

**CS4347 Database Systems  
Final Project Deliverable 2**

**Snoopy Airlines**

Danielle Bryan, Wei Yuan Liew, Tanvi Nair, Dandan Wang, Mandy Hardono, Laura Pinto, Meyli Colmenero, Fernando Portillo, Shreyaa Arun, Hunter Matthews

## 0. Description → Laura

**Project Title:**  
Snoopy Airlines

### GitHub Details

- *All the code is located on GitHub. The code was not added to the zip file due to its size.*  
<https://github.com/Persona5-Joker/4347.005-Snoopy-Airlines>

### Members (10 members): Divine ten

Danielle Bryan [CS], Wei Yuan Liew [CS], Tanvi Nair [DS], Dandan Wang [CS], Mandy Hardono [CE], Laura Pinto [CE], Meyli Colmenero [CE], Fernando Portillo [CE], Hunter Matthews [CS], and Shreyaa Arun [SE].

### Project Task Delegation

Delegation of tasks was decided following the sections of this project:

- *Danielle Bryan*: EER Conceptual data model design and population of database. 3NF redesign diagrams and queries. Front-end creation.
- *Wei Yuan Liew*: Relational data model design. Creation of the views.
- *Tanvi Nair*: Creation and population of database. 3NF redesign diagrams and queries.
- *Hunter Matthews*: Database query execution. 3NF redesign diagrams and queries.
- *Dandan Wang*: Relational data model design. 3NF redesign diagrams and queries.
- *Mandy Hardono*: Background and related work, and Database query execution. 3NF redesign diagrams and queries. Creation of the views.
- *Laura Pinto*: Introduction, Description, References, conclusion, and creation of Views in the database.
- *Meyli Colmenero*: Background, related work, and database query/population and creation testing. Front-end creation.
- *Fernando Portillo*: EER Conceptual data model design, database query execution and Database query testing. Front-end creation.
- *Shreyaa Arun*: Creation and population of database. 3NF redesign diagrams and queries.

### Project Inspiration

This project can be used in a real-world airline system, aiding the backend structure for easily managing flights, aircraft, customers, reservations, passengers, and payments. The inspiration for this project came from the modern industries' increasing reliance on data-driven systems, particularly aviation. The aviation industry has a crucial role in connecting people around the world, but this connectivity heavily depends on the management of large-scale data

related to flights. Therefore, the team decided to choose this project to explore real-world problems like handling reservations, maintaining consistency across multiple related entities, and providing seamless customer interaction which inspired us even more to create a database for a system of this scale. Our motive includes improving operational efficiency in which the system will automate routine tasks and processes, reduce manual effort with minimal errors, and enhance customer service experience which implements features that enable smooth booking, check-in process, and real-time tracking.

## **Timeline of Development**

This project consisted of the following steps to design and develop the project to create the project deliverable 2. Following up with the general time development, the team dedicated another meeting to determine the next steps for project deliverable 2 (steps 5-8).

- **Step 1: Initial Planning and requirements gathering**

- At this beginning stage, the team identified the tasks for each member, meaning that the team was divided into sub-teams to work on each task. Although the team was assigned to a specific task, we were able to help other team members with their tasks. Once the tasks were decided, we identified the key entities that this project needed for the aircraft database, which are the following: Flight, Aircraft, Customer, Reservation, Passenger, and Payment. The team also decided on how to relate these entities together and their primary/foreign keys throughout the process.

- **Step 2: Designing the Normalized ER Diagram and Relational model**

- Once we had the basic idea of what the team wanted to add as entities and relationships, we decided to have a draft of a relational diagram that aided in the visualization of the database schema. The creation of the Entity-Relationship (ER) diagram and Relational model design was done by some of our team members, where they determined the final attributes for each entity and mapped out how the foreign keys would be used to represent all the relationships between the aircraft database.

- **Step 3: Creation and Population of the database**

- After the creation of the ER diagram, the creation and population team worked to create the database using the specifications that had already been set using MySQL. At this stage, all the tables for each entity were created and defined with their corresponding primary and foreign key constraints to maintain the data integrity as well as their data types and any specifications needed for the data (such as cascades, deletes, not null, etc). The tables were created and populated in a specific structure to avoid any redundancy on the tables. The population of the table also ensured diversity as anyone can make a reservation to a flight with a passport or ID.

- **Step 4: Testing and Validation**

- Following the previous steps, the next thing was ensuring that the system worked as intended. All the tables were created and tested with their population data to ensure the correct creation for them. The system was also tested with the test queries, checking whether the relationships worked correctly while maintaining consistency in the data.
- **Step 5: New Session Planning**
  - After the submission of project deliverable 1. The team decided to meet once again to divide the tasks needed for the project deliverable 2. This meeting consisted of all team members selecting the tasks needed to work on by dividing the team again into sub-teams to work on the different tasks together.
- **Step 6: Normalization of the database and database population**
  - The normalization of the database included dividing the passenger reservation and passenger information to avoid redundancy whenever passengers are doing reservations. It is also important because while doing reservation, having these two entities will allow easier manipulation of data.
- **Step 7: Creation of Database Query and Views**
  - After the normalization of the database, the team populated once again the database with a combination of old and new information that allowed the team to start writing the overall creation of the database. New query was also created for the database including the new information from the normalized database, insert, update, delete and view operations were designed to prevent redundancy in the database.
- **Step 8: Front-end User Interface Creation**
  - The front-end user interface creation includes a Snoopy-themed website that allows to visualize the reservation contact information and passenger information such as first name, last name, phone number, and other relevant pieces of information to travel. Booking confirmation is also a page that includes information about the departure and returning flight of the passenger who made the reservation like flight number, departure and arrival time, passenger details, and more. Available flights based on search are also available to find the best flight for individuals looking to travel somewhere. Most importantly, the website offers the booking page where individuals can search for the flight they would like to book. All this implementation was important because of the use of the database to create it.
- **Step 9: Presentation design**
  - The team decided on an airline slides presentation to talk and demonstrate the project that we have been working on during the semester.

## **1. Introduction → Laura**

Airports play a vital role as the main hubs for air travel, connecting people and businesses worldwide[1]. Without modern aircraft, global connectivity would be a challenge for us, which nowadays is an essential key for the growth and development of economies across the world. However, as the aviation industry faces expansion, the need for efficient data management systems has also grown[1].

This project addresses a common challenge in the aviation sector—efficient data management. This includes flight tracking, customer reservation, and overall passenger service. Currently, many existing systems lack the tools to have consistent data which can result in potential problems of inconsistent data such as delays, scheduling errors, and passenger dissatisfaction that can severely impact the airline's ability to operate smoothly and efficiently [3].

Our project, Snoopy Airlines aims to demonstrate how accurate data management can enhance the overall efficiency and safety of airline operations. By creating a well-structured and reliable database, airlines can optimize routes for fuel efficiency, reduce cancellations and delays, and improve passenger experience [2]. Additionally, better data management leads to more smooth operations, allowing airlines to handle customer service reservations and payments with greater ease and accuracy.

Through Snoopy Airlines, we hope to be an example of how a well-designed database can improve the operational efficiency of an airline, making flight management, customer service, and financial transactions more user-friendly and effective. This project highlights the importance of data consistency and efficiency in ensuring smooth and successful airline operations.

