PUNJAB UNIVERSITY COLLEGE OF INFORMATION AND TECHNOLOGY

DATABASE SYSTEMS - POJECT REPORT

SkyLink Airline

Submitted To:

Sir Asif Sohail

Submitted By:

Raima Shahzad (BCSF21M015)

Maheen Munir (BCSF21M020)

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

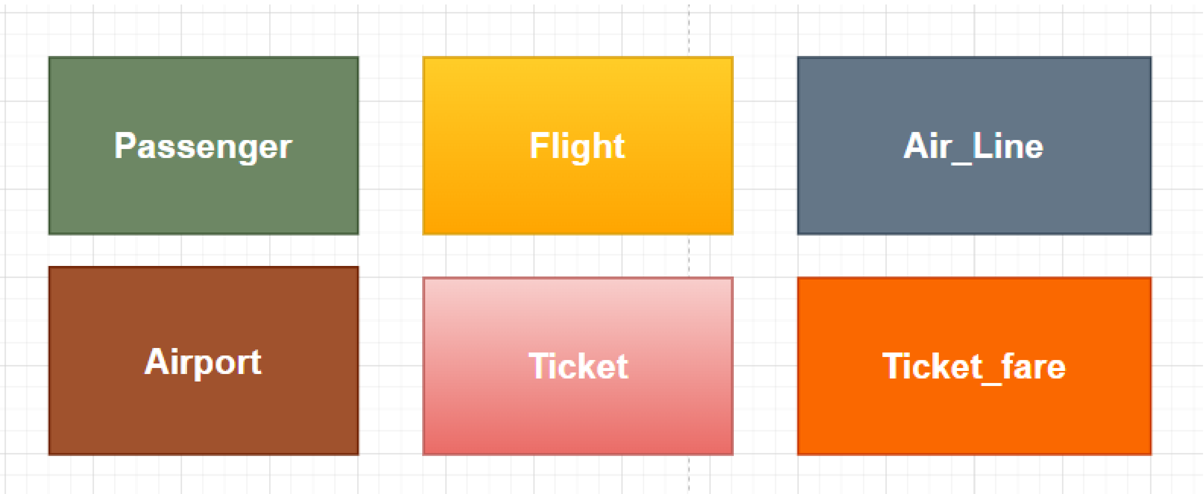
TABLE OF CONTENT

Sr#	Content	Page#
1	Introduction to the working of the system 1.1 Overview 1.2 Key Functionalities 1.3 Work Flow & Interaction 1.4 Data Integrity and Security	3
2	Problems in the existing system	6
3	ERD of the system	7
4	Top Down Approach	8
5	Bottom-Up Approach	9
6	Description of the relations	11
7	CREATE TABLE statements	13
8	Tables Data	18
9	Designing Views	21
10	Relational data model showing the association among different relations	22
11	SELECT statement for common reports to be generated by the system.	24
12	Demonstration of functions, stored procedures, and database triggers.	30

SkyLink Airline

Overview

The airline reservation system is designed to facilitate the booking, management, and monitoring of flights, passengers, tickets, and fares. It serves as an integrated platform that connects various entities involved in air travel, including passengers, flights, airlines, and airports. This system aims to streamline the operations of booking flights, managing passenger information, scheduling flights, and processing fare details.



Key Functionalities

The system provides several critical functionalities that enhance the efficiency and reliability of airline operations:

Booking and Ticket Management:

- **Booking Flights:** Passengers can book tickets for flights based on their preferences, such as class (economy, business, or first class), departure time, and destination.
- **Ticket Generation:** Once a booking is confirmed, a ticket is generated with a unique Ticket_No, which includes details like Seat_No, Booking_Date, Flight_Id, and Passenger_Id.

Passenger Management:

- **Personal Information:** The system maintains a detailed record of passenger information, including Name, Phone, Age, Address, and Passport_No. This ensures that all necessary details are readily available for check-in and security purposes.
- **Ticket Association:** Each passenger's details are linked to their respective tickets, allowing for easy retrieval and management of booking information.

Flight Scheduling and Management:

- **Flight Information:** The system stores comprehensive details about each flight, including Flight_Id, AirLine_Id, Aircraft_Id, Date, Departure_Airport, Arrival_Airport, Departure_Time, Arrival_Time, and Flight_Status.
- **Airport Coordination**: It manages the coordination between different airports, tracking flights that land or take off from various locations.

Airline and Aircraft Management:

- **Airline Details:** Information about different airlines, such as AirLine_Id, Name, Country, and Headquarters, is maintained.
- **Aircraft Information:** The system tracks the Aircraft_Id associated with each flight to manage aircraft assignments effectively.

Fare Management:

- **Class-based Pricing:** The system manages fare details for different classes of service (Economy_Class_Price, Business_Class_Price, and First_Class_Price) associated with each flight.

Workflow and Interaction

The system involves several key interactions and workflows among its entities:

1. **Passenger Booking:**
    - A passenger visits the airline's booking portal or contacts a booking agent.
    - They provide personal details and select their preferred flight.
    - The system checks the availability of seats and confirms the booking.
    - A ticket is generated and linked to the passenger's record.

2. **Flight Management:**
    - Airlines schedule flights and provide details such as departure and arrival times, and associated airports.
    - The system updates the flight schedules and ensures that all flights are accurately reflected in the database.

3. **Airport Coordination:**

    - Airports provide information about their facilities, including available gates, runways, and operational status.
    - The system tracks which flights are landing or taking off from specific airports, ensuring smooth coordination.

4. **Fare Calculation and Management:**
    - The system calculates fares based on the class of service selected by the passenger.
    - It stores and retrieves fare information to ensure accurate billing and financial tracking.

Data Integrity and Security

The system is designed with robust data integrity and security measures to ensure the accuracy and confidentiality of information:

- Primary Keys and Foreign Keys: Each table uses primary keys (e.g., Passenger_Id, Ticket_No, Flight_Id) to uniquely identify records and foreign keys to maintain referential integrity across tables.
- Data Validation: Constraints such as NOT NULL, UNIQUE, and CHECK are applied to ensure valid data entries.
- Access Control: Different levels of access are provided to users based on their roles (e.g., passengers, booking agents, airline staff) to prevent unauthorized access to sensitive information.

2. Problems in the Existing System

Manual Booking and Management

- **Error-Prone:** High risk of human errors in data entry and ticketing.
- **Time-Consuming**: Manual processes lead to longer wait times and inefficiencies.

Inefficient Data Handling

- **Data Inconsistency:** Difficult to maintain accurate and consistent data.

- **Lack of Real-Time Updates:** Delays in updating flight statuses and booking confirmations.

Poor Customer Experience

- **Inconvenient Booking Process:** Limited booking channels and dependency on physical offices or calls.
- **Lack of Personalized Service:** Difficulty in providing tailored services and managing loyalty programs.

Limited Reporting and Analytics

- **Insufficient Reporting:** Challenges in generating insightful operational and financial reports.
- **Difficulty in Tracking and Analysis:** Cumbersome to analyze historical data for trends and optimization.

