CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

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: -**

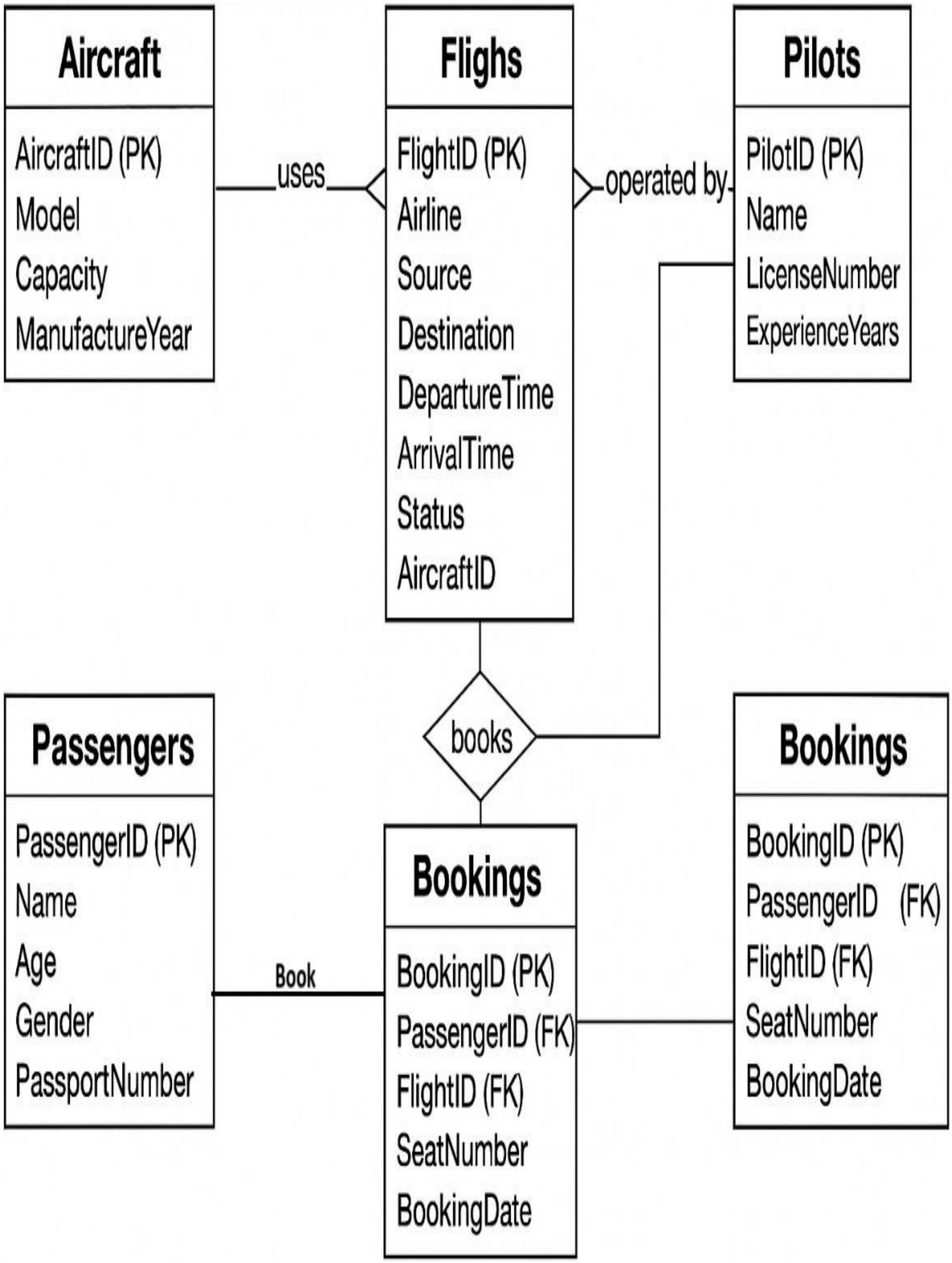
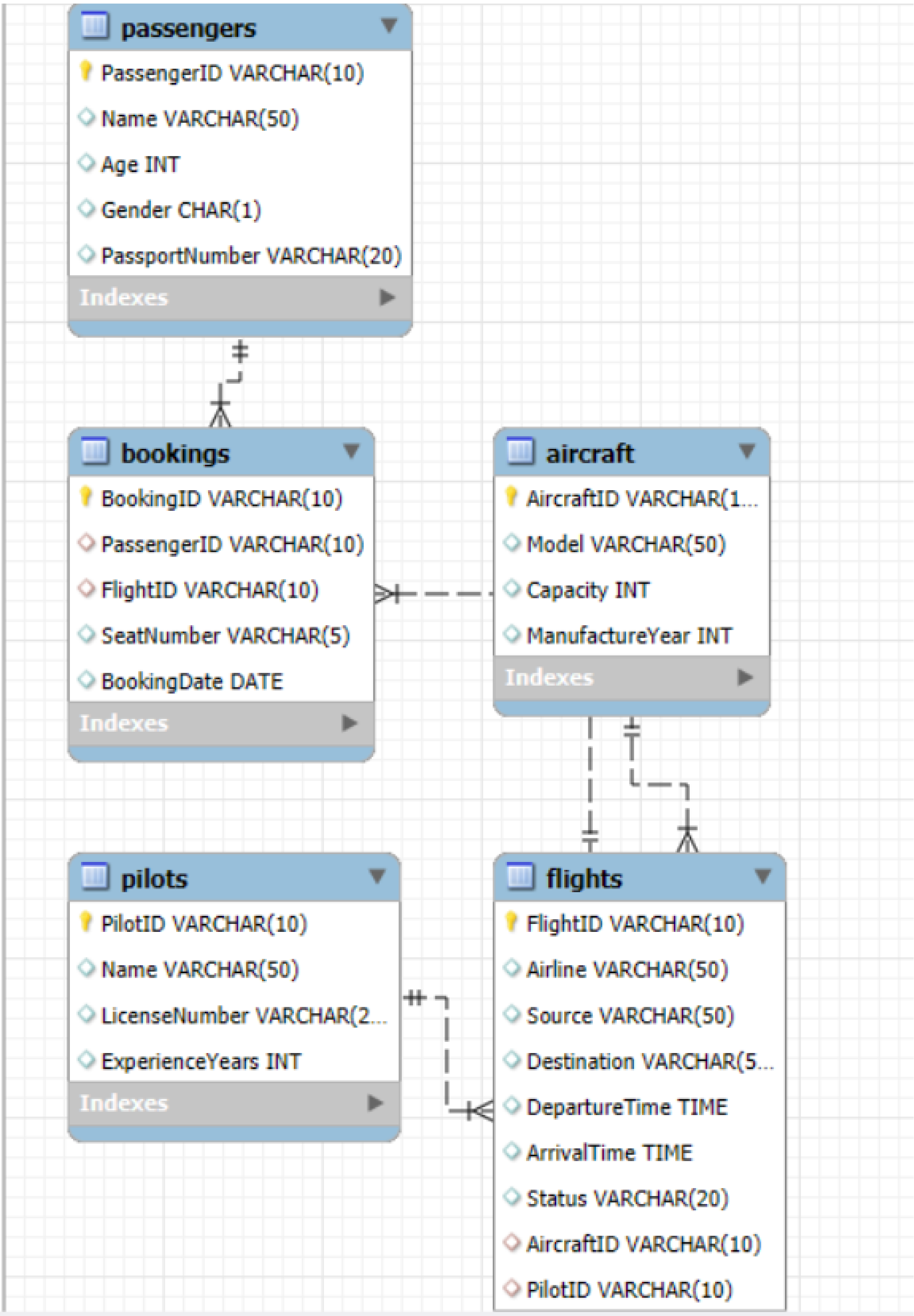

Aircraft
AircraftID (PK)
Model
Capacity
ManufactureYear
uses
Flighs
FlightID (PK)
Airline
Source
Destination
DepartureTime
ArrivalTime
Status
AircraftID
operated by
Pilots
PilotID (PK)
Name
LicenseNumber
ExperienceYears
books
Passengers
PassengerID (PK)
Name
Age
Gender
PassportNumber
Book
Bookings
BookingID (PK)
PassengerID (FK)
FlightID (FK)
SeatNumber
BookingDate
Bookings
BookingID (PK)
PassengerID (FK)
FlightID (FK)
SeatNumber
BookingDate

