**UNIVERSITY INSTITUTE OF COMPUTING**

**CASE STUDY REPORT**

**ON**

**AIRLINE MANAGEMENT SYSTEM**

Program Name: BCA

Subject Name/Code: Database Management System (23CAT-251)

**Submitted by: Submitted to:**

**Name: JYOTI Kumari** **Name: Mr. ARVINDER SINGH**

**UID: 23BCA10071** **Designation: Assistant Professor**

**Section:** 23BCA- 4 "A"

ABSTRACT

**Introduction:**

An **Airline Management System (AFM)** is a powerful, computerized solution that is designed to efficiently manage all the major operations and services offered by an airline company. As the aviation industry continues to grow, managing flight schedules, aircraft details, passenger records, ticket bookings, cancellations, staff assignments, and customer services manually becomes highly complex and time-consuming. Hence, there is a strong need for a centralized and automated system that can handle all these tasks with accuracy and speed.

The **Airline Management System** helps airline operators store, retrieve, and update data in a structured and secure way. The system is built using a **Relational Database Management System (RDBMS)** where all data is stored in related tables, and **SQL (Structured Query Language)** is used to interact with this data. This not only makes the system efficient but also scalable, as it can handle large volumes of data without compromising performance.

The system is capable of managing the end-to-end journey of a passenger — from flight search and booking to check-in and post-flight services. It also manages back-end processes like assigning pilots and crew to flights, keeping track of aircraft maintenance schedules, and generating reports for operational and strategic decision-making.

**Key Techniques Used:**

**1. Relational Database Design**

* The system uses **entity-relationship (ER) modeling** to design the database structure.
* Each real-world object (like Passenger, Flight, Booking) is represented as a **table**.
* **Primary keys** uniquely identify records in each table.
* **Foreign keys** are used to establish **relationships** between tables (e.g., a booking is linked to both a flight and a passenger).

**2. Data Normalization**

* Tables are designed using **normalization techniques** (1NF, 2NF, 3NF) to eliminate data redundancy and improve data integrity.
* This ensures that:
  + Data is logically grouped,
  + No duplicate data is stored,
  + All attributes in a table depend on the primary key.

**3. SQL Query Language**

* **SQL (Structured Query Language)** is used to:
  + **Create** tables (CREATE TABLE)
  + **Insert** data (INSERT INTO)
  + **Update** data (UPDATE)
  + **Delete** records (DELETE)
  + **Retrieve** information (SELECT)
* SQL queries are optimized for performance and quick access to data.

**4. Constraints and Validation**

* To maintain **data accuracy and consistency**, various **constraints** are applied:
  + NOT NULL – ensures that a column cannot have a NULL value.
  + UNIQUE – ensures that all values in a column are different.
  + CHECK – sets a condition that values in a column must satisfy.
  + DEFAULT – sets a default value for a column.
  + FOREIGN KEY – maintains referential integrity between tables.

**5. Relationship Management**

* **One-to-Many**: A flight can have many bookings, but each booking is linked to only one flight.
* **Many-to-One**: Many bookings can be made by one passenger.
* These relationships are built using **foreign key constraints**.

**6. Indexing (Optional/Advanced)**

* Indexes may be created on frequently searched fields like Flight\_ID or Passenger\_ID to **improve search performance** in larger databases.

**System Configuration:**

**Hardware Requirements:**

* **Processor**: Intel Core i3 or higher
* **RAM**: 4 GB (minimum)
* **Storage**: 500 MB of free disk space
* **Display**: 1280x800 resolution or higher (for better UI experience)

**Software Requirements:**

* **Operating System (OS)**:
  + Windows 10/11 (recommended)
  + Linux (Ubuntu or CentOS)
  + macOS (latest version)
* **Database Management System (DBMS)**:
  + MySQL Server 8.0+ (or higher)
  + XAMPP (if running a local server setup)
* **SQL Client**:
  + MySQL Workbench (for GUI-based SQL queries and database management)
  + phpMyAdmin (for web-based interface to manage MySQL databases)
  + CLI (Command Line Interface) (for direct command line SQL queries)
* **Text Editor/IDE**:
  + VS Code (for coding and debugging, highly recommended)
  + Notepad++ (lightweight editor for coding)

SUMMARY

INPUT:

The system takes **SQL commands** and **data entries** as input, such as:

* **Flight Details**: Flight number, departure/arrival cities, date, time, and aircraft type.
* **Passenger Information**: Name, contact details, passport number, and seat preferences.
* **Booking Details**: Passenger booking information, including flight choice, class (economy, business, etc.), and payment status.
* **Ticket Payment**: Payment transaction details, including payment method and amount.
* **Crew Information**: Pilot and cabin crew assignments, including their schedules and qualifications.
* **Flight Status Updates**: Flight delays, cancellations, or status changes.
* **Queries**: SQL commands to retrieve or manipulate stored data, such as searching for available flights, checking booking history, or generating reports.