## **2. Background and Related Work → Mandy, Meyli**

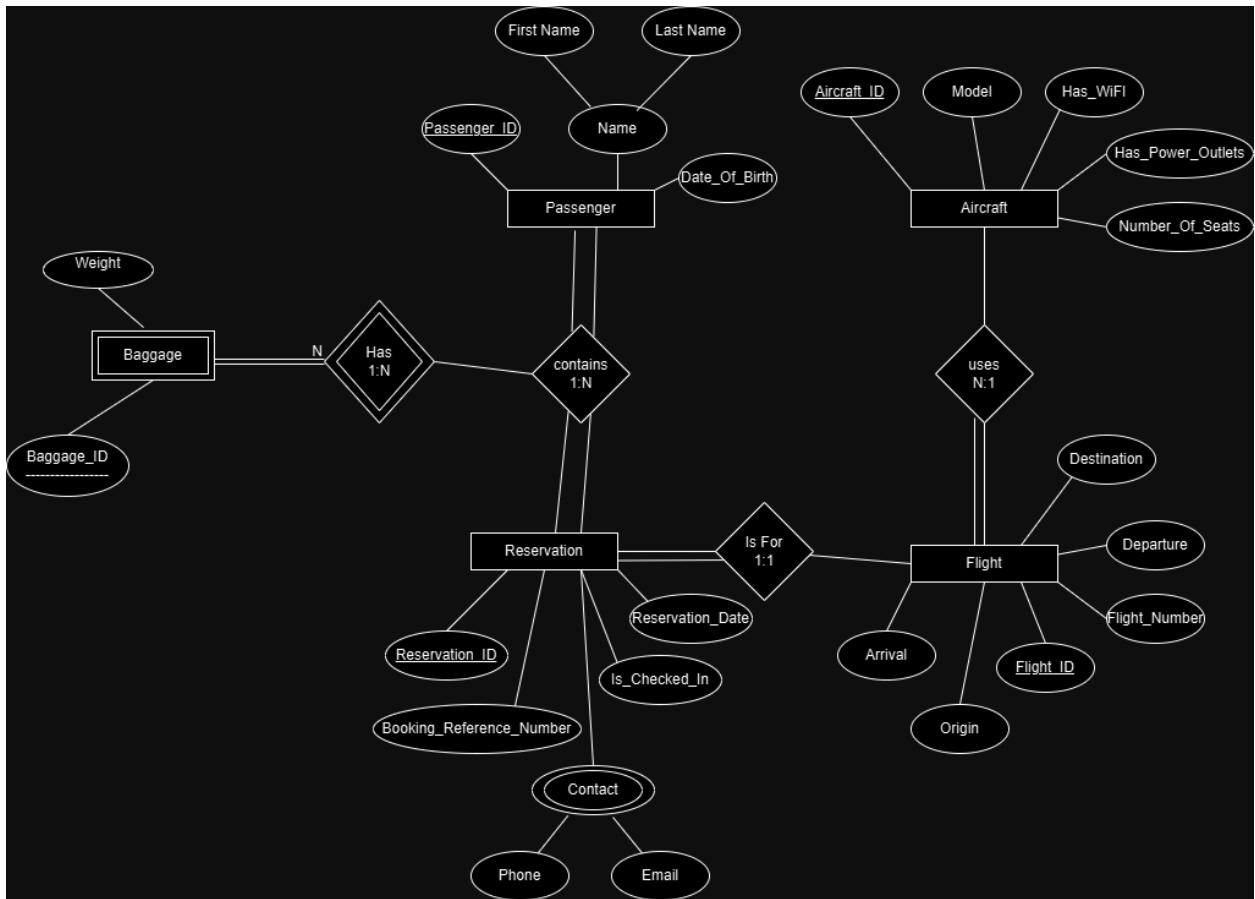
Snoopy Airlines was initially influenced by the American Airlines website. For the Flight Status lookup feature, a user can enter the source and destination cities, then the site will display the matching flights. The information presented includes the flight number, the arrival status, the aircraft name, the departing terminal, baggage, and the arriving terminal [4]. If a flight was completed, the actual arrival and departure times would be included. In Snoopy Airlines, a flight is defined similarly. However, it omits the aircraft model, baggage, and terminal information. In addition to the flight status, airlines store data on payments, reward programs, customer accounts, employees, and more. Though an existing airline website inspired the foundation of Snoopy Airlines, our database focuses on the scenario that every passenger has a ticket and is

ready to board an available flight. The database describes the relationships between a passenger and the basic elements of an airline that allow them to reach their destination.

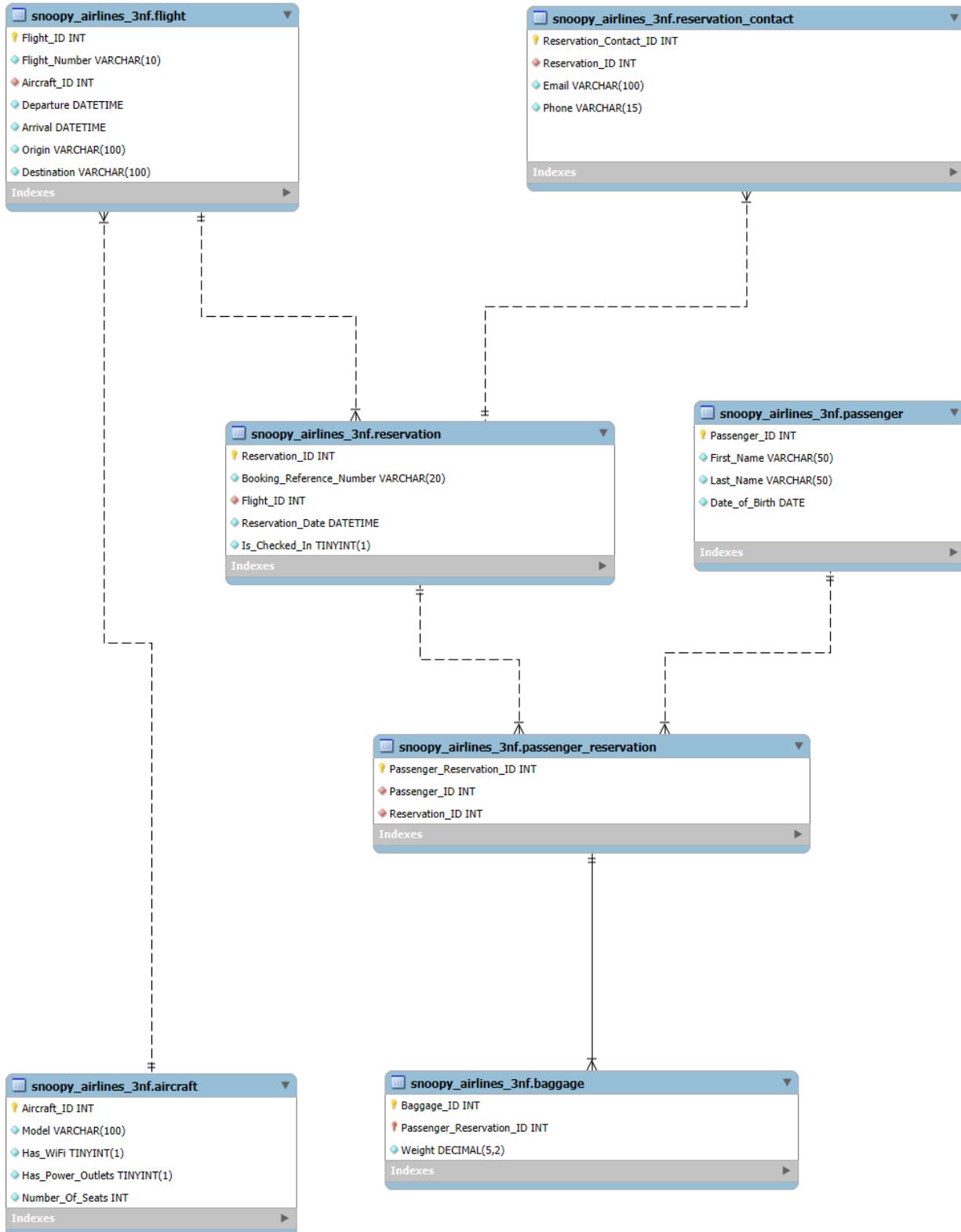
Another example of a fully fleshed out larger scale database can be seen in what GeeksForGeeks labels as a standard database for airline companies [5]. This standard database is operationally focused with detailed relationships on how the airline runs. Example relationships include flights, crew members, aircraft, and reservations; Not just highlight on passengers and reservations like Snoopy Airlines where customer-experience is the main priority hence the focus is on passenger and reservation data. The geeksfor geeks model shows the external relations between the crew and passenger, and more built out entities. For the purpose of this project, minimizing the scope helps by making sure there aren't many entities and relationships to get overwhelmed with and ensuring a more simple and efficient design to manage passengers, their bookings, and their baggage. Snoopy Airlines is providing a more customer focused vision separating the external relations that come with a large scale typical airline database.