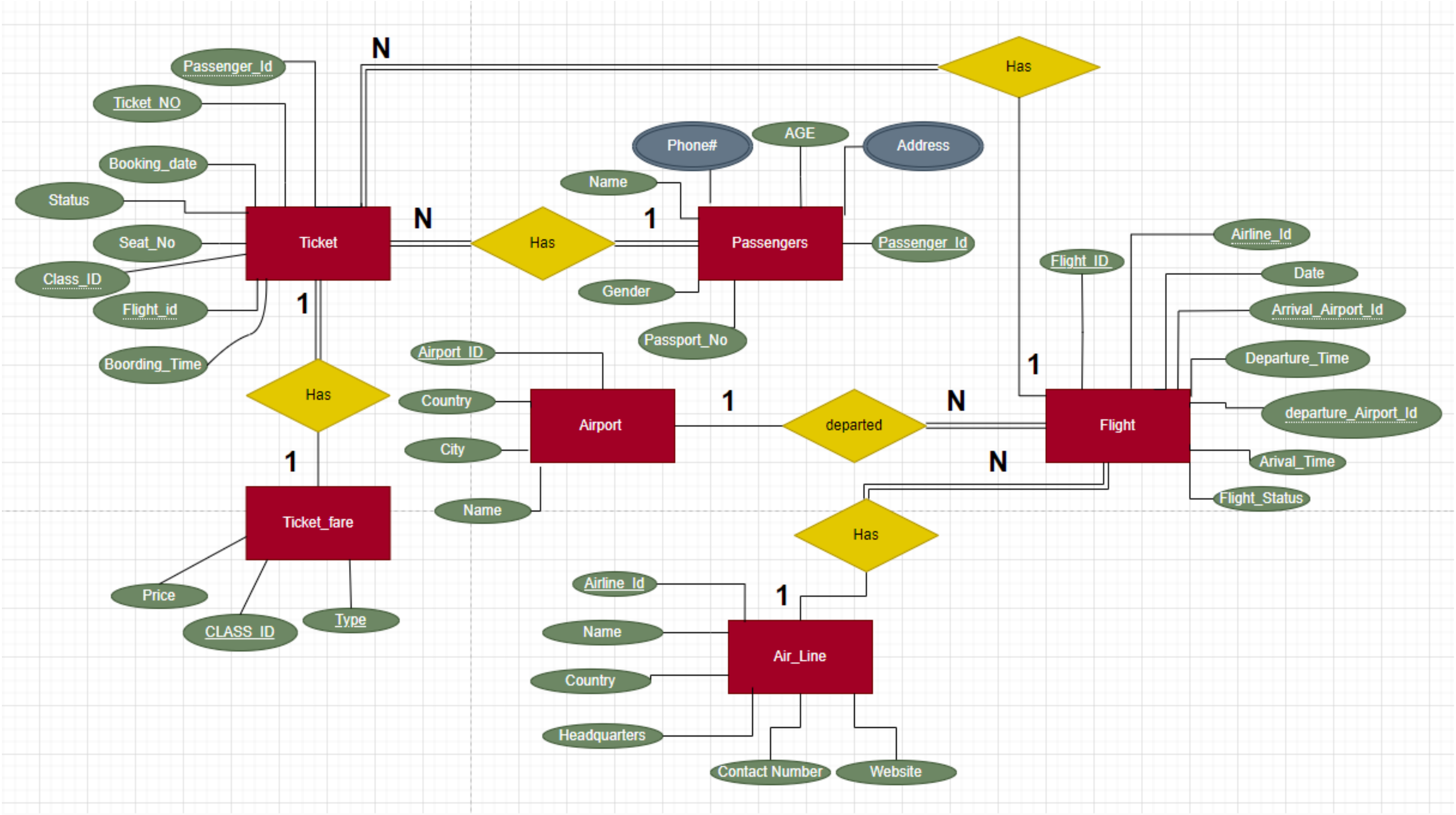
Security and Data Integrity Issues

- **Vulnerability to Data Loss**: Risks of losing data due to accidents or system failures**.**
- **Weak Data Security:** Inadequate protection of sensitive passenger information.

Operational Inefficiencies

- **Resource Management Challenges:** Complex and inefficient management of aircraft, crew, and ground services.
- **Coordination Issues**: Poor communication between airports, airlines, and services affecting operations.

3. Entity-Relationship Daigram (ERD)



Top Down Approach

Identifying_identities:

- Passengers
- Ticket
- Ticket_fare
- Airport
- Flight
- Air_line
- Ticket_Fare

Defining tables (un_normalized):

- **Passengers** (Passenger_Id, Name, Phone#, Age, Address, Gender, Passport_No)
- **Ticket** (Ticket_No, Passenger_Id, Class_ID, Flight_id, Booking_date, Status, Seat_No, Boarding_Time)
- **Ticket_fare** (Class_ID, Price, Type)
- **Airport** (Airport_ID, Name, Country, City)
- **Flight** (Flight_ID, Airline_Id , Date, Departure_Time, Arrival_Time, Departure_Airport_Id, Arrival_Airport_Id, Flight_Status)
- **Air_Line** (Airline_Id, Name, Country, Headquarters, Contact_Number, Website)

NORMALIZATION:

❖Passengers

(Passenger_Id, Name, Phone#, Age, Address, Gender, Passport_No)

- Converting to **(1NF)**
  - Passengers (Passenger_Id, Name, AGE, Gender, Passport_No)
  - Phone_Number (Phone_Number , Passenger_Id)
  - Addresse (Address_Id, Passenger_Id, House# , Street#, City , Country , State)
- No Partial Dependency of any non-prime attribute **(2 NF)**
- No transitive dependency **(3 NF)**

❖AIR_LINE

(Airline_Id, Name, Country, Headquarters, Contact_Number, Website)

- Converting to **(1NF)**
  - Air_Line (Airline_Id, Name, Country, Headquarters, Website)
  - Contact (Contact_Number, Airline_Id,)
- No Partial Dependency of any non-prime attribute **(2 NF)**
- No transitive dependency **(3 NF)**

❖Ticket

(Ticket_No, Passenger_Id, Class_ID, Flight_id, Booking_date, Status, Seat_No, Boarding_Time)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- No transitive dependency (3 NF)

❖Airport

(Airport_ID, Name, Country, City)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- Converting into 3NF
  - Airport(Airport_ID, Name, City)
  - City_Country(City, Country)

❖Flight

(Flight_ID, Airline_Id , F_Date, Departure_Time, Arrival_Time, Departure_Airport_id, Arrival_Airport_id, Flight_Status)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- No transitive dependency (3 NF)

❖Ticket_Fare

(Class_ID, Price, Type)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- No transitive dependency (3 NF)

BOTTOM-UP-APPROACH:

First, We make a bulky relation comprising all attributes.

