

## CSS325 Project Final Report



Thai Airways International

6422770766 Chanikarn Wanla  
6422771194 Kanyarat Monklung  
6422771699 Burassakorn Hannirun  
6422771921 Napatr Sawamipak  
6422771004 Chayaphot Tantichotiwat  
6422771731 Putawan Ponoi

CSS325 Database System

Asst. Prof. Dr. Preecha Tangworakitthaworn

**Sirindhorn International Institute of Technology**

**Thammasat University**

December 2, 2023

## **Abstract**

This comprehensive report encapsulates the progressive journey of the CSS325 Database System group project through its three distinct phases. Phase 1 initiated the project with a focus on data modeling for Thai Airways International, providing insights into business rules, constraints, and the identification of business goals. Phase 2 seamlessly transitioned to the implementation phase, refining Entity-Relationship Diagrams and Extended Entity-Relationship Diagrams to establish a functional database for the airline industry.

Building on these foundations, Phase 3 shifts the focus to data querying, where SQL scripts are meticulously crafted to extract meaningful insights. Each group member contributes a minimum of 10 SELECT queries, combining basic queries that utilize computed columns, built-in functions, and SQL operators with advanced queries that involve subqueries, joins, or set operators. The results aim to empower business administrators with valuable information for strategic decision-making.

Throughout this multifaceted project, the dedication and cooperation of every group member have been instrumental. Special thanks are extended to Assistant Professor Dr. Preecha Tangworakitthaworn for orchestrating this invaluable educational opportunity, reflecting the collaborative spirit and academic excellence of the CSS325 Database System group

# Table of Contents

Sirindhorn International Institute of Technology Thammasat University	1
Abstract	2
Table of Contents	3
1 Introduction	4
2 Company Profile and Background	4
2.1 Information about business domain	4
2.2 History	5
2.2.1 Thai Airways International at present	5
2.2.2 Vision and goal	6
3 Revised ERD/EERD	7
4 Mini ERD/EERD	8
5 8-steps to transform ERD/EERD to relational schema	9
6 Final relational database schema	12
7 Data Dictionary	13
8 Data Definition Language	16
9 Data Manipulation Language	22
10 Basic SQL queries	28
11 Advance SQL queries	51
12 Conclusion	68
13 Reference	69

## **1 Introduction**

This project is our opportunity for our group to practice the data modeling process for the airline industry, the efficient management of information is vital.

A robust database system serves as the backbone, storing and organizing critical data like flight schedules, passenger reservations, and maintenance records. This system streamlines operations, improves decision-making, and ensures safety and customer satisfaction. As the industry evolves, advanced database systems capable of handling massive data volumes become increasingly crucial. In this introduction, we will explore the central role of database systems in aviation, covering their applications, challenges, and future prospects.

We hope that this report will be useful in the future. And we apologize for any mistakes done in this report.

## **2 Company Profile and Background**

Thai Airways International Public Co., Ltd., commonly known as Thai Airways, is the national flag carrier and largest airline of Thailand. It is headquartered in Bangkok and operates both domestic and international flights to various destinations worldwide. Thai Airways is also a member of the Star Alliance, one of the world's largest global airline alliances.

### **2.1 Information about Business Domain**

Thai Airways International Public Co., Ltd., also known as Thai Airways, is a company that works in the aviation industry. Its website, [www.thaiairways.com](http://www.thaiairways.com), serves as the primary online platform for a variety of customer-centric processes. This website provides travelers with a variety of services and information. Thai Airways' digital hub facilitates member registration, booking, payment, check-in, boarding, and additional services processes, all while providing transparent information and enriching the travel experience for its customers.

The website allows customers to easily explore flight details, destinations, and available services. Travelers can reserve flights to both domestic and international destinations using the booking process. Individuals can access these services and unlock exclusive benefits by completing a registration and login

process. The payment process ensures safe and easy transactions for travelers, while the check-in process provides online and convenient options, allowing passengers to choose seats and receive digital boarding passes. The boarding procedure focuses on efficiently managing passenger boarding, ensuring that the correct passengers are on the correct flights, and that all safety and security measures are in place. Finally, the additional service process allows passengers to personalize their journey by selecting options such as special meal requests or seat preferences.

Moreover, The website includes a "Financial Info" section for potential investors looking for more information about the company. This section contains detailed information about financial performance, total revenue, EBITDA, net loss, and annual reports.

## **2.2 History**

Thai Airways International Public Co., Ltd., or Thai Airways, is Thailand's national flag carrier and largest airline, having been founded in 1960. The airline, headquartered in Bangkok, began operations in 1960 as a joint venture between Thai Airways Company (TAC) and Scandinavian Airlines System (SAS). The Thai government took full ownership of the airline in 1977, making it a state-owned enterprise.

Thai Airways has grown into a major player in the global aviation industry over the years. It has steadily expanded its route network, now serving both domestic and international destinations worldwide. Thai Airways has been a member of the Star Alliance since its inception in 1997, strengthening its global presence and offering passengers extensive connectivity.

### **2.2.1 Thai Airways International at present**

Thai Airways International Public Co. Ltd (THAI) is a national company under the Ministry of Transport, listed on the Stock Exchange of Thailand since 1991, and more than 50% owned by the Ministry of Finance. Thai Airways specializes in passenger and cargo transportation and is headquartered at Suvarnabhumi Airport. In addition to its core airline business, the company is also involved in passenger services, in-flight catering, aircraft maintenance, and duty-free retail, with 95% of its revenue coming from passenger and cargo transportation and the remaining 5% from supporting businesses. Thai Airways holds a significant stake in Nok Air and launched Thai Smile, a regional airline, in 2012. In 2023, Thai Smile is scheduled to merge back with Thai Airways. The airline flies to 40 international destinations with a fleet of 46 different aircraft, focusing mainly on Europe and Asia. At the end of 2022, Thai Airways employed approximately 14,400 people.

### **2.2.2 Vision and Goal**

Thai Airways, the national airline, announces its vision to introduce its goal to the public and use it as a guideline for all units to achieve the same goal: "A high quality full service carrier with strong Thai brand connecting Thailand to the world and generating consistently healthy profit margin". THAI aims to offer world-class services and ensure brand value creation in order to create strong and sustainable growth and to enable the Company to compete successfully in the ever changing business environment. The focus is on building shared values to enhance the main value.

In pursuit of their mission, Thai Airways International focuses on several key areas: They prioritize product customization to ensure passenger satisfaction by offering tailored products for a superior customer experience. Additionally, they aim to optimize their online commerce capabilities, streamlining operations to enhance income and profitability, and drive revenue from related ventures. The airline also places a strong emphasis on cost competitiveness in the regional airline market while maintaining highly efficient operations with a focus on safety, serving as a vital hub connecting various cities in Thailand and around the globe.

### 3 A revised ERD/EERD

The initial diagrams in the Thai Airways project underwent a thorough review and possible modifications, which are included in the revised ERD/EERD section. By incorporating changes to relationships, entities, or attributes to achieve improved precision, this crucial step guarantees that the data model accurately reflects the airline's changing needs. Raising the data model's efficacy is the main goal since it will pave the way for a more precise relational database schema in later project stages. The precise changes made to the ERD/EERD diagrams to better align them with the project's changing requirements for increased overall accuracy and effectiveness are detailed in this section.

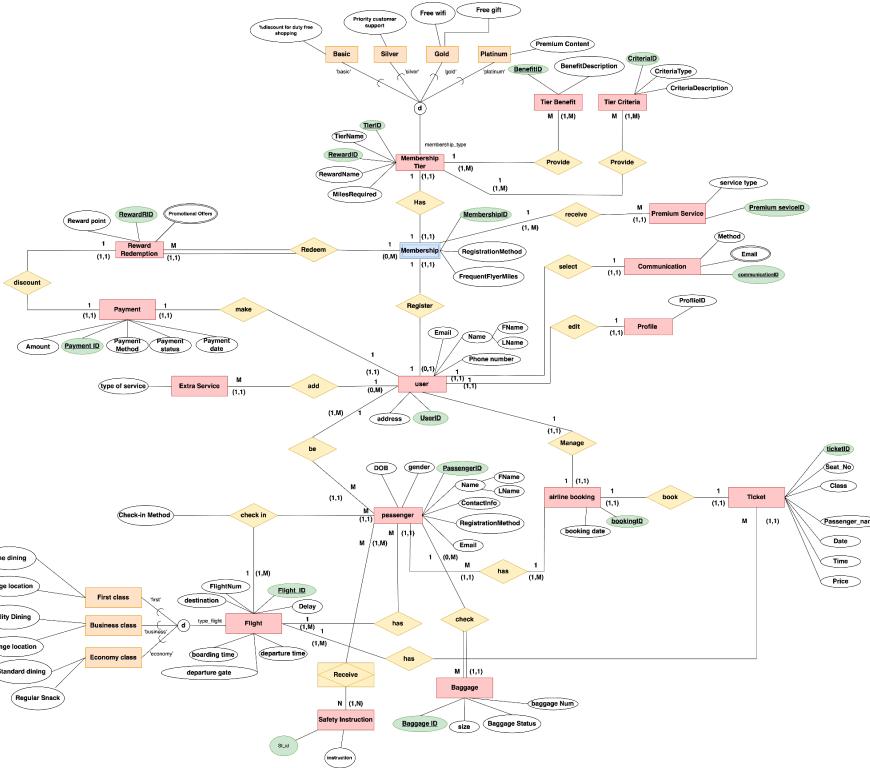


Figure 1 : Revised ERD/EERD

For more detail visit:

[https://drive.google.com/file/d/1RyALOSsrtMMIFmDG\\_YrsPVRwcJ7EbG8V/view?usp=sharing](https://drive.google.com/file/d/1RyALOSsrtMMIFmDG_YrsPVRwcJ7EbG8V/view?usp=sharing)

## 4 Mini ERD/EERD

Following the revision of the Entity-Relationship Diagrams (ERD) and Extended Entity-Relationship Diagrams (EERD) in the Thai Airways project, the next step involves the development of a Mini ERD/EERD. Serving as both a simplification tool and a visual guide for the transformation process, the Mini ERD/EERD distills complex data models into a more comprehensible format. By emphasizing key relationships, entities, and attributes, it enhances team understanding and aids in the systematic conversion of these simplified diagrams into a relational database schema. This strategic simplification ensures efficiency in the transformation process, contributing to the overarching goal of creating a precise and practical database system for Thai Airways.

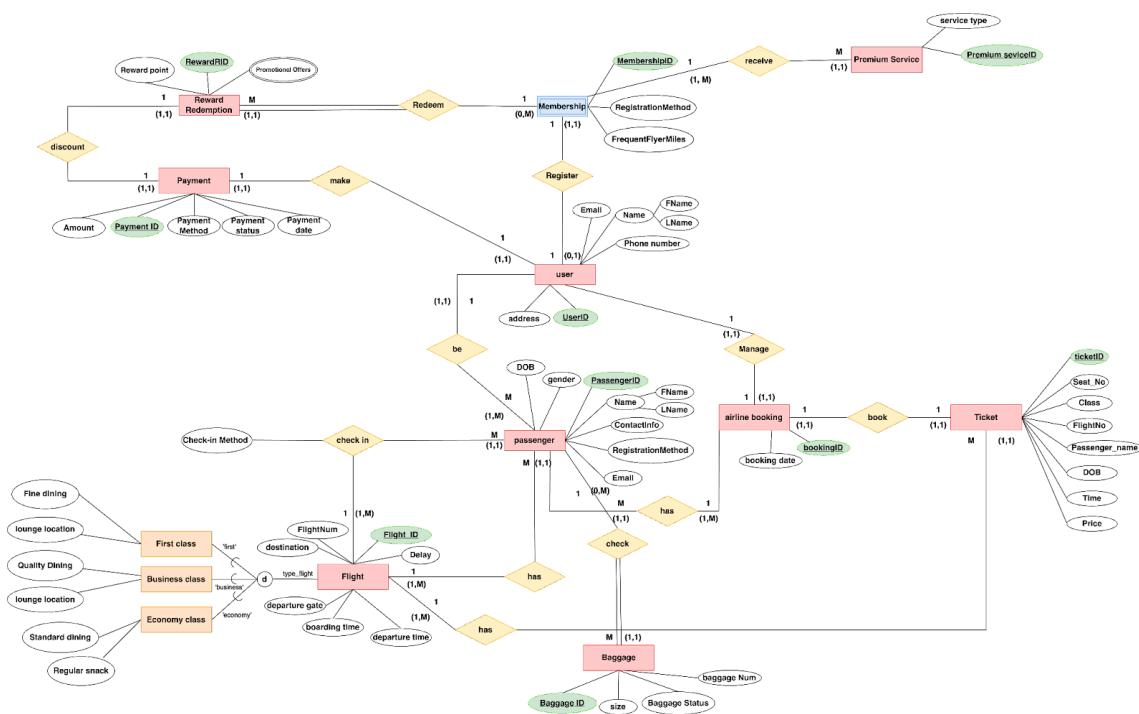


Figure 2 : Mini ERD/EERD

For more detail visit:

[https://drive.google.com/file/d/1RyALOSsrtMMIFmDG\\_YrsPVRwcJ7EbG8V/view?usp=sharing](https://drive.google.com/file/d/1RyALOSsrtMMIFmDG_YrsPVRwcJ7EbG8V/view?usp=sharing)

## 5 8-steps to transform ERD/EERD to relational schema

The step-by-step transformation process is a critical component in the conversion of theoretical Entity-Relationship Diagrams (ERD) and Extended Entity-Relationship Diagrams (EERD) into a functional relational database schema for the Thai Airways project. This systematic approach ensures a smooth transition from conceptual models to a practical database structure. Each step involves a detailed analysis of the diagrams, addressing nuances to accurately represent relationships and entities in the relational schema. The objective is to define tables, primary keys, foreign keys, and any necessary constraints, laying the groundwork for subsequent phases of database development tailored to the specific needs of Thai Airways.