### **3. Design & Implementation (Phase II):**

#### **3.1. Normalization of EER Conceptual Data Model:** → Fernando, Dani



**3.2. Relational Data Model Design Using Normalized EER Diagram:** →  
Dandan, Wei Yuan



### 3.3. Create your Normalized Database and Populate: → Tanvi, Dani, Shreyaa, Laura

Aircraft Table Population

```
-- QUERIES
INSERT INTO Aircraft (Model, Has_WiFi, Has_Power_Outlets, Number_of_Seats) VALUES
('Boeing 737', TRUE, TRUE, 160),
('Airbus A320', TRUE, FALSE, 180),
('Boeing 777', TRUE, TRUE, 300),
('Airbus A330', TRUE, TRUE, 250),
('Embraer 175', FALSE, FALSE, 88),
('Bombardier Q400', FALSE, FALSE, 78),
('Boeing 787', TRUE, TRUE, 242),
('Airbus A350', TRUE, TRUE, 315),
('Cessna 208', FALSE, FALSE, 12),
('Boeing 747', TRUE, TRUE, 416),
('Boeing 767', TRUE, TRUE, 211),
('Airbus A380', TRUE, TRUE, 555),
('Airbus A220', TRUE, TRUE, 140),
('Boeing 757', TRUE, TRUE, 239),
('Boeing 717', TRUE, FALSE, 106),
('Embraer 190', TRUE, TRUE, 100),
('Bombardier CRJ700', FALSE, FALSE, 70),
('Boeing 727', FALSE, FALSE, 189),
('Airbus A310', TRUE, FALSE, 280),
('McDonnell Douglas MD-80', FALSE, FALSE, 172),
('Airbus A321', TRUE, TRUE, 220),
('Boeing 720', FALSE, FALSE, 149),
('Boeing 707', FALSE, FALSE, 174),
('Airbus A319', TRUE, TRUE, 156),
('Embraer E170', FALSE, FALSE, 76);
```

Flight Table Population

```

INSERT INTO Flight (Flight_Number, Aircraft_ID, Departure, Arrival, Origin, Destination) VALUES
('SN100', 1, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Miami'),
('SN101', 2, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Miami', 'Los Angeles'),
('SN102', 3, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Denver'),
('SN103', 4, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Denver', 'Los Angeles'),
('SN104', 5, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Seattle'),
('SN105', 6, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Seattle', 'Los Angeles'),
('SN106', 7, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Houston'),
('SN107', 8, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Houston', 'Los Angeles'),
('SN108', 9, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Boston'),
('SN109', 10, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Boston', 'Los Angeles'),
('SN110', 11, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Washington DC'),
('SN111', 12, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Washington DC', 'Los Angeles'),
('SN112', 13, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Salt Lake City'),
('SN113', 14, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Salt Lake City', 'Los Angeles'),
('SN114', 15, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Portland'),
('SN115', 16, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Portland', 'Los Angeles'),
('SN116', 17, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Nashville'),
('SN117', 18, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Nashville', 'Los Angeles'),
('SN118', 19, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Philadelphia'),
('SN119', 20, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Philadelphia', 'Los Angeles'),
('SN120', 21, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Dallas'),
('SN121', 22, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Dallas', 'Los Angeles'),
('SN122', 23, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Chicago'),
('SN123', 24, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Chicago', 'Los Angeles'),
('SN124', 25, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Atlanta'),
('SN125', 1, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Atlanta', 'Los Angeles'),
('SN126', 2, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Las Vegas'),
('SN127', 3, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Las Vegas', 'Los Angeles'),
('SN128', 4, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'San Francisco'),
('SN129', 5, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'San Francisco', 'Los Angeles'),
('SN130', 6, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'New York'),
('SN131', 7, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'New York', 'Los Angeles'),
('SN132', 8, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Austin'),
('SN133', 9, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Austin', 'Los Angeles'),
('SN134', 10, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Orlando'),
('SN135', 11, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Orlando', 'Los Angeles'),
('SN136', 12, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Phoenix'),
('SN137', 13, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Phoenix', 'Los Angeles'),
('SN138', 14, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'San Diego'),
('SN139', 15, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'San Diego', 'Los Angeles'),
('SN140', 16, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Detroit'),
('SN141', 17, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Detroit', 'Los Angeles'),
('SN142', 18, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Salem'),
('SN143', 19, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Salem', 'Los Angeles'),
('SN144', 20, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Los Angeles', 'Louisville'),
('SN145', 21, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Louisville', 'Los Angeles'),
('SN146', 22, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Los Angeles'),
('SN147', 23, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Los Angeles', 'Miami'),
('SN148', 24, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Denver'),
('SN149', 25, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Denver', 'Miami'),
('SN150', 1, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Seattle'),
('SN151', 2, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Seattle', 'Miami'),
('SN152', 3, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Houston'),
('SN153', 4, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Houston', 'Miami'),
('SN154', 5, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Boston'),
('SN155', 6, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Boston', 'Miami'),
('SN156', 7, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Washington DC'),
('SN157', 8, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Washington DC', 'Miami'),
('SN158', 9, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Salt Lake City'),
('SN159', 10, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Salt Lake City', 'Miami'),
('SN160', 11, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Portland'),
('SN161', 12, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Portland', 'Miami'),
('SN162', 13, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Nashville'),
('SN163', 14, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Nashville', 'Miami'),
('SN164', 15, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Philadelphia'),
('SN165', 16, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Philadelphia', 'Miami'),
('SN166', 17, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Dallas'),
('SN167', 18, '2024-11-15 17:00:00', '2024-11-15 20:00:00', 'Dallas', 'Miami'),
('SN168', 19, '2024-11-15 09:00:00', '2024-11-15 12:00:00', 'Miami', 'Chicago')

