AeroManageX

# Database Management System

# MSCS 542L-256

# Aerotech Titans



Marist College

School of Computer Science and Mathematics

Submitted To:

Dr. Reza Sadeghi

September 13, 2023

Table of Contents

[Project Report of AeroManageX 3](#_Toc147948131)

[1. Team Name 3](#_Toc147948132)

[2. Aerotech Titans 3](#_Toc147948133)

[Team Members 3](#_Toc147948134)

[Description of Team Members 3](#_Toc147948135)

[3. AeroManageX Objective 4](#_Toc147948136)

[4. Review Related Works 4](#_Toc147948137)

[5. The Merits of Our Project 5](#_Toc147948138)

[6. GitHub Repository Address 6](#_Toc147948139)

[7. External ER Diagrams 6](#_Toc147948140)

[Manager Description 6](#_Toc147948141)

[Manager Model 7](#_Toc147948142)

[Booking Agent Description 7](#_Toc147948143)

[Booking Agent Model 8](#_Toc147948144)

[8. Entity Relationship Diagram (ER Diagram) 8](#_Toc147948145)

[Description 8](#_Toc147948146)

[Diagram 9](#_Toc147948147)

[9. Enhanced Entity Relationship Diagram (EER) 9](#_Toc147948148)

[Description 9](#_Toc147948149)

[Diagram 11](#_Toc147948150)

[10. References 12](#_Toc147948151)

# Project Report of AeroManageX

## Team Name

## Aerotech Titans

### Team Members

1. Bashir Dahir [bashir.dahir1@marist.edu](mailto:bashir.dahir1@marist.edu) (Team Head)
2. Nihar Lodaya [nihar.lodaya1@marist.edu](mailto:nihar.lodaya1@marist.edu) (Team Member)
3. Marguerite McGahay [marguerite.mcgahay@marist.edu](mailto:marguerite.mcgahay@marist.edu) (Team Member)

### Description of Team Members

#### Bashir Dahir

I'm a Computer Science student at Marist College, Beacon, New York, in my fifth year, with just two semesters to go before graduation. My academic journey has sharpened my skills in programming and data structures, but my true passion lies in database management systems. Currently, I'm eagerly gearing up for a project focused on airline management systems, where I plan to apply my expertise in database management to create efficient and robust solutions for the aviation industry. Beyond academics, I enjoy problem-solving and community engagement.

#### Nihar Lodaya

Hey there, I'm Nihar Lodaya, and I come from India. I've got a bachelor's degree in computer science from back home, and right now, I'm in the middle of my master's program in Computer Science at Marist College. I'm stoked about working with this bunch of awesome folks on our current project. What really got me excited about my teammates Bashir & Marguerite is how dedicated they are to making this project a success.

#### Marguerite McGahay

I grew up in Poughkeepsie and then attended the University of Delaware, where I graduated in 2021 with a BS in Mathematics and a Minor in Computer Science. While I was there, I was a TA for multiple computer science classes and was also on the Women’s Rowing Team! I currently work at Marist as the Assistant Women’s Rowing Coach. This is my first semester in the Computer Science – Software Development program, so I don’t know many people, but I was excited when Bashir and Nihar extended an invitation for me to be a member of this group. We selected our team head, Bashir, since it was his initial idea to pursue this project, but I truly believe each member of this team has the capability to be able to step into that role if asked of us.

## AeroManageX Objective

The primary objective of the AeroManageX project is to elevate your airline’s existing database infrastructure to be able to better compete with the world’s top airlines. Our company will help you condense the mass amount of information needed to be able to have thousands of planes arrive and depart each day. Our system operates to help your airline in multiple ways, including but not limited to managing flight bookings, optimizing the cost of flights, making sure planes are in the necessary airports based on different flights’ place of departure and arrival, and efficiently assigning pilots to flights that make sense for your company and their schedules.

## Review Related Works

There are many competitions for our company including Ramco Aviation, Airline Suite, and AvPro. Each of these companies has positive and negative aspects. To begin with, Ramco Aviation published on their website that they provide the best aviation software for the following three reasons: “Proven technology champions, usability focusses and in memory planning and optimization” (2). However, Ramco Aviation critics emphasize on their “Security and data control issues and difficulties of data migration” (*2*). Second, Airline Suite promotes that their system is easy to use, scalable and affordable. On the other hand, however, Airline Suite is notorious for having “no history of previous or editions and no import options for Microsoft Excel or file” (1). Finally, AvPro is famous for their “reliability reporting, budget forecasting, and inventory management” (4). On the contrary, AvPro lacks the availability of “instruction manuals and system stability” as there are system glitches (3).