### STEP 1: Transform regular entity type

User							
UserID	FName	LName	Email	PhoneNumber	Address		
Payment							
PaymentID	PaymentMethod	PaymentStatus	Amount	PaymentDate			
Ticket							
TicketID	SeatNo	Class	PassengerName	Date	Time	Price	
Passenger							
PassengerID	FName	LName	ContactInfo	RegisterMethod	Email	DOB	Gender
Premium Service							
PremiumServiceID	ServiceType						
Baggage							
BaggageID	Size	BaggageStatus	BaggageNum				
Flight							
FlightID	FlightNum	Destination	Delay	DepartureGate	BoardingTime	DepartureTime	
Airline Booking							
BookingID	BookingDate						
Reward Redemption							
RewardRID	RewardPoint						

Figure 3 : transform regular entity type

## STEP 2: Transform weak entity

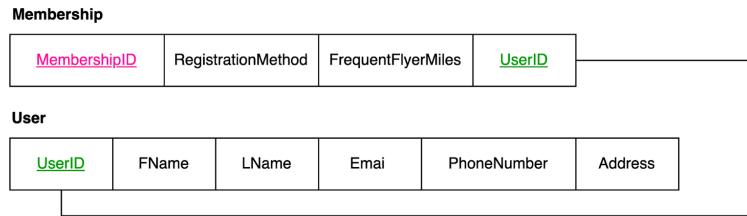


Figure 4 : transform weak entity

## STEP 3: Transform 1:1 relationship

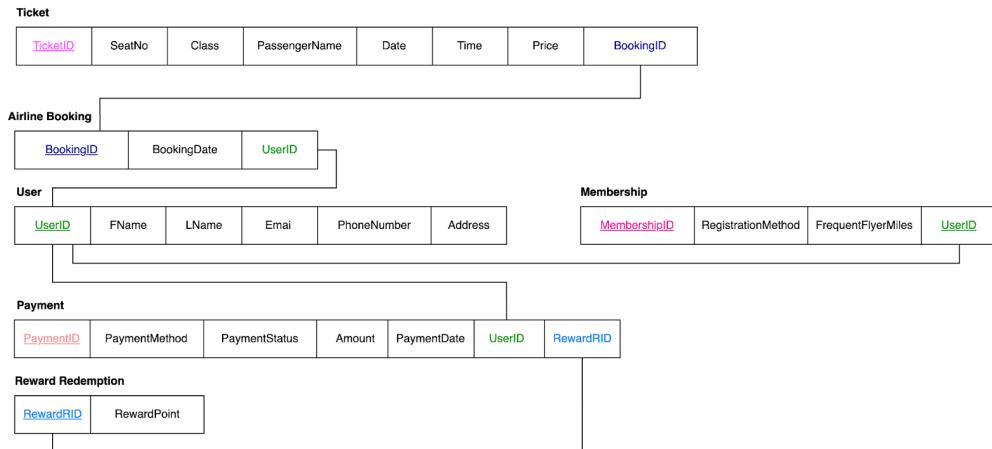


Figure 5 : transform 1:1 relationship

## STEP 4: Transform 1:M relationship



Figure 6 : transform 1:M relationship

## STEP 6: Transform Multi-valued

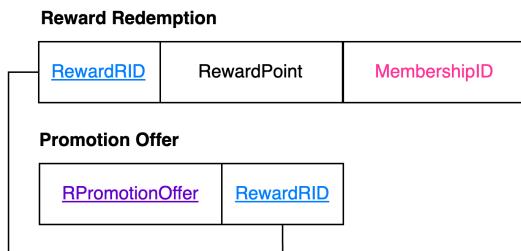


Figure 7 : transform multi-valued

## STEP 8: Transform generalization

Flight											
FlightID	FlightNum	Destination	Delay	DepartureGate	BoardingTime	DepartureTime	Flight_Type	Standard_Dining	Lounge_Location	Quantity_Dining	Fine_Dining

Figure 8 : transform generalization

## 6 Final relation schema

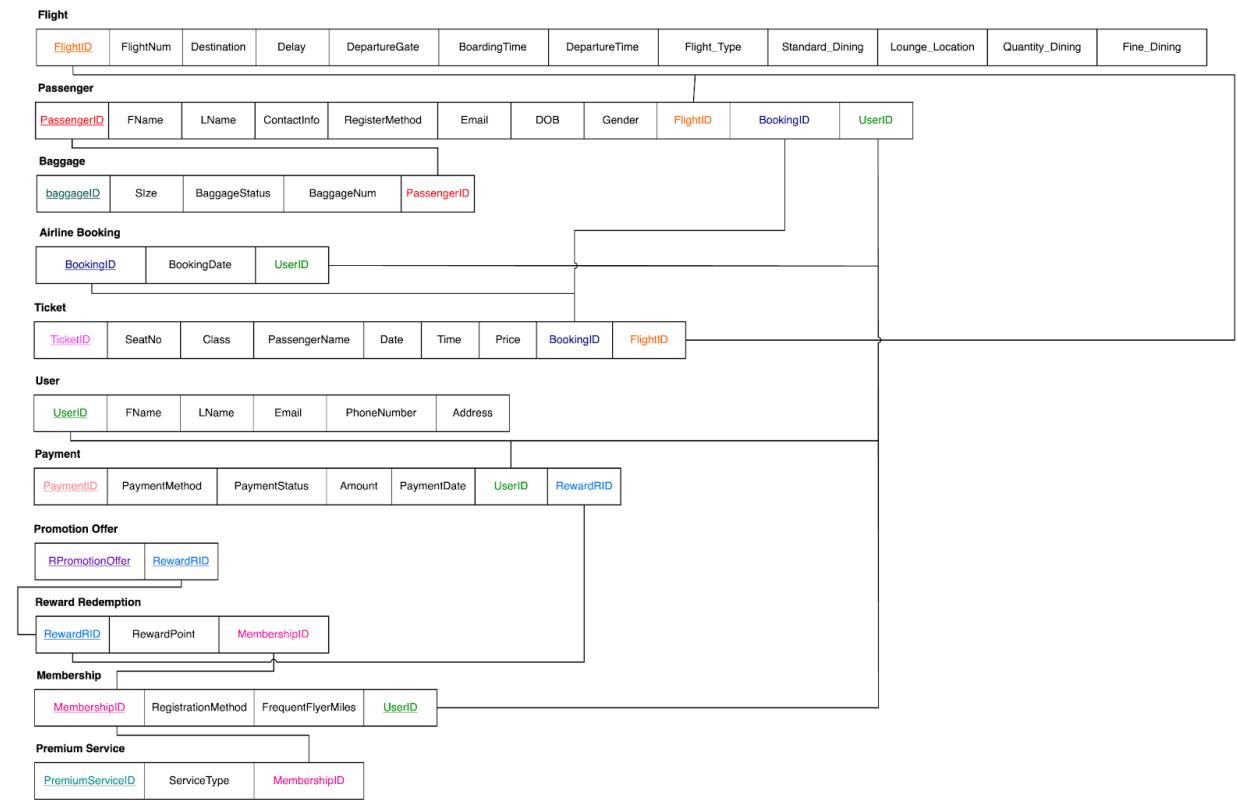


Figure 9 : final relation schema

For more detail visit:

[https://drive.google.com/file/d/1RyALOSrtMMIFmDG\\_YrsPVRwcJ7EbG8V/view?usp=sharing](https://drive.google.com/file/d/1RyALOSrtMMIFmDG_YrsPVRwcJ7EbG8V/view?usp=sharing)

## 7 Data Dictionary

Table Name	Attribute Name	Contents	Type	Format	Nullable	Range	Key	FK reference table
User	UserID	User's ID	int(11)	x			PK	
	FName	User's first name	varchar(50)	Xxxxx				
	LName	User's last name	varchar(50)	Xxxxx				
	Email	User's email	varchar(255)	Xxxx@Xxxx.com				
	PhoneNumber	User's phone number	int(11)	Xxx-xxx-xxxx				
	Address	User's address	varchar(255)	Xxxx				
Passenger	PassengerID	Passenger's ID	int(11)	x			PK	
	FName	Passenger's first name	varchar(50)	Xxxx				
	LName	Passenger's last name	varchar(50)	Xxxx				
	ContactInfo	Passenger's Contact	varchar(50)	Xxxx				
	RegisterM	Register Method	varchar(50)	Xxxx				
	Email	Passenger's email	varchar(50)	Xxxx@Xxxx.com				
	DOB	Date of birth	date	yyyy-mm-dd				
	Gender	Passenger's Sex	varchar(11)	x				
	FlightID	Flight's ID	int(11)					
	BookingID	AirlineBooking's ID	int(11)					
Flight	UserID	User's ID	int(11)					
	FlightID	Flight's ID	int(11)	x			PK	
	FlightNum	Flight Number	varchar(20)	x				
	Destination	Flight's Destination	varchar(50)	Xxxx				
Delay	Delay	Flight's Delay	varchar(50)	Xxxx				

	DepartureGate	Flight's Departure Gate	varchar(50)	Xxxx				
	Boarding Time	Flight's Boarding Time	time	HH:MM XM				
	DepartureTime	Flight's Departure Time	time	HH:MM XM				
	Flight_Type	Flight Type	varchar(50)	Xxxx				
	Standard_Dining	Flight's Standard Dining	varchar(50)	Xxxx				
	Lounge_Location	Flight's Lounge Location	varchar(50)	Xxxx				
	Quantity_Dining	Flight's Quantity Dining	varchar(50)	Xxxx				
	Fine_Dining	Flight's Fine Dining	varchar(50)	Xxxx				
Reward Redemption	RewardRID	Reward Redemption's ID	int(11)	x			PK	
	RewardPoint	Reward Point	int(11)	x				
	MembershipID	Membership's ID	int(11)	x			FK	Membership [Membershi pID]
Baggage	baggagelD	Baggage's ID	int(11)	x			PK	
	Size	Size of baggage	int(11)	x				
	BaggageStatus	Status of Baggage	varchar(255)	Xxxx				
	BaggageNumber	Number of Baggage	int(11)	x				
	PassengerID	Passenger's ID	int(11)	x			FK	Passenger [Passengerl D]
AirlineBooking	BookingID	AirlineBooking's ID	int(11)	x			PK	
	BookingDate	AirlineBooking Date	date	yyyy-mm-dd				
	UserID	User's ID	int(11)	x			FK	User [UserID]
Ticket	TicketID	Ticket's ID	int(11)	x			PK	
	SeatNo	Ticket Seat Number	varchar(20)	xx				

	Class	Ticket Class	varchar(50)	xx				
	PassengerName	Ticket Passenger Name	varchar(255)	Xxxx				
	Date	Date on Ticket	date	yyyy-mm-dd				
	Time	Time on Ticket	time	HH:MM XM				
	Price	Ticket Price	decimal(9,2)	1234567.89				
	BookingID	Booking's ID	int(11)	x			FK	AirlineBooking [BookingID]
	FlightID	Flight's ID	int(11)	x			FK	Flight [FlightID]
Premium Service	Premium ServiceID	Premium Service's ID	int(11)	x			PK	
	ServiceType	Premium Service Type	varchar(255)	Xxxx				
	MembershipID	Membership's ID	int(11)	x			FK	Membership [MembershipID]
Membership	MembershipID	Membership's ID	int(11)	x			PK	
	RegistrationM	Register Method	varchar(50)	Xxxx				
	FrequentFlyerMiles	Miles of flying	int(100)	x				
	UserID	User's ID	int(11)	x			FK	User [UserID]
Promotion Offer	RPromotionOffer	Promotion Offer of Reward Redemption	varchar(255)	Xxxx			PK	
	RewardRID	Reward Redemption's ID	int(11)	x			FK	RewardRedemption [RewardID]
Payment	PaymentID	Payment's ID	int(11)	x			PK	
	PaymentMethod	Method of payment	varchar(50)	Xxxx				
	PaymentStatus	Status of payment	varchar(50)	Xxxx				
	Amount	Amount of money	int(11)	x				
	PaymentDate	Date of payment	date	yyyy-mm-dd				

	UserID	User's ID	int(11)	x			FK	User [UserID]
	RewardRID	Reward Redemption's ID	int(11)	x			FK	RewardRedemption [RewardID]

## 8 Data Definition Language

```
CREATE TABLE `airlinebooking` (
    `bookingID` int(11) NOT NULL,
    `bookingDate` date NOT NULL,
    `userID` int(11) NOT NULL
);

-- AUTO_INCREMENT for table `airlinebooking`
ALTER TABLE `airlinebooking`
    MODIFY `bookingID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;
```

- The `airlinebooking` table stores essential information about airline bookings, featuring columns like bookingID as a unique identifier, bookingDate for the date of the reservation, and userID linking to the user making the booking.

```
CREATE TABLE `baggage` (
    `baggageID` int(11) NOT NULL,
    `size` int(11) NOT NULL,
    `baggageStatus` varchar(255) NOT NULL,
    `passengerID` int(11) NOT NULL,
    `baggageNumber` int(11) NOT NULL
);

-- AUTO_INCREMENT for table `baggage`
ALTER TABLE `baggage`
    MODIFY `baggageID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;
```

- The `baggage` table manages details related to passengers' checked baggage. It includes columns such as baggageID for a unique identifier, size indicating the baggage size, baggageStatus for the status (e.g., checked or lost), passengerID linking to passenger details, and baggageNumber for a specific baggage identifier.

```

CREATE TABLE `flight` (
    `flightID` int(11) NOT NULL,
    `flightNum` varchar(20) NOT NULL,
    `destination` varchar(50) NOT NULL,
    `delay` varchar(50) NOT NULL,
    `departureGate` varchar(50) NOT NULL,
    `boardingTime` time NOT NULL,
    `departureTime` time NOT NULL,
    `flight_Type` varchar(50) NOT NULL,
    `standard_dining` varchar(50) NOT NULL,
    `lounge_location` varchar(50) NOT NULL,
    `quantity_dining` varchar(50) NOT NULL,
    `fine_dining` varchar(50) NOT NULL
);
-- AUTO_INCREMENT for table `flight`
ALTER TABLE `flight`
    MODIFY `flightID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;

```

- The `flight` table contains information about flights, with columns like flightID as a unique identifier, flightNum for the flight number, destination for the flight destination, and various other details such as departure gate, boarding time, departure time, and dining options.

```

CREATE TABLE `membership` (
    `membershipID` int(11) NOT NULL,
    `registrationM` varchar(50) NOT NULL,
    `frequentFlyerMiles` int(100) NOT NULL,
    `userID` int(11) NOT NULL
);
-- AUTO_INCREMENT for table `membership`
ALTER TABLE `membership`
    MODIFY `membershipID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

```

- The `membership` table is dedicated to tracking membership details, featuring columns like membershipID as a unique identifier, registrationM for the membership registration type, frequentFlyerMiles for accumulating miles, and userID linking to user details.

```
CREATE TABLE `passenger` (
    `passengerID` int(11) NOT NULL,
    `Fname` varchar(50) NOT NULL,
    `LName` varchar(50) NOT NULL,
    `contactInfo` varchar(50) NOT NULL,
    `registerM` varchar(50) NOT NULL,
    `email` varchar(50) NOT NULL,
    `dob` date NOT NULL,
    `gender` varchar(11) NOT NULL,
    `flightID` int(11) NOT NULL,
    `bookingID` int(11) NOT NULL,
    `userID` int(11) NOT NULL
);
```

```
-- AUTO_INCREMENT for table `passenger`
ALTER TABLE `passenger`
    MODIFY `passengerID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=61;
```

- The `passenger` table stores information about passengers, including columns such as passengerID as a unique identifier, Fname and LName for first and last names, contactInfo for contact information, and foreign keys linking to flight, booking, and user tables.

```
CREATE TABLE `payment` (
    `paymentID` int(11) NOT NULL,
    `paymentMethod` varchar(50) NOT NULL,
    `paymentStatus` varchar(50) NOT NULL,
    `amount` int(11) NOT NULL,
    `paymentDate` date NOT NULL,
    `userID` int(11) NOT NULL,
    `RewardID` int(11) NOT NULL
);
```

```
-- AUTO_INCREMENT for table `payment`
ALTER TABLE `payment`
    MODIFY `paymentID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=61;
```

- The `payment` table manages payment details, featuring columns like paymentID as a unique identifier, paymentMethod for the payment method, paymentStatus for the payment status, and

amount for the payment amount. It also includes foreign keys linking to the user and rewardredemption tables.

```
▶ ⊖ CREATE TABLE `premium_service` (
    `premium_serviceID` int(11) NOT NULL,
    `serviceType` varchar(255) NOT NULL,
    `membershipID` int(11) NOT NULL
);
```

```
-- AUTO_INCREMENT for table `premium_service`
ALTER TABLE `premium_service`
    MODIFY `premium_serviceID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

- The `premium\_service` table handles premium services associated with memberships. It includes columns such as premium\_serviceID as a unique identifier, serviceType for the type of premium service, and membershipID linking to membership details.