ER DIAGRAM:

An **Entity-Relationship (ER) Diagram** for an **Airline Management System** is a visual tool used to design and represent the structure of the airline’s database. It identifies the key entities involved, such as **Passenger**, **Flight**, **Booking**, **Aircraft**, and **Pilot**, along with their attributes. These relationships are shown with proper cardinality, such as one-to-many or many-to-many, to ensure data consistency and efficient database design. Overall, the ER diagram serves as a blueprint for building a reliable and well-structured airline database system. Here are an **ER DIGRAM** for **AIRLINE MANAGEMENT SYSTEM: -**

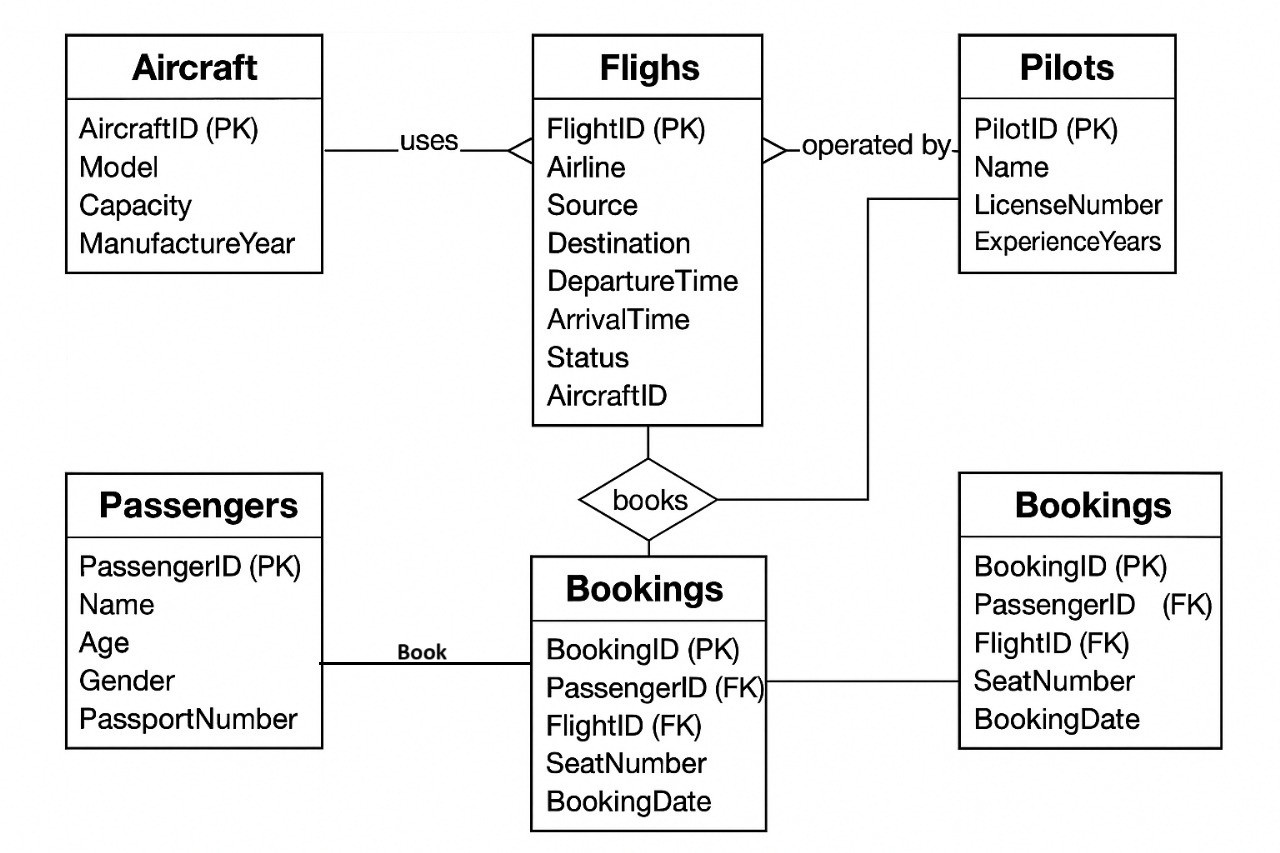
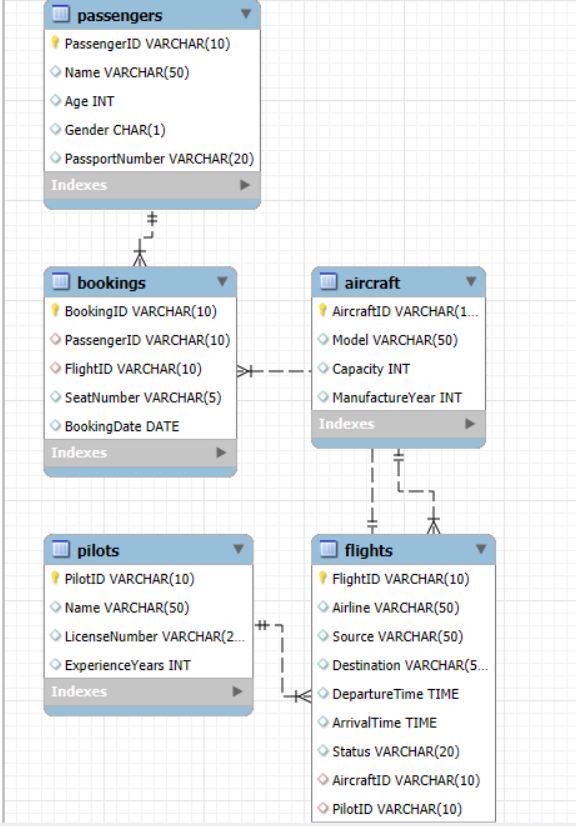
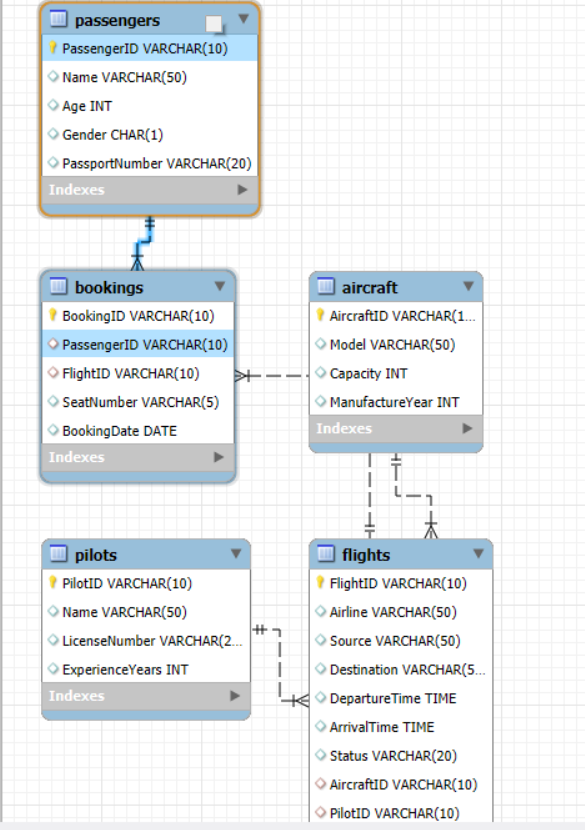
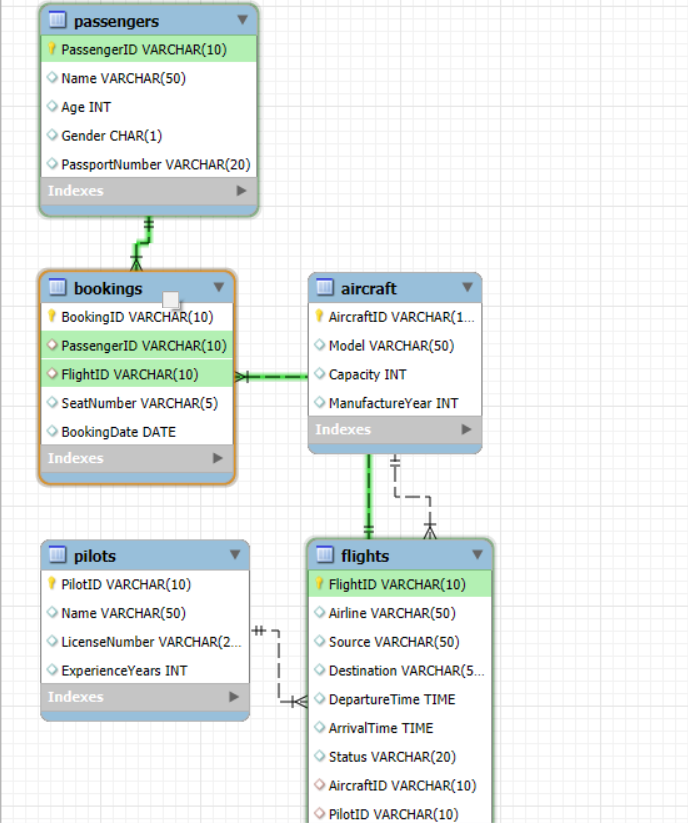
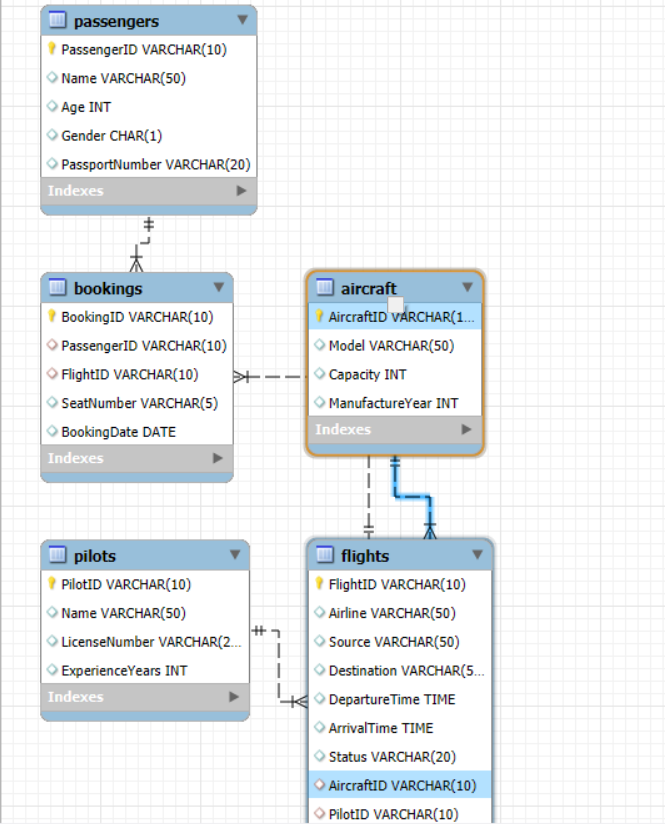