## The Merits of Our Project

Our company will provide airlines with more features than any other competitive company. We provide airlines with the ability to see and organize information which can be categorized into three principal operations:

Our Land Operations:

* Checked baggage: Each passenger has a bag that must be on their flight.
* Staff management: Such as gate-workers, baggage handlers, custodial staff, etc.
* Times of arrival and departure of flights
* Fleet inventory

Air Operations:

* Flight plan: Including non-stop flights and layovers.
* Assigning pilots and airline staff.

Billing Operations:

* Bookings: Secure database for credit card numbers.
* Cost of seats
* Checked luggage (including oversized): If a passenger checks a bag that is over 50lbs, they must pay an extra fee on top of the base cost.

An airline should choose AeroManageX to manage their data because we are security focused and can benefit any size airline, from companies with just a small fleet of planes to global “Aero-Titans"!

## GitHub Repository Address

Here is our GitHub repository’s URL.

<https://github.com/bashirad/MSCS-542L-256_AeroManageX_Aerotech-Titans.git>

## External ER Diagrams

### Manager Description

In the External Entity Relationship Diagram for the Manager Model in our AeroManageX, the primary entities include Flight, Origin Airport, Destination Airport, Pilot, Flight Attendant, Plane, and Airline. Flight is the central entity, while Origin Airport and Destination Airport are directly connected to Flight, representing the departure and arrival locations. Pilots are associated with specific flights as they operate them, and Flight Attendants work on board these flights. The Plane entity represents the aircraft used for the flights, and Airline serves as a parent entity encompassing various flights operated by the airline. This diagram illustrates how these entities are interrelated, providing a clear overview of the airline's operations.