TABLE REALTION:



passengers

- PassengerID VARCHAR(10)
- Name VARCHAR(50)
- Age INT
- Gender CHAR(1)
- PassportNumber VARCHAR(20)
- Indexes

bookings

- BookingID VARCHAR(10)
- PassengerID VARCHAR(10)
- FlightID VARCHAR(10)
- SeatNumber VARCHAR(5)
- BookingDate DATE
- Indexes

aircraft

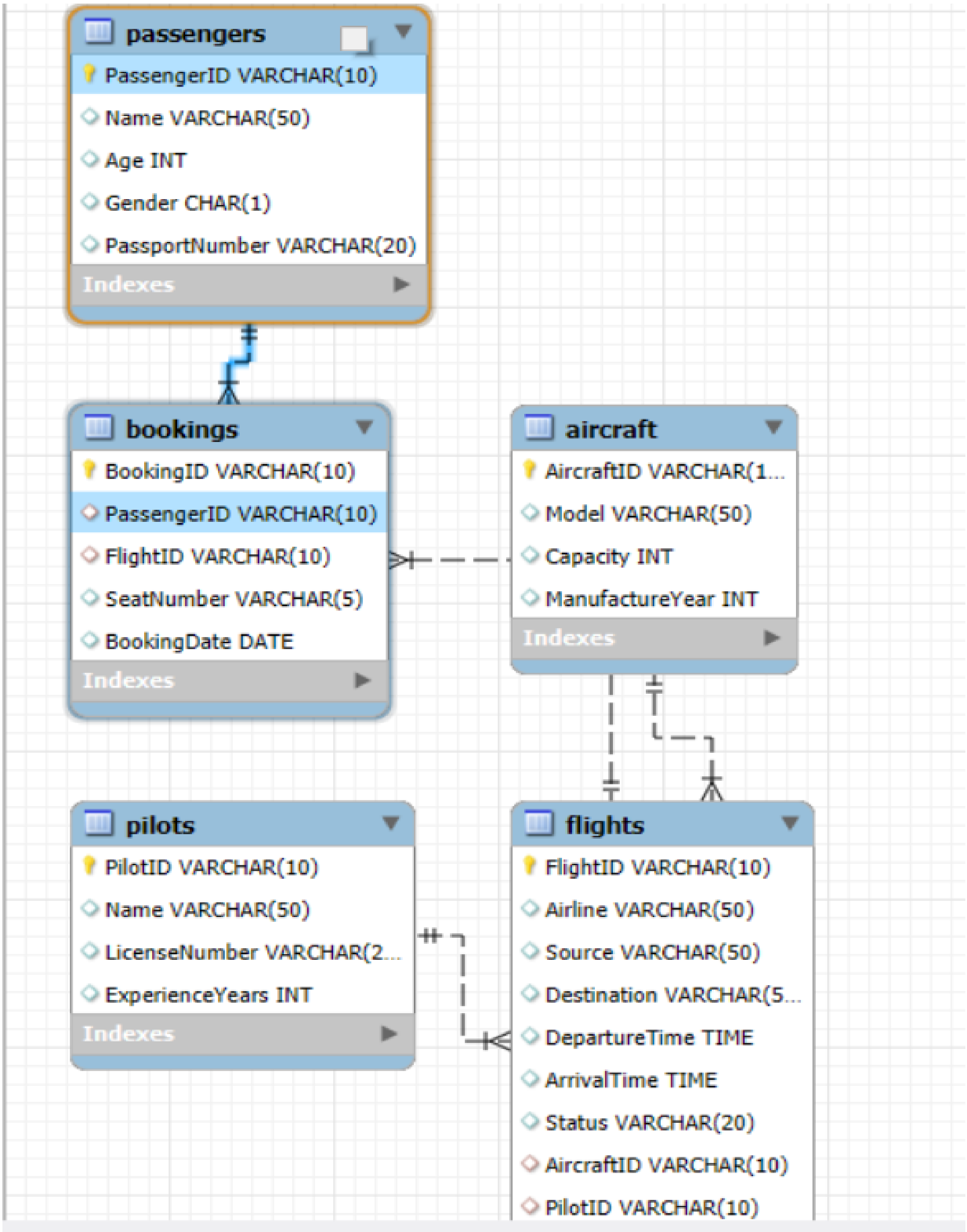
- AircraftID VARCHAR(1...
- Model VARCHAR(50)
- Capacity INT
- ManufactureYear INT
- Indexes

pilots

- PilotID VARCHAR(10)
- Name VARCHAR(50)
- LicenseNumber VARCHAR(2...
- ExperienceYears INT
- Indexes

flights

- FlightID VARCHAR(10)
- Airline VARCHAR(50)
- Source VARCHAR(50)
- Destination VARCHAR(5...
- DepartureTime TIME
- ArrivalTime TIME
- Status VARCHAR(20)
- AircraftID VARCHAR(10)
- PilotID VARCHAR(10)



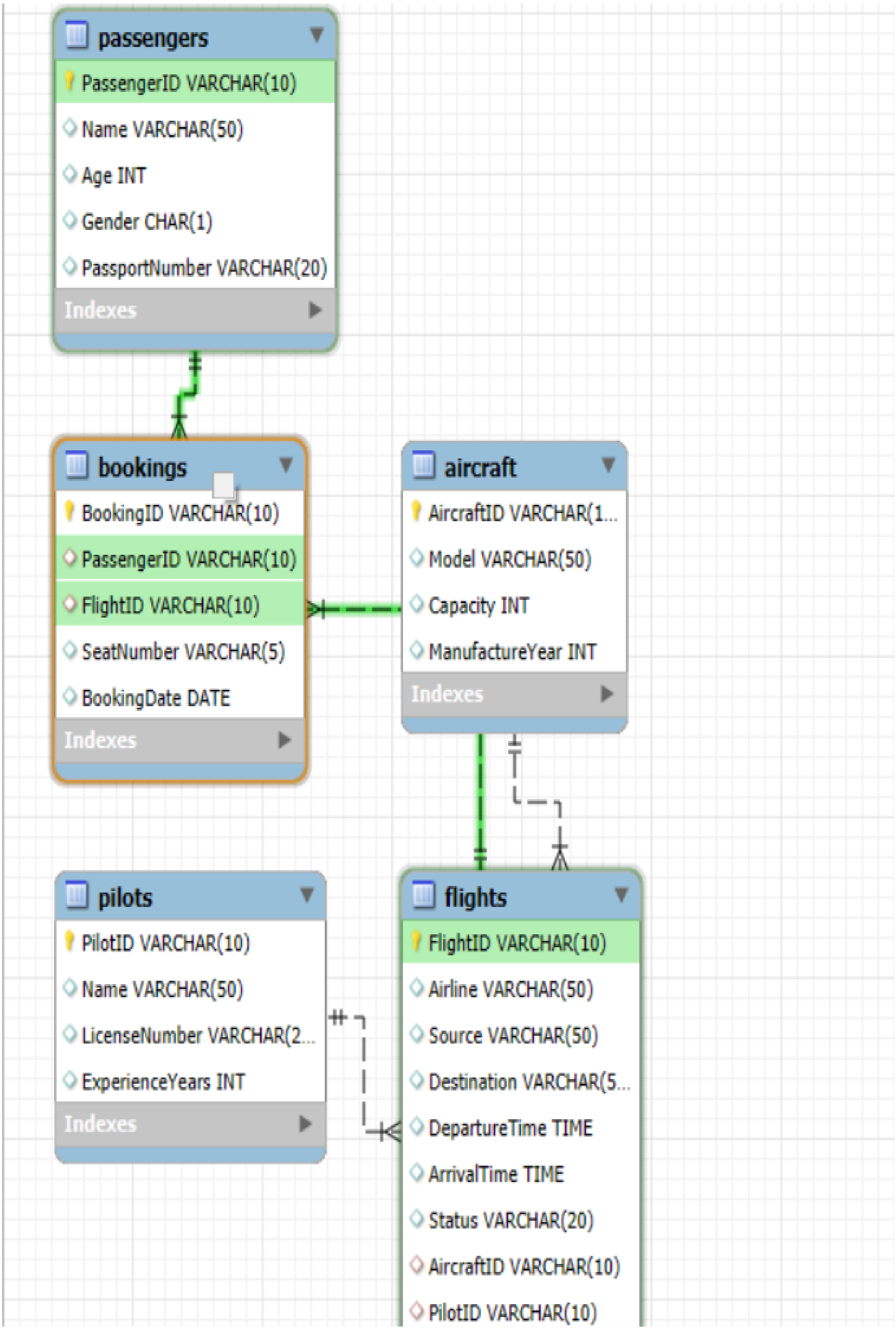
passengers
PassengerID VARCHAR(10)
Name VARCHAR(50)
Age INT
Gender CHAR(1)
PassportNumber VARCHAR(20)

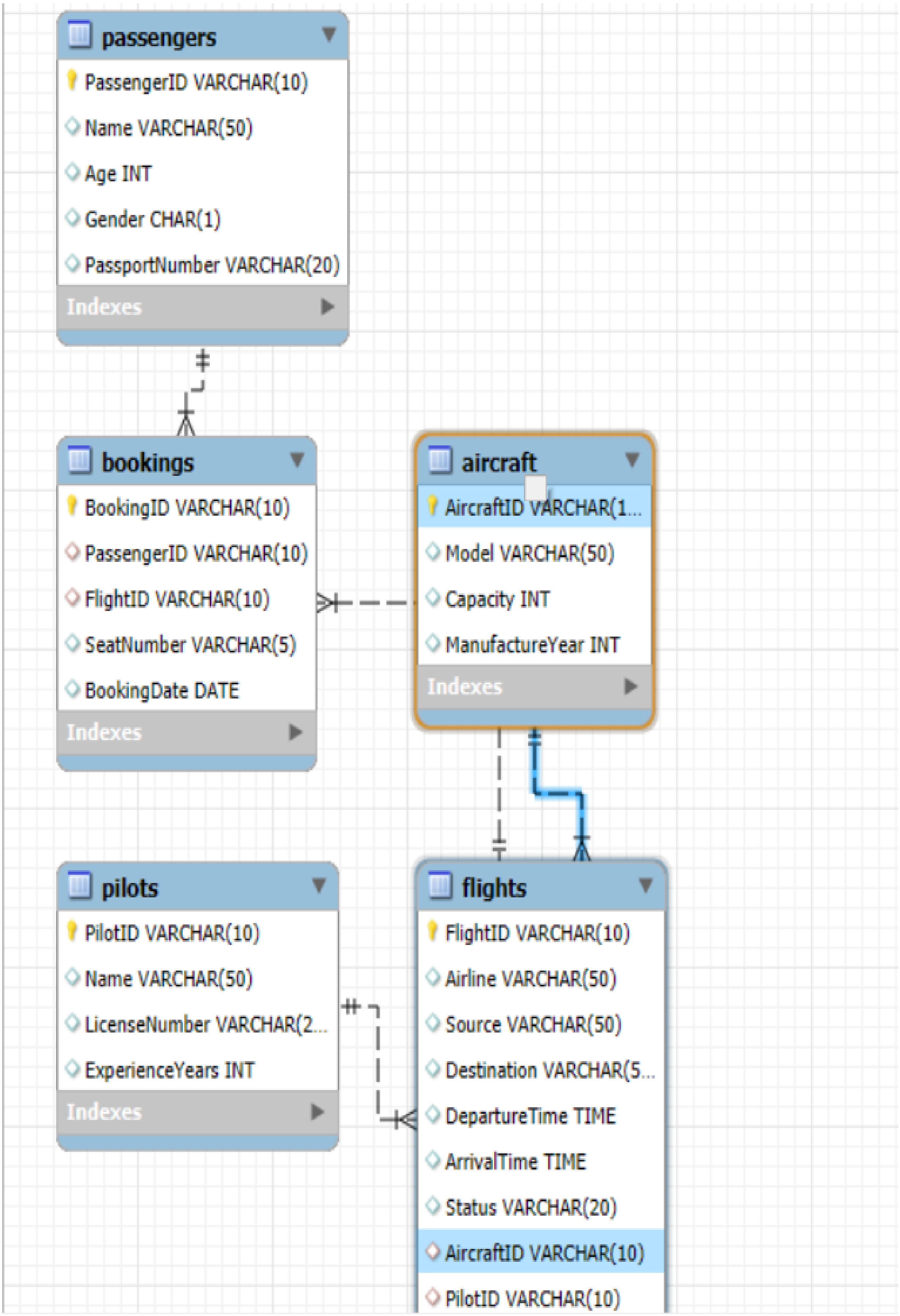
bookings
BookingID VARCHAR(10)
PassengerID VARCHAR(10)
FlightID VARCHAR(10)
SeatNumber VARCHAR(5)
BookingDate DATE