```

## Reservation Table Population

```

INSERT INTO Reservation (Booking_Reference_Number, Flight_ID, Reservation_Date, Is_Checked_In) VALUES
('BR123456', 1, '2024-10-12 10:00:00', FALSE),
('BR234567', 2, '2024-10-13 11:00:00', TRUE),
('BR345678', 3, '2024-10-14 09:30:00', FALSE),
('BR456789', 4, '2024-10-15 12:00:00', TRUE),
('BR567890', 5, '2024-10-16 14:00:00', FALSE),
('BR678901', 6, '2024-10-17 08:00:00', TRUE),
('BR789012', 7, '2024-10-18 15:00:00', FALSE),
('BR890123', 8, '2024-10-19 17:00:00', TRUE),
('BR901234', 9, '2024-10-20 09:00:00', FALSE),
('BR012345', 10, '2024-10-21 18:00:00', TRUE),
('BR112233', 11, '2024-10-22 20:00:00', FALSE),
('BR223344', 12, '2024-10-23 22:00:00', TRUE),
('BR334455', 13, '2024-10-24 07:30:00', FALSE),
('BR445566', 14, '2024-10-25 09:00:00', TRUE),
('BR556677', 15, '2024-10-26 10:30:00', FALSE),
('BR667788', 16, '2024-10-27 12:00:00', TRUE),
('BR778899', 17, '2024-10-28 14:00:00', FALSE),
('BR889900', 18, '2024-10-29 16:00:00', TRUE),
('BR990011', 19, '2024-10-30 08:00:00', FALSE),
('BR001122', 20, '2024-10-31 06:00:00', TRUE);

```

## Reservation\_Contact Table Population

```

INSERT INTO Reservation_Contact (Reservation_ID, Email, Phone)
VALUES
(1, 'john.doe@example.com', '972-555-1234'),
(2, 'jane.smith@example.com', '972-555-5678'),
(3, 'michael.jones@example.com', '214-555-2345'),
(4, 'emily.davis@example.com', '403-555-6789'),
(5, 'david.brown@example.com', '403-555-3456'),
(6, 'linda.wilson@example.com', '972-555-7890'),
(7, 'james.miller@example.com', '972-555-4567'),
(8, 'barbara.moore@example.com', '972-555-8901'),
(9, 'robert.taylor@example.com', '214-555-5679'),
(10, 'patricia.anderson@example.com', '214-555-9012'),
(11, 'charles.thomas@example.com', '872-555-6780'),
(12, 'susan.jackson@example.com', '119-555-0123'),
(13, 'paul.white@example.com', '903-555-3457'),
(14, 'nancy.harris@example.com', '903-555-7891'),
(15, 'mark.martin@example.com', '903-555-4568'),
(16, 'betty.clark@example.com', '214-555-8902'),
(17, 'george.lewis@example.com', '214-555-5670');

```

## Passenger Table Population

```
-- Populating the Passenger table
INSERT INTO Passenger (First_Name, Last_Name, Date_of_Birth) VALUES
('John', 'Doe', '1985-04-12'),
('Jane', 'Smith', '1990-11-23'),
('Michael', 'Jones', '1978-06-05'),
('Emily', 'Davis', '1982-02-17'),
('Danielle', 'Bryan', '1995-09-09'),
('Wei Yuan', 'Liew', '1992-07-22'),
('Tanvi', 'Nair', '2001-01-15'),
('Dandan', 'Wang', '1988-12-01'),
('Mandy', 'Hardono', '1993-03-19'),
('Laura', 'Pinto', '1986-05-30'),
('Meyli', 'Colmenero', '1998-08-24'),
('Fernando', 'Portillo', '1991-10-07'),
('Shreyaa', 'Arun', '1999-04-14'),
('Hunter', 'Matthews', '1987-02-10'),
('Ebru', 'Cankaya', '1994-06-26'),
('Robert', 'Taylor', '1980-09-05'),
('Patricia', 'Anderson', '1975-01-28'),
('George', 'Lewis', '1968-07-18'),
('Jessica', 'Walker', '2000-11-11'),
('Henry', 'King', '1989-02-22');
```

Passenger\_Reservation Population

```
-- Populating the Passenger_Reservation table
INSERT INTO Passenger_Reservation (Passenger_ID, Reservation_ID) VALUES
(1, 1), -- John Doe for Reservation BR123456
(2, 2), -- Jane Smith for Reservation BR234567
(3, 3), -- Michael Jones for Reservation BR345678
(4, 4), -- Emily Davis for Reservation BR456789
(5, 5), -- Danielle Bryan for Reservation BR567890
(6, 6), -- Wei Yuan Liew for Reservation BR678901
(7, 7), -- Tanvi Nair for Reservation BR789012
(8, 8), -- Dandan Wang for Reservation BR890123
(9, 9), -- Mandy Hardono for Reservation BR901234
(10, 10), -- Laura Pinto for Reservation BR012345
(11, 11), -- Meyli Colmenero for Reservation BR112233
(12, 12), -- Fernando Portillo for Reservation BR223344
(13, 13), -- Shreyaa Arun for Reservation BR334455
(14, 14), -- Hunter Matthews for Reservation BR445566
(15, 15), -- Ebru Cankaya for Reservation BR556677
(16, 16), -- Robert Taylor for Reservation BR667788
(17, 17), -- Patricia Anderson for Reservation BR778899
(18, 18), -- George Lewis for Reservation BR889900
(19, 19), -- Jessica Walker for Reservation BR990011
(20, 20), -- Henry King for Reservation BR001122
```

```
-- Adding some passengers to additional reservations to illustrate the many-to-many relationship
(1, 2), -- John Doe for Reservation BR234567
(2, 3), -- Jane Smith for Reservation BR345678
(3, 4), -- Michael Jones for Reservation BR456789
(4, 5), -- Emily Davis for Reservation BR567890
(5, 6); -- Danielle Bryan for Reservation BR678901
```

## Baggage Population

```
-- Inserting baggage entries associated directly with Passenger_Reservation_ID
INSERT INTO Baggage (Passenger_Reservation_ID, Weight) VALUES
(1, 23.50), -- John Doe
(2, 18.75), -- Jane Smith
(3, 21.00), -- Michael Jones
(4, 19.30), -- Emily Davis
(5, 22.80), -- Danielle Bryan
(6, 25.40), -- Wei Yuan Liew
(7, 24.10), -- Tanvi Nair
(8, 20.50), -- Dandan Wang
(9, 23.75), -- Mandy Hardono
(10, 19.20), -- Laura Pinto
(11, 22.90), -- Meyli Colmenero
(12, 26.10), -- Fernando Portillo
(13, 24.30), -- Shreyaa Arun
(14, 25.00), -- Hunter Matthews
(15, 23.40), -- Ebru Cankaya
(16, 22.75), -- Robert Taylor
(17, 24.50), -- Patricia Anderson
(18, 21.30), -- George Lewis
(19, 25.60), -- Jessica Walker
(20, 23.90), -- Henry King
```