```
▶ ⊖ CREATE TABLE `promotionoffer` (
    `RpromotionOffer` int(11) NOT NULL,
    `RewardRID` int(11) NOT NULL
);
```

```
-- AUTO_INCREMENT for table `promotionoffer`
ALTER TABLE `promotionoffer`
    MODIFY `RpromotionOffer` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

- The `promotionoffer` table contains information about promotional offers, featuring columns like RpromotionOffer as a unique identifier and RewardRID linking to reward redemption details.

```

CREATE TABLE `rewardredemption` (
    `rewardRID` int(11) NOT NULL,
    `rewardPoint` int(11) NOT NULL,
    `membershipID` int(11) NOT NULL
);

-- AUTO_INCREMENT for table `rewardredemption`
ALTER TABLE `rewardredemption`
    MODIFY `rewardRID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=81;

```

- The `rewardredemption` table tracks reward redemptions, with columns like rewardRID as a unique identifier, rewardPoint for redeemed points, and foreign keys linking to membership and promotionoffer tables.

```

CREATE TABLE `ticket` (
    `ticketID` int(11) NOT NULL,
    `seatNo` varchar(20) NOT NULL,
    `class` varchar(50) NOT NULL,
    `passengerName` varchar(255) NOT NULL,
    `date` varchar(255) NOT NULL,
    `time` time NOT NULL,
    `price` decimal(10,2) NOT NULL,
    `bookingID` int(11) NOT NULL,
    `flightID` int(11) NOT NULL
);

-- AUTO_INCREMENT for table `ticket`
ALTER TABLE `ticket`
    MODIFY `ticketID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;

```

- The `ticket` table stores details about flight tickets, with columns like ticketID as a unique identifier, seatNo for the seat number, class for the ticket class, and foreign keys linking to airlinebooking and flight tables.

```

CREATE TABLE `user` (
    `userID` int(11) NOT NULL,
    `Fname` varchar(50) NOT NULL,
    `Lname` varchar(50) NOT NULL,
    `email` varchar(50) NOT NULL,
    `phoneNumber` bigint(11) NOT NULL,
    `address` varchar(255) NOT NULL
);

-- AUTO_INCREMENT for table `user`
ALTER TABLE `user`
    MODIFY `userID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

```

- The `user` table contains user information, including columns like userID as a unique identifier, Fname and Lname for the user's first and last names, email for the email address, phoneNumber for the phone number, and address for the user's address.

For more detail visit:

<https://drive.google.com/drive/folders/1wl1kB-SrocXA1sE0umSsRMvQBX5saUJq?usp=sharing>

## 9 Data Manipulation Language

<https://drive.google.com/drive/folders/1wl1kB-SrocXA1sE0umSsRMvQBX5saUJq?usp=sharing>

```
INSERT INTO `user` (`userID`, `Fname`, `Lname`, `email`, `phoneNumber`, `address`) VALUES
(1, 'John', 'Doe', 'john@example.com', 9996754576, '123 Main St, Apt 4B, Springfield, IL 62704'),
(2, 'Alice', 'Smith', 'alice@example.com', 9876543210, '456 Oak Lane, Suite 203, Denver, CO 80202'),
(3, 'Bob', 'Johnson', 'bob@example.com', 5554443333, '789 Maple Avenue, Unit 12, Raleigh, NC 27601'),
(4, 'Emma', 'Wilson', 'emma@example.com', 1112223333, '101 Pine Street, Apt 567, San Francisco, CA 94105'),
(5, 'Michael', 'Brown', 'michael@example.com', 4445556666, '234 Elm Drive, Unit 8D, Atlanta, GA 30301'),
(6, 'Sophia', 'Lee', 'sophia@example.com', 7778889999, '345 Cedar Court, Apt 101, Dallas, TX 75201'),
(7, 'Daniel', 'Garcia', 'daniel@example.com', 9998887777, '678 Birch Street, Suite 304, Seattle, WA 98101'),
(8, 'Olivia', 'Martinez', 'olivia@example.com', 6669991111, '901 Walnut Avenue, Apt 22, Miami, FL 33101'),
(9, 'William', 'Lopez', 'william@example.com', 2223334444, '1123 Spruce Lane, Unit 45, New York, NY 10001'),
(10, 'Ava', 'Gonzalez', 'ava@example.com', 8887776666, '4567 Sycamore Road, Apt 67, Chicago, IL 60601');
```

- `user` table

- `INSERT INTO `membership` (`membershipID`, `registrationM`, `frequentFlyerMiles`, `userID`) VALUES`  
(1, 'Gold', 10000, 1),  
(2, 'Silver', 7500, 2),  
(3, 'Bronze', 5000, 3),  
(4, 'Gold', 11000, 4),  
(5, 'Silver', 8000, 5),  
(6, 'Bronze', 5500, 6),  
(7, 'Gold', 10500, 7),  
(8, 'Silver', 7000, 8),  
(9, 'Bronze', 5200, 9),  
(10, 'Gold', 9500, 10);

- `membership` table

```
INSERT INTO `rewardredemption` (`rewardRID`, `rewardPoint`, `membershipID`) VALUES
(51, 50, 1),
(52, 60, 2),
(53, 70, 3),
(54, 80, 4),
(55, 90, 5),
(56, 100, 6),
(57, 110, 7),
(58, 120, 8),
(59, 130, 9),
(60, 140, 10),
(61, 150, 1),
(62, 160, 2),
(63, 170, 3),
(64, 180, 4),
(65, 190, 5),
(66, 200, 6),
(67, 210, 7),
(68, 220, 8),
(69, 230, 9),
(70, 240, 10),
(71, 250, 1),
(72, 260, 2),
(73, 270, 3),
(74, 280, 4),
(75, 290, 5),
(76, 300, 6),
(77, 310, 7),
(78, 320, 8),
(79, 330, 9),
(80, 340, 10);
```

`rewardredemption` table

```
INSERT INTO `promotionoffer` (`RpromotionOffer`, `RewardRID`) VALUES
(1, 51),
(2, 52),
(3, 53),
(4, 54),
(5, 55),
(6, 56),
(7, 57),
(8, 58),
(9, 59),
(10, 60);
```

`promotionoffer` table

```

INSERT INTO `flight` ('flightID', 'flightnum', 'destination', 'delay', 'departuregate', 'boardingtime', 'departureTime', 'flight_type', 'standard_dining', 'lounge_location', 'quantity_dining', 'fine_dining') VALUES
(1, 'AA123', 'New York', 'No Delay', 'Gate A1', '12:00:00', '13:00:00', 'Domestic', 'Yes', 'Lounge A', 'Full', 'Yes'),
(2, 'DL456', 'Los Angeles', '10 mins', 'Gate 82', '14:30:00', '15:30:00', 'Domestic', 'No', 'Lounge B', 'Limited', 'No'),
(3, 'UA789', 'Chicago', '5 mins', 'Gate C3', '11:00:00', '12:00:00', 'International', 'Yes', 'Lounge C', 'Full', 'Yes'),
(4, 'BA234', 'London', '20 mins', 'Gate D4', '18:45:00', '19:45:00', 'International', 'Yes', 'Lounge D', 'Full', 'Yes'),
(5, 'LH567', 'Tokyo', '15 mins', 'Gate E5', '23:30:00', '00:30:00', 'International', 'No', 'Lounge E', 'Limited', 'No'),
(6, 'AF890', 'Paris', 'No Delay', 'Gate F6', '09:00:00', '10:00:00', 'International', 'Yes', 'Lounge F', 'Full', 'Yes'),
(7, 'SQ123', 'Berlin', '10 mins', 'Gate G7', '21:15:00', '22:15:00', 'International', 'Yes', 'Lounge G', 'Full', 'Yes'),
(8, 'EK456', 'Sydney', '5 mins', 'Gate H8', '03:45:00', 'International', 'No', 'Lounge H', 'Limited', 'No'),
(9, 'QF789', 'Dubai', '30 mins', 'Gate I9', '16:20:00', '17:20:00', 'International', 'Yes', 'Lounge I', 'Full', 'Yes'),
(10, 'TK234', 'Rome', '15 mins', 'Gate J10', '07:30:00', '08:30:00', 'International', 'No', 'Lounge J', 'Limited', 'No'),
(11, 'CX567', 'Madrid', 'No Delay', 'Gate K11', '11:45:00', '12:45:00', 'International', 'Yes', 'Lounge K', 'Full', 'Yes'),
(12, 'KL890', 'Beijing', '20 mins', 'Gate L12', '19:00:00', '20:00:00', 'International', 'Yes', 'Lounge L', 'Full', 'Yes'),
(13, 'JL123', 'Humait', '25 mins', 'Gate M13', '02:40:00', '03:40:00', 'International', 'Yes', 'Lounge M', 'Full', 'Yes'),
(14, 'AV456', 'Seoul', '10 mins', 'Gate N14', '15:55:00', '16:55:00', 'International', 'No', 'Lounge N', 'Limited', 'No'),
(15, 'EV789', 'Moscow', '15 mins', 'Gate O15', '09:10:00', '10:10:00', 'International', 'Yes', 'Lounge O', 'Full', 'Yes'),
(16, 'SU234', 'Bangkok', '5 mins', 'Gate P16', '22:50:00', '23:50:00', 'International', 'Yes', 'Lounge P', 'Full', 'Yes'),
(17, 'GY567', 'Istanbul', '30 mins', 'Gate Q17', '04:15:00', '05:15:00', 'International', 'Yes', 'Lounge Q', 'Full', 'Yes'),
(18, 'AV890', 'Toronto', 'No Delay', 'Gate R18', '11:20:00', '12:20:00', 'Domestic', 'Yes', 'Lounge R', 'Full', 'Yes'),
(19, 'QF123', 'Vancouver', '20 mins', 'Gate S19', '18:00:00', '19:00:00', 'Domestic', 'Yes', 'Lounge S', 'Full', 'Yes'),
(20, 'EK456', 'Montreal', '15 mins', 'Gate T20', '09:05:00', '10:05:00', 'Domestic', 'Yes', 'Lounge T', 'Full', 'Yes'),
(21, 'UH789', 'Calgary', 'No Delay', 'Gate U21', '12:30:00', '13:30:00', 'Domestic', 'Yes', 'Lounge U', 'Full', 'Yes'),
(22, 'BA234', 'Edmonton', '10 mins', 'Gate V22', '14:40:00', '15:40:00', 'Domestic', 'No', 'Lounge V', 'Limited', 'No'),
(23, 'AA567', 'Quebec', '25 mins', 'Gate W23', '19:25:00', '20:25:00', 'Domestic', 'Yes', 'Lounge W', 'Full', 'Yes'),
(24, 'QF890', 'Ottawa', '15 mins', 'Gate X24', '08:15:00', '09:15:00', 'Domestic', 'Yes', 'Lounge X', 'Full', 'Yes'),
(25, 'KL123', 'Halifax', 'No Delay', 'Gate Y25', '10:55:00', '11:55:00', 'Domestic', 'Yes', 'Lounge Y', 'Full', 'Yes'),
(26, 'SG456', 'Saskatoon', '20 mins', 'Gate Z26', '17:50:00', '18:50:00', 'Domestic', 'No', 'Lounge Z', 'Limited', 'No'),
(27, 'CK789', 'Regina', '10 mins', 'Gate AA27', '13:00:00', '14:00:00', 'Domestic', 'Yes', 'Lounge AA', 'Full', 'Yes'),
(28, 'DL234', 'Winnipeg', '5 mins', 'Gate AB28', '16:35:00', '17:35:00', 'Domestic', 'Yes', 'Lounge AB', 'Full', 'Yes'),
(29, 'TK567', 'Charlottetown', '15 mins', 'Gate AC29', '09:40:00', '10:40:00', 'Domestic', 'No', 'Lounge AC', 'Limited', 'No'),
(30, 'AF890', 'Whitehorse', 'No Delay', 'Gate AD30', '12:50:00', '13:50:00', 'Domestic', 'Yes', 'Lounge AD', 'Full', 'Yes');

```

## 'flight' table

```

INSERT INTO `airlinebooking` ('bookingID', 'bookingDate', 'userID') VALUES
(1, '2023-11-15', 1),
(2, '2023-11-16', 2),
(3, '2023-11-17', 3),
(4, '2023-11-18', 4),
(5, '2023-11-19', 5),
(6, '2023-11-20', 6),
(7, '2023-11-21', 7),
(8, '2023-11-22', 8),
(9, '2023-11-23', 9),
(10, '2023-11-24', 10),
(11, '2023-11-25', 1),
(12, '2023-11-26', 2),
(13, '2023-11-27', 3),
(14, '2023-11-28', 4),
(15, '2023-11-29', 5),
(16, '2023-11-30', 6),
(17, '2023-12-01', 7),
(18, '2023-12-02', 8),
(19, '2023-12-03', 9),
(20, '2023-12-04', 10),
(21, '2023-12-05', 1),
(22, '2023-12-06', 2),
(23, '2023-12-07', 3),
(24, '2023-12-08', 4),
(25, '2023-12-09', 5),
(26, '2023-12-10', 6),
(27, '2023-12-11', 7),
(28, '2023-12-12', 8),
(29, '2023-12-13', 9),
(30, '2023-12-14', 10);