- **RELATION** ➡ **(**Passenger_Id, Name, Phone#, Age, Address, Gender, Passport_No, Ticket_No, Passenger_Id, Class_ID, Flight_id, Booking_date, Status, Seat_No, Boarding_Time,Class_ID, Price, Type,Airport_ID, Name, Country, City, Flight_ID, Airline_Id, Departure_Airport_Id, F_Date, Departure_Time, Arrival_Time, Departure_Airport_Id, Arrival_Airport_Id, Flight_Status,Airline_Id, Name, Country, Headquarters, Contact_Number, Website)

- As mentioned above, the relation consists of all the attributes in our present ERD, we will construct a sub-relation from the above and perform

❖RELATION ⟹ 1

Passenger (Passenger_Id, Name, Phone#, Age, Address, Gender, Passport_No)

- Converting to **(1NF)**
  - Passengers (Passenger_Id, Name, AGE, Gender, Passport_No)
  - Phone_Number (Phone_Number , Passenger_Id)
  - Addresse (Address_Id, Passenger_Id, House# , Street#, City , Country , State)
- No Partial Dependency of any non-prime attribute **(2 NF)**
- No transitive dependency **(3 NF)**

❖RELATION ⟹ 2

Air_Line (Airline_Id, Name, Country, Headquarters, Contact_Number, Website)

- Converting to **(1NF)**
  - Air_Line (Airline_Id, Name, Country, Headquarters, Website)
  - Contact (Contact_Number, Airline_Id,)
- No Partial Dependency of any non-prime attribute **(2 NF)**
- No transitive dependency **(3 NF)**

❖RELATION ⟹ 3

Ticket (Ticket_No, Passenger_Id, Class_ID, Flight_id, Booking_date, Status, Seat_No, Boarding_Time)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- No transitive dependency (3 NF)

❖RELATION ⟹ 4

Airport (Airport_ID, Name, Country, City)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- Converting into 3NF
  - Airport(Airport_ID, Name, City)
  - City_Country(City, Country)

❖RELATION ➡ 5

Flight (Flight_ID, Airline_Id , F_Date, Departure_Time, Arrival_Time, Departure_Airport_id, Arrival_Airport_id, Flight_Status)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- No transitive dependency (3 NF)

❖RELATION ➡ 6

Ticket_Fare (Class_ID, Price, Type)

- No Duplicating Data as there is only one PK (1 NF)
- No Partial Dependency of any non-prime attribute (2 NF)
- No transitive dependency (3 NF)

5. Description of the relations

Table Name: Passenger

Attribute	Data Type	Size	Constraints
Passenger_Id	VARCHAR2	10	Primary Key
Name	VARCHAR2	50	Not Null
Age	NUMBER		
Gender	VARCHAR2	1	Check ('M', 'F')
Passport_No	VARCHAR2	15	Unique

Table Name: Phone_Number

Attribute	Data Type	Size	Constraints
Phone_Number	VARCHAR2	15	Primary Key
Passenger_Id	VARCHAR2	10	Foreign Key (Passengers.Passenger_Id)

Table Name: Addresse

Attribute	Data Type	Size	Constraints
Address_Id	VARCHAR2	10	Primary Key
Passenger_Id	VARCHAR2	10	Foreign Key (Passengers.Passenger_Id)
House#	VARCHAR2	10	

Attribute	Data Type	Size	Constraints
Street#	VARCHAR2	10	
City	VARCHAR2	50	
Country	VARCHAR2	50	
State	VARCHAR2	50	

Table Name: Air_Line

Attribute	Data Type	Size	Constraints
Airline_Id	VARCHAR2	10	Primary Key
Name	VARCHAR2	50	Not Null
Country	VARCHAR2	50	
Headquarters	VARCHAR2	100	
Website	VARCHAR2	100	

Table Name: Contact

Attribute	Data Type	Size	Constraints
Contact_Number	VARCHAR2	15	Primary Key
Airline_Id	VARCHAR2	10	Foreign Key (Air_Line.Airline_Id)

Table Name: Airport

Attribute	Data Type	Size	Constraints
Airport_ID	VARCHAR2	10	Primary Key
Name	VARCHAR2	50	Not Null
City	VARCHAR2	50	Foreign Key (City_Country.City)

Table Name: City_Country

Attribute	Data Type	Size	Constraints
City	VARCHAR2	50	Primary Key
Country	VARCHAR2	50	

Table Name: Ticket

Attribute	Data Type	Size	Constraints
Ticket_No	VARCHAR2	10	Primary Key
Passenger_Id	VARCHAR2	10	Foreign Key (Passengers.Passenger_Id)
Class_ID	VARCHAR2	10	Foreign Key (Ticket_Fare.Class_ID)
Flight_Id	VARCHAR2	10	Foreign Key (Flight.Flight_ID)
Booking_Date	DATE		
Status	VARCHAR2	10	Check ('Booked', 'Cancelled', 'Completed')
Seat_No	VARCHAR2	10	
Boarding_Time	TIMESTAMP		

Table Name: Flight

Attribute	Data Type	Size	Constraints
Flight_ID	VARCHAR2	10	Primary Key
Airline_Id	VARCHAR2	10	Foreign Key (Air_Line.Airline_Id)
F_Date	DATE		
Departure_Time	TIMESTAMP		
Arrival_Time	TIMESTAMP		
Departure_Airport_id	Number	50	Foreign Key (Airport.Id)
Arrival_Airport_id	Number	50	Foreign Key (Airport.id)
Flight_Status	VARCHAR2	10	Check ('On Time', 'Delayed', 'Cancelled')

Table Name: Ticket_Fare

Attribute	Data Type	Size	Constraints
Class_ID	VARCHAR2	10	Primary Key
Price	NUMBER	10,2	
Type	VARCHAR2	10	Check ('Economy', 'Business', 'First Class')

6.CREATE TABLE statements

Table Name: Passenger

```
CREATE TABLE Passenger (
   Passenger_Id VARCHAR2(10) PRIMARY KEY,
   Name VARCHAR2(50) NOT NULL,
   Age INTEGER CHECK (Age > 0),
   Passport_No VARCHAR2(50) UNIQUE,
   Gender VARCHAR2(10) CHECK (Gender IN ('Male', 'Female', 'Other'))
);
```

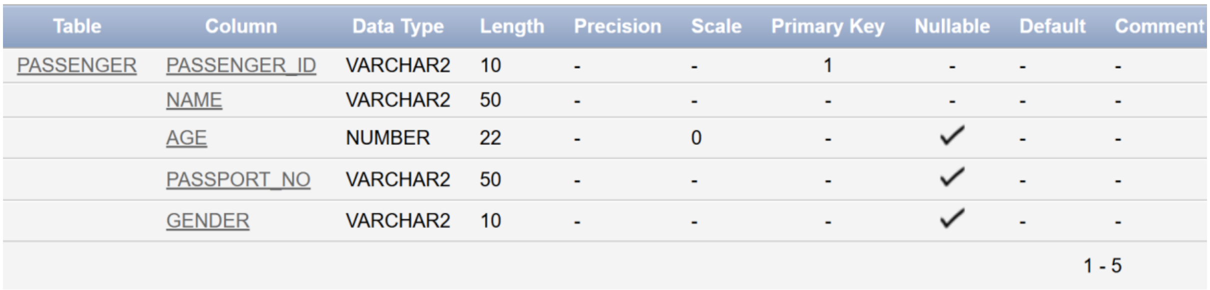


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PASSENGER	PASSENGER_ID	VARCHAR2	10	-	-	1	-	-	-
	NAME	VARCHAR2	50	-	-	-	-	-	-
	AGE	NUMBER	22	-	0	-	✓	-	-
	PASSPORT_NO	VARCHAR2	50	-	-	-	✓	-	-
	GENDER	VARCHAR2	10	-	-	-	✓	-	-

1 - 5

Table Name: Phone_Number

```
CREATE TABLE Phone_Number(
    Phone_Number VARCHAR2(15) PRIMARY KEY,
    Passenger_Id VARCHAR2(10),
    FOREIGN KEY (Passenger_Id) REFERENCES Passenger(Passenger_Id) ON DELETE CASCADE
);
```