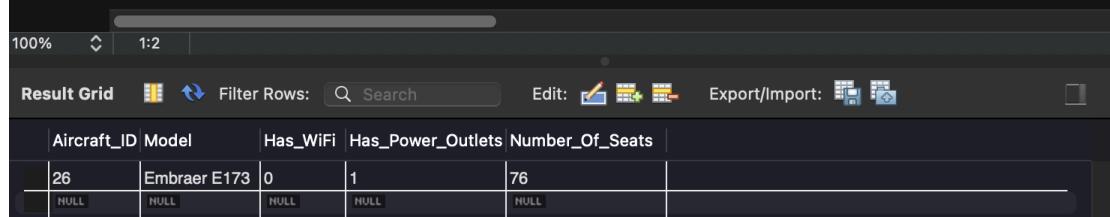
```
-- Additional entries
(21, 23.50),    -- John Doe
(22, 18.75),    -- Jane Smith
(23, 21.00),    -- Michael Jones
(24, 19.30),    -- Emily Davis
(25, 22.80);   -- Danielle Bryan
```

### 3.4. Database Query Execution on your Normalized Database (from inside your SQL client): → Mandy, Hunter, Meyli, Fernando

#### 3.4.1. Aircraft Table:

**INSERT:** Insert new aircraft called Embraer E173:

```
3 •  INSERT INTO Aircraft (Model, Has_WiFi, Has_Power_Outlets, Number_of_Seats) VALUES
4      ('Embraer E173', FALSE, TRUE, 76);
5 •  SELECT * FROM Aircraft WHERE Model = "Embraer E173";
```



The screenshot shows the MySQL Workbench interface with a query editor window. The query has been executed, and the results are displayed in a grid. The grid has columns labeled Aircraft\_ID, Model, Has\_WiFi, Has\_Power\_Outlets, and Number\_of\_Seats. There is one row of data: Aircraft\_ID 26, Model 'Embraer E173', Has\_WiFi 0, Has\_Power\_Outlets 1, and Number\_of\_Seats 76. Other columns in the grid are labeled 'NULL'.

Aircraft_ID	Model	Has_WiFi	Has_Power_Outlets	Number_of_Seats	
26	Embraer E173	0	1	76	

**QUERY:** Display Aircraft ID and models of planes that have Wifi:

```

62      -- QUERY: Display the Aircraft ID and Model of planes that have WiFi
63 •  SELECT Aircraft_ID, Model
64    FROM Aircraft
65   WHERE Has_WiFi = TRUE;
~~

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	Aircraft_ID	Model
▶	1	Boeing 737
	2	Airbus A320
	3	Boeing 777
	4	Airbus A330
	7	Boeing 787
	8	Airbus A350
	10	Boeing 747
	11	Boeing 767
	12	Airbus A380
	13	Airbus A220
	14	Boeing 757
	15	Boeing 717
	16	Embraer 190
	19	Airbus A310
	21	Airbus A321
	24	Airbus A319
*	NULL	NULL

## UPDATE: Change capacity to 50 for all Boeing aircrafts:

```

68      -- UPDATE: Change capacity to 50 for all planes from the Aircraft table that are Boeing aircrafts, then display result
69 •  UPDATE Aircraft
70   SET Number_Of_Seats = 50
71   WHERE Model LIKE 'Boeing%';
72 •  SELECT * FROM Aircraft WHERE Model LIKE 'Boeing%';
~~

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	Aircraft_ID	Model	Has_WiFi	Has_Power_Outlets	Number_Of_Seats
▶	1	Boeing 737	1	1	50
	3	Boeing 777	1	1	50
	7	Boeing 787	1	1	50
	10	Boeing 747	1	1	50
	11	Boeing 767	1	1	50
	14	Boeing 757	1	1	50
	15	Boeing 717	1	0	50
	18	Boeing 727	0	0	50
	22	Boeing 720	0	0	50
	23	Boeing 707	0	0	50
-	NULL	NULL	NULL	NULL	NULL

## DELETE: Delete aircraft entries with no power outlets:

```

74      -- DELETE: Delete Aircraft entries with no power outlets, then display Aircraft table
75 •  DELETE FROM Aircraft
76 WHERE Has_Power_Outlets = FALSE;
77 •  SELECT * FROM Aircraft;
78

```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The grid displays the 'Aircraft' table with the following data:

Aircraft_ID	Model	Has_WiFi	Has_Power_Outlets	Number_Of_Seats
1	Boeing 737	1	1	50
3	Boeing 777	1	1	50
4	Airbus A330	1	1	250
7	Boeing 787	1	1	50
8	Airbus A350	1	1	315
10	Boeing 747	1	1	50
11	Boeing 767	1	1	50
12	Airbus A380	1	1	555
13	Airbus A220	1	1	140
14	Boeing 757	1	1	50
16	Embraer 190	1	1	100
21	Airbus A321	1	1	220
24	Airbus A319	1	1	156
26	Embraer E173	0	1	76
*	NUL	NUL	NUL	NUL

### 3.4.2. Baggage Table :

Cannot insert without violating an integrity constraint for Passenger\_Reservation.

**QUERY:** Show Baggage entries where the baggage weight is heavier than average:

```

86 •  SELECT *
87     FROM Baggage
88 WHERE Weight > (SELECT AVG(Weight) FROM Baggage);

```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The grid displays the 'Baggage' table with the following data:

Baggage_ID	Passenger_Reservation_ID	Weight
1	1	23.50
3	3	23.50
7	7	24.10
11	11	22.90
12	12	26.10
13	13	24.30
14	14	25.00
16	16	22.75
*	NUL	NUL

**UPDATE:** Change baggage weight to 22.5 for passenger\_reservation\_ID = 3:

```
81 • UPDATE Baggage
82     SET Weight = 22.5
83     WHERE Passenger_Reservation_ID = 3;
84 • SELECT * FROM Baggage WHERE Passenger_Reservation_ID = 3;
85
86
87
88
```

	Baggage_ID	Passenger_Reservation_ID	Weight
▶	3	3	22.50
*	NULL	NULL	NULL

**DELETE:** Delete Baggage entry where the passenger\_reservation\_ID = 2:

```
90 • DELETE FROM Baggage
91     WHERE Passenger_Reservation_ID = 2;
92 • SELECT * FROM Baggage;
```

	Baggage_ID	Passenger_Reservation_ID	Weight
▶	1	1	23.50
	4	4	19.30
	7	7	24.10
	8	8	20.50
	10	10	19.20
	11	11	22.90
	12	12	26.10
	13	13	24.30
	14	14	25.00
	16	16	22.75
	22	22	18.75
	23	23	21.00
*	NULL	NULL	NULL

### 3.4.3. Passenger Table:

**INSERT:** Insert new passenger data:

```
97 •     INSERT INTO Passenger VALUES (22, 'Coco', 'Hardono', '2011-03-17');
```

```
98 •     SELECT * FROM Passenger WHERE Passenger_ID = 22;
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The grid displays the following data:

	Passenger_ID	First_Name	Last_Name	Date_of_Birth
▶	22	Coco	Hardono	2011-03-17
*	HULL	NULL	NULL	NULL

**QUERY:** Display the last names of passengers that start with 'M':