```

## 'airlinebooking' table

```

INSERT INTO `passenger` (`passengerID`, `Fname`, `LName`, `contactInfo`, `registerM`, `email`, `dob`, `gender`, `flightID`, `bookingID`, `userID`) VALUES
(31, 'John', 'Doe', '555-123-4567', 'Gold', 'john@example.com', '1990-05-05', 'Male', 1, 1, 1),
(32, 'Alice', 'Smith', '555-987-6543', 'Silver', 'alice@example.com', '1992-08-15', 'Female', 2, 2, 2),
(33, 'Robert', 'Johnson', '555-111-3333', 'Bronze', 'robert@example.com', '1985-11-20', 'Male', 3, 3, 3),
(34, 'Emily', 'Brown', '555-222-4444', 'Platinum', 'emily@example.com', '1988-03-12', 'Female', 4, 4, 4),
(35, 'William', 'Taylor', '555-333-5555', 'Gold', 'william@example.com', '1995-06-30', 'Male', 5, 5, 5),
(36, 'Olivia', 'Miller', '555-444-6666', 'Silver', 'olivia@example.com', '1993-09-25', 'Female', 6, 6, 6),
(37, 'Michael', 'Wilson', '555-555-7777', 'Bronze', 'michael@example.com', '1980-12-05', 'Male', 7, 7, 7),
(38, 'Sophia', 'Garcia', '555-666-8888', 'Platinum', 'sophia@example.com', '1987-02-10', 'Female', 8, 8, 8),
(39, 'Daniel', 'Martinez', '555-777-9999', 'Gold', 'daniel@example.com', '1991-04-17', 'Male', 9, 9, 9),
(40, 'Ava', 'Lopez', '555-888-0000', 'Silver', 'ava@example.com', '1994-07-22', 'Female', 10, 10, 10),
(41, 'Ethan', 'Hernandez', '555-999-1111', 'Bronze', 'ethan@example.com', '1983-10-18', 'Male', 11, 11, 1),
(42, 'Mia', 'Gonzalez', '555-111-2222', 'Platinum', 'mia@example.com', '1989-01-19', 'Female', 12, 12, 2),
(43, 'Alexander', 'Perez', '555-222-3333', 'Gold', 'alexander@example.com', '1996-08-08', 'Male', 13, 13, 3),
(44, 'Charlotte', 'Smith', '555-333-4444', 'Silver', 'charlotte@example.com', '1992-02-28', 'Female', 14, 14, 4),
(45, 'Matthew', 'Rodriguez', '555-444-5555', 'Bronze', 'matthew@example.com', '1984-05-07', 'Male', 15, 15, 5),
(46, 'Amelia', 'Lee', '555-555-6666', 'Platinum', 'amelia@example.com', '1986-12-25', 'Female', 16, 16, 6),
(47, 'David', 'Garcia', '555-666-7777', 'Gold', 'david@example.com', '1997-09-03', 'Male', 17, 17, 7),
(48, 'Abigail', 'Hernandez', '555-777-8888', 'Silver', 'abigail@example.com', '1993-10-11', 'Female', 18, 18, 8),
(49, 'James', 'Martinez', '555-888-9999', 'Bronze', 'james@example.com', '1981-01-30', 'Male', 19, 19, 9),
(50, 'Harper', 'Lopez', '555-999-0000', 'Platinum', 'harper@example.com', '1988-06-19', 'Female', 20, 20, 10),
(51, 'Logan', 'Hernandez', '555-000-1111', 'Gold', 'logan@example.com', '1995-03-14', 'Male', 21, 21, 1),
(52, 'Evelyn', 'Gonzalez', '555-111-2222', 'Silver', 'evelyn@example.com', '1982-04-23', 'Female', 22, 22, 2),
(53, 'Benjamin', 'Perez', '555-222-3333', 'Bronze', 'benjamin@example.com', '1990-11-01', 'Male', 23, 23, 3),
(54, 'Liam', 'Rodriguez', '555-333-4444', 'Platinum', 'liam@example.com', '1987-07-09', 'Male', 24, 24, 4),
(55, 'Victoria', 'Lee', '555-444-5555', 'Gold', 'victoria@example.com', '1998-02-20', 'Female', 25, 25, 5),
(56, 'Lucas', 'Garcia', '555-555-6666', 'Silver', 'lucas@example.com', '1984-09-29', 'Male', 26, 26, 6),
(57, 'Avery', 'Hernandez', '555-666-7777', 'Bronze', 'avery@example.com', '1991-12-15', 'Female', 27, 27, 7),
(58, 'Jackson', 'Martinez', '555-777-8888', 'Platinum', 'jackson@example.com', '1989-05-02', 'Male', 28, 28, 8),
(59, 'Madison', 'Lopez', '555-888-9999', 'Gold', 'madison@example.com', '1996-06-11', 'Female', 29, 29, 9),
(60, 'Grayson', 'Johnson', '555-999-0000', 'Silver', 'grayson@example.com', '1988-03-26', 'Male', 30, 30, 10);

```

## 'passenger' table

```

INSERT INTO `baggage` (`baggageID`, `size`, `baggageStatus`, `passengerID`, `baggageNumber`) VALUES
(1, 25, 'Checked', 31, 1),
(2, 30, 'Carry-on', 32, 2),
(3, 20, 'Checked', 33, 3),
(4, 22, 'Carry-on', 34, 4),
(5, 27, 'Checked', 35, 5),
(6, 18, 'Carry-on', 36, 6),
(7, 21, 'Checked', 37, 7),
(8, 26, 'Carry-on', 38, 8),
(9, 24, 'Checked', 39, 9),
(10, 19, 'Carry-on', 40, 10),
(11, 23, 'Checked', 41, 11),
(12, 28, 'Carry-on', 42, 12),
(13, 17, 'Checked', 43, 13),
(14, 29, 'Carry-on', 44, 14),
(15, 16, 'Checked', 45, 15),
(16, 30, 'Carry-on', 46, 16),
(17, 15, 'Checked', 47, 17),
(18, 26, 'Carry-on', 48, 18),
(19, 14, 'Checked', 49, 19),
(20, 27, 'Carry-on', 50, 20),
(21, 13, 'Checked', 51, 21),
(22, 28, 'Carry-on', 52, 22),
(23, 12, 'Checked', 53, 23),
(24, 29, 'Carry-on', 54, 24),
(25, 11, 'Checked', 55, 25),
(26, 30, 'Carry-on', 56, 26),
(27, 10, 'Checked', 57, 27),
(28, 25, 'Carry-on', 58, 28),
(29, 9, 'Checked', 59, 29),
(30, 24, 'Carry-on', 60, 30);

```

## 'baggage' table

```

INSERT INTO `payment` (`paymentID`, `paymentMethod`, `paymetnStatus`, `amount`, `paymentDate`, `userID`, `RewardID`) VALUES
(31, 'Credit Card', 'Paid', 350, '2023-11-15', 1, 51),
(32, 'Debit Card', 'Paid', 780, '2023-11-16', 2, 52),
(33, 'PayPal', 'Paid', 1250, '2023-11-17', 3, 53),
(34, 'Credit Card', 'Paid', 410, '2023-11-18', 4, 54),
(35, 'Debit Card', 'Paid', 820, '2023-11-19', 5, 55),
(36, 'PayPal', 'Paid', 1320, '2023-11-20', 6, 56),
(37, 'Credit Card', 'Paid', 390, '2023-11-21', 7, 57),
(38, 'Debit Card', 'Paid', 800, '2023-11-22', 8, 58),
(39, 'PayPal', 'Paid', 1300, '2023-11-23', 9, 59),
(40, 'Credit Card', 'Paid', 380, '2023-11-24', 10, 60),
(41, 'Debit Card', 'Paid', 800, '2023-11-25', 1, 61),
(42, 'PayPal', 'Paid', 1300, '2023-11-26', 2, 62),
(43, 'Credit Card', 'Paid', 360, '2023-11-27', 3, 63),
(44, 'Debit Card', 'Paid', 790, '2023-11-28', 4, 64),
(45, 'PayPal', 'Paid', 1280, '2023-11-29', 5, 65),
(46, 'Credit Card', 'Paid', 350, '2023-11-30', 6, 66),
(47, 'Debit Card', 'Paid', 800, '2023-12-01', 7, 67),
(48, 'PayPal', 'Paid', 1320, '2023-12-02', 8, 68),
(49, 'Credit Card', 'Paid', 390, '2023-12-03', 9, 69),
(50, 'Debit Card', 'Paid', 800, '2023-12-04', 10, 70),
(51, 'PayPal', 'Paid', 1320, '2023-12-05', 1, 71),
(52, 'Credit Card', 'Paid', 370, '2023-12-06', 2, 72),
(53, 'Debit Card', 'Paid', 800, '2023-12-07', 3, 73),
(54, 'PayPal', 'Paid', 1300, '2023-12-08', 4, 74),
(55, 'Credit Card', 'Paid', 360, '2023-12-09', 5, 75),
(56, 'Debit Card', 'Paid', 790, '2023-12-10', 6, 76),
(57, 'PayPal', 'Paid', 1320, '2023-12-11', 7, 77),
(58, 'Credit Card', 'Paid', 380, '2023-12-12', 8, 78),
(59, 'Debit Card', 'Paid', 790, '2023-12-13', 9, 79),
(60, 'PayPal', 'Paid', 1300, '2023-12-14', 10, 80);

```

## 'payment' table

```

INSERT INTO `premium_service` (`premium_serviceID`, `serviceType`, `membershipID`) VALUES
(1, 'Priority Boarding', 1),
(2, 'Lounge Access', 2),
(3, 'Extra Baggage Allowance', 3),
(4, 'Priority Boarding', 4),
(5, 'Lounge Access', 5),
(6, 'Extra Baggage Allowance', 6),
(7, 'Priority Boarding', 7),
(8, 'Lounge Access', 8),
(9, 'Extra Baggage Allowance', 9),
(10, 'Priority Boarding', 10);

```

## 'premium\_service' table

```

INSERT INTO 'ticket' ('ticketID', 'seatNo', 'class', 'passengerName', 'date', 'time', 'price', 'bookingID', 'flightID') VALUES
(1, '31D', 'Economy', 'John Doe', '2023-11-15', '14:00:00', '350.00', 1, 1),
(2, '16B', 'Business', 'Alice Smith', '2023-11-16', '09:30:00', '780.00', 2, 2),
(3, '11A', 'First Class', 'Robert Johnson', '2023-11-17', '10:45:00', '1250.00', 3, 3),
(4, '40E', 'Economy', 'Emily Brown', '2023-11-18', '17:30:00', '410.00', 4, 4),
(5, '11B', 'Business', 'William Taylor', '2023-11-19', '21:00:00', '820.00', 5, 5),
(6, '12B', 'First Class', 'Olivia Miller', '2023-11-20', '08:00:00', '1320.00', 6, 6),
(7, '45A', 'Economy', 'Michael Wilson', '2023-11-21', '12:15:00', '390.00', 7, 7),
(8, '14J', 'Business', 'Sophia Garcia', '2023-11-22', '19:45:00', '800.00', 8, 8),
(9, '13E', 'First Class', 'Daniel Martinez', '2023-11-23', '04:20:00', '1300.00', 9, 9),
(10, '57C', 'Economy', 'Ava Lopez', '2023-11-24', '14:30:00', '380.00', 10, 10),
(11, '15F', 'Business', 'Ethan Hernandez', '2023-11-25', '11:45:00', '800.00', 11, 11),
(12, '12G', 'First Class', 'Mia Gonzalez', '2023-11-26', '18:00:00', '1300.00', 12, 12),
(13, '65B', 'Economy', 'Alexander Perez', '2023-11-27', '01:40:00', '360.00', 13, 13),
(14, '13K', 'Business', 'Charlotte Smith', '2023-11-28', '15:55:00', '790.00', 14, 14),
(15, '14F', 'First Class', 'Matthew Rodriguez', '2023-11-29', '10:10:00', '1280.00', 15, 15),
(16, '47E', 'Economy', 'Amelia Lee', '2023-11-30', '23:50:00', '350.00', 16, 16),
(17, '16A', 'Business', 'David Garcia', '2023-12-01', '05:15:00', '800.00', 17, 17),
(18, '123', 'First Class', 'Abigail Hernandez', '2023-12-02', '12:20:00', '1320.00', 18, 18),
(19, '23A', 'Economy', 'James Martinez', '2023-12-03', '19:00:00', '390.00', 19, 19),
(20, '19A', 'Business', 'Harper Lopez', '2023-12-04', '10:05:00', '800.00', 20, 20),
(21, '12K', 'First Class', 'Logan Hernandez', '2023-12-05', '13:30:00', '1320.00', 21, 21),
(22, '33B', 'Economy', 'Evelyn Gonzalez', '2023-12-06', '15:40:00', '370.00', 22, 22),
(23, '16D', 'Business', 'Benjamin Perez', '2023-12-07', '20:25:00', '800.00', 23, 23),
(24, '14G', 'First Class', 'Liam Rodriguez', '2023-12-08', '09:15:00', '1300.00', 24, 24),
(25, '41H', 'Economy', 'Victoria Lee', '2023-12-09', '11:55:00', '360.00', 25, 25),
(26, '36H', 'Economy', 'Lucas Garcia', '2023-12-10', '18:50:00', '790.00', 26, 26),
(27, '18J', 'First Class', 'Avery Hernandez', '2023-12-11', '14:00:00', '1320.00', 27, 27),
(28, '49A', 'Economy', 'Jackson Martinez', '2023-12-12', '17:35:00', '380.00', 28, 28),
(29, '21B', 'Business', 'Madison Lopez', '2023-12-13', '18:40:00', '790.00', 29, 29),
(30, '56C', 'Economy', 'Grayson Johnson', '2023-12-14', '13:50:00', '1300.00', 30, 30);

```

## 'ticket' table

## 10 SQL Basic Queries

1. Find the total number of flights for each departure gate

**SQL statement:**

```
SELECT departureGate, COUNT(*) AS TotalFlights  
FROM flight  
GROUP BY departureGate;
```

**The results of the query:**

	departureGate	TotalFlights
▶	Gate A1	1
	Gate B2	1
	Gate C3	1
	Gate D4	1
	Gate E5	1
	Gate F6	1
	Gate G7	1
	Gate H8	1
	Gate I9	1
	Gate J10	1
	Gate K11	1
	Gate L12	1
	Gate M13	1
	Gate N14	1
	Gate O15	1
	Gate P16	1
	Gate Q17	1
	Gate R18	1

2. Find the average size of baggage for each baggage status:

**SQL statement:**

```
SELECT baggageStatus, AVG(size) AS AverageBaggageSize  
FROM baggage  
GROUP BY baggageStatus;
```

**The results of the query:**

	baggageStatus	AverageBaggageSize
▶	Checked	17.1333
	Carry-on	26.0667

**3. Find users who made payments using a specific payment method:**

**SQL statement:**

```
SELECT u.*, p.*  
FROM user u, payment p  
WHERE u.userID = p.userID AND p.paymentMethod = 'PayPal';
```

**The results of the query:**

	userID	Fname	Lname	email	phoneNumber	address	paymentID	paymentMethod	paymetnStatus	amount	paymentDate	userID	RewardID
▶	3	Bob	Johnson	bob@example.com	5554443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	33	PayPal	Paid	1250	2023-11-17	3	53
6	Sophia	Lee	sophia@example.com	7778889999	345 Cedar Court, Apt 101, Dallas, TX 75201	36	PayPal	Paid	1320	2023-11-20	6	56	
9	William	Lopez	wiliam@example.com	2223344444	1123 Spruce Lane, Unit 45, New York, NY 10001	39	PayPal	Paid	1300	2023-11-23	9	59	
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	42	PayPal	Paid	1300	2023-11-26	2	62	
5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 80, Atlanta, GA 30301	45	PayPal	Paid	1280	2023-11-29	5	65	
8	Olivia	Martinez	olivia@example.com	6669991111	901 Walnut Avenue, Apt 22, Miami, FL 33101	48	PayPal	Paid	1320	2023-12-02	8	68	
1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704	51	PayPal	Paid	1320	2023-12-05	1	71	
4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	54	PayPal	Paid	1300	2023-12-08	4	74	
7	Daniel	Garcia	daniel@example.com	9998887777	678 Birch Street, Suite 304, Seattle, WA 98101	57	PayPal	Paid	1320	2023-12-11	7	77	
10	Ava	Gonzalez	ava@example.com	8887776666	4567 Sycamore Road, Apt 67, Chicago, IL 60601	60	PayPal	Paid	1300	2023-12-14	10	80	