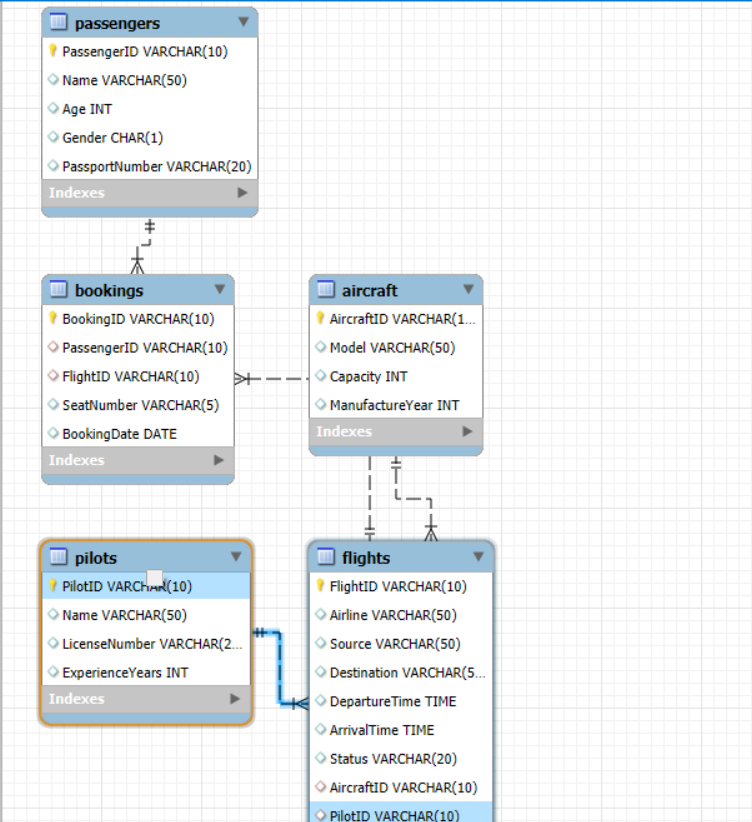
TABLE REALTION:

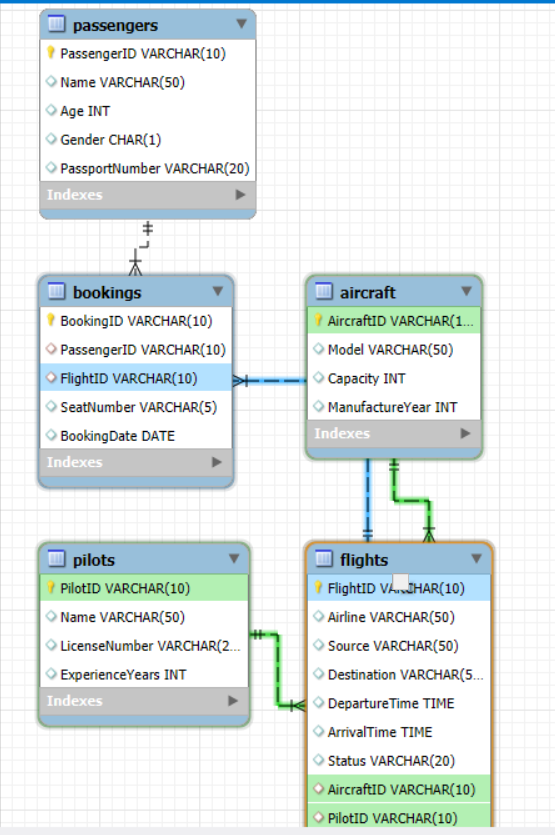






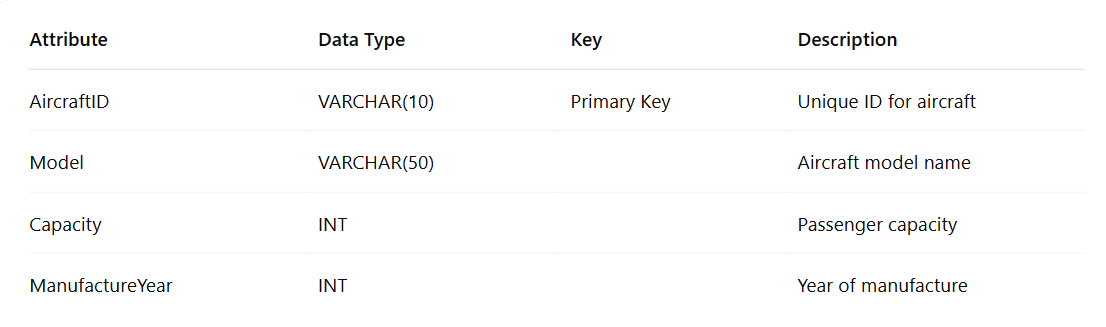






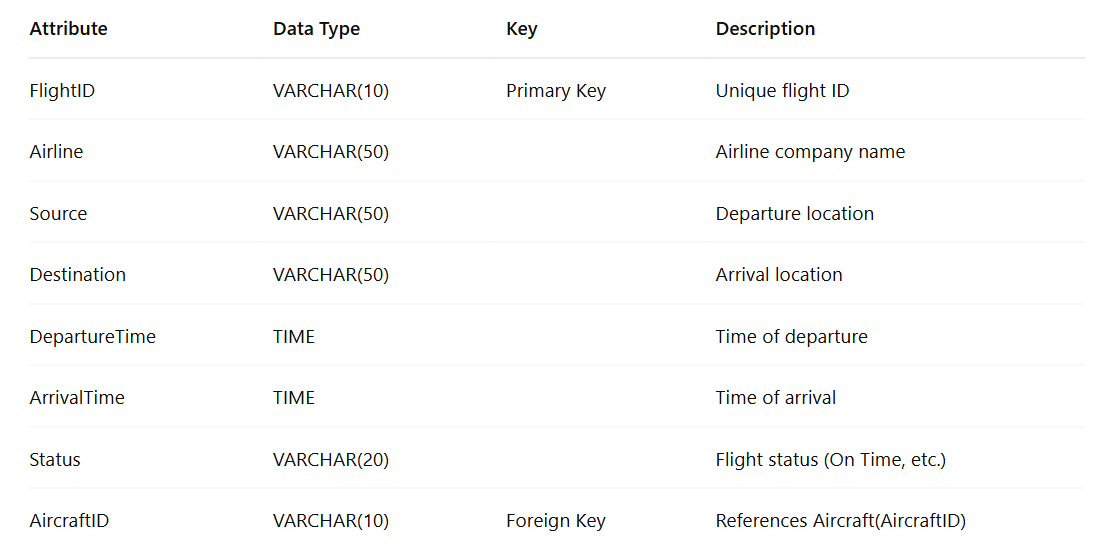
TABULAR FORMAT:

**1. Aircraft**

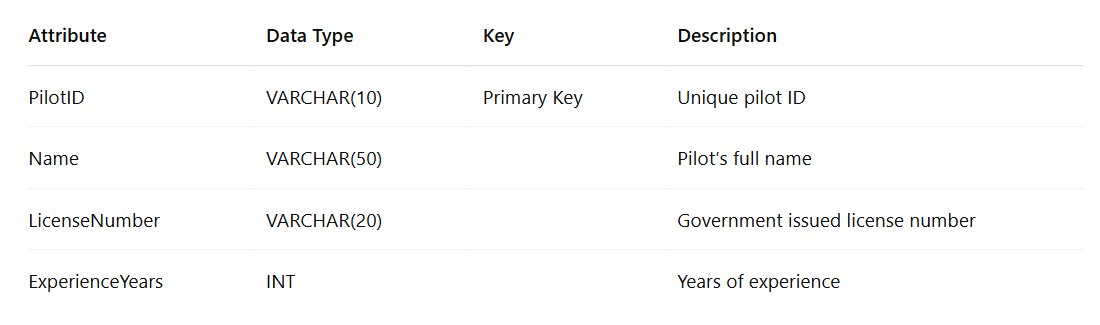


|  |
| --- |
|  |

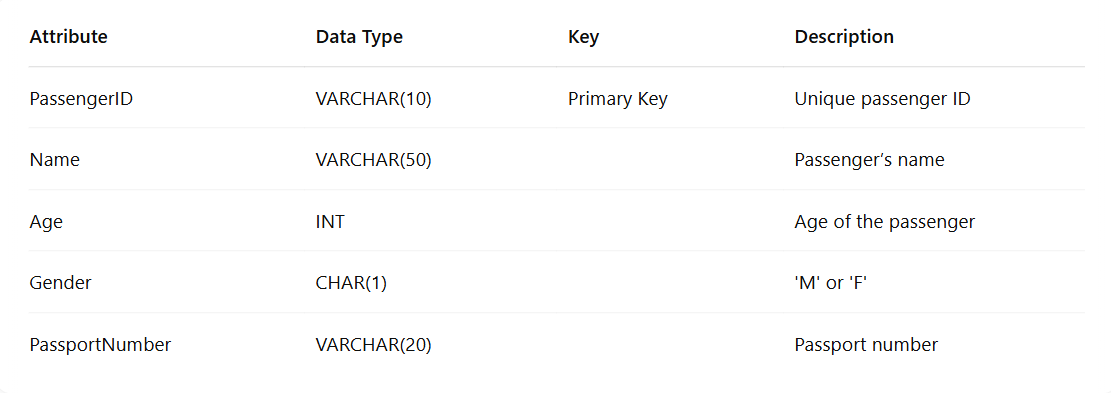
**2. Flights**

****

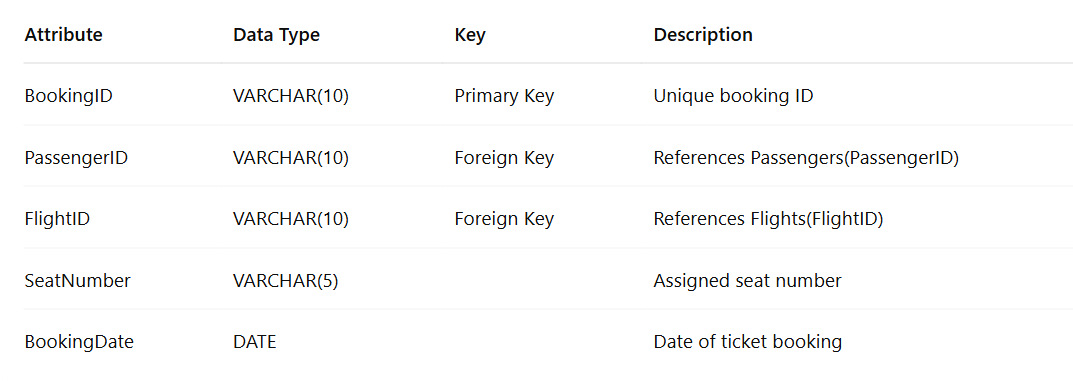
**3. Pilots**

****

**4. Passengers**

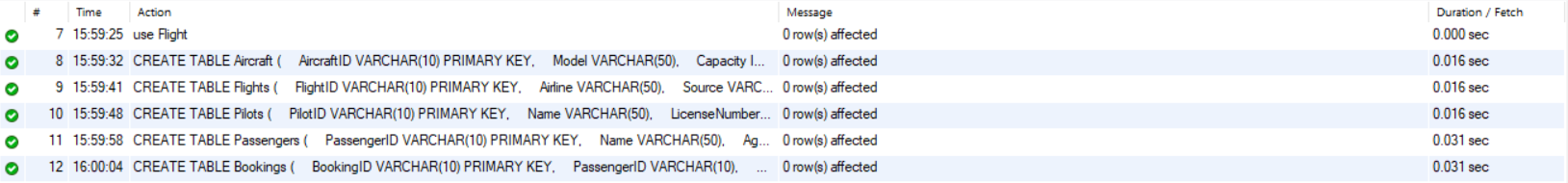
****

**5. Bookings**

****

IMPLEMENTATION:

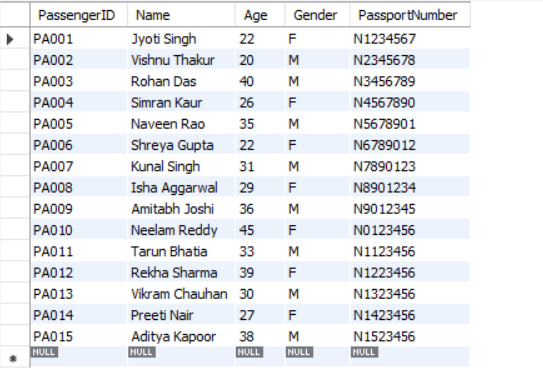
Tables created:



SQL Queries Performed with Output:

1. Show all passengers

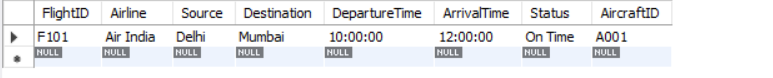
SELECT \* FROM Passenger;



2. Display all flights from Delhi to Mumbai

SELECT \* FROM Flights

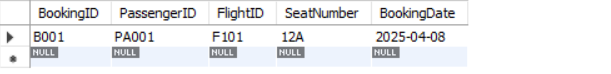
WHERE Source = 'Delhi' AND Destination = 'Mumbai';



3. Show all bookings made by a passenger with ID "PA001"

SELECT \* FROM Bookings

WHERE Passenger\_ID = "PA001";

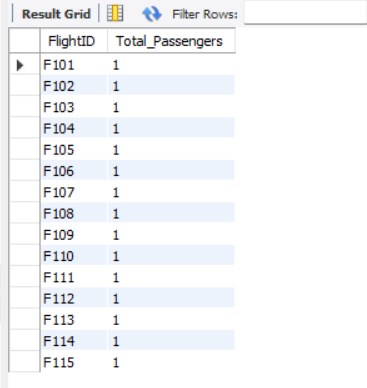


4. Count how many passengers have booked each flight

SELECT FlightID, COUNT(\*) AS Total\_Passengers

FROM Bookings

GROUP BY FlightID;