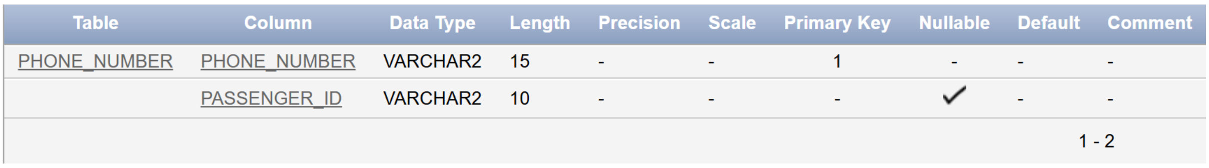


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PHONE_NUMBER	PHONE_NUMBER	VARCHAR2	15	-	-	1	-	-	-
	PASSENGER_ID	VARCHAR2	10	-	-	-	✓	-	-

1 - 2

Table Name:Address

```
CREATE TABLE Addresse (
    Address_Id VARCHAR2(10) PRIMARY KEY,
    Passenger_Id VARCHAR2(10),
    House# VARCHAR2(10),
    Street# VARCHAR2(10),
    City VARCHAR2(50),
    Country VARCHAR2(50),
    State VARCHAR2(50),
    FOREIGN KEY (Passenger_Id) REFERENCES Passenger(Passenger_Id) ON DELETE CASCADE
);
```

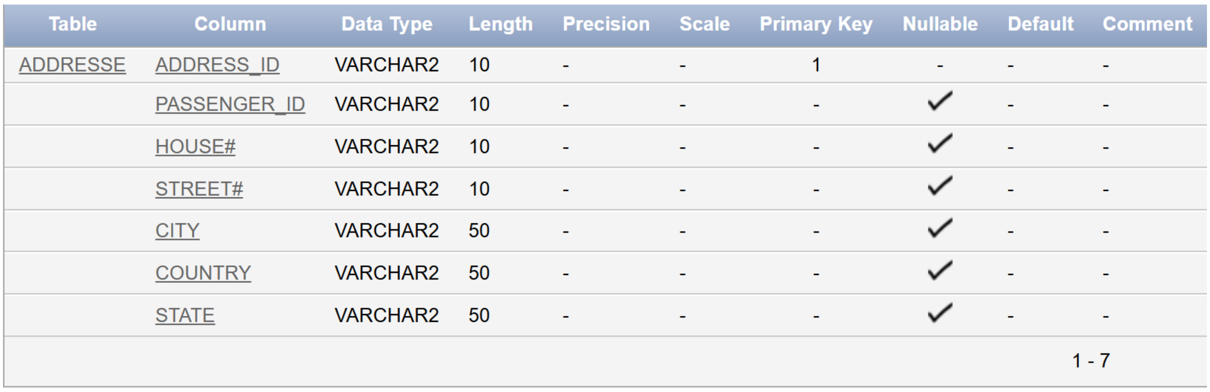


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADDRESSE	ADDRESS_ID	VARCHAR2	10	-	-	1	-	-	-
	PASSENGER_ID	VARCHAR2	10	-	-	-	✓	-	-
	HOUSE#	VARCHAR2	10	-	-	-	✓	-	-
	STREET#	VARCHAR2	10	-	-	-	✓	-	-
	CITY	VARCHAR2	50	-	-	-	✓	-	-
	COUNTRY	VARCHAR2	50	-	-	-	✓	-	-
	STATE	VARCHAR2	50	-	-	-	✓	-	-

1 - 7

Table Name: Air_Line

```
CREATE TABLE Air_Line (
    Airline_Id VARCHAR2(10) PRIMARY KEY,
    Name VARCHAR2(50) NOT NULL,

    Country VARCHAR2(50),

    Headquarters VARCHAR2(100),

    Website VARCHAR2(100)

);
```

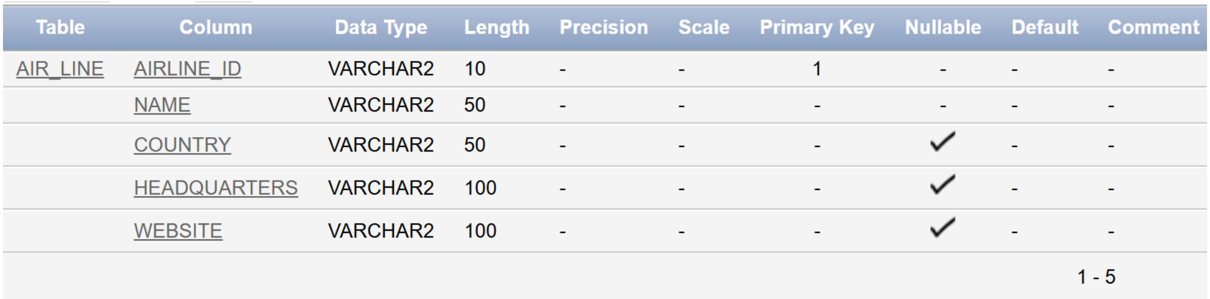


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
AIR_LINE	AIRLINE_ID	VARCHAR2	10	-	-	1	-	-	-
	NAME	VARCHAR2	50	-	-	-	-	-	-
	COUNTRY	VARCHAR2	50	-	-	-	✓	-	-
	HEADQUARTERS	VARCHAR2	100	-	-	-	✓	-	-
	WEBSITE	VARCHAR2	100	-	-	-	✓	-	-

1 - 5

Table Name: Contact

```
CREATE TABLE Contact (

    Contact_Number VARCHAR2(15) PRIMARY KEY,

    Airline_Id VARCHAR2(10),

    FOREIGN KEY (Airline_Id) REFERENCES Air_Line(Airline_Id) ON DELETE CASCADE

);
```

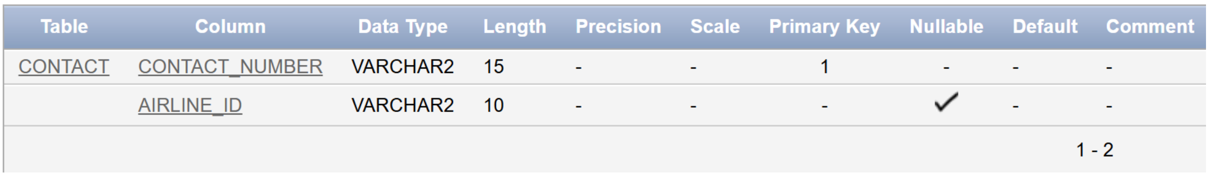


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CONTACT	CONTACT_NUMBER	VARCHAR2	15	-	-	1	-	-	-
	AIRLINE_ID	VARCHAR2	10	-	-	-	✓	-	-

1 - 2

Table Name: Airport

```
CREATE TABLE Airport (

      VARCHAR2(10) PRIMARY KEY,

    Name VARCHAR2(50) NOT NULL,

    City VARCHAR2(50),

    FOREIGN KEY (City) REFERENCES City_Country(City) ON DELETE CASCADE

);
```

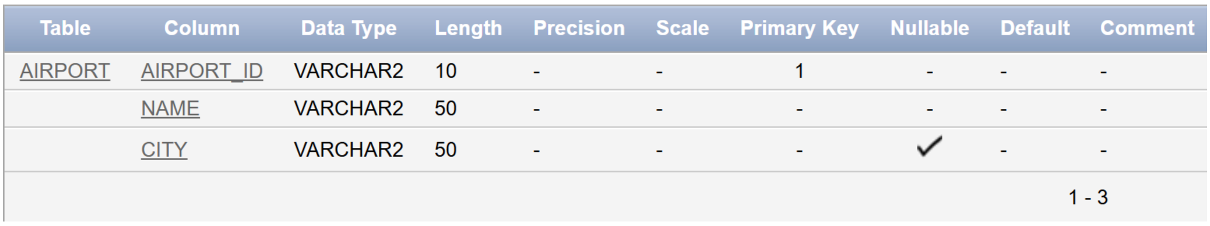


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
AIRPORT	AIRPORT_ID	VARCHAR2	10	-	-	1	-	-	-
	NAME	VARCHAR2	50	-	-	-	-	-	-
	CITY	VARCHAR2	50	-	-	-	✓	-	-