```
100 •     SELECT Last_Name
```

```
101         FROM Passenger
```

```
102     WHERE Last_Name LIKE 'M%';
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The grid displays the following data:

	Last_Name
▶	Matthews

**UPDATE:** Update the last name of passenger with ID 1:

```
104 • UPDATE Passenger  
105     SET Last_Name = 'Neumann'  
106     WHERE Passenger_ID = 1;  
107 • SELECT * FROM Passenger WHERE Passenger_ID = 1;  
...
```

	Passenger_ID	First_Name	Last_Name	Date_of_Birth
▶	1	John	Neumann	1985-04-12
*	HULL	HULL	HULL	HULL

**DELETE:** Delete the entry with passenger\_ID = 22:

```
109 • DELETE FROM Passenger  
110     WHERE Passenger_ID = 22;  
111 • SELECT * FROM Passenger WHERE Passenger_ID = 22;  
112  
113 • SELECT * FROM Passenger_Reservation;  
...
```

	Passenger_ID	First_Name	Last_Name	Date_of_Birth
*	HULL	HULL	HULL	HULL

### 3.4.4 Passenger\_Reservation Table Update

```
39929 • UPDATE Reservation_Contact  
39930   SET Phone = '972-555-9999'  
39931   WHERE Reservation_ID = 7;  
39932 • SELECT * FROM Reservation_Contact  
39933 WHERE Reservation_ID = 7;  
39934
```

Result Grid				
	Reservation_Contact_ID	Reservation_ID	Email	Phone
▶	7	7	james.miller@example.com	972-555-9999
*	NULL	NULL	NULL	NULL

### Delete

```
90 • DELETE FROM Baggage  
91 WHERE Passenger_Reservation_ID = 3;  
92 • SELECT * FROM Baggage;
```

Result Grid			
	Baggage_ID	Passenger_Reservation_ID	Weight
▶	1	1	23.50
	4	4	19.30
	7	7	24.10
	8	8	20.50
	10	10	19.20
	11	11	22.90
	12	12	26.10
	13	13	24.30
	14	14	25.00
	16	16	22.75
	22	22	18.75
	23	23	21.00
*	NULL	NULL	NULL

```

109 •   DELETE FROM Passenger
110      WHERE Passenger_ID = 22;
111 •   SELECT * FROM Passenger WHERE Passenger_ID = 22;
...

```

	Passenger_ID	First_Name	Last_Name	Date_of_Birth
*	NULL	NULL	NULL	NULL

### 3.4.5 Reservation Table Query

#### Insert

```

39913 • INSERT INTO Reservation (Booking_Reference_Number, Flight_ID, Reservation_Date, Is_Checked_In
39914 VALUES
39915 ('BR123456', 21, '2024-11-17 10:00:00', FALSE),
39916 ('BR654321', 22, '2024-11-18 14:30:00', TRUE),
39917 ('BR789123', 23, '2024-11-19 08:15:00', FALSE);
39918 • SELECT * FROM Reservation;
39919

```

Reservation_ID	Booking_Reference_Number	Flight_ID	Reservation_Date	Is_Checked_In
20	BR001122	20	2024-10-31 06:00:00	1
21	BR123456	21	2024-11-17 10:00:00	0
22	BR654321	22	2024-11-18 14:30:00	1
23	BR789123	23	2024-11-19 08:15:00	0
*	NULL	NULL	NULL	NULL

#### Delete

```

39907 • DELETE FROM Reservation
39908 WHERE Booking_Reference_Number = 'BR123456';
39909 • SELECT *
39910 FROM Reservation
39911 WHERE Booking_Reference_Number = 'BR123456';

```

Reservation_ID	Booking_Reference_Number	Flight_ID	Reservation_Date	Is_Checked_In
*	NULL	NULL	NULL	NULL

#### Update

```

39901 • UPDATE Reservation
39902   SET Is_Checked_In = TRUE
39903   WHERE Booking_Reference_Number = 'BR123456';
39904 • SELECT *
39905   FROM Reservation
39906   WHERE Booking_Reference_Number = 'BR123456';
39907

```

Result Grid					
	Reservation_ID	Booking_Reference_Number	Flight_ID	Reservation_Date	Is_Checked_In
▶	1	BR123456	1	2024-10-12 10:00:00	1
●	NULL	NULL	NULL	NULL	NULL

### 3.4.6 Reservation\_Contact Table

Query

Insert

```

39919 • INSERT INTO Reservation_Contact (Reservation_ID, Email, Phone)
39920   VALUES
39921   (18, 'anna.walker@example.com', '972-555-1111'),
39922   (19, 'kevin.hall@example.com', '214-555-2222');
39923 • SELECT * FROM Reservation_Contact
39924   WHERE Reservation_ID IN (18, 19);
39925

```

Result Grid				
	Reservation_Contact_ID	Reservation_ID	Email	Phone
▶	18	18	anna.walker@example.com	972-555-1111
●	19	19	kevin.hall@example.com	214-555-2222
●	NULL	NULL	NULL	NULL

Delete

```

39925 • DELETE FROM Reservation_Contact
39926   WHERE Reservation_ID = 6;
39927 • SELECT * FROM Reservation_Contact
39928   WHERE Reservation_ID = 6;
39929

```

Result Grid				
	Reservation_Contact_ID	Reservation_ID	Email	Phone
*	NULL	NULL	NULL	NULL

Update

```

39929 • UPDATE Reservation_Contact
39930   SET Phone = '972-555-9999'
39931   WHERE Reservation_ID = 7;
39932 • SELECT * FROM Reservation_Contact
39933   WHERE Reservation_ID = 7;
39934

```

	Reservation_Contact_ID	Reservation_ID	Email	Phone
▶	7	7	james.miller@example.com	972-555-9999
*	NULL	NULL	NULL	NULL

### 3.4.7 Flight Table

**INSERT:** Insert a new flight into the Flight table:

```

117 • INSERT INTO Flight VALUES (3005,'SN3005', 7, '2024-11-17 02:00:00', '2024-11-17 05:00:00', 'Dallas', 'New York City');
118 • SELECT * FROM Flight WHERE Flight_Number = 'SN3005';
119

```

	Flight_ID	Flight_Number	Aircraft_ID	Departure	Arrival	Origin	Destination
▶	3005	SN3005	7	2024-11-17 02:00:00	2024-11-17 05:00:00	Dallas	New York City
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**QUERY:** Getting all the flights departing from Dallas, Texas:

```

120 •  SELECT * FROM Flight WHERE Origin = 'Dallas';
121
122

```

	Flight_ID	Flight_Number	Aircraft_ID	Departure	Arrival	Origin	Destination
▶	114	SN213	14	2024-11-15 17:00:00	2024-11-15 20:00:00	Dallas	Denver
	160	SN259	10	2024-11-15 17:00:00	2024-11-15 20:00:00	Dallas	Seattle
	436	SN535	11	2024-11-15 17:00:00	2024-11-15 20:00:00	Dallas	Nashville
	482	SN581	7	2024-11-15 17:00:00	2024-11-15 20:00:00	Dallas	Philadelphia
	507	SN606	7	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Los Angeles
	511	SN610	11	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Denver
	513	SN612	13	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Seattle
	521	SN620	21	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Salt Lake City
	529	SN628	4	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Chicago
	533	SN632	8	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Las Vegas
	535	SN634	10	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	San Francisco
	537	SN636	12	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	New York
	539	SN638	14	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Austin
	541	SN640	16	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Orlando
	549	SN648	24	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Salem
	551	SN650	1	2024-11-15 09:00:00	2024-11-15 12:00:00	Dallas	Louisville
	576	SN675	1	2024-11-15 17:00:00	2024-11-15 20:00:00	Dallas	Chicago

