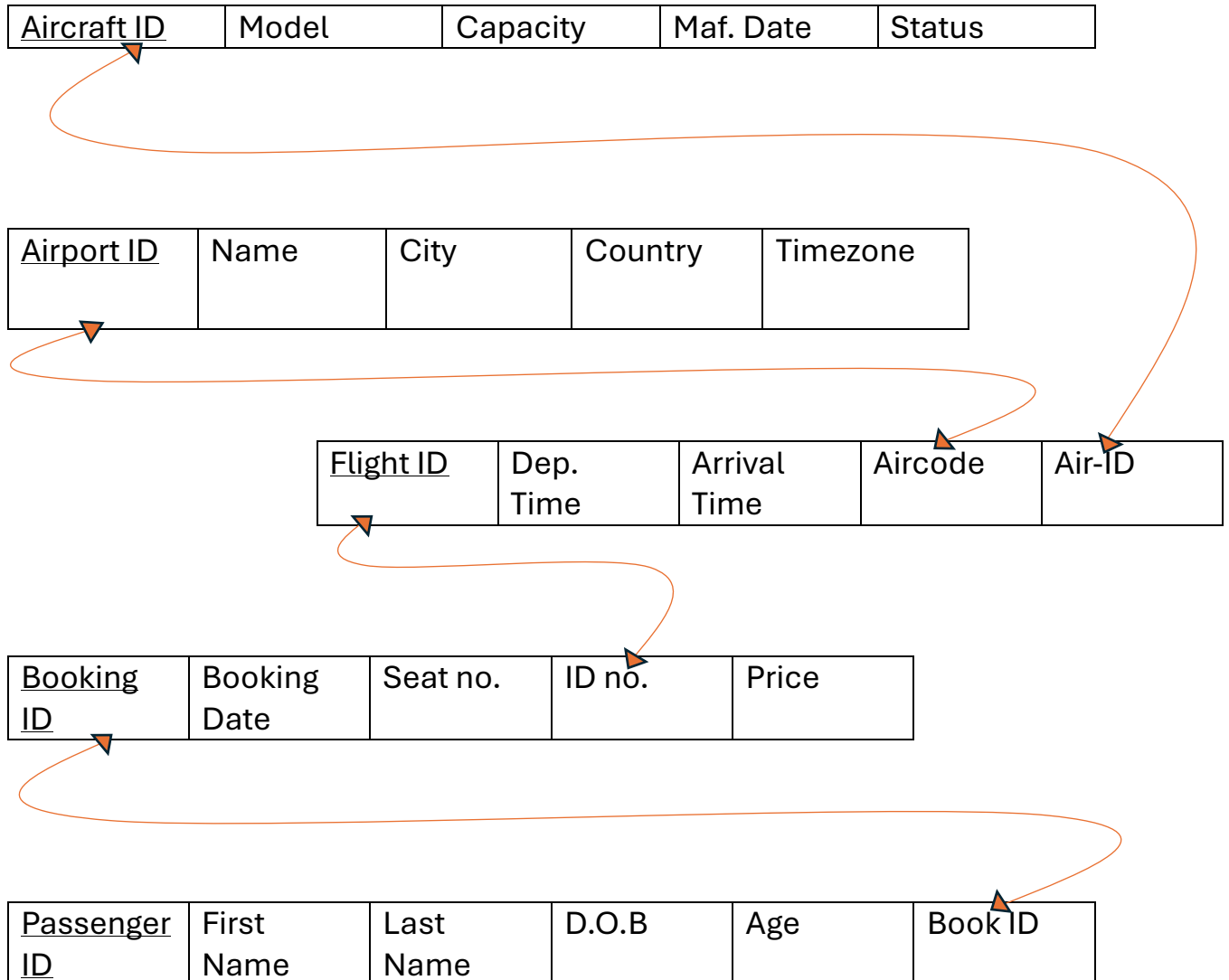
<u>Aircraft ID</u>	Model	Capacity	Maf. Date	Status
--------------------	-------	----------	-----------	--------

<u>Airport ID</u>	Name	City	Country	Timezone
-------------------	------	------	---------	----------

<u>Flight ID</u>	Dep. Time	Arrival Time	Aircode	Air-ID
------------------	-----------	--------------	---------	--------

<u>Booking ID</u>	Booking Date	Seat no.	ID no.	Price
-------------------	--------------	----------	--------	-------

<u>Passenger ID</u>	First Name	Last Name	D.O.B	Age	Book ID
---------------------	------------	-----------	-------	-----	---------



Databases or Tables

Passenger Table

Passenger_ID	First_Name	Last_Name	D.O.B	Age	Book_ID
1	John	Doe	1990-05-14	34	101
2	Alice	Smith	1985-11-20	39	102
3	Micheal	Brown	1992-07-08	32	103

Flight Table

Flight_ID	Dep_Time	Arrival_Time	Aircode	Air_ID
201	10:30:00	14:00:00	AA101	301
202	15:45:00	18:20:00	BA202	302
203	08:00:00	11:30:00	LH303	303

Aircraft Table

Aircraft_ID	Model	Capacity	Maf_Date	Status
301	Boeing 737	180	2015-06-15	Active
302	Airbus A320	200	2018-03-22	Active
303	Boeing 787	250	2020-09-10	Under Maintenance

Booking Table

Booking_ID	Booking_Date	Seat_no	ID_no	Price
101	2025-04-01	12A	1	350.00
102	2025-04-02	7C	2	450.00
103	2025-04-03	18B	3	600.00

Airport Table

Airport_ID	Name	City	Country	Timezone
401	Heathrow Airport	London	UK	GMT+0
402	JFK International	New York	USA	GMT-5
403	Changi Airport	Singapore	Singapore	GMT+8

SQL Queries

1. DDL (Data Definition Language) Queries

These queries create the database schema.