1 - 3

Table Name: City_Country

```
CREATE TABLE City_Country (

   City VARCHAR2(50) PRIMARY KEY,

   Country VARCHAR2(50)

);
```

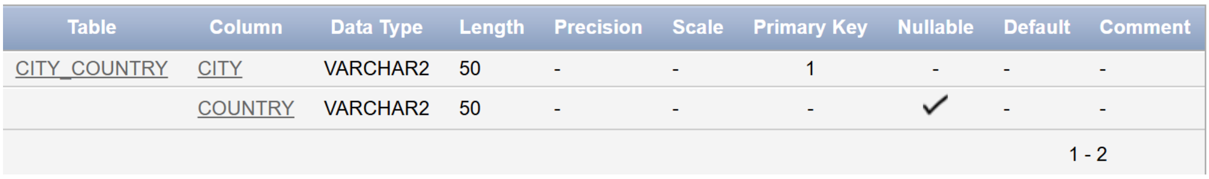


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CITY_COUNTRY	CITY	VARCHAR2	50	-	-	1	-	-	-
	COUNTRY	VARCHAR2	50	-	-	-	✓	-	-

1 - 2

Table Name: Ticket

```
CREATE TABLE Ticket (

   Ticket_No VARCHAR2(10) PRIMARY KEY,

   Passenger_Id VARCHAR2(10),

   Class_ID VARCHAR2(10),

   Flight_Id VARCHAR2(10),

   Booking_Date DATE,

   Status VARCHAR2(10) CHECK (Status IN ('Booked', 'Cancelled', 'Completed')),

   Seat_No VARCHAR2(10),

   Boarding_Time TIMESTAMP,

   FOREIGN KEY (Passenger_Id) REFERENCES Passenger(Passenger_Id) ON DELETE CASCADE,

   FOREIGN KEY (Class_ID) REFERENCES Ticket_Fare(Class_ID) ON DELETE CASCADE,

   FOREIGN KEY (Flight_Id) REFERENCES Flight(Flight_ID) ON DELETE CASCADE


);
```

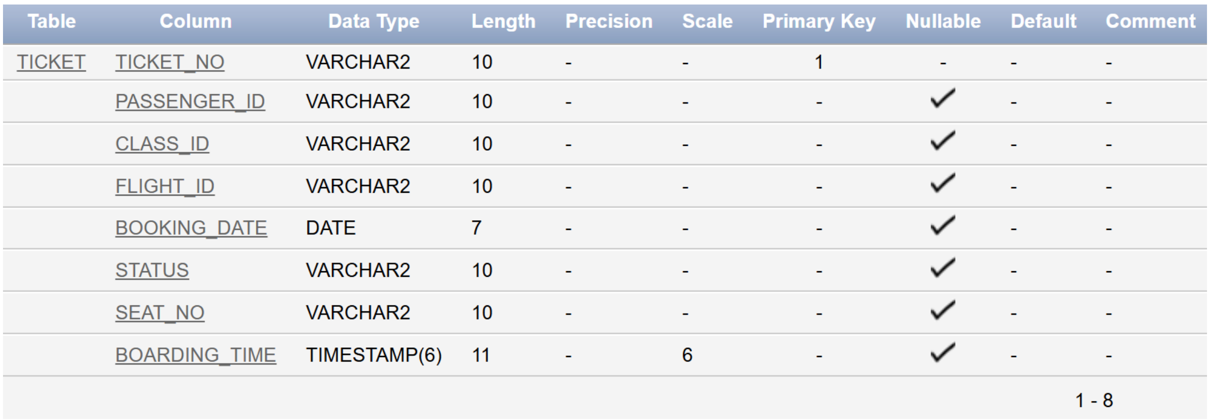


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET	TICKET_NO	VARCHAR2	10	-	-	1	-	-	-
	PASSENGER_ID	VARCHAR2	10	-	-	-	✓	-	-
	CLASS_ID	VARCHAR2	10	-	-	-	✓	-	-
	FLIGHT_ID	VARCHAR2	10	-	-	-	✓	-	-
	BOOKING_DATE	DATE	7	-	-	-	✓	-	-
	STATUS	VARCHAR2	10	-	-	-	✓	-	-
	SEAT_NO	VARCHAR2	10	-	-	-	✓	-	-
	BOARDING_TIME	TIMESTAMP(6)	11	-	6	-	✓	-	-

1 - 8

Table Name: Ticket_Fare

```
CREATE TABLE Ticket_Fare (
Class_ID VARCHAR2(10) PRIMARY KEY,
   Price NUMBER(10,2),
   Type VARCHAR2(10) CHECK (Type IN ('Economy', 'Business', 'First Class')));
```

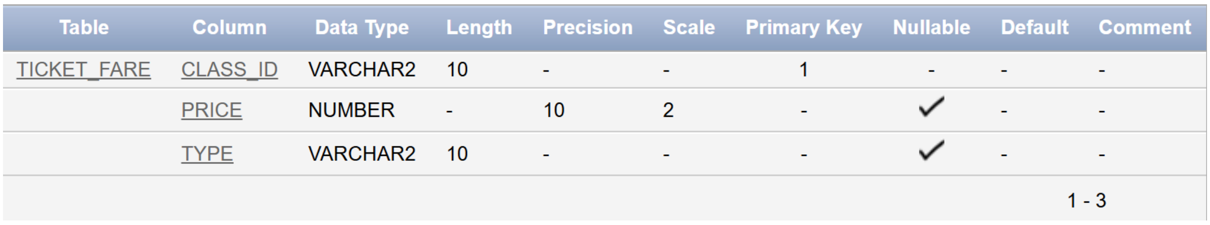


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TICKET_FARE	CLASS_ID	VARCHAR2	10	-	-	1	-	-	-
	PRICE	NUMBER	-	10	2	-	✓	-	-
	TYPE	VARCHAR2	10	-	-	-	✓	-	-

1 - 3

Table Name: Flight table

```
CREATE TABLE Flight (
   Flight_ID VARCHAR2(10) PRIMARY KEY,
   Airline_Id VARCHAR2(10),
   F_Date DATE,
   Departure_Time TIMESTAMP,
   Arrival_Time TIMESTAMP,
   Departure_Airport_id  VARCHAR2(10),
   Arrival_Airport_id  VARCHAR2(10),
   Flight_Status VARCHAR2(10) CHECK (Flight_Status IN ('On Time', 'Delayed', 'Cancelled')),
   FOREIGN KEY (Airline_Id) REFERENCES Air_Line(Airline_Id) ON DELETE CASCADE,
   FOREIGN KEY (Departure_Airport_id) REFERENCES Airport(Airport_ID) ON DELETE CASCADE,
   FOREIGN KEY (Arrival_Airport_id) REFERENCES Airport(Airport_ID) ON DELETE CASCADE
);
```