**4. Count the number of passengers for each flight:**

**SQL statement:**

```
SELECT flightID, COUNT(passengerID) AS PassengerCount  
FROM passenger  
GROUP BY flightID;
```

**The results of the query:**

	flightID	PassengerCount
▶	1	1
2		1
3		1
4		1
5		1
6		1
7		1
8		1
9		1
10		1
11		1
12		1
13		1
14		1
15		1
16		1
17		1
18		1

**5.** Retrieve details of baggage for a specific passenger:

**SQL statement:**

```
SELECT * FROM baggage  
WHERE passengerID = 31;
```

**The results of the query:**

	baggageID	size	baggageStatus	passengerID	baggageNumber
▶	1	25	Checked	31	1
*	NULL	NULL	NULL	NULL	NULL

**6.** Find the total number of flights with no delays

**SQL statement:**

```
SELECT COUNT(*) AS OnTimeFlights  
FROM flight  
WHERE delay = 'No Delay';
```

**The results of the query:**

	OnTimeFlights
▶	7

**7.** List all bookings made on a specific date:

**SQL statement:**

```
SELECT * FROM airlinebooking  
WHERE bookingDate = '2023-11-20';
```

**The results of the query:**

	bookingID	bookingDate	userID
▶	6	2023-11-20	6
*	NULL	NULL	NULL

**8.** Calculate the total number of premium service subscriptions:

**SQL statement:**

```
SELECT COUNT(*) AS TotalSubscriptions  
FROM premium_service;
```

**The results of the query:**

	TotalSubscriptions
▶	10

**9. Retrieve flight information for a specific flight number:**

**SQL statement:**

```
SELECT * FROM flight
```

```
WHERE flightNum = 'AA123';
```

**The results of the query:**

	flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining
▶	1	AA123	New York	No Delay	Gate A1	12:00:00	13:00:00	Domestic	Yes	Lounge A	Full	Yes
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

**10. Retrieve passengers who booked a flight and their respective flight details:**

**SQL statement:**

```
SELECT pa.* , f.*
```

```
FROM passenger pa, flight f
```

```
WHERE pa.flightID = f.flightID;
```

**The results of the query:**

	passengerID	FName	UName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID	flightID	flightNum	destination	delay	departureGate
▶	31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1	AA123	New York	No Delay	Gate A1	
32	Alice	Smith		555-987-6543	Silver	alice@example.com	1992-08-15	Female	2	2	2	DL456	Los Angeles	10 mins	Gate B2	
33	Robert	Johnson		555-111-3333	Bronze	robert@example.com	1985-11-20	Male	3	3	3	UA899	Chicago	5 mins	Gate C3	
34	Emily	Brown		555-222-4444	Platinum	emily@example.com	1988-03-12	Female	4	4	4	BA234	London	20 mins	Gate D4	
35	William	Taylor		555-333-5555	Gold	william@example.com	1995-06-30	Male	5	5	5	LH567	Tokyo	15 mins	Gate E5	
36	Olivia	Miller		555-444-6666	Silver	olivia@example.com	1993-09-25	Female	6	6	6	AF890	Paris	No Delay	Gate F6	
37	Michael	Wilson		555-555-7777	Bronze	michael@example.com	1980-12-05	Male	7	7	7	SQ123	Berlin	25 mins	Gate G7	
38	Sophia	Garcia		555-666-8888	Platinum	sophia@example.com	1987-02-10	Female	8	8	8	EK456	Sydney	10 mins	Gate H8	
39	Daniel	Martinez		555-777-9999	Gold	daniel@example.com	1991-04-17	Male	9	9	9	QF789	Dubai	30 mins	Gate I9	
40	Ava	Lopez		555-888-0000	Silver	ava@example.com	1994-07-22	Female	10	10	10	TK234	Rome	15 mins	Gate J10	
41	Ethan	Hernan...		555-999-1111	Bronze	ethan@example.com	1983-10-18	Male	11	11	11	CX567	Madrid	No Delay	Gate K11	
42	Mia	Gonzalez		555-111-2222	Platinum	mia@example.com	1989-01-13	Female	12	12	12	KL890	Beijing	20 mins	Gate L12	
43	Alex...	Perez		555-222-3333	Gold	alexander@example....	1996-08-08	Male	13	13	13	JL123	Mumbai	25 mins	Gate M13	
44	Charlotte	Smith		555-333-4444	Silver	charlotte@example.com	1992-02-28	Female	14	14	14	AY456	Seoul	10 mins	Gate N14	
45	Matthew	Rodriguez		555-444-5555	Bronze	matthew@example.com	1984-05-07	Male	15	15	15	EY789	Moscow	15 mins	Gate O15	
46	Amelia	Lee		555-555-6666	Platinum	amelia@example.com	1986-12-25	Female	16	16	16	SW234	Bangkok	5 mins	Gate P16	
47	David	Garcia		555-666-7777	Gold	david@example.com	1997-09-03	Male	17	17	17	GY567	Istanbul	30 mins	Gate Q17	

boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining
12:00:00	13:00:00	Domestic	Yes	Lounge A	Full	Yes
14:30:00	15:30:00	Domestic	No	Lounge B	Limited	No
11:00:00	12:00:00	International	Yes	Lounge C	Full	Yes
18:45:00	19:45:00	International	Yes	Lounge D	Full	Yes
23:30:00	00:30:00	International	No	Lounge E	Limited	No
09:00:00	10:00:00	International	Yes	Lounge F	Full	Yes
21:15:00	22:15:00	International	Yes	Lounge G	Full	Yes
03:45:00	04:45:00	International	No	Lounge H	Limited	No
16:20:00	17:20:00	International	Yes	Lounge I	Full	Yes
07:30:00	08:30:00	International	No	Lounge J	Limited	No
11:45:00	12:45:00	International	Yes	Lounge K	Full	Yes
19:00:00	20:00:00	International	Yes	Lounge L	Full	Yes
02:40:00	03:40:00	International	Yes	Lounge M	Full	Yes
15:55:00	16:55:00	International	No	Lounge N	Limited	No
09:10:00	10:10:00	International	Yes	Lounge O	Full	Yes
22:50:00	23:50:00	International	Yes	Lounge P	Full	Yes
04:15:00	05:15:00	International	Yes	Lounge Q	Full	Yes

**11.** List all users who have a Gold membership:

**SQL statement:**

```
SELECT *
FROM user
WHERE userID IN (SELECT userID FROM membership WHERE registrationM = 'Gold');
```

**The results of the query:**

	userID	Fname	Lname	email	phoneNumber	address
▶	1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704
	4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...
	7	Daniel	Garcia	daniel@example.com	9998887777	678 Birch Street, Suite 304, Seattle, WA 98101
	10	Ava	Gonzalez	ava@example.com	8887776666	4567 Sycamore Road, Apt 67, Chicago, IL 60601
*	NULL	NULL	NULL	NULL	NULL	NULL

**12.** Retrieve flight information for a specific destination and departure gate:

**SQL statement:**

```
SELECT *  
FROM flight  
WHERE destination = 'New York' AND departureGate = 'Gate A1';
```

**The results of the query:**

	flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining
▶	1	AA123	New York	No Delay	Gate A1	12:00:00	13:00:00	Domestic	Yes	Lounge A	Full	Yes
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

**13.** Calculate the average size of baggage for each passenger:

**SQL statement:**

```
SELECT passengerID, AVG(size) AS AverageBaggageSize  
FROM baggage  
GROUP BY passengerID;
```

**The results of the query:**

	passengerID	AverageBaggageSize
▶	31	25.0000
	32	30.0000
	33	20.0000
	34	22.0000
	35	27.0000
	36	18.0000
	37	21.0000
	38	26.0000
	39	24.0000
	40	19.0000
	41	23.0000
	42	28.0000
	43	17.0000
	44	29.0000
	45	16.0000
	46	30.0000
	47	15.0000
	48	26.0000

**14.** Find the number of flights for each type of dining:

**SQL statement:**

```
SELECT flight_Type, COUNT(*) AS FlightCount  
FROM flight  
GROUP BY flight_Type;
```

**The results of the query:**

	flight_Type	FlightCount
▶	Domestic	15
	International	15

**15.** Retrieve information about users with phone numbers ending in '123':

**SQL statement:**

```
SELECT *  
FROM user  
WHERE phoneNumber LIKE '%666';
```

**The results of the query:**

	userID	Fname	Lname	email	phoneNumber	address
▶	5	Michael	Brown	michael@example.com	444556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301
	10	Ava	Gonzalez	ava@example.com	8887776666	4567 Sycamore Road, Apt 67, Chicago, IL 60601
*	NULL	NULL	NULL	NULL	NULL	NULL

**16.** List all flights departing after a specific time:

**SQL statement:**

```
SELECT *  
FROM flight  
WHERE departureTime > '15:00:00';
```

**The results of the query:**

	flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining
▶	2	DL456	Los Angeles	10 mins	Gate B2	14:30:00	15:30:00	Domestic	No	Lounge B	Limited	No
	4	BA234	London	20 mins	Gate D4	18:45:00	19:45:00	International	Yes	Lounge D	Full	Yes
	7	SQ123	Berlin	25 mins	Gate G7	21:15:00	22:15:00	International	Yes	Lounge G	Full	Yes
	9	QF789	Dubai	30 mins	Gate I9	16:20:00	17:20:00	International	Yes	Lounge I	Full	Yes
	12	KL890	Beijing	20 mins	Gate L12	19:00:00	20:00:00	International	Yes	Lounge L	Full	Yes
	14	AY456	Seoul	10 mins	Gate N14	15:55:00	16:55:00	International	No	Lounge N	Limited	No
	16	SW234	Bangkok	5 mins	Gate P16	22:50:00	23:50:00	International	Yes	Lounge P	Full	Yes
	19	QR123	Vancouver	20 mins	Gate S19	18:00:00	19:00:00	Domestic	Yes	Lounge S	Full	Yes
	22	BA234	Edmonton	10 mins	Gate V22	14:40:00	15:40:00	Domestic	No	Lounge V	Limited	No
	23	AA567	Quebec	25 mins	Gate W23	19:25:00	20:25:00	Domestic	Yes	Lounge W	Full	Yes
	26	SQ456	Saskatoon	20 mins	Gate Z26	17:50:00	18:50:00	Domestic	No	Lounge Z	Limited	No
*	28	DL234	Winnipeg	5 mins	Gate AB28	16:35:00	17:35:00	Domestic	Yes	Lounge AB	Full	Yes
	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

**17.** Find the total amount spent by each user on payments:

**SQL statement:**

```
SELECT userID, SUM(amount) AS TotalAmountSpent  
FROM payment  
GROUP BY userID;
```

**The results of the query:**

	userID	TotalAmountSpent
▶	1	2470
	2	2450
	3	2410
	4	2500
	5	2460
	6	2460
	7	2510
	8	2500
	9	2480
	10	2480

**18.** Retrieve information about flights with a specific departure gate:

**SQL statement:**

```
SELECT *
FROM flight
WHERE departureGate = 'Gate B2';
```

**The results of the query:**

	flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining
▶	2	DL456	Los Angeles	10 mins	Gate B2	14:30:00	15:30:00	Domestic	No	Lounge B	Limited	No
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

**19.** Count the number of passengers for each gender:

**SQL statement:**

```
SELECT gender, COUNT(*) AS PassengerCount
FROM passenger
GROUP BY gender;
```

**The results of the query:**

	gender	PassengerCount
▶	Male	16
	Female	14

**20. Find flights with fine dining and their booked passengers:**

**SQL statement:**

```
SELECT f.* , pa.*  
FROM flight f, passenger pa  
WHERE f.flightID = pa.flightID AND f.fine_dining = 'Yes';
```

**The results of the query:**

flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining	passengerID	Fname	LName
9	QF789	Dubai	30 mins	Gate I9	16:20:00	17:20:00	International	Yes	Lounge I	Full	Yes	39	Daniel	Martinez
11	CX567	Madrid	No Delay	Gate K11	11:45:00	12:45:00	International	Yes	Lounge K	Full	Yes	41	Ethan	Hernandez
12	KL890	Beijing	20 mins	Gate L12	19:00:00	20:00:00	International	Yes	Lounge L	Full	Yes	42	Mia	Gonzalez
13	JL123	Mumbai	25 mins	Gate M13	02:40:00	03:40:00	International	Yes	Lounge M	Full	Yes	43	Alexander	Perez
15	EY789	Moscow	15 mins	Gate O15	09:10:00	10:10:00	International	Yes	Lounge O	Full	Yes	45	Matthew	Rodriguez
16	SW234	Bangkok	5 mins	Gate P16	22:50:00	23:50:00	International	Yes	Lounge P	Full	Yes	46	Amelia	Lee
17	GY567	Istanbul	30 mins	Gate Q17	04:15:00	05:15:00	International	Yes	Lounge Q	Full	Yes	47	David	Garcia
18	AY890	Toronto	No Delay	Gate R18	11:20:00	12:20:00	Domestic	Yes	Lounge R	Full	Yes	48	Abigail	Hernandez
19	QR123	Vancouver	20 mins	Gate S19	18:00:00	19:00:00	Domestic	Yes	Lounge S	Full	Yes	49	James	Martinez
20	EK456	Montreal	15 mins	Gate T20	09:05:00	10:05:00	Domestic		Lounge T	Full	Yes	50	Harper	Lopez
21	LH789	Calgary	No Delay	Gate U21	12:30:00	13:30:00	Domestic	Yes	Lounge U	Full	Yes	51	Logan	Hernandez
23	AA567	Quebec	25 mins	Gate W23	19:25:00	20:25:00	Domestic	Yes	Lounge W	Full	Yes	53	Benjamin	Perez
24	QF890	Ottawa	15 mins	Gate X24	08:15:00	09:15:00	Domestic	Yes	Lounge X	Full	Yes	54	Liam	Rodriguez
25	KL123	Halifax	No Delay	Gate Y25	10:55:00	11:55:00	Domestic	Yes	Lounge Y	Full	Yes	55	Victoria	Lee
27	CX789	Regina	10 mins	Gate AA27	13:00:00	14:00:00	Domestic	Yes	Lounge AA	Full	Yes	57	Avery	Hernandez
28	DL234	Winnipeg	5 mins	Gate AB28	16:35:00	17:35:00	Domestic	Yes	Lounge AB	Full	Yes	58	Jackson	Martinez
30	AF890	Whitehorse	No Delay	Gate AD30	12:50:00	13:50:00	Domestic	Yes	Lounge AD	Full	Yes	60	Grayson	Johnson

contactInfo	registerM	email	dob	gender	flightID	bookingID	userID
555-777-9999	Gold	daniel@example.com	1991-04-17	Male	9	9	9
555-999-1111	Bronze	ethan@example.com	1983-10-18	Male	11	11	1
555-111-2222	Platinum	mia@example.com	1989-01-13	Female	12	12	2
555-222-3333	Gold	alexander@example.com	1996-08-08	Male	13	13	3
555-444-5555	Bronze	matthew@example.com	1984-05-07	Male	15	15	5
555-555-6666	Platinum	amelia@example.com	1986-12-25	Female	16	16	6
555-666-7777	Gold	david@example.com	1997-09-03	Male	17	17	7
555-777-8888	Silver	abigail@example.com	1993-10-11	Female	18	18	8
555-888-9999	Bronze	james@example.com	1981-01-30	Male	19	19	9
555-999-0000	Platinum	harper@example.com	1988-06-19	Female	20	20	10
555-000-1111	Gold	logan@example.com	1995-03-14	Male	21	21	1
555-222-3333	Bronze	benjamin@example.com	1990-11-01	Male	23	23	3
555-333-4444	Platinum	liam@example.com	1987-07-09	Male	24	24	4
555-444-5555	Gold	victoria@example.com	1998-02-20	Female	25	25	5
555-666-7777	Bronze	avery@example.com	1991-12-15	Female	27	27	7
555-777-8888	Platinum	jackson@example.com	1989-05-02	Male	28	28	8
555-999-0000	Silver	grayson@example.com	1986-03-26	Male	30	30	10