### Manager Model

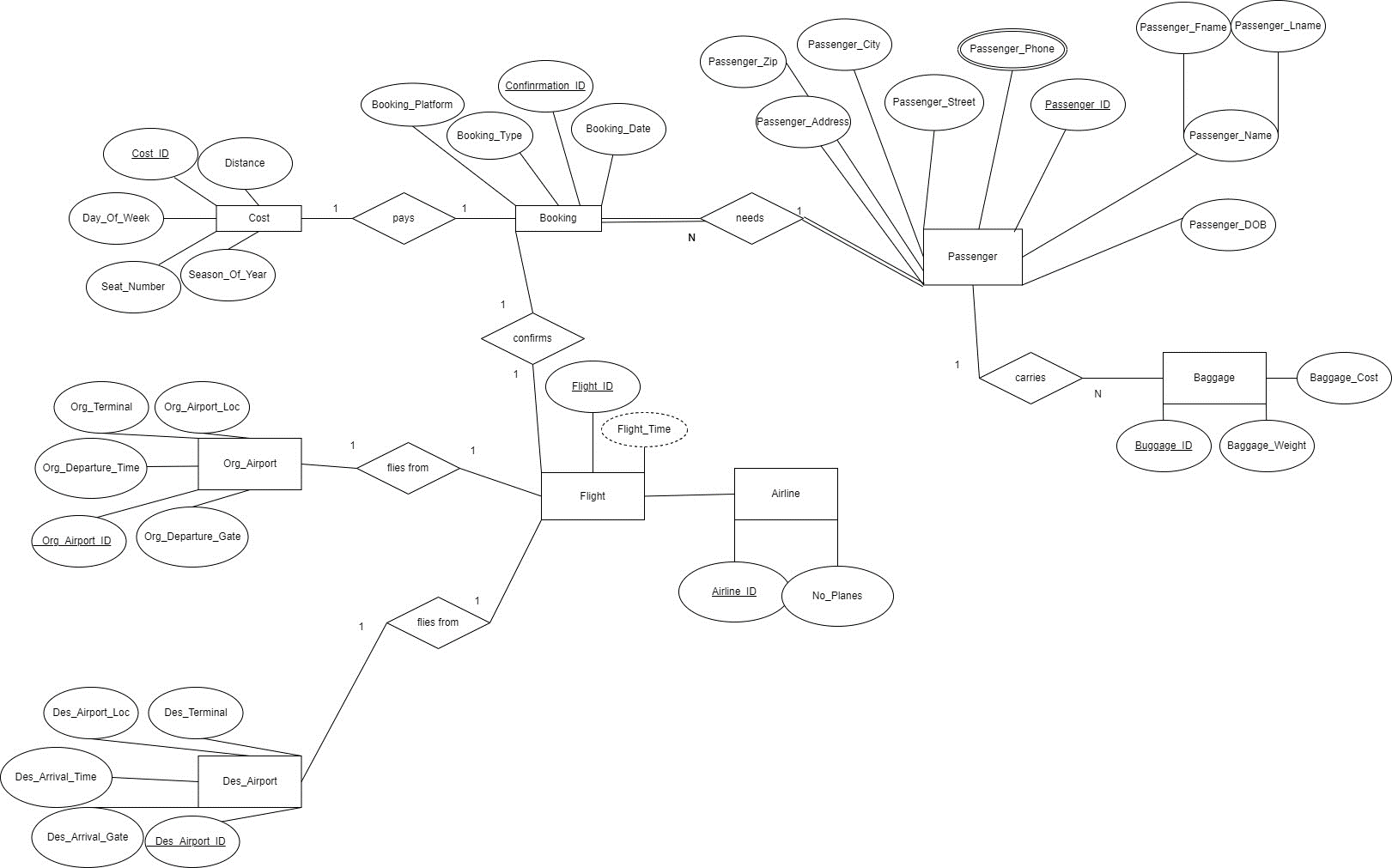


Figure 1

### Booking Agent Description

In the External Entity Relationship Diagram for the Booking Agent Model in the AeroManageX, a web of interconnected entities and relationships emerges. Cost is linked to Booking, reflecting the financial transaction between booking agents and reservations. Booking necessitates Passenger, showing the connection between bookings and the passengers they are made for, while Passenger also carries Baggage, signifying the belongings associated with passengers. Bookings confirm Flights, illustrating the reservation process, and Flights are tied to both Origin and Destination Airports, representing the departure and arrival points of the journeys.

### Booking Agent Model

Figure 2

## Entity Relationship Diagram (ER Diagram)

### Description

Creating this project involves a thoughtful selection of entities, attributes, relationships, participations, and cardinalities to represent a simplified domain. In this case, the chosen entities include "Cost," "Baggage," "Booking," "Plane," "Flight," "Flight\_Attendant," "Org\_Airport," "Des\_Airport," "Pilot," "Passenger," and "Airline." Each of these entities represents key elements in the domain of Airline travel. Attributes are then identified for each entity to capture relevant information; for instance, "Passenger" have attributes like Passenger\_Fname, Passenger\_Lname, Passenger\_Address, etc. While "Cost" has attributes like Confirmation\_ID, Distance, Season\_Of\_Year, etc. Relationships are established between entities to represent how they are connected, such as the relationship between "Booking" and "Passenger" to show that a passenger can make a booking.

### DiagramA diagram of a company Description automatically generated

Figure 3

## Enhanced Entity Relationship Diagram (EER)

### Description

|  |  |  |  |
| --- | --- | --- | --- |
| # | Entity | Key | Relationships |
| 1 | Flight | Flight\_ID | Each flight has one origin airport, one destination airport, multiple bookings for their passengers, as well as a one plane, one pilot, and multiple flight attendants |
| 2 | Plane | Plane\_ID | Each flight must have a plane, which owned by an airline |
| 3 | Airline | Airline\_ID | Each airline has multiple planes in its fleet |
| 4 | Cost | Cost\_ID | Cost is related to Booking via a 1-1 relationship. |
| 5 | Booking | Confirmation\_ID | Each booking is related to a passenger, which results in a cost of booking. When a passenger books, they are then connected to a flight |
| 6 | Passenger | Passenger\_ID | Each passenger can have one or more bags |
| 7 | Baggage | Baggage\_ID | Each bag is related to one passenger |
| 8 | Org\_Airport | Org\_Airport\_ID | Each flight must depart from one origin airport |
| 9 | Des\_Airport | Des\_Airport\_ID | Each flight must arrive at one destination airport. |
| 10 | Pilot | Pilot\_ID | Each flight must have one pilot. A pilot can have multiple flights in one day |
| 11 | Flight Attendant | Flight\_Att\_ID | Each flight can have multiple flight attendants and a flight attendant can have multiple flights in one day |