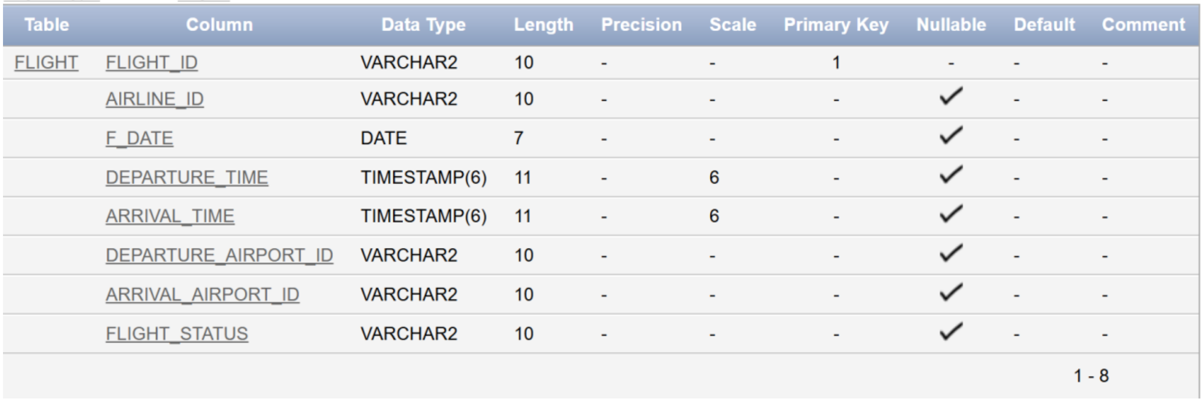
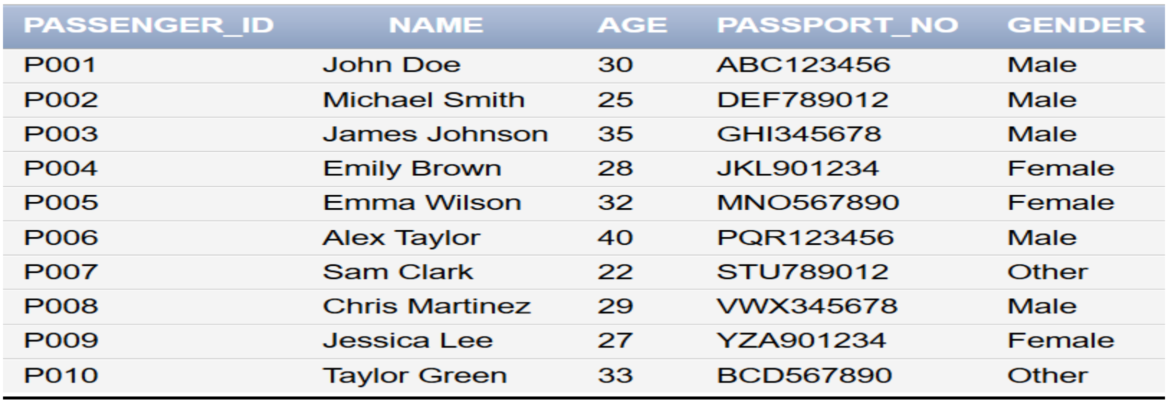


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FLIGHT	FLIGHT_ID	VARCHAR2	10	-	-	1	-	-	-
	AIRLINE_ID	VARCHAR2	10	-	-	-	✓	-	-
	F_DATE	DATE	7	-	-	-	✓	-	-
	DEPARTURE_TIME	TIMESTAMP(6)	11	-	6	-	✓	-	-
	ARRIVAL_TIME	TIMESTAMP(6)	11	-	6	-	✓	-	-
	DEPARTURE_AIRPORT_ID	VARCHAR2	10	-	-	-	✓	-	-
	ARRIVAL_AIRPORT_ID	VARCHAR2	10	-	-	-	✓	-	-
	FLIGHT_STATUS	VARCHAR2	10	-	-	-	✓	-	-

1 - 8

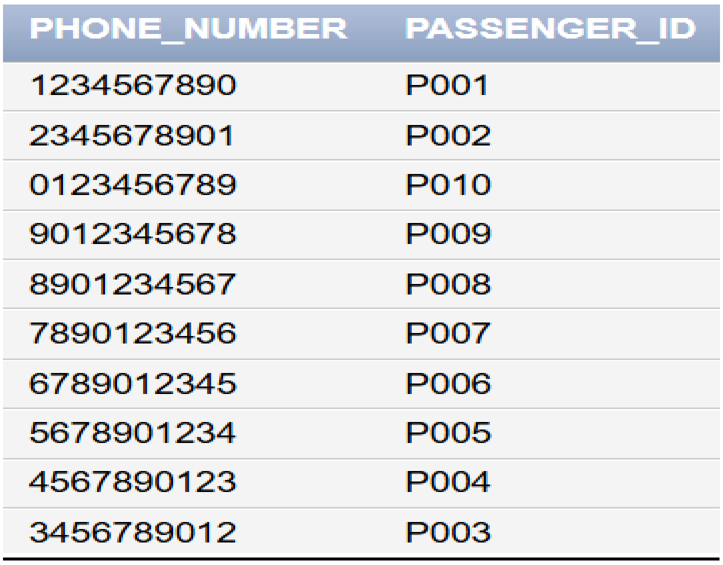
Tables Data

1. Passenger



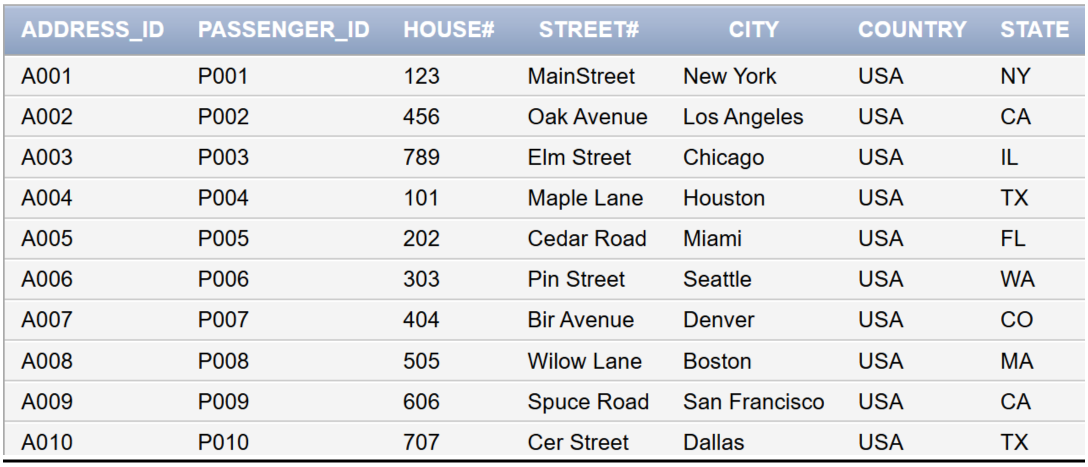
PASSENGER_ID	NAME	AGE	PASSPORT_NO	GENDER
P001	John Doe	30	ABC123456	Male
P002	Michael Smith	25	DEF789012	Male
P003	James Johnson	35	GHI345678	Male
P004	Emily Brown	28	JKL901234	Female
P005	Emma Wilson	32	MNO567890	Female
P006	Alex Taylor	40	PQR123456	Male
P007	Sam Clark	22	STU789012	Other
P008	Chris Martinez	29	VWX345678	Male
P009	Jessica Lee	27	YZA901234	Female
P010	Taylor Green	33	BCD567890	Other