## 21. Retrieve users with their total spending and payment details:

**SQL statement:**

```
SELECT u.* , p.*
```

```
FROM user u, payment p
```

```
WHERE u.userID = p.userID;
```

**The results of the query:**

	userID	Fname	Lname	email	phoneNumber	address	paymentID	paymentMethod	paymetnStatus	amount	paymentDate	userID	RewardID
▶	1	John	Doe	john@example.com	9996754576	123 Main St, Apt #B, Springfield, IL 62704	31	Credit Card	Paid	350	2023-11-15	1	51
1	John	Doe	john@example.com	9996754576	123 Main St, Apt #B, Springfield, IL 62704	41	Debit Card	Paid	800	2023-11-25	1	61	
1	John	Doe	john@example.com	9996754576	123 Main St, Apt #B, Springfield, IL 62704	51	PayPal	Paid	1320	2023-12-05	1	71	
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	32	Debit Card	Paid	780	2023-11-16	2	52	
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	42	PayPal	Paid	1300	2023-11-26	2	62	
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	52	Credit Card	Paid	370	2023-12-06	2	72	
3	Bob	Johnson	bob@example.com	555443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	33	PayPal	Paid	1250	2023-11-17	3	53	
3	Bob	Johnson	bob@example.com	555443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	43	Credit Card	Paid	360	2023-11-27	3	63	
3	Bob	Johnson	bob@example.com	555443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	53	Debit Card	Paid	800	2023-12-07	3	73	
4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	34	Credit Card	Paid	410	2023-11-18	4	54	
4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	44	Debit Card	Paid	790	2023-11-28	4	64	
4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	54	PayPal	Paid	1300	2023-12-08	4	74	
5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	35	Debit Card	Paid	820	2023-11-19	5	55	
5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	45	PayPal	Paid	1280	2023-11-29	5	65	
5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	55	Credit Card	Paid	360	2023-12-09	5	75	
6	Sophia	Lee	sophia@example.com	7778899999	345 Cedar Court, Apt 101, Dallas, TX 75201	36	PayPal	Paid	1320	2023-11-20	6	56	
6	Sophia	Lee	sophia@example.com	7778899999	345 Cedar Court, Apt 101, Dallas, TX 75201	46	Credit Card	Paid	350	2023-11-30	6	66	
6	Sophia	Lee	sophia@example.com	7778899999	345 Cedar Court, Apt 101, Dallas, TX 75201	56	Debit Card	Paid	790	2023-12-10	6	76	

## 22. List passengers and their baggage details:

**SQL statement:**

```
SELECT pa.* , b.*
```

```
FROM passenger pa, baggage b
```

```
WHERE pa.passengerID = b.passengerID;
```

**The results of the query:**

	passengerID	Fname	Lname	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID	baggageID	size	baggageStatus	passengerID	baggageNumber
▶	31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1	1	25	Checked	31	1
32	John	Smith	555-087-6543	Silver	alice@example.com	1992-08-15	Female	2	2	2	2	30	Carry-on	32	2	
33	Robert	Johnson	555-111-3333	Bronze	rober@example.com	1985-11-20	Male	3	3	3	3	20	Checked	33	3	
34	Emily	Brown	555-222-4444	Platinum	emily@example.com	1988-03-12	Female	4	4	4	4	22	Carry-on	34	4	
35	William	Taylor	555-333-5555	Gold	william@example.com	1995-06-15	Male	5	5	5	5	27	Checked	35	5	
36	Olivia	Miller	555-444-6666	Silver	olivia@example.com	1993-09-25	Female	6	6	6	6	18	Carry-on	36	6	
37	Michael	Wilson	555-335-7777	Bronze	michael@example.com	1980-12-05	Male	7	7	7	7	21	Checked	37	7	
38	Sophia	Garcia	555-666-8888	Platinum	sophia@example.com	1987-02-10	Female	8	8	8	8	26	Carry-on	38	8	
39	Daniel	Martinez	555-777-9999	Gold	daniel@example.com	1991-04-17	Male	9	9	9	9	24	Checked	39	9	
40	Ava	Lopez	555-888-0000	Silver	ava@example.com	1994-07-22	Female	10	10	10	10	19	Carry-on	40	10	
41	Ethan	Hernan...	555-999-1111	Bronze	ethan@example.com	1983-10-18	1994-07-22	11	1	11	11	23	Checked	41	11	
42	Mia	Gonzalez	555-111-2222	Platinum	mia@example.com	1989-01-13	Female	12	12	2	12	28	Carry-on	42	12	
43	Alex...	Perez	555-222-3333	Gold	alexander@example...	1996-08-01	Male	13	13	3	13	17	Checked	43	13	
44	Charlie...	Smith	555-333-4444	Silver	charlotte@example.com	1992-02-28	Female	14	4	14	14	29	Carry-on	44	14	
45	Matthew	Rodriguez	555-444-5555	Bronze	matthew@example.com	1984-05-07	Male	15	5	15	15	16	Checked	45	15	
46	Amelia	Lee	555-555-6666	Platinum	amelia@example.com	1986-12-25	Female	16	6	16	16	30	Carry-on	46	16	
47	David	Garcia	555-666-7777	Gold	david@example.com	1997-09-03	Male	17	7	17	17	15	Checked	47	17	
48	Abigail	Herman...	555-777-8888	Silver	abigail@example.com	1993-10-11	Female	18	8	18	18	26	Carry-on	48	18	

**23.** List all flights with delayed departures:

**SQL statement:**

```
SELECT *  
FROM flight  
WHERE delay != 'No Delay';
```

**The results of the query:**

flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining
2	DL456	Los Angeles	10 mins	Gate B2	14:30:00	15:30:00	Domestic	No	Lounge B	Limited	No
3	UA789	Chicago	5 mins	Gate C3	11:00:00	12:00:00	International	Yes	Lounge C	Full	Yes
4	BA234	London	20 mins	Gate D4	18:45:00	19:45:00	International	Yes	Lounge D	Full	Yes
5	LH567	Tokyo	15 mins	Gate E5	23:30:00	00:30:00	International	No	Lounge E	Limited	No
7	SQ123	Berlin	25 mins	Gate G7	21:15:00	22:15:00	International	Yes	Lounge G	Full	Yes
8	EK456	Sydney	10 mins	Gate H8	03:45:00	04:45:00	International	No	Lounge H	Limited	No
9	QF789	Dubai	30 mins	Gate I9	16:20:00	17:20:00	International	Yes	Lounge I	Full	Yes
10	TK234	Rome	15 mins	Gate J10	07:30:00	08:30:00	International	No	Lounge J	Limited	No
12	KL890	Beijing	20 mins	Gate L12	19:00:00	20:00:00	International	Yes	Lounge L	Full	Yes
13	JL123	Mumbai	25 mins	Gate M13	02:40:00	03:40:00	International	Yes	Lounge M	Full	Yes
14	AY456	Seoul	10 mins	Gate N14	15:55:00	16:55:00	International	No	Lounge N	Limited	No
15	EY789	Moscow	15 mins	Gate O15	09:10:00	10:10:00	International	Yes	Lounge O	Full	Yes
16	SW234	Bangkok	5 mins	Gate P16	22:50:00	23:50:00	International	Yes	Lounge P	Full	Yes
17	GY567	Istanbul	30 mins	Gate Q17	04:15:00	05:15:00	International	Yes	Lounge Q	Full	Yes
19	QR123	Vancouver	20 mins	Gate S19	18:00:00	19:00:00	Domestic	Yes	Lounge S	Full	Yes
20	EK456	Montreal	15 mins	Gate T20	09:05:00	10:05:00	Domestic	No	Lounge T	Full	Yes
22	BA234	Edmonton	10 mins	Gate V22	14:40:00	15:40:00	Domestic	No	Lounge V	Limited	No
23	AA567	Quebec	25 mins	Gate W23	19:25:00	20:25:00	Domestic	Yes	Lounge W	Full	Yes

**24.** Find the total number of passengers for each flight class:

**SQL statement:**

```
SELECT class, COUNT(*) AS TotalPassengers  
FROM ticket  
GROUP BY class;
```

**The results of the query:**

class	TotalPassengers
Economy	12
Business	9
First Class	9

**25.** Retrieve information about passengers with a specific gender and age range:

**SQL statement:**

```
SELECT *  
FROM passenger  
WHERE gender = 'Male' AND YEAR(NOW()) - YEAR(dob) BETWEEN 25 AND 35;
```

**The results of the query:**

	passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID
▶	31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1
	35	William	Taylor	555-333-5555	Gold	william@example.com	1995-06-30	Male	5	5	5
	39	Daniel	Martinez	555-777-9999	Gold	daniel@example.com	1991-04-17	Male	9	9	9
	43	Alexander	Perez	555-222-3333	Gold	alexander@example.com	1996-08-08	Male	13	13	3
	47	David	Garcia	555-666-7777	Gold	david@example.com	1997-09-03	Male	17	17	7
	51	Logan	Hernandez	555-000-1111	Gold	logan@example.com	1995-03-14	Male	21	21	1
	53	Benjamin	Perez	555-222-3333	Bronze	benjamin@example.com	1990-11-01	Male	23	23	3
	58	Jackson	Martinez	555-777-8888	Platinum	jackson@example.com	1989-05-02	Male	28	28	8
*	NUL	NUL	NUL	NUL	NUL	NUL	NUL	NUL	NUL	NUL	NUL

**26.** List users who booked a flight and their corresponding departure gates:

**SQL statement:**

```
SELECT u.*, f.departureGate  
FROM user u, airlinebooking ab, passenger pa, flight f  
WHERE u.userID = ab.userID  
AND ab.bookingID = pa.bookingID  
AND pa.flightID = f.flightID;
```

### The results of the query

	userID	Fname	Lname	email	phoneNumber	address	departureGate
▶	1	John	Doe	john@example.com	9996754576	123 Main St, Apt #B, Springfield, IL 62704	Gate A1
	1	John	Doe	john@example.com	9996754576	123 Main St, Apt #B, Springfield, IL 62704	Gate K11
	1	John	Doe	john@example.com	9996754576	123 Main St, Apt #B, Springfield, IL 62704	Gate U21
	2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	Gate B2
	2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	Gate L12
	2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	Gate V22
	3	Bob	Johnson	bob@example.com	5554443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	Gate C3
	3	Bob	Johnson	bob@example.com	5554443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	Gate M13
	3	Bob	Johnson	bob@example.com	5554443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	Gate W23
	4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	Gate D4
	4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	Gate N14
	4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	Gate X24
	5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	Gate E5
	5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	Gate O15
	5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	Gate Y25
	6	Sophia	Lee	sophia@example.com	7778889999	345 Cedar Court, Apt 101, Dallas, TX 75201	Gate F6
	6	Sophia	Lee	sophia@example.com	7778889999	345 Cedar Court, Apt 101, Dallas, TX 75201	Gate P16
	6	Sophia	Lee	sophia@example.com	7778889999	345 Cedar Court, Apt 101, Dallas, TX 75201	Gate Z26

**27.** Retrieve users who have not booked any flights:

#### SQL statement:

```
SELECT u.*  
  
FROM user u  
  
WHERE u.userID NOT IN (SELECT DISTINCT userID FROM airlinebooking);
```

### The results of the query

	userID	Fname	Lname	email	phoneNumber	address
*	HULL	HULL	HULL	HULL	HULL	HULL

**28.** Calculate the total number of delayed flights for each destination:

#### SQL statement:

```
SELECT destination, COUNT(*) AS DelayedFlights  
  
FROM flight  
  
WHERE delay != 'No Delay'  
  
GROUP BY destination;
```

### The results of the query

	destination	DelayedFlights
▶	Los Angeles	1
*	Chicago	1
	London	1
	Tokyo	1
	Berlin	1
	Sydney	1
	Dubai	1
	Rome	1
	Beijing	1
	Mumbai	1
	Seoul	1
	Moscow	1
	Bangkok	1
	Istanbul	1
	Vancouver	1
	Montreal	1
	Edmonton	1
	Quebec	1

29. Retrieve information about passengers with a specific booking ID:

**SQL statement:**

```
SELECT *
FROM passenger
WHERE bookingID = 1;
```

### The results of the query

	passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID
▶	31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1
*	HULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**30. Find flights with more than 100 available seats**

**SQL statement:**

```
SELECT f.*  
FROM flight f, passenger p  
WHERE f.flightID = p.flightID  
GROUP BY f.flightID  
HAVING COUNT(p.passengerID) < 100;
```

**The results of the query**

	flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining
▶	1	AA123	New York	No Delay	Gate A1	12:00:00	13:00:00	Domestic	Yes	Lounge A	Full	Yes
2	DL456	Los Angeles	10 mins	Gate B2	14:30:00	15:30:00	Domestic	No	Lounge B	Limited	No	
3	UA789	Chicago	5 mins	Gate C3	11:00:00	12:00:00	International	Yes	Lounge C	Full	Yes	
4	BA234	London	20 mins	Gate D4	18:45:00	19:45:00	International	Yes	Lounge D	Full	Yes	
5	LH567	Tokyo	15 mins	Gate E5	23:30:00	00:30:00	International	No	Lounge E	Limited	No	
6	AF890	Paris	No Delay	Gate F6	09:00:00	10:00:00	International	Yes	Lounge F	Full	Yes	
7	SQ123	Berlin	25 mins	Gate G7	21:15:00	22:15:00	International	Yes	Lounge G	Full	Yes	
8	EK456	Sydney	10 mins	Gate H8	03:45:00	04:45:00	International	No	Lounge H	Limited	No	
9	QF789	Dubai	30 mins	Gate I9	16:20:00	17:20:00	International	Yes	Lounge I	Full	Yes	
10	TK234	Rome	15 mins	Gate J10	07:30:00	08:30:00	International	No	Lounge J	Limited	No	
11	CX567	Madrid	No Delay	Gate K11	11:45:00	12:45:00	International	Yes	Lounge K	Full	Yes	
12	KL890	Beijing	20 mins	Gate L12	19:00:00	20:00:00	International	Yes	Lounge L	Full	Yes	
13	JL123	Mumbai	25 mins	Gate M13	02:40:00	03:40:00	International	Yes	Lounge M	Full	Yes	
14	AY456	Seoul	10 mins	Gate N14	15:55:00	16:55:00	International	No	Lounge N	Limited	No	
15	EY789	Moscow	15 mins	Gate O15	09:10:00	10:10:00	International	Yes	Lounge O	Full	Yes	
16	SW234	Bangkok	5 mins	Gate P16	22:50:00	23:50:00	International	Yes	Lounge P	Full	Yes	
17	GY567	Istanbul	30 mins	Gate Q17	04:15:00	05:15:00	International	Yes	Lounge Q	Full	Yes	
18	AY890	Toronto	No Delay	Gate R18	11:20:00	12:20:00	Domestic	Yes	Lounge R	Full	Yes	