Figure 4

### Diagram

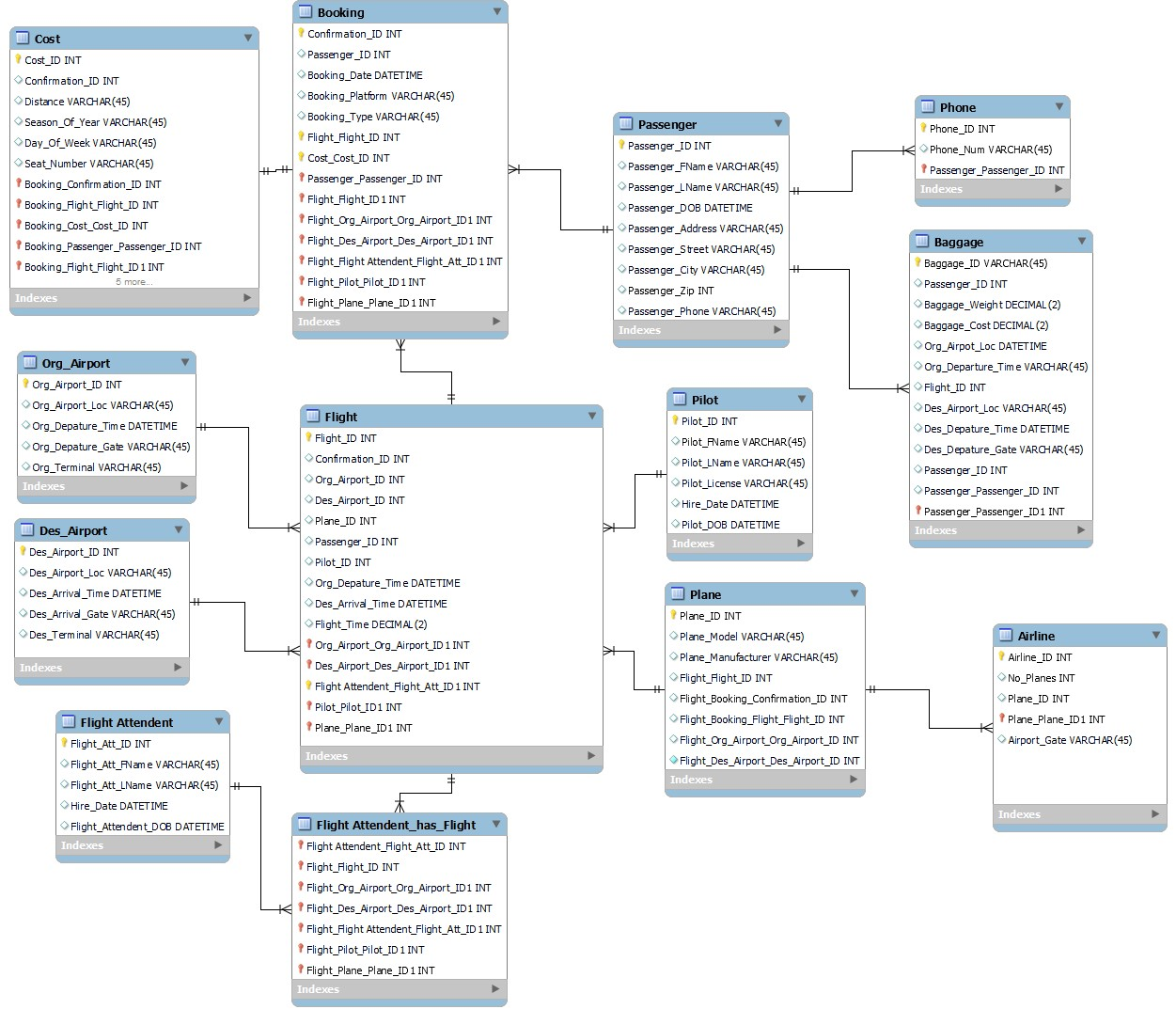


Figure 5

## Database Development

DROP DATABASE IF EXISTS mydb;

CREATE DATABASE IF NOT EXISTS mydb DEFAULT CHARACTER SET utf8;

USE mydb;

-- GET RID OF ALL FOREIGN KEYS EXCEPT BOOKING'S CONFIRMATION\_ID

-- Table mydb.Cost

DROP TABLE IF EXISTS mydb.Cost;

CREATE TABLE IF NOT EXISTS mydb.Cost (

Cost\_ID INT NOT NULL,

Amount INT NOT NULL,

Distance VARCHAR(45) NULL,

Season\_Of\_Year VARCHAR(45) NULL,

Day\_Of\_Week VARCHAR(45) NULL,

Seat\_Number VARCHAR(45) NULL,

PRIMARY KEY (Cost\_ID)

);

### **Table: Cost**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Cost\_ID** | INT | This attribute serves as the primary key for the "Cost" table, uniquely identifying each cost record. It is an integer (INT) data type. |
| **Amount** | INT | The "Amount" attribute represents the cost or price associated with a specific item or service. It is of integer (INT) data type and is required, meaning it cannot be null. |
| **Distance** | VARCHAR | This attribute stores textual information, likely related to the distance associated with the cost. It is of VARCHAR(45) data type, allowing up to 45 characters, and it is not marked as required (nullable). |
| **Season\_Of\_Year** | VARCHAR | The "Season\_Of\_Year" attribute stores information related to the season of the year associated with the cost. It is of VARCHAR(45) data type, allowing up to 45 characters, and it is not marked as required (nullable). |
| **Day\_Of\_Week** | VARCHAR | This attribute represents information related to the day of the week associated with the cost. It is of VARCHAR(45) data type, allowing up to 45 characters, and it is not marked as required (nullable). |
| **Seat\_Number** | VARCHAR | The "Seat\_Number" attribute stores information related to seat numbers associated with the cost. It is of VARCHAR(45) data type, allowing up to 45 characters, and it is not marked as required (nullable). |

-- Table mydb.Airline

DROP TABLE IF EXISTS mydb.Airline;

CREATE TABLE IF NOT EXISTS mydb.Airline (

Airline\_ID INT NOT NULL,

No\_Planes INT NULL,

Plane\_ID INT NULL,

Airport\_Gate VARCHAR(45) NULL,

Num\_Employees INT,

PRIMARY KEY (Airline\_ID),

);