2. Phone_Number



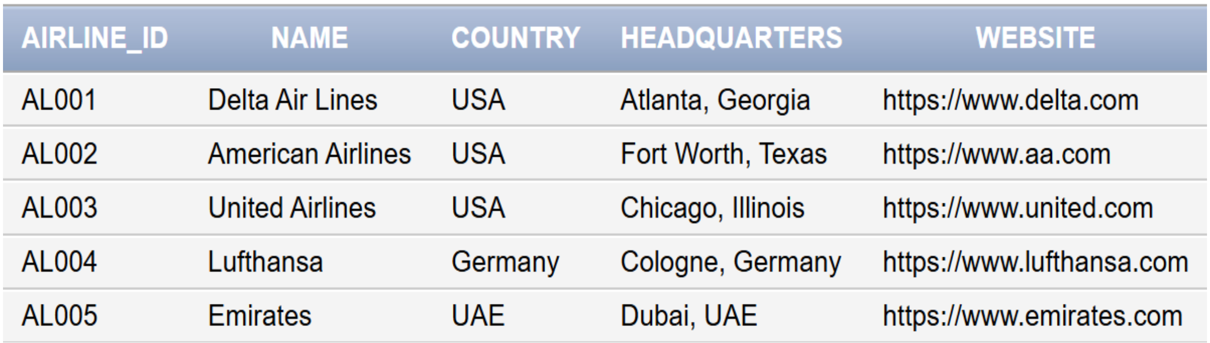
PHONE_NUMBER	PASSENGER_ID
1234567890	P001
2345678901	P002
0123456789	P010
9012345678	P009
8901234567	P008
7890123456	P007
6789012345	P006
5678901234	P005
4567890123	P004
3456789012	P003

3. Address



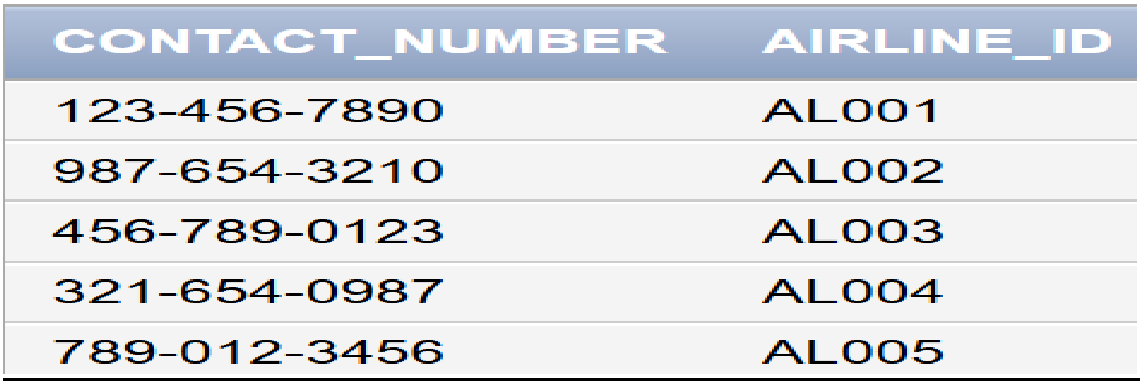
ADDRESS_ID	PASSENGER_ID	HOUSE#	STREET#	CITY	COUNTRY	STATE
A001	P001	123	MainStreet	New York	USA	NY
A002	P002	456	Oak Avenue	Los Angeles	USA	CA
A003	P003	789	Elm Street	Chicago	USA	IL
A004	P004	101	Maple Lane	Houston	USA	TX
A005	P005	202	Cedar Road	Miami	USA	FL
A006	P006	303	Pin Street	Seattle	USA	WA
A007	P007	404	Bir Avenue	Denver	USA	CO
A008	P008	505	Wilow Lane	Boston	USA	MA
A009	P009	606	Spuce Road	San Francisco	USA	CA
A010	P010	707	Cer Street	Dallas	USA	TX

4. Air_Line



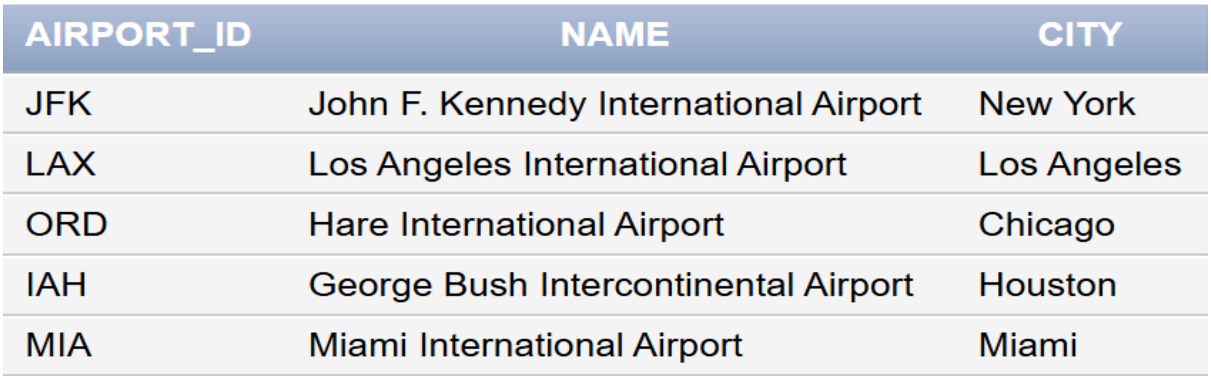
AIRLINE_ID	NAME	COUNTRY	HEADQUARTERS	WEBSITE
AL001	Delta Air Lines	USA	Atlanta, Georgia	https://www.delta.com
AL002	American Airlines	USA	Fort Worth, Texas	https://www.aa.com
AL003	United Airlines	USA	Chicago, Illinois	https://www.united.com
AL004	Lufthansa	Germany	Cologne, Germany	https://www.lufthansa.com
AL005	Emirates	UAE	Dubai, UAE	https://www.emirates.com

5. Contact



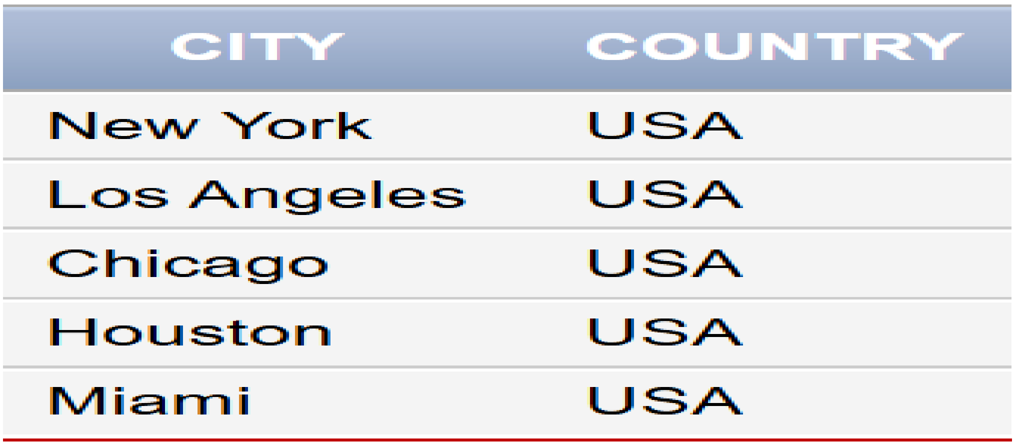
CONTACT_NUMBER	AIRLINE_ID
123-456-7890	AL001
987-654-3210	AL002
456-789-0123	AL003
321-654-0987	AL004
789-012-3456	AL005