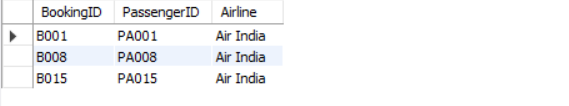
5. Get all bookings for 'Air India' flights

SELECT B.BookingID, B.PassengerID, F.Airline

FROM Bookings B

JOIN Flights F ON B.FlightID = F.FlightID

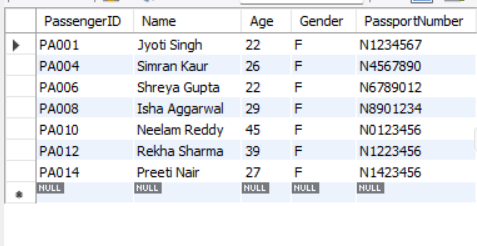
WHERE F.Airline = 'Air India';



6. Show all female passengers

SELECT \* FROM Passengers

WHERE Gender = 'F';



7. Find the average aircraft capacity

SELECT AVG(Capacity) AS AvgCapacity

FROM Aircraft;

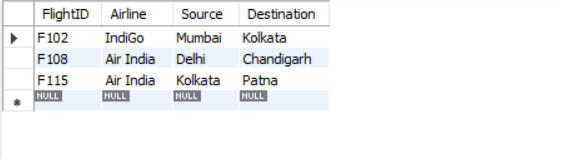


8. List Flights that are currently ‘Delayed’

SELECT FlightID, Airline, Source, Destination

FROM Flights

WHERE Status = 'Delayed';



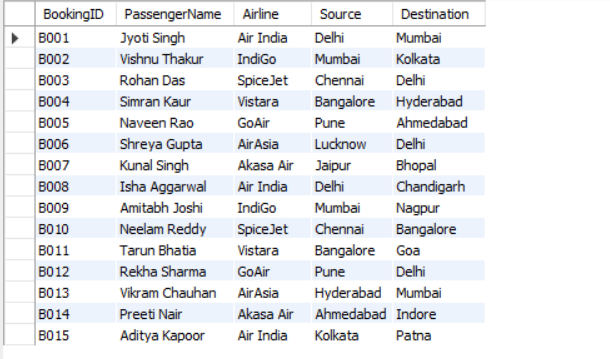
9. Show all bookings with passenger and flight details

SELECT B.BookingID, P.Name AS PassengerName, F.Airline, F.Source, F.Destination

FROM Bookings B

JOIN Passengers P ON B.PassengerID = P.PassengerID

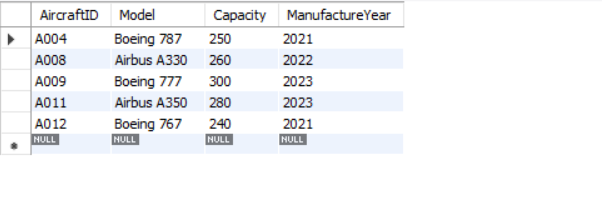
JOIN Flights F ON B.FlightID = F.FlightID;



10. Get the details of aircraft manufactured after 2020

SELECT \* FROM Aircraft

WHERE ManufactureYear > 2020;

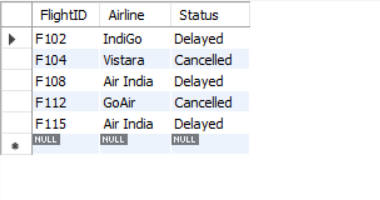


11. List all flights that are either ‘Cancelled’ or ‘Delayed’

SELECT FlightID, Airline, Status

FROM Flights

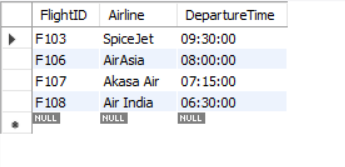
WHERE Status IN ('Cancelled', 'Delayed');



12. Find flights that depart before 10:00 AM

SELECT FlightID, Airline, DepartureTime

FROM Flights WHERE DepartureTime < '10:00:00';



13. Show passenger names with their booked flight destinations

SELECT P.Name AS PassengerName, F.Destination

FROM Bookings B

JOIN Passengers P ON B.PassengerID = P.PassengerID

JOIN Flights F ON B.FlightID = F.FlightID;



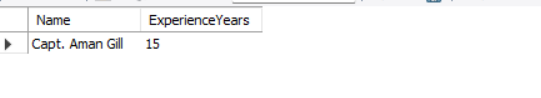
14. Show the name of the pilot with the highest experience

SELECT Name, ExperienceYears

FROM Pilots

ORDER BY ExperienceYears DESC

LIMIT 1;