### **Table: Airline**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Airline\_ID** | INT | This attribute serves as the primary key for the "Airline" table, uniquely identifying each airline record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **No\_Planes** | INT | The "No\_Planes" attribute represents the number of planes associated with the airline. It is of integer (INT) data type and is not marked as required (nullable). |
| **Plane\_ID** | INT | This attribute is used to store the ID of a specific plane associated with the airline. It is of integer (INT) data type and is not marked as required (nullable). |
| **Airport\_Gate** | VARCHAR | The "Airport\_Gate" attribute stores textual information related to the airport gate used by the airline. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Num\_Employees** | INT | This attribute represents the number of employees working for the airline. It is of integer (INT) data type and is required, meaning it cannot be null. |

-- Table mydb.Passenger

DROP TABLE IF EXISTS mydb.Passenger;

CREATE TABLE IF NOT EXISTS mydb.Passenger (

Passenger\_ID INT NOT NULL,

Passenger\_FName VARCHAR(45) NULL,

Passenger\_LName VARCHAR(45) NULL,

Passenger\_DOB DATETIME NULL,

Passenger\_Address VARCHAR(45) NULL,

Passenger\_Street VARCHAR(45) NULL,

Passenger\_City VARCHAR(45) NULL,

Passenger\_Zip INT NULL,

Passenger\_Phone VARCHAR(45) NULL,

PRIMARY KEY (Passenger\_ID)

);

### **Table: Passenger**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Passenger\_ID** | INT | This attribute serves as the primary key for the "Passenger" table, uniquely identifying each passenger record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **Passenger\_FName** | VARCHAR | The "Passenger\_FName" attribute represents the first name of the passenger. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Passenger\_LName** | VARCHAR | This attribute is used to store the last name of the passenger. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Passenger\_DOB** | DATETIME | The "Passenger\_DOB" attribute is used to store the date of birth of the passenger. It is of DATETIME data type and is not marked as required (nullable). |
| **Passenger\_Address** | VARCHAR | This attribute stores the primary address of the passenger. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Passenger\_Street** | VARCHAR | The "Passenger\_Street" attribute represents the street information associated with the passenger's address. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Passenger\_City** | VARCHAR | This attribute is used to store the city information associated with the passenger's address. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Passenger\_Zip** | INT | The "Passenger\_Zip" attribute is used to store the zip code associated with the passenger's address. It is of integer (INT) data type and is not marked as required (nullable). |
| **Passenger\_Phone** | VARCHAR | This attribute represents the phone number of the passenger. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |

-- Table mydb.Pilot

DROP TABLE IF EXISTS mydb.Pilot;

CREATE TABLE IF NOT EXISTS mydb.Pilot (

Pilot\_ID INT NOT NULL,

Pilot\_FName VARCHAR(45) NULL,

Pilot\_LName VARCHAR(45) NULL,

Pilot\_License VARCHAR(45) NULL,

Hire\_Date DATETIME NULL,

Pilot\_DOB DATETIME NULL,

PRIMARY KEY (Pilot\_ID)

);

### **Table: Pilot**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Pilot\_ID** | INT | This attribute serves as the primary key for the "Pilot" table, uniquely identifying each pilot record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **Pilot\_FName** | VARCHAR | The "Pilot\_FName" attribute represents the first name of the pilot. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Pilot\_LName** | VARCHAR | This attribute is used to store the last name of the pilot. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Pilot\_License** | VARCHAR | The "Pilot\_License" attribute represents the license information associated with the pilot. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Hire\_Date** | DATETIME | This attribute is used to store the hire date of the pilot. It is of DATETIME data type and is not marked as required (nullable). |
| **Pilot\_DOB** | DATETIME | The "Pilot\_DOB" attribute is used to store the date of birth of the pilot. It is of DATETIME data type and is not marked as required (nullable). |

-- Table mydb.Des\_Airport

DROP TABLE IF EXISTS mydb.Des\_Airport;

CREATE TABLE IF NOT EXISTS mydb.Des\_Airport (

Des\_Airport\_ID INT NOT NULL,

Des\_Airport\_Loc VARCHAR(45) NULL,

Des\_Arrival\_Time DATETIME NULL,

Des\_Arrival\_Gate VARCHAR(45) NULL,

Des\_Terminal VARCHAR(45) NULL,

PRIMARY KEY (Des\_Airport\_ID)

);

### **Table: Des\_Airport**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Des\_Airport\_ID** | INT | This attribute serves as the primary key for the "Des\_Airport" table, uniquely identifying each destination airport record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **Des\_Airport\_Loc** | VARCHAR | The "Des\_Airport\_Loc" attribute represents the location of the destination airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Des\_Arrival\_Time** | DATETIME | This attribute is used to store the arrival time at the destination airport. It is of DATETIME data type and is not marked as required (nullable). |
| **Des\_Arrival\_Gate** | VARCHAR | The "Des\_Arrival\_Gate" attribute represents the arrival gate at the destination airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Des\_Terminal** | VARCHAR | This attribute is used to store the terminal information at the destination airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |

-- Table mydb.Org\_Airport

DROP TABLE IF EXISTS mydb.Org\_Airport;

CREATE TABLE IF NOT EXISTS mydb.Org\_Airport (

Org\_Airport\_ID INT NOT NULL,

Org\_Airport\_Loc VARCHAR(45) NULL,

Org\_Departure\_Time DATETIME NULL,

Org\_Departure\_Gate VARCHAR(45) NULL,

Org\_Terminal VARCHAR(45) NULL,

PRIMARY KEY (Org\_Airport\_ID)

);

### **Table: Org\_Airport**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Org\_Airport\_ID** | INT | This attribute serves as the primary key for the "Org\_Airport" table, uniquely identifying each originating airport record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **Org\_Airport\_Loc** | VARCHAR | The "Org\_Airport\_Loc" attribute represents the location of the originating airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Org\_Departure\_Time** | DATETIME | This attribute is used to store the departure time from the originating airport. It is of DATETIME data type and is not marked as required (nullable). |
| **Org\_Departure\_Gate** | VARCHAR | The "Org\_Departure\_Gate" attribute represents the departure gate at the originating airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Org\_Terminal** | VARCHAR | This attribute is used to store the terminal information at the originating airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |

-- Table mydb.Flight Attendent

DROP TABLE IF EXISTS mydb.Flight\_Attendent;

CREATE TABLE IF NOT EXISTS mydb.Flight\_Attendent (

Flight\_Att\_ID INT NOT NULL,

Flight\_Att\_FName VARCHAR(45) NULL,

Flight\_Att\_LName VARCHAR(45) NULL,

Hire\_Date DATETIME NULL,

Flight\_Attendent\_DOB DATETIME NULL,

PRIMARY KEY (Flight\_Att\_ID)

);

### **Table: Flight\_Attendent**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Flight\_Att\_ID** | INT | This attribute serves as the primary key for the "Flight\_Attendent" table, uniquely identifying each flight attendant's record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **Flight\_Att\_FName** | VARCHAR | The "Flight\_Att\_FName" attribute represents the first name of the flight attendant. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Flight\_Att\_LName** | VARCHAR | This attribute represents the last name of the flight attendant. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Hire\_Date** | DATETIME | The "Hire\_Date" attribute is used to store the date when the flight attendant was hired. It is of DATETIME data type and is not marked as required (nullable). |
| **Flight\_Attendant\_DOB** | DATETIME | This attribute represents the date of birth of the flight attendant. It is of DATETIME data type and is not marked as required (nullable). |

-- Table mydb.Plane

DROP TABLE IF EXISTS mydb.Plane;

CREATE TABLE IF NOT EXISTS mydb.Plane (

Plane\_ID INT NOT NULL,

Plane\_Model VARCHAR(45) NULL,

Plane\_Manufacturer VARCHAR(45) NULL,

Plane\_Size VARCHAR(20),

Plane\_Engine\_Power VARCHAR(20),

PRIMARY KEY (Plane\_ID),

);

### **Table: Plane**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Plane\_ID** | INT | This attribute serves as the primary key for the "Plane" table, uniquely identifying each plane's record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **Plane\_Model** | VARCHAR | The "Plane\_Model" attribute represents the model of the plane. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Plane\_Manufacturer** | VARCHAR | This attribute represents the manufacturer of the plane. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Plane\_Size** | VARCHAR | The "Plane\_Size" attribute describes the size or category of the plane. It is of VARCHAR(20) data type and is not marked as required (nullable). |
| **Plane\_Engine\_Power** | VARCHAR | This attribute represents the engine power of the plane. It is of VARCHAR(20) data type and is not marked as required (nullable). |

-- Table mydb.Booking

DROP TABLE IF EXISTS mydb.Booking;

CREATE TABLE IF NOT EXISTS mydb.Booking (

Confirmation\_ID INT NOT NULL,

Booking\_Date DATETIME NULL,

Booking\_Platform VARCHAR(45) NULL,

Booking\_Type VARCHAR(45) NULL,

PRIMARY KEY (Confirmation\_ID),

FOREIGN KEY (Passenger\_ID) REFERENCES Passenger (Passenger\_ID),

);