aircraft
AircraftID VARCHAR(1...
Model VARCHAR(50)
Capacity INT
ManufactureYear INT

pilots
PilotID VARCHAR(10)
Name VARCHAR(50)
LicenseNumber VARCHAR(2...
ExperienceYears INT

flights
FlightID VARCHAR(10)
Airline VARCHAR(50)
Source VARCHAR(50)
Destination VARCHAR(5...
DepartureTime TIME
ArrivalTime TIME
Status VARCHAR(20)
AircraftID VARCHAR(10)
PilotID VARCHAR(10)


passengers
PassengerID VARCHAR(10)
Name VARCHAR(50)
Age INT
Gender CHAR(1)
PassportNumber VARCHAR(20)
Indexes
bookings
BookingID VARCHAR(10)
PassengerID VARCHAR(10)
FlightID VARCHAR(10)
SeatNumber VARCHAR(5)
BookingDate DATE
Indexes
aircraft
AircraftID VARCHAR(1...
Model VARCHAR(50)
Capacity INT
ManufactureYear INT
Indexes
pilots
PilotID VARCHAR(10)
Name VARCHAR(50)
LicenseNumber VARCHAR(2...
ExperienceYears INT
Indexes
flights
FlightID VARCHAR(10)
Airline VARCHAR(50)
Source VARCHAR(50)
Destination VARCHAR(5...
DepartureTime TIME
ArrivalTime TIME
Status VARCHAR(20)
AircraftID VARCHAR(10)
PilotID VARCHAR(10)



passengers
- PassengerID VARCHAR(10)
- Name VARCHAR(50)
- Age INT
- Gender CHAR(1)
- PassportNumber VARCHAR(20)

Indexes

bookings
- BookingID VARCHAR(10)
- PassengerID VARCHAR(10)
- FlightID VARCHAR(10)
- SeatNumber VARCHAR(5)
- BookingDate DATE

Indexes

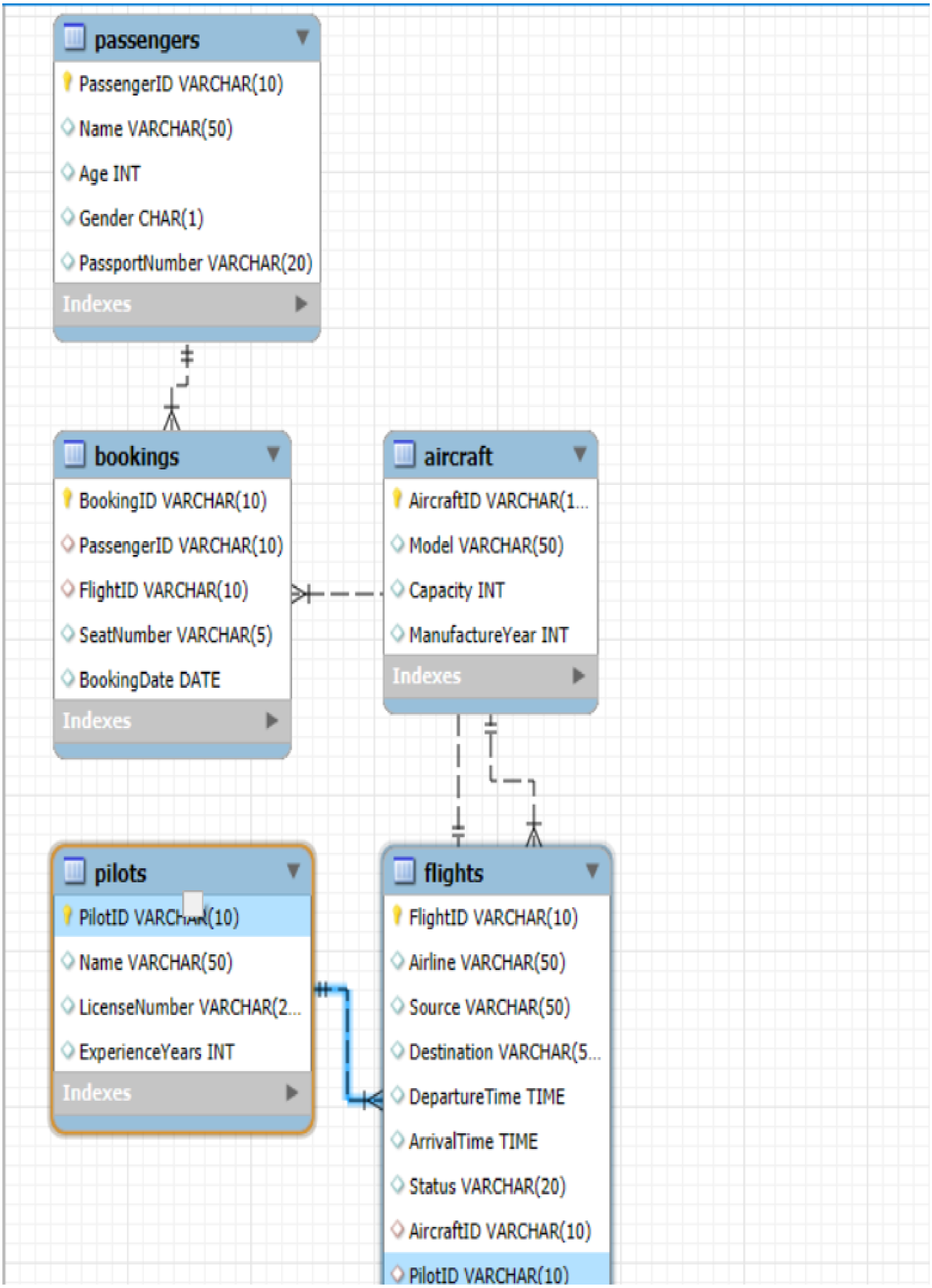
aircraft
- AircraftID VARCHAR(1...
- Model VARCHAR(50)
- Capacity INT
- ManufactureYear INT

Indexes

pilots
- PilotID VARCHAR(10)
- Name VARCHAR(50)
- LicenseNumber VARCHAR(2...
- ExperienceYears INT

Indexes

flights
- FlightID VARCHAR(10)
- Airline VARCHAR(50)
- Source VARCHAR(50)
- Destination VARCHAR(5...
- DepartureTime TIME
- ArrivalTime TIME
- Status VARCHAR(20)
- AircraftID VARCHAR(10)
- PilotID VARCHAR(10)