**UPDATE:** Changing the arrival time of flight number SN3005:

```

122 •  UPDATE Flight
123      SET Arrival = '2024-11-17 07:00:00'
124      WHERE Flight_Number = 'SN3005';
125 •  SELECT * FROM Flight WHERE Flight_Number = 'SN3005';

```

	Flight_ID	Flight_Number	Aircraft_ID	Departure	Arrival	Origin	Destination
▶	3005	SN3005	7	2024-11-17 02:00:00	2024-11-17 07:00:00	Dallas	New York City
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**DELETE:** Deleting flight number SN3005:

```
127 •    DELETE FROM Flight  
128      WHERE Flight_Number = 'SN3005';  
129 •    SELECT * FROM Flight WHERE Flight_Number = 'SN3005';  
130
```

Result Grid    Filter Rows: <input type="text"/> Edit:     Export/Import:							
	Flight_ID	Flight_Number	Aircraft_ID	Departure	Arrival	Origin	Destination
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 3.5. Create View:

**3.5.1 View Table:** Creates a view of the passenger data and the checked-in reservations associated with each passenger.

	Passenger_ID	First_Name	Last_Name	Date_Of_Birth	Reservation_ID	Booking_Reference_Number	Is_Checked_In	Flight_ID	Flight_Number	Aircraft_ID	Departure	Arrival	Origin	Destination	Total_Baggage	Total_Baggage_Weight
1	John	Doe		1980-04-12	1	BR123456	0	1	SN100	1	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Miami	1	23.50
2	Jane	Smith		1990-11-23	2	BR234567	1	2	SN101	2	2024-11-15 12:00:00	2024-11-15 20:00:00	Miami	Los Angeles	1	18.75
1	John	Doe		1980-04-12	2	BR234567	1	2	SN101	2	2024-11-15 17:00:00	2024-11-15 20:00:00	Miami	Los Angeles	1	23.50
3	Michael	Jones		1978-06-05	3	BR345678	0	3	SN102	3	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Denver	1	21.00
2	Jane	Smith		1990-11-23	3	BR345678	0	3	SN102	3	2024-11-15 12:00:00	2024-11-15 20:00:00	Los Angeles	Denver	1	18.75
4	Emily	Davis		1980-02-17	4	BR456789	1	4	SN103	4	2024-11-15 17:00:00	2024-11-15 20:00:00	Denver	Los Angeles	1	19.30
3	Michael	Jones		1978-06-05	4	BR456789	1	4	SN103	4	2024-11-15 17:00:00	2024-11-15 20:00:00	Denver	Los Angeles	1	21.00
5	Danielle	Bryan		1995-09-09	5	BR567890	0	5	SN104	5	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Seattle	1	22.80
4	Emily	Davis		1980-02-17	5	BR567890	0	5	SN104	5	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Seattle	1	19.30
6	Wei Yuan	Lew		1990-07-22	6	BR678901	1	6	SN105	6	2024-11-15 17:00:00	2024-11-15 20:00:00	Seattle	Los Angeles	1	25.40
5	Danielle	Bryan		1995-09-09	6	BR678901	1	6	SN105	6	2024-11-15 17:00:00	2024-11-15 20:00:00	Seattle	Los Angeles	1	22.80
7	Tanvi	Nar		2001-01-15	7	BR789012	0	7	SN106	7	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Houston	1	24.10
8	Dandan	Wang		1988-12-01	8	BR890123	1	8	SN107	8	2024-11-15 17:00:00	2024-11-15 20:00:00	Houston	Los Angeles	1	20.50
9	Mandy	Hardono		1993-03-19	9	BR901234	0	9	SN108	9	2024-11-15 17:00:00	2024-11-15 12:00:00	Los Angeles	Boston	1	23.75
10	Laura	Pinto		1986-05-30	10	BR012345	1	10	SN109	10	2024-11-15 17:00:00	2024-11-15 20:00:00	Boston	Los Angeles	1	19.20
11	Meyli	Colmenero		1998-08-24	11	BR112233	0	11	SN110	11	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Washington	1	22.90
12	Fernando	Portillo		1991-10-07	12	BR223344	1	12	SN111	12	2024-11-15 17:00:00	2024-11-15 20:00:00	Washington	Los Angeles	1	26.10
13	Shreyaa	Arun		1999-04-14	13	BR334455	0	13	SN112	13	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Salt Lake City	1	24.30
14	Hunter	Matthews		1987-02-10	14	BR445566	1	14	SN113	14	2024-11-15 17:00:00	2024-11-15 20:00:00	Salt Lake City	Los Angeles	1	25.00
15	Ebru	Cankaya		1994-06-26	15	BR556677	0	15	SN114	15	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Portland	1	23.40
16	Robert	Taylor		1980-09-05	16	BR667788	1	16	SN115	16	2024-11-15 17:00:00	2024-11-15 20:00:00	Portland	Los Angeles	1	22.75
▶ 17	Patricia	Anderson		1975-01-28	17	BR778899	0	17	SN116	17	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Nashville	1	24.50
18	George	Lewis		1968-07-18	18	BR889900	1	18	SN117	18	2024-11-15 17:00:00	2024-11-15 20:00:00	Nashville	Los Angeles	1	21.30
19	Jessica	Walker		2000-11-11	19	BR990011	0	19	SN118	19	2024-11-15 09:00:00	2024-11-15 12:00:00	Los Angeles	Philadelphia	1	25.60
20	Henry	King		1989-02-22	20	BR001122	1	20	SN119	20	2024-11-15 17:00:00	2024-11-15 20:00:00	Philadelphia	Los Angeles	1	23.90

**3.5.2 View Table:** Creates a view of all Boeing flights, listing flight numbers, aircraft models, flight origins with destinations

	Flight_Number	Model	Origin	Destination
▶	SN123	Boeing 737	New York	Los Angeles
	SN789	Boeing 777	Dallas	Denver
	SN104	Boeing 787	Atlanta	Washington DC
	SN107	Boeing 747	Austin	Nashville
	SN108	Boeing 767	Detroit	Philadelphia
	SN111	Boeing 757	San Francisco	Chicago
	SN112	Boeing 717	Orlando	Atlanta
	SN115	Boeing 727	Los Angeles	San Francisco
	SN119	Boeing 720	Seattle	Atlanta
	SN120	Boeing 707	Miami	Atlanta

**3.5.3 View Table:** Creates a view of all Airbus flights, with relevant flight information.

	Flight_Number	Model	Origin	Destination
▶	SN456	Airbus A320	Chicago	Miami
	SN101	Airbus A330	San Francisco	Seattle
	SN105	Airbus A350	Phoenix	Salt Lake City
	SN109	Airbus A380	Las Vegas	Los Angeles
	SN110	Airbus A220	Houston	Dallas
	SN116	Airbus A310	Miami	Houston
	SN118	Airbus A321	Dallas	New York
	SN121	Airbus A319	Salem	Austin