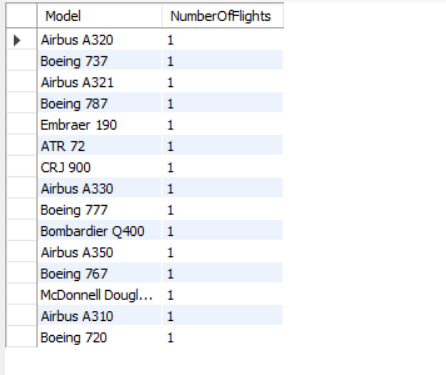
15. Display the number of flights associated with each aircraft model

SELECT A.Model, COUNT(F.FlightID) AS NumberOfFlights

FROM Aircraft A

JOIN Flights F ON A.AircraftID = F.AircraftID

GROUP BY A.Model;



16. List all passengers booked on ‘Air India’ flights

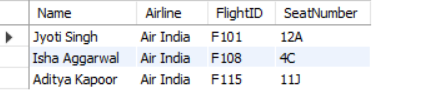
SELECT P.Name, F.Airline, F.FlightID, B.SeatNumber

FROM Passengers P

JOIN Bookings B ON P.PassengerID = B.PassengerID

JOIN Flights F ON B.FlightID = F.FlightID

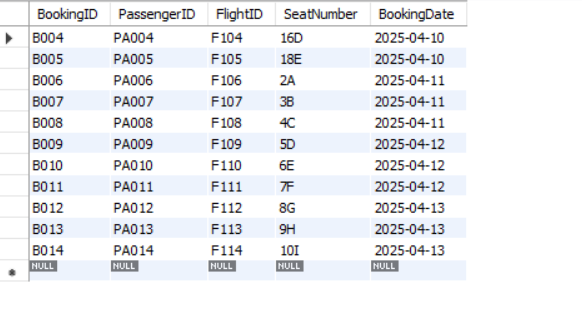
WHERE F.Airline = 'Air India';



17. List all bookings made between '2025-04-10' and '2025-04-13'

SELECT \* FROM Bookings

WHERE BookingDate BETWEEN '2025-04-10' AND '2025-04-13';

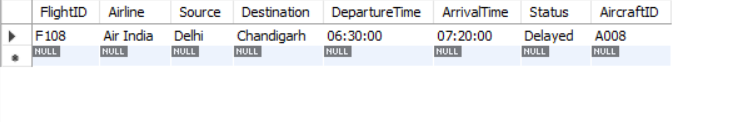


18. **Find the flight that arrives the earliest**

SELECT \* FROM Flights

ORDER BY ArrivalTime ASC

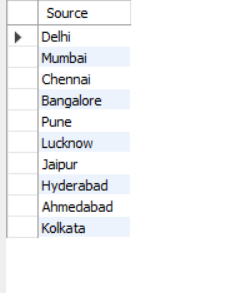
LIMIT 1;



19. **List all distinct flight sources**

**SELECT DISTINCT Source**

**FROM Flights;**

****

SUMMARY:

This project presents a comprehensive **Airline Management System** built using **SQL and relational database concepts**, designed to replicate the key operational elements of a real-world airline database. It consists of **five interconnected entities**:

1. **Aircraft Table** – Stores details about the aircraft fleet such as ID, model, capacity, and manufacture year.
2. **Flights Table** – Holds scheduled flights including airline name, source and destination, departure/arrival times, status, and the aircraft used.
3. **Pilots Table** – Contains pilot profiles including license numbers and years of flying experience.
4. **Passengers Table** – Records information about passengers like age, gender, and passport details.
5. **Bookings Table** – Maintains reservation records mapping passengers to specific flights, seats, and booking dates.

Throughout the project, we applied **DDL (Data Definition Language)** commands to create structured tables, and **DML (Data Manipulation Language)** statements to insert and manage data. By executing **over 19 SQL queries**, we were able to perform data retrieval, filtering, joining, and aggregation. These queries helped us answer critical business questions like:

* Which flights are delayed or cancelled?
* Which aircraft models are used by specific airlines?
* What are the passenger details on a particular flight?
* How many bookings exist for each day?
* Who is the most experienced pilot?

All relationships are clearly defined using **primary and foreign keys**, which ensures **referential integrity** and accurate data mapping.

**CONCLUSION:**

This project proves the importance and power of **Database Management Systems (DBMS)** in modern industries like aviation. A well-designed relational database not only stores information securely but also provides tools for making **data-driven decisions**.

Through this project, we learned:

* How to **model a real-world system** using database concepts.
* The use of **normalization** to avoid redundancy and maintain consistency.
* Practical skills in **writing and optimizing SQL queries** for various purposes.
* How to implement **data relationships** using foreign keys.

The Flight Management System created in this project serves as a **miniature version of real airline databases**. It is scalable and can be extended to include additional modules such as:

* **Crew scheduling**
* **Flight maintenance logs**
* **Airport management**
* **E-ticketing systems**
* **Payment gateways and billing**

In conclusion, this DBMS project is not only a successful academic exercise but also a stepping stone toward building more complex, real-time database applications for large-scale industries.

**GITHUB LINK:**

<https://github.com/Kkukiii/AIRLINE_MANAGEMENT_SYSTEM>