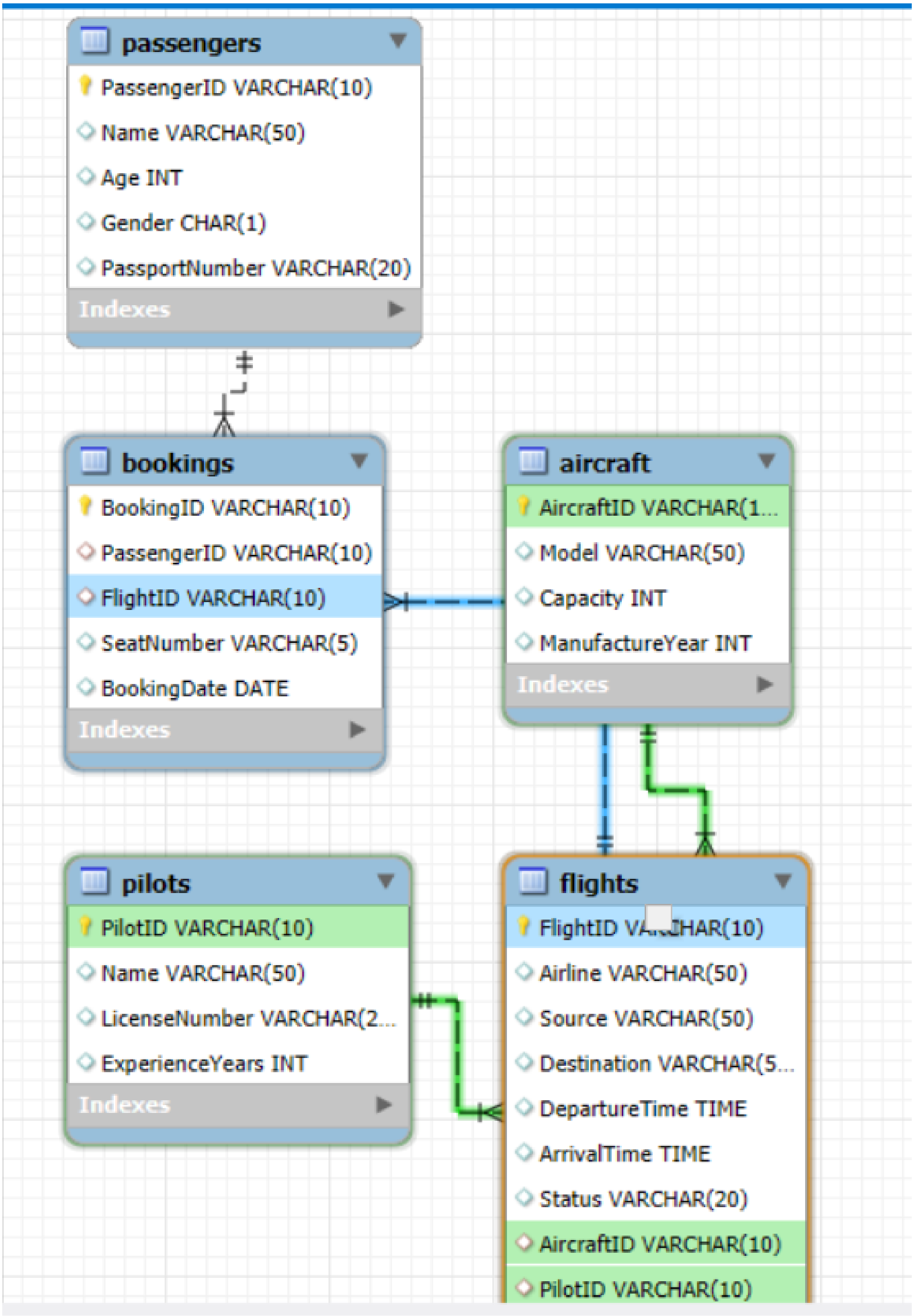
passengers
- PassengerID VARCHAR(10)
- Name VARCHAR(50)
- Age INT
- Gender CHAR(1)
- PassportNumber VARCHAR(20)
- Indexes

bookings
- BookingID VARCHAR(10)
- PassengerID VARCHAR(10)
- FlightID VARCHAR(10)
- SeatNumber VARCHAR(5)
- BookingDate DATE
- Indexes

aircraft
- AircraftID VARCHAR(1...
- Model VARCHAR(50)
- Capacity INT
- ManufactureYear INT
- Indexes

pilots
- PilotID VARCHAR(10)
- Name VARCHAR(50)
- LicenseNumber VARCHAR(2...
- ExperienceYears INT
- Indexes

flights
- FlightID VARCHAR(10)
- Airline VARCHAR(50)
- Source VARCHAR(50)
- Destination VARCHAR(5...
- DepartureTime TIME
- ArrivalTime TIME
- Status VARCHAR(20)
- AircraftID VARCHAR(10)
- PilotID VARCHAR(10)



passengers
- PassengerID VARCHAR(10)
- Name VARCHAR(50)
- Age INT
- Gender CHAR(1)
- PassportNumber VARCHAR(20)
- Indexes

bookings
- BookingID VARCHAR(10)
- PassengerID VARCHAR(10)
- FlightID VARCHAR(10)
- SeatNumber VARCHAR(5)
- BookingDate DATE
- Indexes

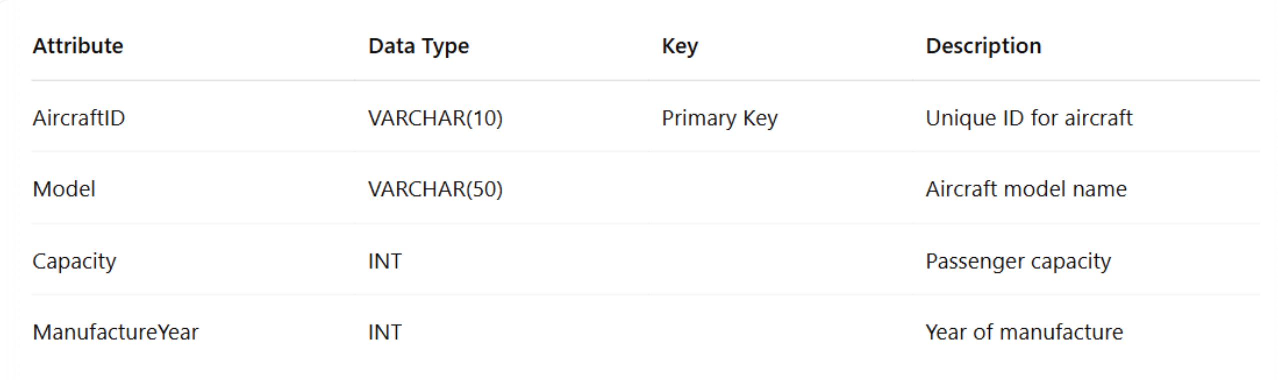
aircraft
- AircraftID VARCHAR(1...
- Model VARCHAR(50)
- Capacity INT
- ManufactureYear INT
- Indexes

pilots
- PilotID VARCHAR(10)
- Name VARCHAR(50)
- LicenseNumber VARCHAR(2...
- ExperienceYears INT
- Indexes

flights
- FlightID VARCHAR(10)
- Airline VARCHAR(50)
- Source VARCHAR(50)
- Destination VARCHAR(5...
- DepartureTime TIME
- ArrivalTime TIME
- Status VARCHAR(20)
- AircraftID VARCHAR(10)
- PilotID VARCHAR(10)

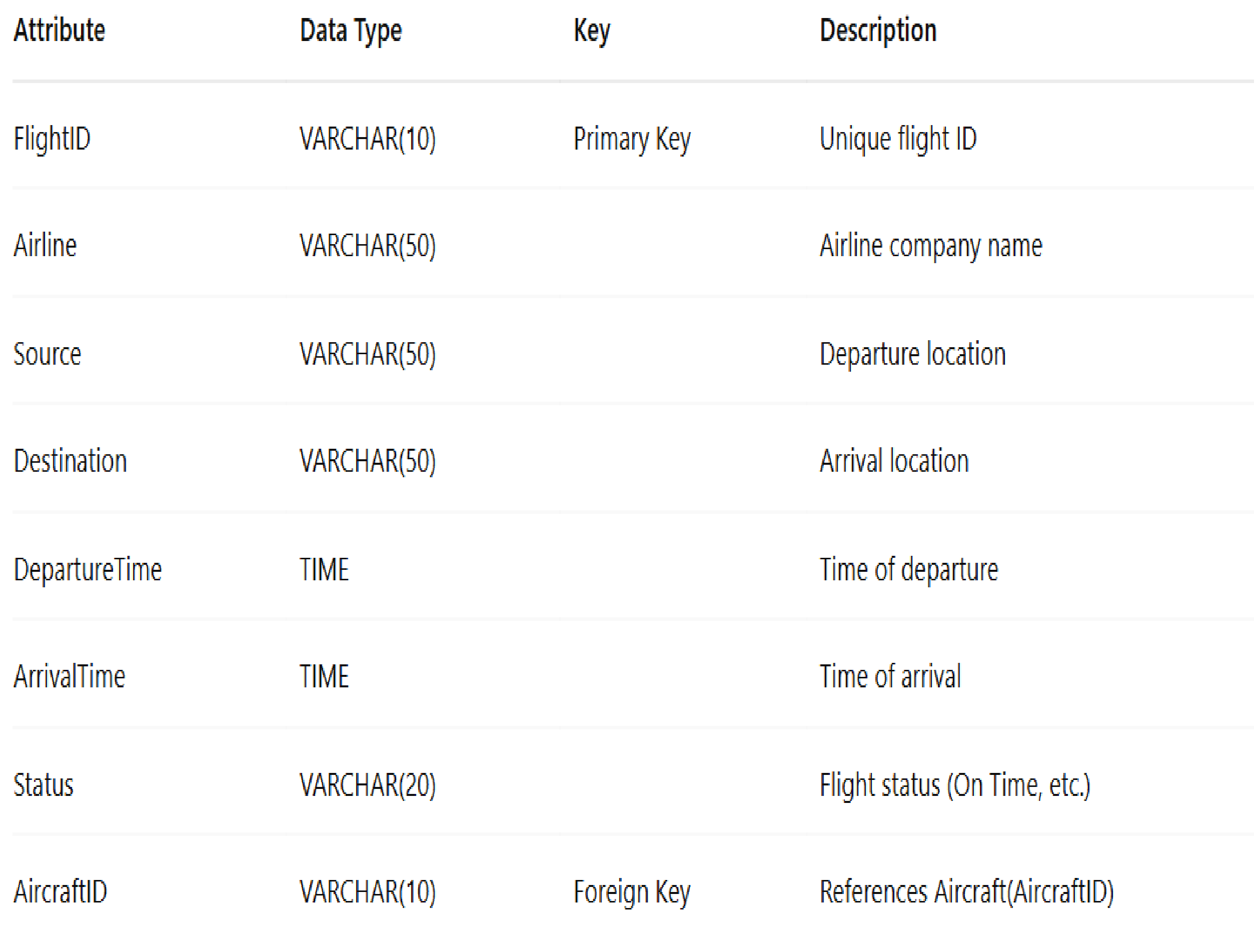
TABULAR FORMAT:

1. Aircraft



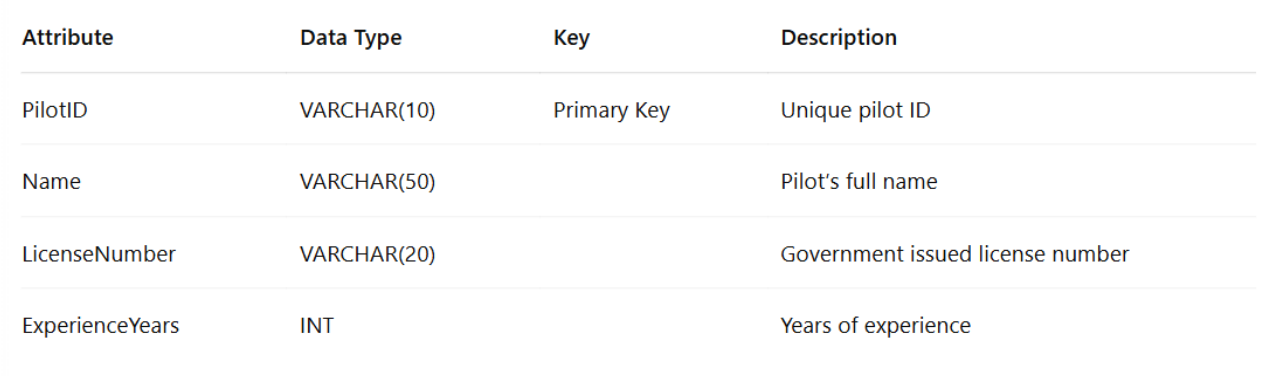
Attribute	Data Type	Key	Description
AircraftID	VARCHAR(10)	Primary Key	Unique ID for aircraft
Model	VARCHAR(50)		Aircraft model name
Capacity	INT		Passenger capacity
ManufactureYear	INT		Year of manufacture

2. Flights



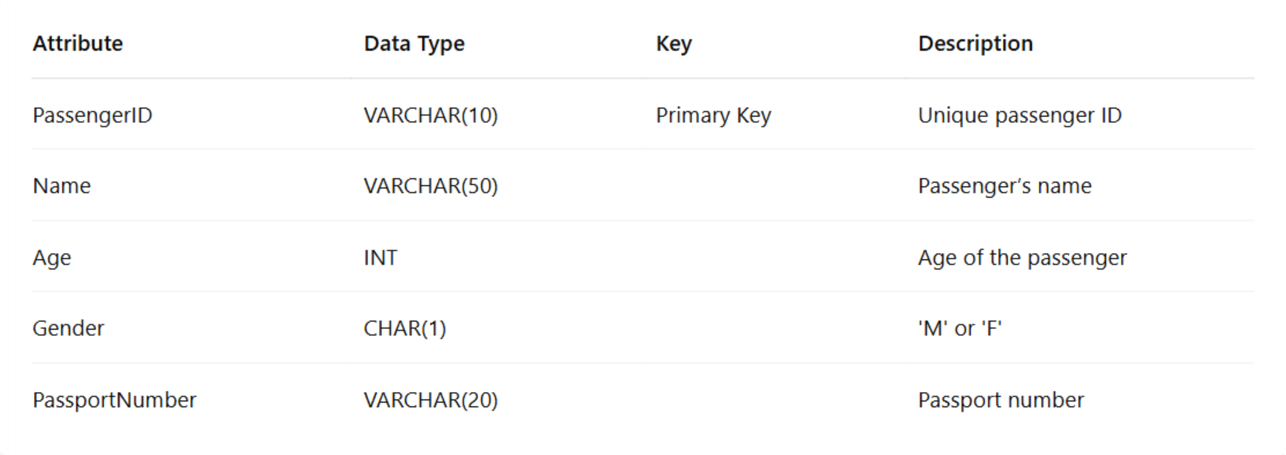
Attribute	Data Type	Key	Description
FlightID	VARCHAR(10)	Primary Key	Unique flight ID
Airline	VARCHAR(50)		Airline company name
Source	VARCHAR(50)		Departure location
Destination	VARCHAR(50)		Arrival location
DepartureTime	TIME		Time of departure
ArrivalTime	TIME		Time of arrival
Status	VARCHAR(20)		Flight status (On Time, etc.)
AircraftID	VARCHAR(10)	Foreign Key	References Aircraft(AircraftID)

3. Pilots



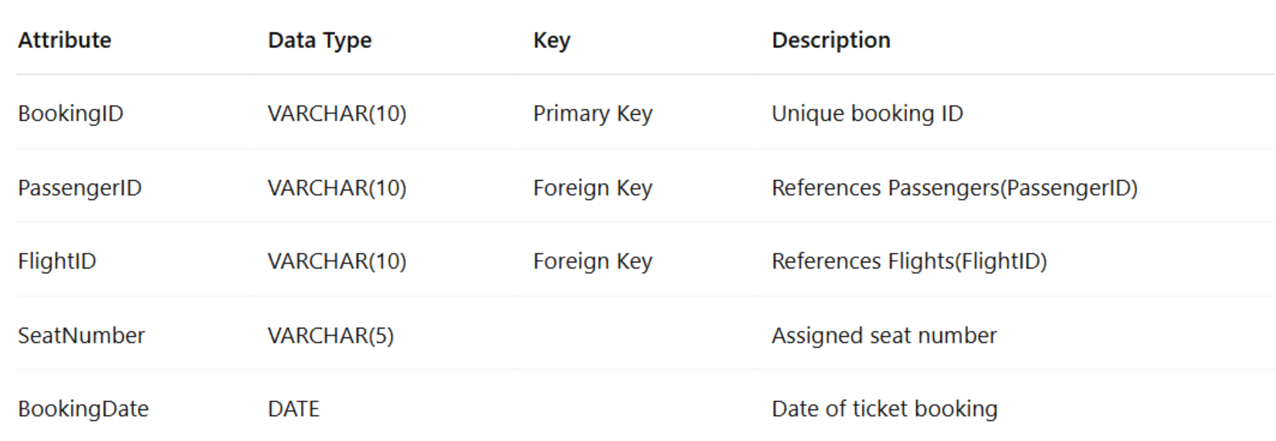
Attribute	Data Type	Key	Description
PilotID	VARCHAR(10)	Primary Key	Unique pilot ID
Name	VARCHAR(50)		Pilot’s full name
LicenseNumber	VARCHAR(20)		Government issued license number
ExperienceYears	INT		Years of experience

4. Passengers



Attribute	Data Type	Key	Description
PassengerID	VARCHAR(10)	Primary Key	Unique passenger ID
Name	VARCHAR(50)		Passenger’s name
Age	INT		Age of the passenger
Gender	CHAR(1)		'M' or 'F'
PassportNumber	VARCHAR(20)		Passport number

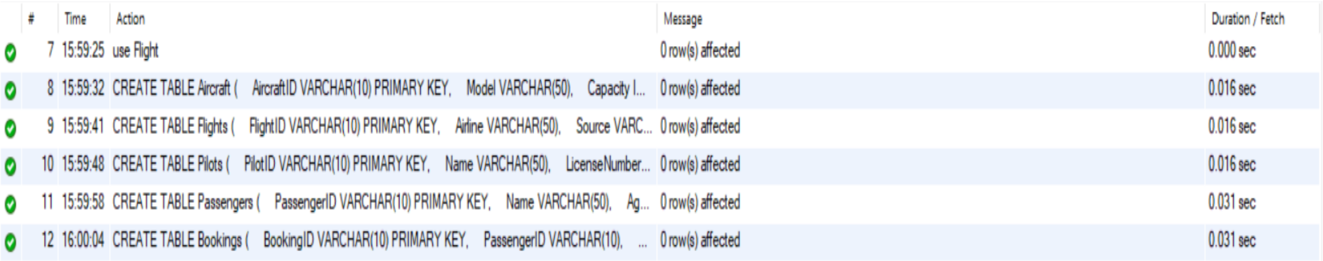
5. Bookings



Attribute	Data Type	Key	Description
BookingID	VARCHAR(10)	Primary Key	Unique booking ID
PassengerID	VARCHAR(10)	Foreign Key	References Passengers(PassengerID)
FlightID	VARCHAR(10)	Foreign Key	References Flights(FlightID)
SeatNumber	VARCHAR(5)		Assigned seat number
BookingDate	DATE		Date of ticket booking

IMPLEMENTATION:

Tables created:

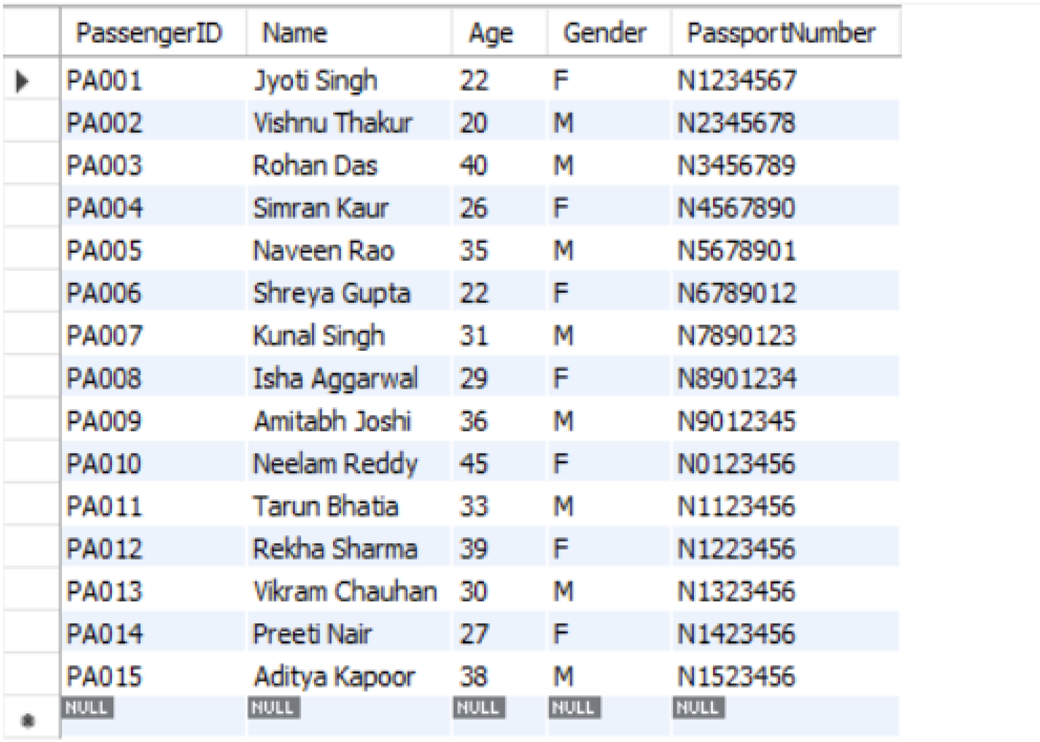


	#	Time	Action	Message	Duration / Fetch
✓	7	15:59:25	use Flight	0 row(s) affected	0.000 sec
✓	8	15:59:32	CREATE TABLE Aircraft (AircraftID VARCHAR(10) PRIMARY KEY, Model VARCHAR(50), Capacity I...	0 row(s) affected	0.016 sec
✓	9	15:59:41	CREATE TABLE Flights (FlightID VARCHAR(10) PRIMARY KEY, Airline VARCHAR(50), Source VARC...	0 row(s) affected	0.016 sec
✓	10	15:59:48	CREATE TABLE Pilots (PilotID VARCHAR(10) PRIMARY KEY, Name VARCHAR(50), LicenseNumber...	0 row(s) affected	0.016 sec
✓	11	15:59:58	CREATE TABLE Passengers (PassengerID VARCHAR(10) PRIMARY KEY, Name VARCHAR(50), Ag...	0 row(s) affected	0.031 sec
✓	12	16:00:04	CREATE TABLE Bookings (BookingID VARCHAR(10) PRIMARY KEY, PassengerID VARCHAR(10), ...	0 row(s) affected	0.031 sec

SQL Queries Performed with Output:

1. Show all passengers

SELECT * FROM Passenger;

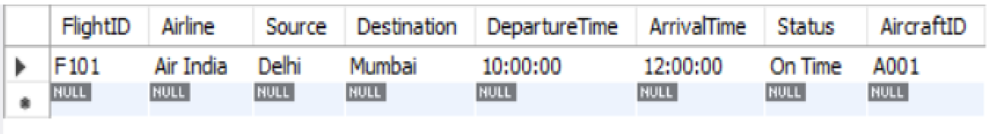


	PassengerID	Name	Age	Gender	PassportNumber
▶	PA001	Jyoti Singh	22	F	N1234567
	PA002	Vishnu Thakur	20	M	N2345678
	PA003	Rohan Das	40	M	N3456789
	PA004	Simran Kaur	26	F	N4567890
	PA005	Naveen Rao	35	M	N5678901
	PA006	Shreya Gupta	22	F	N6789012
	PA007	Kunal Singh	31	M	N7890123
	PA008	Isha Aggarwal	29	F	N8901234
	PA009	Amitabh Joshi	36	M	N9012345
	PA010	Neelam Reddy	45	F	N0123456
	PA011	Tarun Bhatia	33	M	N1123456
	PA012	Rekha Sharma	39	F	N1223456
	PA013	Vikram Chauhan	30	M	N1323456
	PA014	Preeti Nair	27	F	N1423456
	PA015	Aditya Kapoor	38	M	N1523456
*	NULL	NULL	NULL	NULL	NULL

2. Display all flights from Delhi to Mumbai

SELECT * FROM Flights

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

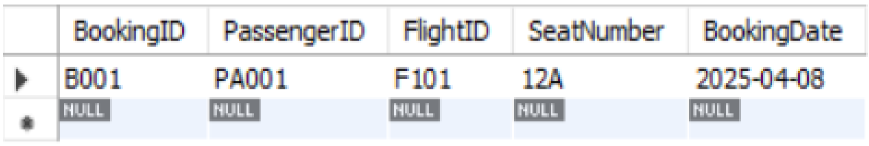


	FlightID	Airline	Source	Destination	DepartureTime	ArrivalTime	Status	AircraftID
▶	F101	Air India	Delhi	Mumbai	10:00:00	12:00:00	On Time	A001
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
SELECT * FROM Bookings

WHERE Passenger_ID = "PA001";
```



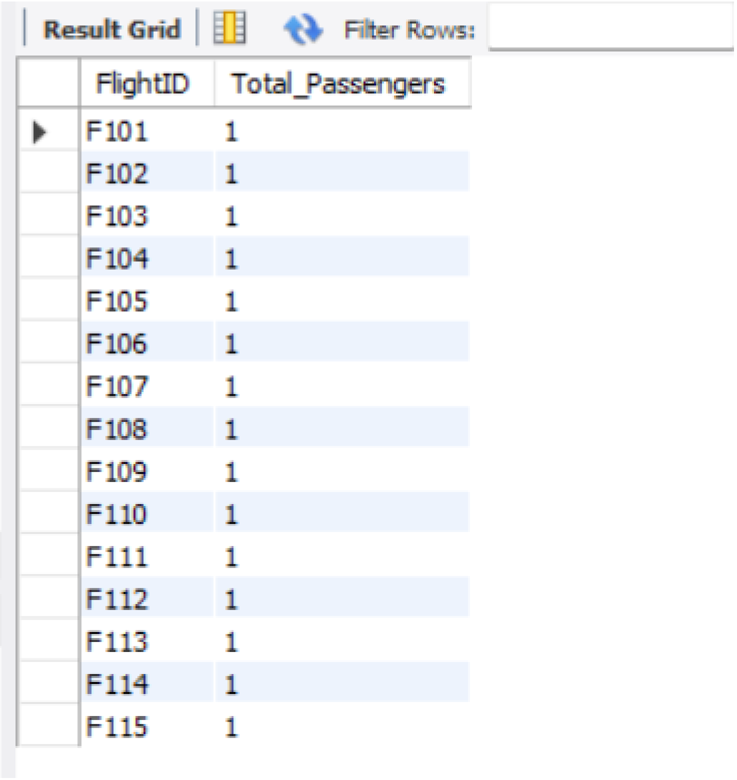
	BookingID	PassengerID	FlightID	SeatNumber	BookingDate
▶	B001	PA001	F101	12A	2025-04-08
*	NULL	NULL	NULL	NULL	NULL

4. Count how many passengers have booked each flight