**3.5.4 View Table:** Creates a view of all Embraer Flights.

	Flight_Number	Model	Origin	Destination
▶	SN102	Embraer 175	Las Vegas	Houston
	SN113	Embraer 190	Phoenix	Las Vegas
	SN122	Embraer E170	Louisville	Houston

**3.5.5 View Table:** Creates a view of all Cessna Flights.

	Flight_Number	Model	Origin	Destination
▶	SN106	Cessna 208	San Diego	Portland

**3.5.6 View Table:** Creates a view of all Bombardier Flights.

	Flight_Number	Model	Origin	Destination
▶	SN103	Bombardier Q400	Orlando	Boston
	SN114	Bombardier CRJ700	New York	Washington DC

**3.5.7 View Table:** Creates a view of all McDonnell Douglas Flights.

	Flight_Number	Model	Origin	Destination
▶	SN117	McDonnell Douglas MD-80	Denver	Seattle

## 4. Front End User Interface:

Snoopy Airlines  Home

**Manage Trip / Check-in**  
View or update your trip details, check in for your flight, and access boarding passes.

Booking Reference Number  
BR390637

Search



Explore the World  
Book your next adventure with us

**Explore our highlighted aircraft models**  
Select a company to view their available aircraft models.

Boeing

**Aircraft for Boeing**

Boeing 737

Wifi: No  
Power Outlets: Yes  
Seats: 160

Boeing 777

Snoopy Airlines  Home

**Available Flights Based on Your Search**  
Here are the available outbound and return flights based on your search criteria.

**Pick your Outbound Flight**

**SN6148**  
Airbus A319 - 156 Seats - Has Wifi - Has Power Outlets

Origin	Destination
Dallas	Chicago

Departure Nov 20, 2024 09:00 AM Arrival Nov 20, 2024 12:00 PM

**SN6195**  
Airbus A321 - 220 Seats - Has Wifi - Has Power Outlets

Origin	Destination
Dallas	Chicago

Departure Nov 20, 2024 05:00 PM Arrival Nov 20, 2024 08:00 PM

**Pick your Returning Flight**

**SN9461**  
Airbus A380 - 555 Seats - Has Wifi - Has Power Outlets

Origin	Destination
Chicago	Dallas

**Reservation Contact Information**

Please fill in details for Reservation Contacts.

**Contact 1**

Phone

2 3 9 • 1 2 3 • 1 2 3 4

Email

john@gmail.com

[Remove Contact](#)[Add Contact](#)**Passenger Information**

Please fill in the details for 1 passenger(s).

**Passenger 1**

First Name

John

Last Name

Hunger

Date of Birth

12/06/1998 **Booking Confirmation**

Thank you for booking with us! Here are your booking details:

**Departure Flight**

Booking ID: BR390637

Flight: SN6148 from Dallas to Chicago

Departure Time: 11/20/2024, 9:00:00 AM

Arrival Time: 11/20/2024, 12:00:00 PM

**Passenger Details:**

- John Hunger (DOB: 12/6/1998) (1x Baggage)

**Returning Flight**

Booking ID: BR478909

Flight: SN9461 from Chicago to Dallas

Departure Time: 11/23/2024, 5:00:00 PM

**Passenger Details:**

- John Hunger (DOB: 12/6/1998) (1x Baggage)

[Go Back Home](#)

## Manage Your Trip

Review and manage your flight details, passenger information, and options for check-in.

Check-in ahead of time by clicking the button below!

**Flight Information**  
Booking Reference: BR390637

**Flight:** SN6148 from Dallas to Chicago  
**Departure:** 11/20/2024, 9:00:00 AM  
**Arrival:** 11/20/2024, 12:00:00 PM

**John Hunger (1x Baggage)**  
DOB: 12/6/1998

[Edit Name](#)

[Cancel Trip](#)

[Check In for Flight](#)

[Back to Home](#)



## Manage Your Trip

Review and manage your flight details, passenger information, and options for check-in.

**Flight Information**  
Booking Reference: BR390637

**Flight:** SN6148 from Dallas to Chicago  
**Departure:** 11/20/2024, 9:00:00 AM  
**Arrival:** 11/20/2024, 12:00:00 PM

**John Hunger (1x Baggage)**  
DOB: 12/6/1998

[Edit Name](#)

[Cancel Trip](#)

[Back to Home](#)



**Manage Your Trip**

Review and manage your flight details, passenger information, and options for check-in.

**Flight Information**

Booking Reference: BR390637

Flight: SN6148 from Dallas to Chicago

Departure: 11/20/2024, 9:00:00 AM

Arrival: 11/20/2024, 12:00:00 PM

**Fernando Hunger (1x Baggage)**

DOB: 12/6/1998

Edit Name

Cancel Trip

Back to Home

**Snoopy Airlines**  [Home](#)

**Oops!**

We couldn't load your booking details.

There was an error retrieving the information for this booking.

Please try again later, or return to the homepage to start over.

Back to Home

## 5. Conclusion and Future Work:

In conclusion, Snoopy Airlines showcases the significant impact of effective data management and creation in the aviation industry. By focussing on accurate and consistent data, our project highlights how a well-designed database can address key challenges such as scheduling errors, passenger information, and operational inefficiencies.

As the aviation industry continues to expand, the importance of efficient data management systems will only grow, making projects like Snoopy Airlines critical to future advancements. We believe that our work can serve as a foundation for more comprehensive systems in the future, paving the way for airlines to operate with safety and reliability.

Working on this project allowed us to understand the importance of database management and security. As a team, we were able to work together to create this project while learning from our mistakes, communicating with the team and learning/implementing the different topics to further understand the course and importance of these topics.

Database security could be a future implementation in this project. Although this project was developed with careful consideration of multiple factors, security could be one of those factors that can be worked on, especially when it comes to airline databases because most of the information like passenger and booking ID are highly sensitive and confidential. Therefore, implementing that would be beneficial to airline databases and website design.

## 6. References → Laura

[1]S. I. Marketing, “Airport Operational Database: Why it Matters for Aviation,” *Airport Gurus*, Mar. 02, 2023.

<https://www.airportgurus.com/en/airport-operational-database-why-it-matters-for-aviation/>

[2]“How Data Unlocks Huge Benefits in the Aviation Industry | Aviation Week Network,” *aviationweek.com*.

<https://aviationweek.com/air-transport/how-data-unlocks-huge-benefits-aviation-industry>

[3]A. Data, “EXSYN Aviation Solutions,” *EXSYN Aviation Solutions*, Jul. 27, 2022.

<https://www.exsyn.com/blog/what-is-aircraft-data-management-and-why-its-important> (accessed Oct. 13, 2024).

[4] “Flight status,” aa.com. <https://www.aa.com/travelInformation/flights/status> (accessed Oct. 15, 2024)

[5] “How to Design Database For Airline Companies?,” GeeksforGeeks.org, May 24, 2024.

<https://www.geeksforgeeks.org/design-database-for-airline-companies/> (accesses Oct. 15, 2024)