### **Table: Booking**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Confirmation\_ID** | INT | This attribute serves as the primary key for the "Booking" table, uniquely identifying each booking's record. It is an integer (INT) data type and is required, meaning it cannot be null. |
| **Booking\_Date** | DATETIME | The "Booking\_Date" attribute represents the date and time when the booking was made. It is of DATETIME data type and is not marked as required (nullable). |
| **Booking\_Platform** | VARCHAR | This attribute describes the platform or source through which the booking was made. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Booking\_Type** | VARCHAR | The "Booking\_Type" attribute specifies the type of booking made, such as "economy," "business," etc. It is of VARCHAR(45) data type and is not marked as required (nullable). |
| **Passenger\_ID** | INT | This foreign key is used to establish a relationship with the "Passenger" table, specifically referencing the "Passenger\_ID" attribute in the "Booking" table. It represents the passenger associated with the booking. |

-- Table mydb.Baggage

DROP TABLE IF EXISTS mydb.Baggage;

CREATE TABLE IF NOT EXISTS mydb.Baggage (

Baggage\_ID VARCHAR(45) NOT NULL,

Passenger\_ID INT NULL,

Baggage\_Weight DECIMAL(2) NULL,

Baggage\_Cost DECIMAL(2) NULL,

Org\_Airport\_Loc DATETIME NULL,

Org\_Departure\_Time VARCHAR(45) NULL,

Flight\_ID INT NULL,

Des\_Airport\_Loc VARCHAR(45) NULL,

Des\_Departure\_Time DATETIME NULL,

Des\_Departure\_Gate VARCHAR(45) NULL,

PRIMARY KEY (Baggage\_ID),

FOREIGN KEY (Passenger\_ID) REFERENCES Passenger (Passenger\_ID)

);

### **Table: Baggage**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Baggage\_ID** | VARCHAR | This attribute serves as the primary key for the "Baggage" table, uniquely identifying each piece of baggage. It is of VARCHAR(45) data type, allowing up to 45 characters, and it is required, meaning it cannot be null. |
| **Passenger\_ID** | INT | The "Passenger\_ID" attribute represents the passenger associated with the baggage. It is an integer (INT) data type and is not marked as required (nullable). This attribute is linked to the "Passenger" table through a foreign key relationship. |
| **Baggage\_Weight** | DECIMAL | This attribute indicates the weight of the baggage, represented as a decimal value with two decimal places. It is not marked as required (nullable). |
| **Baggage\_Cost** | DECIMAL | The "Baggage\_Cost" attribute specifies the cost associated with the baggage. It is represented as a decimal value with two decimal places and is not marked as required (nullable). |
| **Org\_Airport\_Loc** | DATETIME | This attribute represents the location of the originating airport. It is of DATETIME data type and is not marked as required (nullable). The data type choice may be inappropriate for this attribute, and it might need to be adjusted based on the actual data it represents. |
| **Org\_Departure\_Time** | VARCHAR | The "Org\_Departure\_Time" attribute denotes the departure time of the flight from the originating airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Flight\_ID** | INT | This attribute represents the ID of the flight associated with the baggage. It is an integer (INT) data type and is not marked as required (nullable). |
| **Des\_Airport\_Loc** | VARCHAR | The "Des\_Airport\_Loc" attribute specifies the location of the destination airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |
| **Des\_Departure\_Time** | DATETIME | This attribute represents the departure time of the flight from the destination airport. It is of DATETIME data type and is not marked as required (nullable). |
| **Des\_Departure\_Gate** | VARCHAR | The "Des\_Departure\_Gate" attribute indicates the departure gate at the destination airport. It is of VARCHAR(45) data type, allowing up to 45 characters, and is not marked as required (nullable). |

-- Table mydb.Flight

DROP TABLE IF EXISTS mydb.Flight;

CREATE TABLE IF NOT EXISTS mydb.Flight (

Flight\_ID INT NOT NULL,

Org\_Departure\_Time DATETIME NULL,

Des\_Arrival\_Time DATETIME NULL,

Flight\_Time DECIMAL(2) NULL,

PRIMARY KEY (Flight\_ID),

FOREIGN KEY (Passenger\_ID) REFERENCES Passenger(Passenger\_ID),

FOREIGN KEY (Confirmation\_ID) REFERENCES Booking(Confirmation\_ID),

FOREIGN KEY (Org\_Airport\_ID) REFERENCES Org\_Airport (Org\_Airport\_ID),

FOREIGN KEY (Des\_Airport\_ID) REFERENCES Des\_Airport (Des\_Airport\_ID),

FOREIGN KEY (Plane\_ID) REFERENCES Plane (Plane\_ID),

FOREIGN KEY (Pilot\_ID) REFERENCES Pilot (Pilot\_ID),

FOREIGN KEY (Flight\_Att\_ID) REFERENCES Flight\_Attendent (Flight\_Att\_ID)

);