```
SELECT FlightID, COUNT(*) AS Total_Passengers

FROM Bookings

GROUP BY FlightID;
```



	FlightID	Total_Passengers
▶	F101	1
	F102	1
	F103	1
	F104	1
	F105	1
	F106	1
	F107	1
	F108	1
	F109	1
	F110	1
	F111	1
	F112	1
	F113	1
	F114	1
	F115	1

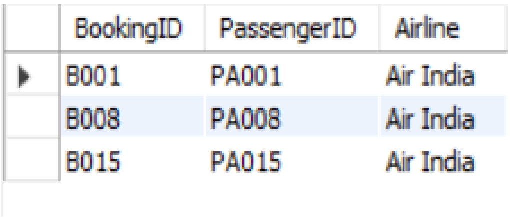
5. Get all bookings for 'Air India' flights

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

FROM Bookings B

JOIN Flights F ON B.FlightID = F.FlightID

WHERE F.Airline = 'Air India';

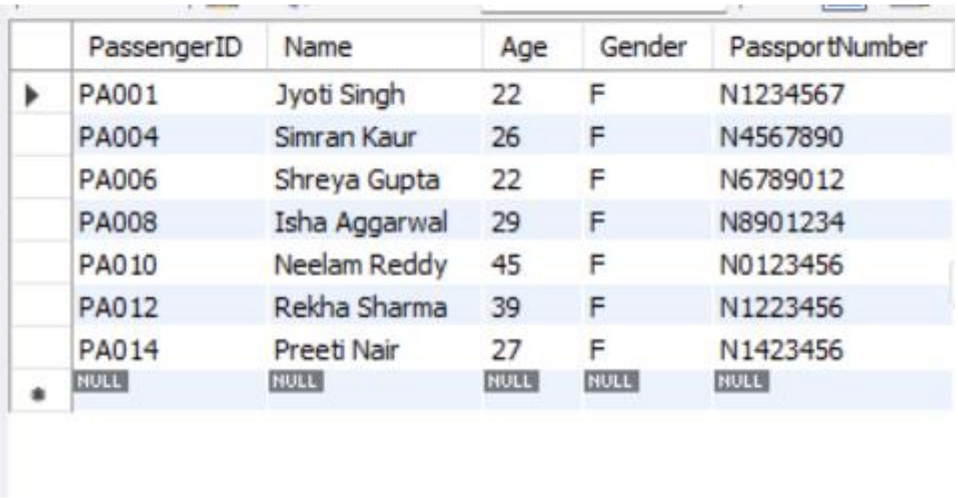


BookingID	PassengerID	Airline
B001	PA001	Air India
B008	PA008	Air India
B015	PA015	Air India

6. Show all female passengers

SELECT * FROM Passengers

WHERE Gender = 'F';

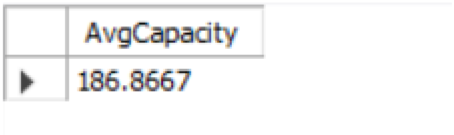


PassengerID	Name	Age	Gender	PassportNumber
PA001	Jyoti Singh	22	F	N1234567
PA004	Simran Kaur	26	F	N4567890
PA006	Shreya Gupta	22	F	N6789012
PA008	Isha Aggarwal	29	F	N8901234
PA010	Neelam Reddy	45	F	N0123456
PA012	Rekha Sharma	39	F	N1223456
PA014	Preeti Nair	27	F	N1423456
NULL	NULL	NULL	NULL	NULL

7. Find the average aircraft capacity

SELECT AVG(Capacity) AS AvgCapacity

FROM Aircraft;



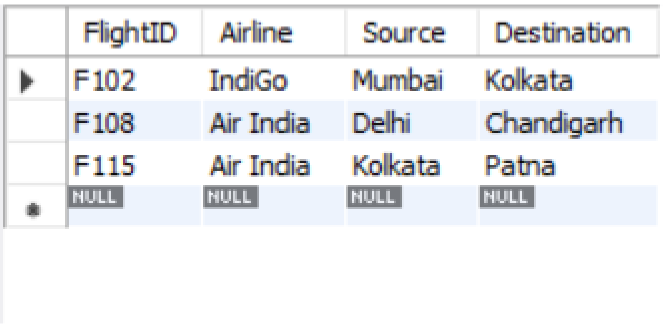
AvgCapacity
186.8667

8. List Flights that are currently 'Delayed'

SELECT FlightID, Airline, Source, Destination

FROM Flights

WHERE Status = 'Delayed';



FlightID	Airline	Source	Destination
F102	IndiGo	Mumbai	Kolkata
F108	Air India	Delhi	Chandigarh
F115	Air India	Kolkata	Patna
NULL	NULL	NULL	NULL

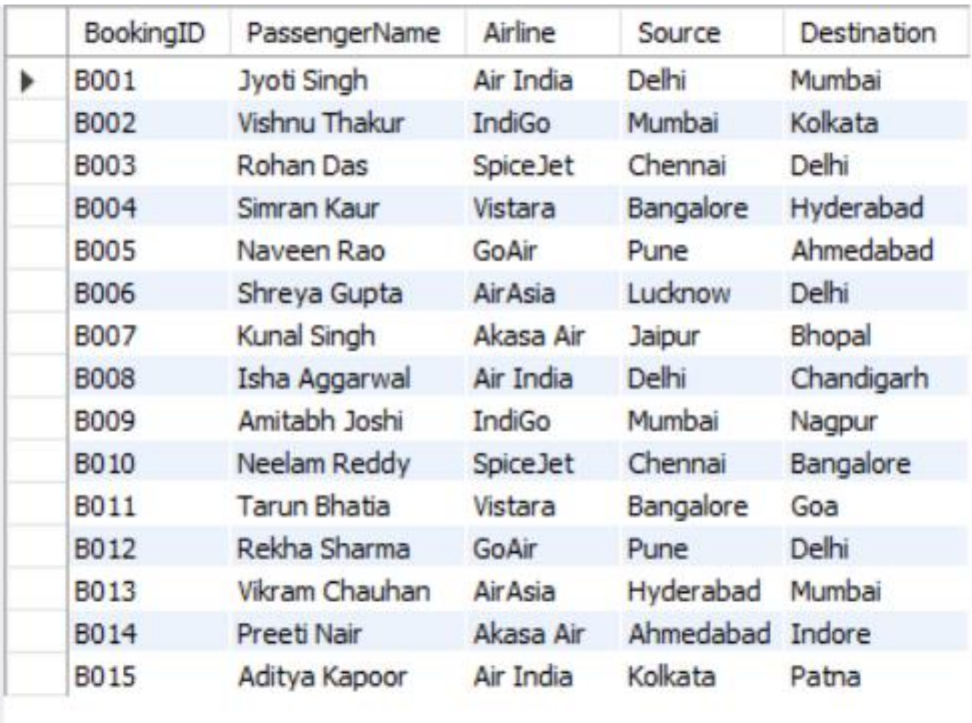
9. Show all bookings with passenger and flight details

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;

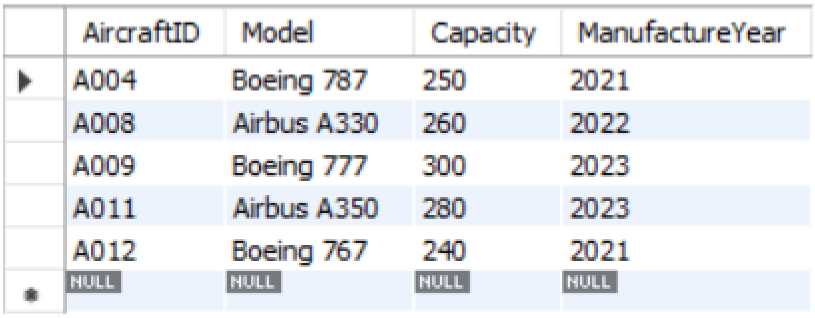


BookingID	PassengerName	Airline	Source	Destination
B001	Jyoti Singh	Air India	Delhi	Mumbai
B002	Vishnu Thakur	IndiGo	Mumbai	Kolkata
B003	Rohan Das	SpiceJet	Chennai	Delhi
B004	Simran Kaur	Vistara	Bangalore	Hyderabad
B005	Naveen Rao	GoAir	Pune	Ahmedabad
B006	Shreya Gupta	AirAsia	Lucknow	Delhi
B007	Kunal Singh	Akasa Air	Jaipur	Bhopal
B008	Isha Aggarwal	Air India	Delhi	Chandigarh
B009	Amitabh Joshi	IndiGo	Mumbai	Nagpur
B010	Neelam Reddy	SpiceJet	Chennai	Bangalore
B011	Tarun Bhatia	Vistara	Bangalore	Goa
B012	Rekha Sharma	GoAir	Pune	Delhi
B013	Vikram Chauhan	AirAsia	Hyderabad	Mumbai
B014	Preeti Nair	Akasa Air	Ahmedabad	Indore
B015	Aditya Kapoor	Air India	Kolkata	Patna

10. Get the details of aircraft manufactured after 2020

SELECT * FROM Aircraft

WHERE ManufactureYear > 2020;



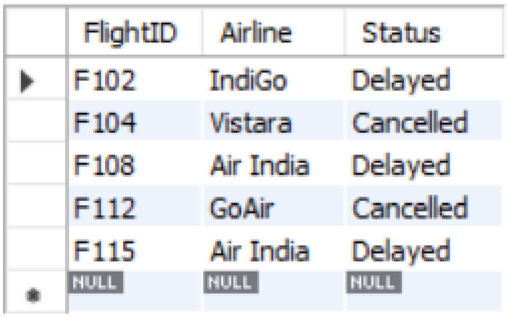
AircraftID	Model	Capacity	ManufactureYear
A004	Boeing 787	250	2021
A008	Airbus A330	260	2022
A009	Boeing 777	300	2023
A011	Airbus A350	280	2023
A012	Boeing 767	240	2021
NULL	NULL	NULL	NULL

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

SELECT FlightID, Airline, Status

FROM Flights

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

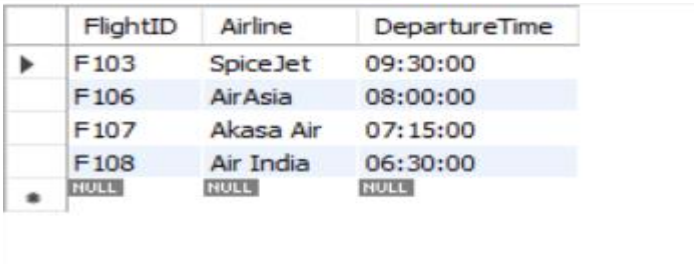


FlightID	Airline	Status
F102	IndiGo	Delayed
F104	Vistara	Cancelled
F108	Air India	Delayed
F112	GoAir	Cancelled
F115	Air India	Delayed
NULL	NULL	NULL

12. Find flights that depart before 10:00 AM

SELECT FlightID, Airline, DepartureTime

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



FlightID	Airline	DepartureTime
F103	SpiceJet	09:30:00
F106	AirAsia	08:00:00
F107	Akasa Air	07:15:00
F108	Air India	06:30:00
NULL	NULL	NULL

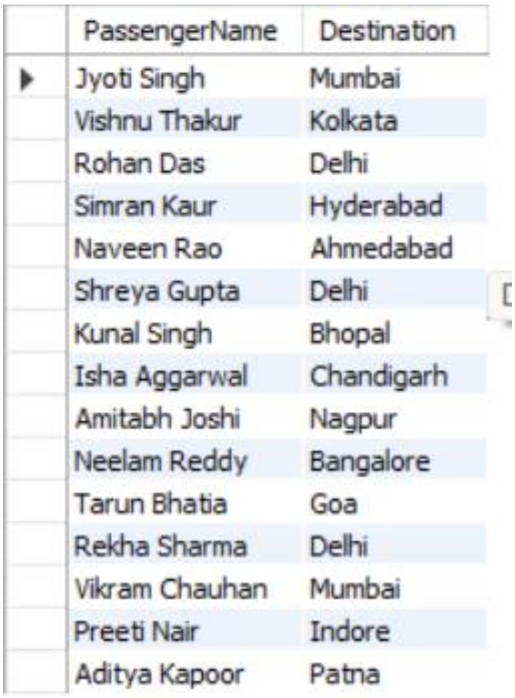
13. Show passenger names with their booked flight destinations