**31. Calculate the average age of passengers for each flight**

**SQL statement:**

```
SELECT flightID, AVG(YEAR(NOW()) - YEAR(dob)) AS AveragePassengerAge  
FROM passenger  
GROUP BY flightID;
```

**The results of the query:**

flightID	AveragePassengerAge
1	33.0000
2	31.0000
3	38.0000
4	35.0000
5	28.0000
6	30.0000
7	43.0000
8	36.0000
9	32.0000
10	29.0000
11	40.0000
12	34.0000
13	27.0000
14	31.0000
15	39.0000
16	37.0000

**32.** Calculate the total number of flights for each destination with fine dining

**SQL statement:**

```
SELECT destination, COUNT(*) AS FineDiningFlights  
FROM flight  
WHERE fine_dining = 'Yes'  
GROUP BY destination;
```

**The results of the query:**

	destination	FineDiningFlights
▶	New York	1
	Chicago	1
	London	1
	Paris	1
	Berlin	1
	Dubai	1
	Madrid	1
	Beijing	1
	Mumbai	1
	Moscow	1
	Bangkok	1
	Istanbul	1
	Toronto	1
	Vancouver	1
	Montreal	1
	Calgary	1

**33.** List users who made payments with the highest and lowest amounts

**SQL statement:**

```
SELECT userID, MIN(amount) AS MinPaymentAmount, MAX(amount) AS  
MaxPaymentAmount  
FROM payment  
GROUP BY userID;
```

**The results of the query:**

	userID	MinPaymentAmount	MaxPaymentAmount
▶	1	350	1320
	2	370	1300
	3	360	1250
	4	410	1300
	5	360	1280
	6	350	1320
	7	390	1320
	8	380	1320
	9	390	1300
	10	380	1300

**34.** Count the number of passengers for each flight class, excluding first class

**SQL statement:**

```
SELECT class, COUNT(*) AS TotalPassengers FROM ticket  
WHERE class != 'First Class'  
GROUP BY class;
```

**The results of the query:**

	class	TotalPassengers
▶	Economy	12
	Business	9

**35. Find the flight with the earliest departure time**

**SQL statement:**

```
SELECT flightID, MIN(departureTime) AS EarliestDepartureTime  
FROM flight  
GROUP BY flightID  
ORDER BY EarliestDepartureTime ASC;
```

**The results of the query:**

	flightID	EarliestDepartureTime
▶	5	00:30:00
	13	03:40:00
	8	04:45:00
	17	05:15:00
	10	08:30:00
	24	09:15:00
	6	10:00:00
	20	10:05:00
	15	10:10:00
	29	10:40:00
	25	11:55:00
	3	12:00:00
	18	12:20:00
	11	12:45:00
	1	13:00:00
	21	13:30:00

**36. Calculate the average payment amount for each payment method**

**SQL statement:**

```
SELECT paymentMethod, AVG(amount) AS AveragePaymentAmount  
FROM payment  
GROUP BY paymentMethod;
```

**The results of the query:**

	paymentMethod	AveragePaymentAmount
▶	Credit Card	374.0000
	Debit Card	797.0000
	PayPal	1301.0000

**37.** Count the number of passengers for each flight destination

**SQL statement:**

```
SELECT destination, COUNT(passengerID) AS PassengerCount  
FROM flight, passenger  
WHERE flight.flightID = passenger.flightID  
GROUP BY destination;
```

**The results of the query:**

	destination	PassengerCount
▶	New York	1
	Los Angeles	1
	Chicago	1
	London	1
	Tokyo	1
	Paris	1
	Berlin	1
	Sydney	1
	Dubai	1
	Rome	1
	Madrid	1
	Beijing	1
	Mumbai	1
	Seoul	1
	Moscow	1
	Bangkok	1

**38. Retrieve the flight with the most delayed departures**

**SQL statement:**

```
SELECT flightID, COUNT(*) AS DelayedDepartures  
FROM flight  
WHERE delay != 'No Delay'  
GROUP BY flightID  
ORDER BY DelayedDepartures DESC  
LIMIT 1;
```

**The results of the query:**

	flightID	DelayedDepartures
▶	2	1

**39. Retrieve the passenger with the highest age**

**SQL statement:**

```
SELECT *  
FROM passenger  
WHERE dob = (SELECT MIN(dob) FROM passenger);
```

**The results of the query:**

	passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID
▶	37	Michael	Wilson	555-555-7777	Bronze	michael@example.com	1980-12-05	Male	7	7	7
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**40.** Retrieve the passenger with the lowest age

**SQL statement:**

```
SELECT *  
FROM passenger  
WHERE dob = (SELECT Max(dob) FROM passenger);
```

**The results of the query:**

	passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID
▶	55	Victoria	Lee	555-444-5555	Gold	victoria@example.com	1998-02-20	Female	25	25	5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

For more details visit:

<https://drive.google.com/drive/folders/1wl1kB-SrocXA1sE0umSsRMvQBX5saUJq?usp=sharing>

## 11 SQL Advance Queries

1. Find the total number of bookings per destination

**SQL statement:**

```
SELECT f.destination, COUNT(ab.bookingID) as total_bookings  
FROM flight f LEFT JOIN airlinebooking ab ON f.flightID = ab.bookingID  
GROUP BY f.destination;
```

**The results of the query:**

destination	total_bookin...
New York	1
Los Angeles	1
Chicago	1
London	1
Tokyo	1
Paris	1
Berlin	1
Sydney	1
Dubai	1
Rome	1
Madrid	1
Beijing	1
Mumbai	1
Seoul	1

2. Retrieve the baggage information for a specific passenger

**SQL statement:**

```
SELECT f.destination, COUNT(ab.bookingID) as total_bookings  
FROM flight f LEFT JOIN airlinebooking ab ON f.flightID = ab.bookingID  
GROUP BY f.destination;
```

**The results of the query:**

baggageID	size	baggageStatus	passengerID	baggageNumb...
1	25	Checked	31	1
11	23	Checked	41	11
21	13	Checked	51	21

3. Calculate the average delay for each destination

**SQL statement:**

```
SELECT f.destination, AVG(CAST(f.delay AS SIGNED)) as avg_delay  
FROM flight f  
GROUP BY f.destination;
```

**The results of the query:**

destination	avg_delay
New York	0.0000
Los Angeles	10.0000
Chicago	5.0000
London	20.0000
Tokyo	15.0000
Paris	0.0000

4. List all users with premium services

**SQL statement:**

```
SELECT u.*, ps.serviceType FROM user u  
INNER JOIN membership m ON u.userID = m.userID  
INNER JOIN premium_service ps ON m.membershipID = ps.membershipID;
```

**The results of the query:**

	userID	Fname	Lname	email	phoneNumber	address	serviceType
1	John	Doe		john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704	Priority Boarding
2	Alice	Smith		alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	Lounge Access
3	Bob	Johnson		bob@example.com	5554443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	Extra Baggage Allowance
4	Emma	Wilson		emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	Priority Boarding
5	Michael	Brown		michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	Lounge Access
6	Sophia	Lee		sophia@example.com	7778889999	345 Cedar Court, Apt 101, Dallas, TX 75201	Extra Baggage Allowance
7	Daniel	Garcia		daniel@example.com	9998887777	678 Birch Street, Suite 304, Seattle, WA 98101	Priority Boarding
8	Olivia	Martinez		olivia@example.com	6669991111	901 Walnut Avenue, Apt 22, Miami, FL 33101	Lounge Access
9	William	Lopez		william@example.com	2223334444	1123 Spruce Lane, Unit 45, New York, NY 10001	Extra Baggage Allowance
10	Ava	Gonzalez		ava@example.com	8887776666	4567 Sycamore Road, Apt 67, Chicago, IL 60601	Priority Boarding

5. Find the passengers who booked flights with fine dining

**SQL statement:**

```
SELECT p.* FROM passenger p
INNER JOIN flight f ON p.flightID = f.flightID
WHERE f.fine_dining = 'Yes';
```

**The results of the query:**

	passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID
31	John	Doe		555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1
33	Robert	Johnson		555-111-3333	Bronze	robert@example.com	1985-11-20	Male	3	3	3
34	Emily	Brown		555-222-4444	Platinum	emily@example.com	1988-03-12	Female	4	4	4
36	Olivia	Miller		555-444-6666	Silver	olivia@example.com	1993-09-25	Female	6	6	6
37	Michael	Wilson		555-555-7777	Bronze	michael@example.com	1980-12-05	Male	7	7	7
39	Daniel	Martinez		555-777-9999	Gold	daniel@example.com	1991-04-17	Male	9	9	9

6. Find the total amount spent by a user

**SQL statement:**

```
SELECT u.userID, u.Fname, u.Lname, SUM(p.amount) as total_amount_spent
FROM user u
LEFT JOIN payment p ON u.userID = p.userID
GROUP BY u.userID;
```

**The results of the query:**

userID	Fname	Lname	total_amount_sp...
1	John	Doe	2470
2	Alice	Smith	2450
3	Bob	Johnson	2410
4	Emma	Wilson	2500
5	Michael	Brown	2460
6	Sophia	Lee	2460
7	Daniel	Garcia	2510
8	Olivia	Martinez	2500
9	William	Lopez	2480
10	Ava	Gonzalez	2480

7. List all users who redeemed rewards

**SQL statement:**

```
SELECT u.*FROM user u  
INNER JOIN payment p ON u.userID = p.userID  
WHERE p.RewardID IS NOT NULL;
```

**The results of the query:**

userID	Fname	Lname	email	phoneNumber	address
1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704
1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704
1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202

8. calculate the total number of passengers per flight class

**SQL statement:**

```

SELECT t.class, COUNT(p.passengerID) as total_passengers
FROM ticket t
INNER JOIN passenger p ON t.bookingID = p.bookingID
GROUP BY t.class;

```

**The results of the query:**

class	total_passeng...
Economy	12
Business	9
First Class	9

- 9.** Find users who have not made any bookings

**SQL statement:**

```

SELECT u.* FROM user u
LEFT JOIN airlinebooking ab ON u.userID = ab.userID
WHERE ab.bookingID IS NULL;

```

**The results of the query:**

userID	Fname	Lname	email	phoneNumber	address

- 10.** List all passengers with their corresponding baggage information

**SQL statement:**

```

SELECT p.* , b.*

FROM passenger p

LEFT JOIN baggage b ON p.passengerID = b.passengerID;

```

**The results of the query:**

passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID	baggagelD	size	baggageStatus	passengerID	baggageNum
31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1	1	25	Checked	31	1
32	Alice	Smith	555-987-6543	Silver	alice@example.com	1992-08-15	Female	2	2	2	2	30	Carry-on	32	2
33	Robert	Johnson	555-111-3333	Bronze	robert@example.com	1985-11-20	Male	3	3	3	3	20	Checked	33	3
34	Emily	Brown	555-222-4444	Platinum	emily@example.com	1988-03-12	Female	4	4	4	4	22	Carry-on	34	4
35	William	Taylor	555-333-5555	Gold	william@example.com	1995-06-30	Male	5	5	5	5	27	Checked	35	5
36	Olivia	Miller	555-444-6666	Silver	olivia@example.com	1993-09-25	Female	6	6	6	6	18	Carry-on	36	6

**11. Retrieve the total revenue generated by each flight**

**SQL statement:**

```

SELECT f.flightID, f.flightNum, SUM(t.price) as total_revenue FROM flight f

LEFT JOIN ticket t ON f.flightID = t.flightID

GROUP BY f.flightID, f.flightNum;

```

**The results of the query:**

flightID	flightNum	total_revenue
1	AA123	350.00
2	DL456	780.00
3	UA789	1250.00
4	BA234	410.00
5	LH567	820.00
6	AF890	1320.00
7	SQ123	390.00
8	EK456	800.00
9	QF789	1300.00
10	TK234	380.00

**12. List all passengers with membership information**

**SQL statement:**

```
SELECT p.* , m.* FROM passenger p  
LEFT JOIN membership m ON p.userID = m.userID;
```

**The results of the query:**

	passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID	membershipID	registrationDate	frequentFlyerMiles	userID
	31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1	1	Gold	10000	1
	32	Alice	Smith	555-987-6543	Silver	alice@example.com	1992-08-15	Female	2	2	2	2	Silver	7500	2
	33	Robert	Johnson	555-111-3333	Bronze	robert@example.com	1985-11-20	Male	3	3	3	3	Bronze	5000	3
	34	Emily	Brown	555-222-4444	Platinum	emily@example.com	1988-03-12	Female	4	4	4	4	Gold	11000	4
	35	William	Taylor	555-333-5555	Gold	william@example.com	1995-06-30	Male	5	5	5	5	Silver	8000	5
	36	Olivia	Miller	555-444-6666	Silver	olivia@example.com	1993-09-25	Female	6	6	6	6	Bronze	5500	6

- 13.** Find the users who made payments using a specific payment method such as credit card

**SQL statement:**

```
SELECT u.* , p.paymentMethod FROM user u  
INNER JOIN payment p ON u.userID = p.userID  
WHERE p.paymentMethod = 'Credit Card';
```

**The results of the query:**

userID	Fname	Lname	email	phoneNumber	address	paymentMeth...
1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704	Credit Card
4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	Credit Card
7	Daniel	Garcia	daniel@example.com	9998887777	678 Birch Street, Suite 304, Seattle, WA 98101	Credit Card
10	Ava	Gonzalez	ava@example.com	8887776666	4567 Sycamore Road, Apt 67, Chicago, IL 60601	Credit Card
3	Bob	Johnson	bob@example.com	5554443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	Credit Card
6	Sophia	Lee	sophia@example.com	7778889999	345 Cedar Court, Apt 101, Dallas, TX 75201	Credit Card
9	William	Lopez	william@example.com	2223334444	1123 Spruce Lane, Unit 45, New York, NY 10001	Credit Card
2	Alice	Smith	alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	Credit Card
5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	Credit Card
8	Olivia	Martinez	olivia@example.com	6669991111	901 Walnut Avenue, Apt 22, Miami, FL 33101	Credit Card

- 14.** Retrieve the flights with the highest total revenue

**SQL statement:**

```
SELECT f.flightID, f.flightNum, SUM(t.price) as total_revenue  
FROM flight f  
LEFT JOIN ticket t ON f.flightID = t.flightID  
GROUP BY f.flightID, f.flightNum  
ORDER BY total_revenue DESC  
LIMIT 5;
```