### **Table: Flight**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Flight\_ID** | INT | This attribute serves as the primary key for the "Flight" table, uniquely identifying each flight. It is of INT data type, and it is required, meaning it cannot be null. |
| **Org\_Departure\_Time** | DATETIME | The "Org\_Departure\_Time" attribute represents the departure time of the flight from the originating airport. It is of DATETIME data type and is not marked as required (nullable). |
| **Des\_Arrival\_Time** | DATETIME | This attribute denotes the arrival time of the flight at the destination airport. It is of DATETIME data type and is not marked as required (nullable). |
| **Flight\_Time** | DECIMAL | The "Flight\_Time" attribute specifies the duration of the flight in decimal format with two decimal places. It is not marked as required (nullable). |
| **Passenger\_ID** | INT | This foreign key establishes a relationship with the "Passenger" table, specifically referencing the "Passenger\_ID" attribute in the "Flight" table. It associates flights with the corresponding passenger. |
| **Confirmation\_ID** | INT | This foreign key relates to the "Confirmation\_ID" attribute in the "Booking" table, allowing flights to be associated with specific bookings. |
| **Org\_Airport\_ID** | INT | This foreign key links to the "Org\_Airport\_ID" attribute in the "Org\_Airport" table, indicating the originating airport of the flight. |
| **Des\_Airport\_ID** | INT | This foreign key references the "Des\_Airport\_ID" attribute in the "Des\_Airport" table, specifying the destination airport of the flight. |
| **Plane\_ID** | INT | This foreign key is associated with the "Plane\_ID" attribute in the "Plane" table, allowing flights to be linked to specific planes. |
| **Pilot\_ID** | INT | This foreign key relates to the "Pilot\_ID" attribute in the "Pilot" table, indicating the pilot responsible for the flight. |
| **Flight\_Att\_ID** | INT | This foreign key establishes a relationship with the "Flight\_Attendent" table, specifically referencing the "Flight\_Att\_ID" attribute in the "Flight" table. It associates flight attendants with specific flights. |

-- Table mydb.Flight\_Attendent\_has\_Flight

DROP TABLE IF EXISTS mydb.Flight\_Attendent\_has\_Flight;

CREATE TABLE IF NOT EXISTS mydb.Flight\_Attendent\_has\_Flight (

PRIMARY KEY (Flight\_Att\_ID, Flight\_ID, Org\_Airport\_ID, Des\_Airport\_ID, Pilot\_ID, Plane\_ID),

FOREIGN KEY (Flight\_Att\_ID) REFERENCES Flight\_Attendent (Flight\_Att\_ID),

FOREIGN KEY (Flight\_ID) REFERENCES Flight (Flight\_ID),

FOREIGN KEY (Org\_Airport\_ID) REFERENCES Org\_Airport (Org\_Airport\_ID),

FOREIGN KEY (Des\_Airport\_ID) REFERENCES Des\_Airport (Des\_Airport\_ID),

FOREIGN KEY (Pilot\_ID) REFERENCES Pilot (Pilot\_ID),

FOREIGN KEY (Plane\_ID) REFERENCES Plane (Plane\_ID)

);

### **Table: Flight\_Attendent\_has\_Flight**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| **Flight\_Att\_ID** | INT | This attribute is part of the primary key and establishes a relationship with the "Flight\_Attendent" table. It uniquely identifies the flight attendant involved in the flight. |
| **Flight\_ID** | INT | This attribute is part of the primary key and references the "Flight" table. It uniquely identifies the flight in which the flight attendant is assigned. |
| **Org\_Airport\_ID** | INT | This foreign key is associated with the "Org\_Airport" table, indicating the originating airport of the flight. |
| **Des\_Airport\_ID** | INT | This foreign key references the "Des\_Airport" table, specifying the destination airport of the flight. |
| **Pilot\_ID** | INT | This foreign key establishes a relationship with the "Pilot" table, indicating the pilot responsible for the flight. |
| **Plane\_ID** | INT | This foreign key is associated with the "Plane" table, allowing the association of flight attendants with specific planes used for flights. |

## References

1. Airline Suite. (n.d.). *Airline Suite Reviews - Pros & Cons, Ratings & More*. GetApp. <https://www.getapp.com/operations-management-software/a/airline-suite/reviews/>
2. *Aviation Maintenance Operations & Aircraft Management Software: RAMCO systems*. Aviation Maintenance Operations & Aircraft Management Software | Ramco Systems. (n.d.). <https://www.ramco.com/products/aviation-software/?utm_source=capterra&utm_medium=capterra_ppc&utm_campaign=capterra_aviation&utm_channel=capterra>
3. AvPro. (n.d.-a). *Aircraft Maintenance Software: Wellington FL: AvPro software*. avpro. <https://www.avprosoftware.com/>
4. AvPro. (n.d.). *Fly Safer. work smarter.* C.A.L.M. Systems | Airline Suite | Aviation Management Software. <https://info.gartnerdigitalmarkets.com/calm-systems-gdm-lp?utm_source=capterra>
5. Trustradius. (2021, June 16). *Pros and cons of Ramco ERP 2023*. TrustRadius. <https://www.trustradius.com/products/ramco-erp/reviews?qs=pros-and-cons#reviews>