```
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;
```



PassengerName	Destination
Jyoti Singh	Mumbai
Vishnu Thakur	Kolkata
Rohan Das	Delhi
Simran Kaur	Hyderabad
Naveen Rao	Ahmedabad
Shreya Gupta	Delhi
Kunal Singh	Bhopal
Isha Aggarwal	Chandigarh
Amitabh Joshi	Nagpur
Neelam Reddy	Bangalore
Tarun Bhatia	Goa
Rekha Sharma	Delhi
Vikram Chauhan	Mumbai
Preeti Nair	Indore
Aditya Kapoor	Patna

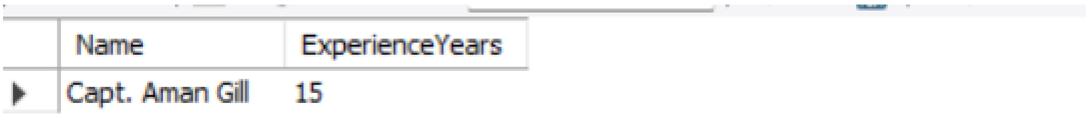
14. Show the name of the pilot with the highest experience

```
SELECT Name, ExperienceYears

FROM Pilots

ORDER BY ExperienceYears DESC

LIMIT 1;
```



Name	ExperienceYears
Capt. Aman Gill	15

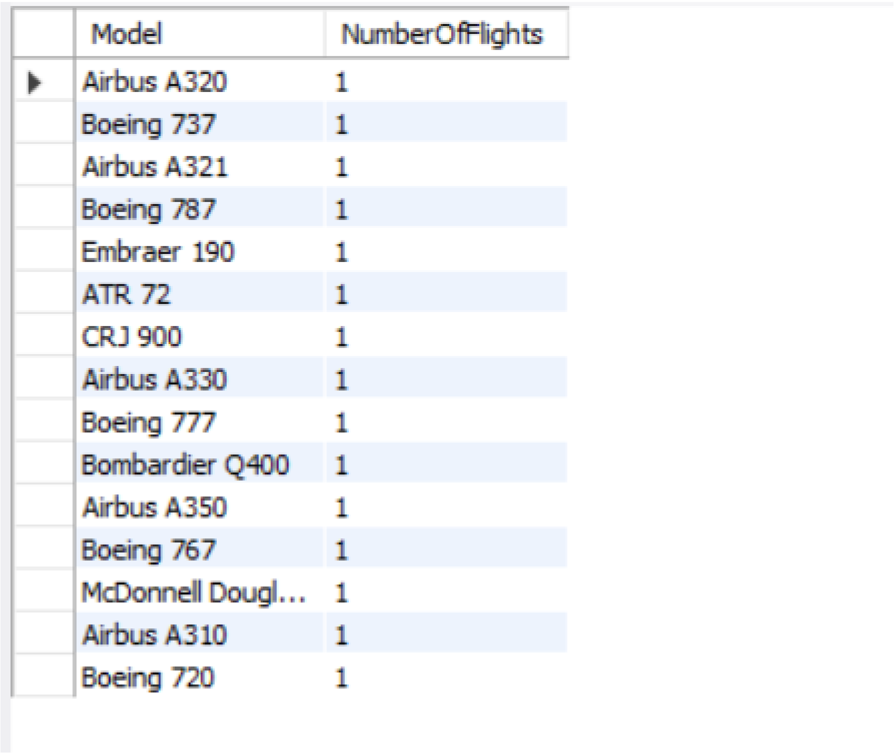
15. Display the number of flights associated with each aircraft model

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

FROM Aircraft A

JOIN Flights F ON A.AircraftID = F.AircraftID

GROUP BY A.Model;
```



Model	NumberOfFlights
Airbus A320	1
Boeing 737	1
Airbus A321	1
Boeing 787	1
Embraer 190	1
ATR 72	1
CRJ 900	1
Airbus A330	1
Boeing 777	1
Bombardier Q400	1
Airbus A350	1
Boeing 767	1
McDonnell Dougl...	1
Airbus A310	1
Boeing 720	1

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

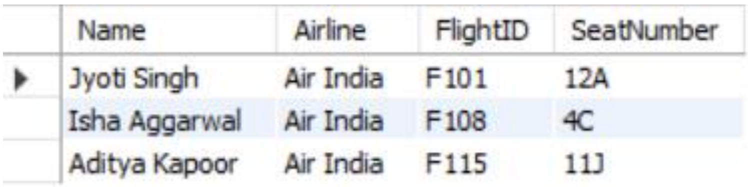
```
SELECT P.Name, F.Airline, F.FlightID, B.SeatNumber

FROM Passengers P

JOIN Bookings B ON P.PassengerID = B.PassengerID

JOIN Flights F ON B.FlightID = F.FlightID

WHERE F.Airline = 'Air India';
```

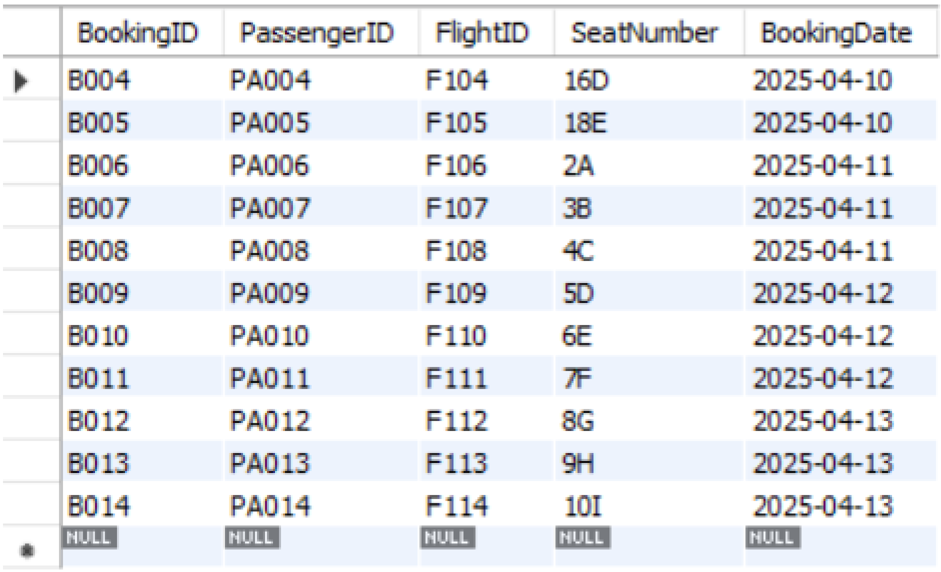


Name	Airline	FlightID	SeatNumber
Jyoti Singh	Air India	F101	12A
Isha Aggarwal	Air India	F108	4C
Aditya Kapoor	Air India	F115	11J

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';
```



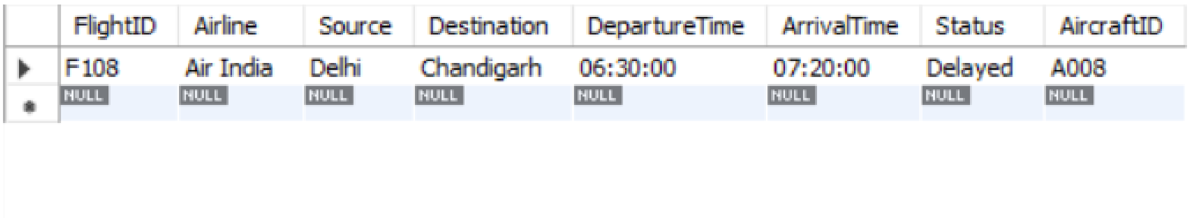
BookingID	PassengerID	FlightID	SeatNumber	BookingDate
B004	PA004	F104	16D	2025-04-10
B005	PA005	F105	18E	2025-04-10
B006	PA006	F106	2A	2025-04-11
B007	PA007	F107	3B	2025-04-11
B008	PA008	F108	4C	2025-04-11
B009	PA009	F109	5D	2025-04-12
B010	PA010	F110	6E	2025-04-12
B011	PA011	F111	7F	2025-04-12
B012	PA012	F112	8G	2025-04-13
B013	PA013	F113	9H	2025-04-13
B014	PA014	F114	10I	2025-04-13
NULL	NULL	NULL	NULL	NULL

18. Find the flight that arrives the earliest

```
SELECT * FROM Flights

ORDER BY ArrivalTime ASC

LIMIT 1;
```

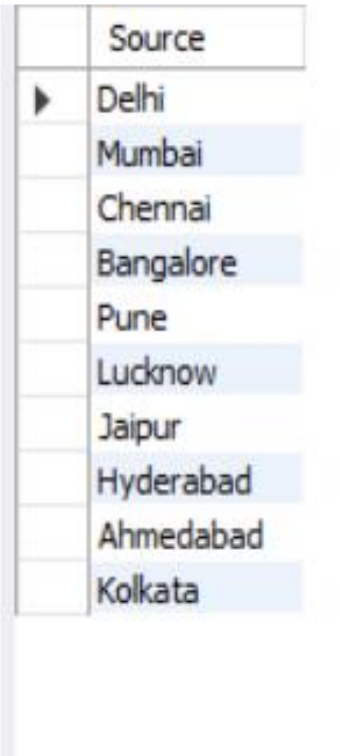


FlightID	Airline	Source	Destination	DepartureTime	ArrivalTime	Status	AircraftID
F108	Air India	Delhi	Chandigarh	06:30:00	07:20:00	Delayed	A008
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

19. List all distinct flight sources

```
SELECT DISTINCT Source

FROM Flights;
```



Source
Delhi
Mumbai
Chennai
Bangalore
Pune
Lucknow
Jaipur
Hyderabad
Ahmedabad
Kolkata

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