**The results of the query:**

	flightID	flightNum	total_revenue
27	CX789	1320.00	
21	LH789	1320.00	
18	AY890	1320.00	
6	AF890	1320.00	
12	KL890	1300.00	

15. List all users who booked a flight with a specific destination such as New York

**SQL statement:**

```
SELECT u.* FROM user u  
INNER JOIN passenger p ON u.userID = p.userID  
INNER JOIN flight f ON p.flightID = f.flightID  
WHERE f.destination = 'New York';
```

**The results of the query:**

userID	Fname	Lname	email	phoneNumber	address
1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704

**16.** Retrieve the total number of users with premium services

**SQL statement:**

```
SELECT COUNT(DISTINCT u.userID) as total_users_with_premium_services
FROM user u
INNER JOIN membership m ON u.userID = m.userID
INNER JOIN premium_service ps ON m.membershipID = ps.membershipID;
```

**The results of the query:**

total_users_with_premium_servi...
10

**17.** Find the average delay for flights departing from each gate

**SQL statement:**

```
SELECT departureGate, AVG(CAST(delay AS SIGNED)) as avg_delay
FROM flight
GROUP BY departureGate;
```

**The results of the query:**

departureGate	avg_delay
Gate A1	0.0000
Gate B2	10.0000
Gate C3	5.0000
Gate D4	20.0000
Gate E5	15.0000
Gate F6	0.0000
Gate G7	25.0000
Gate H8	10.0000
Gate I9	30.0000
Gate J10	15.0000
Gate K11	0.0000

**18.** Retrieve the top 3 users with the highest total payments

**SQL statement:**

```
SELECT u.*, SUM(p.amount) as total_payments FROM user u
LEFT JOIN payment p ON u.userID = p.userID
GROUP BY u.userID
ORDER BY total_payments DESC
LIMIT 3;
```

**The results of the query:**

userID	Fname	Lname	email	phoneNumber	address	total_payments
7	Daniel	Garcia	daniel@example.com	9998887777	678 Birch Street, Suite 304, Seattle, WA 98101	2510
8	Olivia	Martinez	olivia@example.com	6669991111	901 Walnut Avenue, Apt 22, Miami, FL 33101	2500
4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	2500

**19.** Retrieve the flights with the highest and lowest delays

**SQL statement:**

```
SELECT flightID, destination, delay FROM flight  
WHERE delay = (SELECT MAX(delay) FROM flight)  
OR delay = (SELECT MIN(delay) FROM flight);
```

**The results of the query:**

	flightID	destination	delay
	1	New York	No Delay
	2	Los Angeles	10 mins
	6	Paris	No Delay
	8	Sydney	10 mins
	11	Madrid	No Delay
	14	Seoul	10 mins
	18	Toronto	No Delay
	21	Calgary	No Delay
	22	Edmonton	10 mins
	25	Halifax	No Delay
	27	Regina	10 mins
	30	Whitehorse	No Delay

**20.** Find the average size of baggage for each passenger

**SQL statement:**

```
SELECT p.passengerID, p.Fname, p.LName, AVG(b.size) as avg_baggage_size  
FROM passenger p  
LEFT JOIN baggage b ON p.passengerID = b.passengerID  
GROUP BY p.passengerID, p.Fname, p.LName;
```

**The results of the query:**

passengerID	Fname	LName	avg_baggage_size
31	John	Doe	25.0000
32	Alice	Smith	30.0000
33	Robert	Johnson	20.0000
34	Emily	Brown	22.0000
35	William	Taylor	27.0000
36	Olivia	Miller	18.0000
37	Michael	Wilson	21.0000

**21.** Retrieve the top 5 users with the highest frequent flyer miles

**SQL statement:**

```
SELECT u.*, m.frequentFlyerMiles  
FROM user u  
LEFT JOIN membership m ON u.userID = m.userID  
ORDER BY m.frequentFlyerMiles DESC  
LIMIT 5;
```

**The results of the query:**

userID	Fname	Lname	email	phoneNumber	address	frequentFlyerMil...
4	Emma	Wilson	emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	11000
7	Daniel	Garcia	daniel@example.com	9998887777	678 Birch Street, Suite 304, Seattle, WA 98101	10500
1	John	Doe	john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704	10000
10	Ava	Gonzalez	ava@example.com	8887776666	4567 Sycamore Road, Apt 67, Chicago, IL 60601	9500
5	Michael	Brown	michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	8000

**22.** List all flights with the count of passengers on each flight

**SQL statement:**

```
SELECT f.*, COUNT(p.passengerID) as total_passengers FROM flight f  
LEFT JOIN passenger p ON f.flightID = p.flightID  
GROUP BY f.flightID, f.flightNum, f.destination;
```

**The results of the query:**

	flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location	quantity_dining	fine_dining	total_passenger
1	AA123	New York	No Delay	Gate A1	12:00:00	13:00:00		Domestic	Yes	Lounge A	Full	Yes	1
2	DL456	Los Angeles	10 mins	Gate B2	14:30:00	15:30:00		Domestic	No	Lounge B	Limited	No	1
3	UA789	Chicago	5 mins	Gate C3	11:00:00	12:00:00		International	Yes	Lounge C	Full	Yes	1
4	BA234	London	20 mins	Gate D4	18:45:00	19:45:00		International	Yes	Lounge D	Full	Yes	1
5	LH567	Tokyo	15 mins	Gate E5	23:30:00	00:30:00		International	No	Lounge E	Limited	No	1
6	AF890	Paris	No Delay	Gate F6	09:00:00	10:00:00		International	Yes	Lounge F	Full	Yes	1
7	SQ123	Berlin	25 mins	Gate G7	21:15:00	22:15:00		International	Yes	Lounge G	Full	Yes	1

**23.** Retrieve the total number of bookings per month

**SQL statement:**

```
SELECT DATE_FORMAT(ab.bookingDate, '%Y-%m') as month,  
COUNT(ab.bookingID) as total_bookings  
FROM airlinebooking ab  
GROUP BY month;
```

**The results of the query:**

month	total_bookings
2023-11	16
2023-12	14

**24.** List all passengers who booked a flight with a specific departure gate such as Gate A1

**SQL statement:**

```
SELECT p.* FROM passenger p  
INNER JOIN flight f ON p.flightID = f.flightID  
WHERE f.departureGate = 'Gate A1';
```

**The results of the query:**

passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID
31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1

**25.** Retrieve the passengers who redeemed rewards with their reward points

**SQL statement:**

```
SELECT p.*, rr.rewardPoint FROM passenger p  
INNER JOIN rewardredemption rr ON p.userID = rr.membershipID;
```

**The results of the query:**

passengerID	Fname	LName	contactInfo	registerM	email	dob	gender	flightID	bookingID	userID	rewardPoint
31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1	50
31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1	150
31	John	Doe	555-123-4567	Gold	john@example.com	1990-05-05	Male	1	1	1	250
32	Alice	Smith	555-987-6543	Silver	alice@example.com	1992-08-15	Female	2	2	2	60
32	Alice	Smith	555-987-6543	Silver	alice@example.com	1992-08-15	Female	2	2	2	160
32	Alice	Smith	555-987-6543	Silver	alice@example.com	1992-08-15	Female	2	2	2	260
33	Robert	Johnson	555-111-3333	Bronze	robert@example.com	1985-11-20	Male	3	3	3	70

**26.** List all flights with the number of passengers who booked fine dining

**SQL statement:**

```
SELECT f.*, COUNT(p.passengerID) as fine_dining_passengers FROM flight f  
LEFT JOIN passenger p ON f.flightID = p.flightID AND f.fine_dining = 'Yes'  
GROUP BY f.flightID, f.flightNum, f.destination;
```

**The results of the query:**

	flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dining	lounge_location
1	AA123	New York	No Delay	Gate A1	12:00:00	13:00:00		Domestic	Yes	Lounge A
2	DL456	Los Angeles	10 mins	Gate B2	14:30:00	15:30:00		Domestic	No	Lounge B
3	UA789	Chicago	5 mins	Gate C3	11:00:00	12:00:00		International	Yes	Lounge C
4	BA234	London	20 mins	Gate D4	18:45:00	19:45:00		International	Yes	Lounge D
5	LH567	Tokyo	15 mins	Gate E5	23:30:00	00:30:00		International	No	Lounge E
6	AF890	Paris	No Delay	Gate F6	09:00:00	10:00:00		International	Yes	Lounge F
7	SQ123	Berlin	25 mins	Gate G7	21:15:00	22:15:00		International	Yes	Lounge G
8	EK456	Sydney	10 mins	Gate H8	03:45:00	04:45:00		International	No	Lounge H
9	QF789	Dubai	30 mins	Gate I9	16:20:00	17:20:00		International	Yes	Lounge I

**27.** Retrieve the top 3 destinations with the highest average delay

**SQL statement:**

```
SELECT f.destination, AVG(CAST(f.delay AS SIGNED)) as avg_delay FROM flight f  
GROUP BY f.destination  
ORDER BY avg_delay DESC  
LIMIT 3;
```

**The results of the query:**

destination	avg_delay
Dubai	30.0000
Istanbul	30.0000
Berlin	25.0000

**28.** Find the total number of bookings and payments per user

**SQL statement:**

```
SELECT u.userID, u.Fname, u.Lname, COUNT(DISTINCT ab.bookingID) as total_bookings,  
COUNT(DISTINCT p.paymentID) as total_payments FROM user u  
LEFT JOIN airlinebooking ab ON u.userID = ab.userID  
LEFT JOIN payment p ON u.userID = p.userID  
GROUP BY u.userID;
```

**The results of the query:**

userID	Fname	Lname	total_bookin...	total_payme...
1	John	Doe	3	3
2	Alice	Smith	3	3
3	Bob	Johnson	3	3
4	Emma	Wilson	3	3
5	Michael	Brown	3	3
6	Sophia	Lee	3	3
7	Daniel	Garcia	3	3
8	Olivia	Martinez	3	3
9	William	Lopez	3	3
10	Ava	Gonzalez	3	3

**29.** List all users who have booked flights to multiple destinations

**SQL statement:**

```
SELECT u.*, COUNT(DISTINCT f.destination) as unique_destinations FROM user u  
LEFT JOIN airlinebooking ab ON u.userID = ab.userID  
LEFT JOIN passenger p ON ab.bookingID = p.bookingID  
LEFT JOIN flight f ON p.flightID = f.flightID  
GROUP BY u.userID  
HAVING unique_destinations > 1;
```

**The results of the query:**

	userID	Fname	Lname	email	phoneNumber	address	unique_destination
1	John	Doe		john@example.com	9996754576	123 Main St, Apt 4B, Springfield, IL 62704	3
2	Alice	Smith		alice@example.com	9876543210	456 Oak Lane, Suite 203, Denver, CO 80202	3
3	Bob	Johnson		bob@example.com	5554443333	789 Maple Avenue, Unit 12, Raleigh, NC 27601	3
4	Emma	Wilson		emma@example.com	1112223333	101 Pine Street, Apt 567, San Francisco, CA 94...	3
5	Michael	Brown		michael@example.com	4445556666	234 Elm Drive, Unit 8D, Atlanta, GA 30301	3
6	Sophia	Lee		sophia@example.com	7778889999	345 Cedar Court, Apt 101, Dallas, TX 75201	3
7	Daniel	Garcia		daniel@example.com	9998887777	678 Birch Street, Suite 304, Seattle, WA 98101	3
8	Olivia	Martinez		olivia@example.com	6669991111	901 Walnut Avenue, Apt 22, Miami, FL 33101	3
9	William	Lopez		william@example.com	2223334444	1123 Spruce Lane, Unit 45, New York, NY 10001	3
10	Ava	Gonzalez		ava@example.com	8887776666	4567 Sycamore Road, Apt 67, Chicago, IL 60601	3

- 30.** List all flights with the number of passengers who booked a ticket in first class

**SQL statement:**

```
SELECT f.*, COUNT(DISTINCT t.ticketID) as first_class_passengers FROM flight f
LEFT JOIN ticket t ON f.flightID = t.flightID AND t.class = 'First Class'
GROUP BY f.flightID, f.flightNum, f.destination;
```

**The results of the query:**

flightID	flightNum	destination	delay	departureGate	boardingTime	departureTime	flight_Type	standard_dinner	lounge_location
1	AA123	New York	No Delay	Gate A1	12:00:00	13:00:00	Domestic	Yes	Lounge A
2	DL456	Los Angeles	10 mins	Gate B2	14:30:00	15:30:00	Domestic	No	Lounge B
3	UA789	Chicago	5 mins	Gate C3	11:00:00	12:00:00	International	Yes	Lounge C
4	BA234	London	20 mins	Gate D4	18:45:00	19:45:00	International	Yes	Lounge D
5	LH567	Tokyo	15 mins	Gate E5	23:30:00	00:30:00	International	No	Lounge E
6	AF890	Paris	No Delay	Gate F6	09:00:00	10:00:00	International	Yes	Lounge F

For more details visit:

<https://drive.google.com/drive/folders/1wl1kB-SrocXA1sE0umSsRMvQBX5saUJq?usp=sharing>

## **12 Conclusion**

Our project unfolded in two distinct phases, providing an in-depth exploration of Thai Airways International Public Co., Ltd.'s operational intricacies and database implementation journey.

In phase one, business domain analysis, delving into the airline's core processes—Member registration/login, Booking, Payment, Check-in, Boarding, and Additional services—we identified critical business rules shaping Thai Airways' operations. The main points for users were the ease with which they could register for membership, manage booking, book tickets, and check in to their flight. We also looked into how users' membership tier determines the premium services they can use and also get reward redemption to discount their payment. Our group utilized these rules as foundational elements to construct a comprehensive Entity-Relationship Diagram (ERD/EERD). This diagram not only visually represented the interplay between different processes but also served as a blueprint for subsequent phases.

In phase two, database implementation, our focus shifted to translating the complex ERD/EERD into a streamlined Mini ERD/EERD. Employing a step-by-step transformation approach, we simplified the structures for enhanced clarity. The culmination of this phase manifested in the creation of a final relational schema, laying the groundwork for database implementation. The relational schema, when translated into Data Definition Language (DDL) and Data Manipulation Language (DML) scripts, ensured the seamless execution of the MySQL database.

To maximize practicality, we formulated SQL queries tailored to the identified business processes. These queries, spanning basic and advanced levels, aimed to assist administrators in meaningful decision-making. For instance, queries like finding passengers who booked flights with the

same destination or retrieving baggage information for specific passengers were crafted to align with our refined understanding of Thai Airways' operational nuances.

In essence, our project encapsulates a comprehensive journey—from dissecting Thai Airways' intricate business processes and rules to transforming them into a functional database and extracting meaningful insights through tailored SQL queries. This holistic approach not only enhances the airline's operational efficiency but also empowers administrators with valuable tools for informed decision-making.

## 13 References

- [1] Thai Airways International Public Co., Ltd., accessed September 4, 2023.  
<https://www.thaiairways.com/>
- [2] Thai Airways International Public Co., Ltd., accessed September 4, 2023.  
[https://www.thaiairways.com/en\\_TH/rop/index.page](https://www.thaiairways.com/en_TH/rop/index.page)
- [3] Reference 3 information.