Create Table

```
CREATE TABLE Aircraft (  
    Aircraft_ID INT PRIMARY KEY,  
    Model VARCHAR(50),  
    Capacity INT,  
    Maf_Date DATE,  
    Status VARCHAR(20)  
);
```

Output:

Field	Type	Null	Key	Default	Extra
Aircraft_ID	int	NO	PRI	NULL	
Model	varchar(50)	YES		NULL	
Capacity	int	YES		NULL	
Maf_Date	date	YES		NULL	
Status	varchar(20)	YES		NULL	

```
CREATE TABLE Booking (  
    Booking_ID INT PRIMARY KEY,  
    Booking_Date DATE,  
    Seat_no VARCHAR(5),  
    ID_no INT,  
    Price DECIMAL(10,2)  
);
```

Output:

Field	Type	Null	Key	Default	Extra
Booking_ID	int	NO	PRI	NULL	
Booking_Date	date	YES		NULL	
Seat_no	varchar(5)	YES		NULL	
ID_no	int	YES		NULL	
Price	decimal(10,2)	YES		NULL	

```
CREATE TABLE Airport (  
    Airport_ID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    City VARCHAR(50),  
    Country VARCHAR(50),  
    Timezone VARCHAR(50)  
);
```

Output:

Field	Type	Null	Key	Default	Extra
Airport_ID	int	NO	PRI	NULL	
Name	varchar(50)	YES		NULL	
City	varchar(50)	YES		NULL	
Country	varchar(50)	YES		NULL	
Timezone	varchar(50)	YES		NULL	


```
CREATE TABLE Passenger (
    Passenger_ID INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    D_O_B DATE,
    Age INT,
    Book_ID INT,
    FOREIGN KEY (Book_ID) REFERENCES
Booking(Booking_ID)
);
```

Output:

Field	Type	Null	Key	Default	Extra
Passenger_ID	int	NO	PRI	NULL	
First_Name	varchar(50)	YES		NULL	
Last_Name	varchar(50)	YES		NULL	
D_O_B	date	YES		NULL	
Age	int	YES		NULL	
Book_ID	int	YES	MUL	NULL	

```
CREATE TABLE Flight (
    Flight_ID INT PRIMARY KEY,
    Dep_Time TIME,
    Arrival_Time TIME,
    Aircode VARCHAR(10),
    Air_ID INT,
    FOREIGN KEY (Air_ID) REFERENCES
Aircraft(Aircraft_ID)
);
```

Output:

Field	Type	Null	Key	Default	Extra
Flight_ID	int	NO	PRI	NULL	
Dep_Time	time	YES		NULL	
Arrival_Time	time	YES		NULL	
Aircode	varchar(10)	YES		NULL	
Air_ID	int	YES	MUL	NULL	

2. DML (Data Manipulation Language) Queries

These queries insert, update, and delete data.

Insert Data

```
INSERT INTO Booking (Booking_ID, Booking_Date,
Seat_no, ID_no, Price)
VALUES
(101, '2025-04-01', '12A', 1, 350.00),
(102, '2025-04-02', '7C', 2, 450.00),
(103, '2025-04-03', '18B', 3, 600.00);
```

Output:

Booking_ID	Booking_Date	Seat_no	ID_no	Price
101	2025-04-01	12A	1	350.00
102	2025-04-02	7C	2	450.00
103	2025-04-03	18B	3	600.00

```
INSERT INTO Passenger (Passenger_ID, First_Name,
Last_Name, D_O_B, Age, Book_ID)
VALUES
(1, 'John', 'Doe', '1990-05-14', 34, 101),
(2, 'Alice', 'Smith', '1985-11-20', 39, 102),
(3, 'Michael', 'Brown', '1992-07-08', 32, 103);
```

Output:

Passenger_ID	First_Name	Last_Name	D_O_B	Age	Book_ID
1	John	Doe	1990-05-14	34	101
2	Alice	Smith	1985-11-20	39	102
3	Michael	Brown	1992-07-08	32	103

```
INSERT INTO Aircraft(Aircraft_ID, Model, Capacity,
Maf_Date, Status)
VALUES
(301, 'Boeing737', 180, '2015-06-15', 'Active'),
(302, 'AirbusA320', 200, '2018-03-22', 'Active'),
(303, 'Boeing787', 250, '2020-09-10', 'Under
Maintenance');
```

Output:

Aircraft_ID	Model	Capacity	Maf_Date	Status
301	Boeing737	180	2015-06-15	Active
302	AirbusA320	200	2018-03-22	Active
303	Boeing787	250	2020-09-10	Under Maintenance

```

INSERT INTO Airport(Airport_ID, Name, City, Country,
Timezone)
VALUES
(401, 'Heathrow Airport', 'London', 'UK', 'GMT+0'),
(402, 'JFK International', 'New York', 'USA', 'GMT-
5'),
(403, 'Changi Airport', 'Singapore', 'Singapore',
'GMT+8');

```

Output:

Airport_ID	Name	City	Country	Timezone
401	Heathrow Airport	London	UK	GMT+0
402	JFK International	New York	USA	GMT-5
403	Changi Airport	Singapore	Singapore	GMT+8

```

INSERT INTO Flight (Flight_ID, Dep_Time,
Arrival_Time, Aircode, Air_ID)
VALUES
(201, '10:30:00', '14:00:00', 'AA101', 301),
(202, '15:45:00', '18:20:00', 'BA202', 302),
(203, '08:00:00', '11:30:00', 'LH303', 303);

```

Output:

Flight_ID	Dep_Time	Arrival_Time	Aircode	Air_ID
201	10:30:00	14:00:00	AA101	301
202	15:45:00	18:20:00	BA202	302
203	08:00:00	11:30:00	LH303	303

Update Data

Modify existing records.

```
UPDATE Passenger
SET Age = 35
WHERE Passenger_ID = 1;
```

Output:

Passenger_ID	First_Name	Last_Name	D_O_B	Age	Book_ID
1	John	Doe	1990-05-14	35	101
2	Alice	Smith	1985-11-20	39	102
3	Michael	Brown	1992-07-08	32	103

Delete Data

Remove specific records.

```
DELETE FROM Passenger WHERE Book_ID = 103;
DELETE FROM Booking WHERE Booking_ID = 103;
```

Output:

Booking_ID	Booking_Date	Seat_no	ID_no	Price
101	2025-04-01	12A	1	350.00
102	2025-04-02	7C	2	450.00

3. Retrieval Queries

Fetching relevant data.

Get all flight details

```
SELECT * FROM Flight;
```

Output:

Flight_ID	Dep_Time	Arrival_Time	Aircode	Air_ID
201	10:30:00	14:00:00	AA101	301
202	15:45:00	18:20:00	BA202	302
203	08:00:00	11:30:00	LH303	303

Find passengers who booked flights after April 1, 2025

```
SELECT First_Name, Last_Name, Booking_Date
FROM Passenger
JOIN Booking ON Passenger.Book_ID =
Booking.Booking_ID
WHERE Booking.Booking_Date > '2025-04-01';
```

Output:

First_Name	Last_Name	Booking_Date
Alice	Smith	2025-04-02

Find flights departing from a specific airport

```
SELECT Flight_ID, Dep_Time, Arrival_Time  
FROM Flight  
WHERE Aircode = 'AA101';
```

Output:

Flight_ID	Dep_Time	Arrival_Time
201	10:30:00	14:00:00

4. Joins & Views

Join Passengers with their Bookings

```
SELECT Passenger.First_Name, Passenger.Last_Name,  
Booking.Seat_no, Booking.Price  
FROM Passenger  
JOIN Booking ON Passenger.Book_ID =  
Booking.Booking_ID;
```

Output:

First_Name	Last_Name	Seat_no	Price
John	Doe	12A	350.00
Alice	Smith	7C	450.00

Create a View for Active Flights

```
CREATE VIEW ActiveFlights AS
SELECT Flight_ID, Dep_Time, Arrival_Time, Aircode
FROM Flight
WHERE Dep_Time IS NOT NULL;
```

Output:

```
mysql> CREATE VIEW ActiveFlights AS
-> SELECT Flight_ID, Dep_Time, Arrival_Time, Aircode
-> FROM Flight
-> WHERE Dep_Time IS NOT NULL;
Query OK, 0 rows affected (0.01 sec)
```

5.COUNT Queries

Count Total Passengers

```
SELECT COUNT(*) AS Total_Passengers FROM Passenger;
```

Output:

Total_Passengers
2

Count Total Flights

```
SELECT COUNT(*) AS Total_Flights FROM Flight;
```

Output:

Total_Flights
3

6.SUM Queries

Calculate Total Revenue from Bookings

```
SELECT SUM(Price) AS Total_Revenue FROM Booking;
```

Output:

Total_Revenue
800.00

Calculate Total Capacity of All Aircrafts

```
SELECT SUM(Capacity) AS Total_Seat_Capacity FROM Aircraft;
```

Output:

Total_Seat_Capacity
630

Conclusion for the Airline Booking System

- The system successfully stores passenger, flight, aircraft, booking, and airport data.
- SQL queries allow for efficient data retrieval, revenue tracking, and capacity management.
- Aggregate functions like **COUNT**, **SUM**, and **AVG** help analyze flight trends and profitability.
- Future improvements can include adding stored procedures for automated booking processing and triggers for flight status updates.