6. Airport



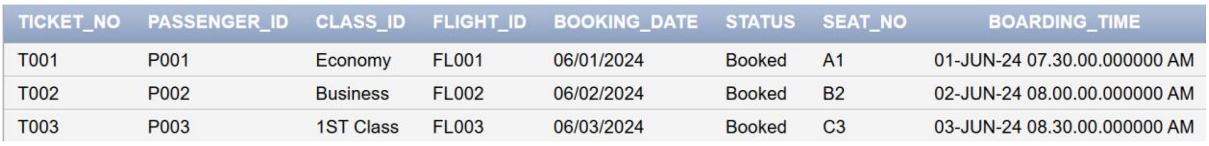
AIRPORT_ID	NAME	CITY
JFK	John F. Kennedy International Airport	New York
LAX	Los Angeles International Airport	Los Angeles
ORD	Hare International Airport	Chicago
IAH	George Bush Intercontinental Airport	Houston
MIA	Miami International Airport	Miami

7. City_Country



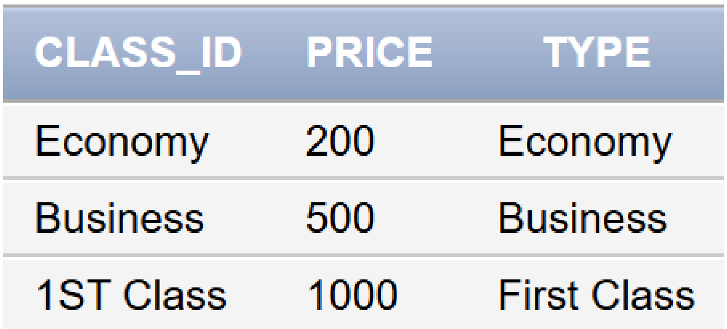
CITY	COUNTRY
New York	USA
Los Angeles	USA
Chicago	USA
Houston	USA
Miami	USA

8. Ticket



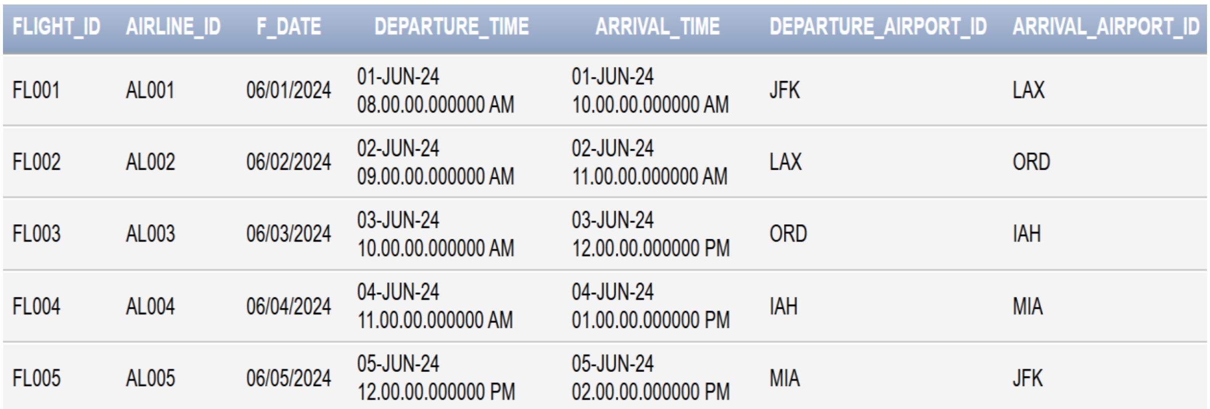
TICKET_NO	PASSENGER_ID	CLASS_ID	FLIGHT_ID	BOOKING_DATE	STATUS	SEAT_NO	BOARDING_TIME
T001	P001	Economy	FL001	06/01/2024	Booked	A1	01-JUN-24 07.30.00.000000 AM
T002	P002	Business	FL002	06/02/2024	Booked	B2	02-JUN-24 08.00.00.000000 AM
T003	P003	1ST Class	FL003	06/03/2024	Booked	C3	03-JUN-24 08.30.00.000000 AM

9. Ticket_Fare



CLASS_ID	PRICE	TYPE
Economy	200	Economy
Business	500	Business
1ST Class	1000	First Class

10. Flight table



FLIGHT_ID	AIRLINE_ID	F_DATE	DEPARTURE_TIME	ARRIVAL_TIME	DEPARTURE_AIRPORT_ID	ARRIVAL_AIRPORT_ID
FL001	AL001	06/01/2024	01-JUN-24 08.00.00.000000 AM	01-JUN-24 10.00.00.000000 AM	JFK	LAX
FL002	AL002	06/02/2024	02-JUN-24 09.00.00.000000 AM	02-JUN-24 11.00.00.000000 AM	LAX	ORD
FL003	AL003	06/03/2024	03-JUN-24 10.00.00.000000 AM	03-JUN-24 12.00.00.000000 PM	ORD	IAH
FL004	AL004	06/04/2024	04-JUN-24 11.00.00.000000 AM	04-JUN-24 01.00.00.000000 PM	IAH	MIA
FL005	AL005	06/05/2024	05-JUN-24 12.00.00.000000 PM	05-JUN-24 02.00.00.000000 PM	MIA	JFK

8. Most important VIEWS

Passenger Details View:

```
CREATE VIEW Passenger_Details AS
SELECT p.Passenger_Id, p.Name, p.Age, p.Passport_No, p.Gender, 
pn.Phone_Number,
       a.House#, a.Street#, a.City, a.Country, a.State
FROM Passenger p
LEFT JOIN Phone_Number pn ON p.Passenger_Id = pn.Passenger_Id
LEFT JOIN Addresse a ON p.Passenger_Id = a.Passenger_Id;
```

Flight Information View:

```
CREATE VIEW Flight_Information AS
SELECT f.Flight_ID, f.F_Date, f.Departure_Time, f.Arrival_Time,
       dep.Name AS Departure_Airport, arr.Name AS Arrival_Airport,
       al.Name AS Airline, al.Country AS Airline_Country
FROM Flight f
INNER JOIN Airport dep ON f.Departure_Airport_id = dep.Airport_id
INNER JOIN Airport arr ON f.Arrival_Airport_id = arr.Airport_id
INNER JOIN Air_Line al ON f.Airline_Id = al.Airline_Id;
```

Ticket Status View:

```
CREATE VIEW Ticket_Status AS
SELECT t.Ticket_No, p.Name AS Passenger_Name, f.Flight_ID, f.F_Date,
       t.Status AS Ticket_Status, t.Seat_No
FROM Ticket t
INNER JOIN Passenger p ON t.Passenger_Id = p.Passenger_Id
INNER JOIN Flight f ON t.Flight_Id = f.Flight_ID;
```

Ticket Pricing View:

```
CREATE VIEW Ticket_Pricing AS
SELECT t.Ticket_No, tf.Type AS Class_Type, tf.Price AS Ticket_Price
FROM Ticket t
INNER JOIN Ticket_Fare tf ON t.Class_ID = tf.Class_ID;
```

Airport Details View:

```
CREATE VIEW Airport_Details AS
SELECT a.Airport_id, a.Name AS Airport_Name, a.City, cc.Country
FROM Airport a
INNER JOIN City_Country cc ON a.City = cc.City;
```

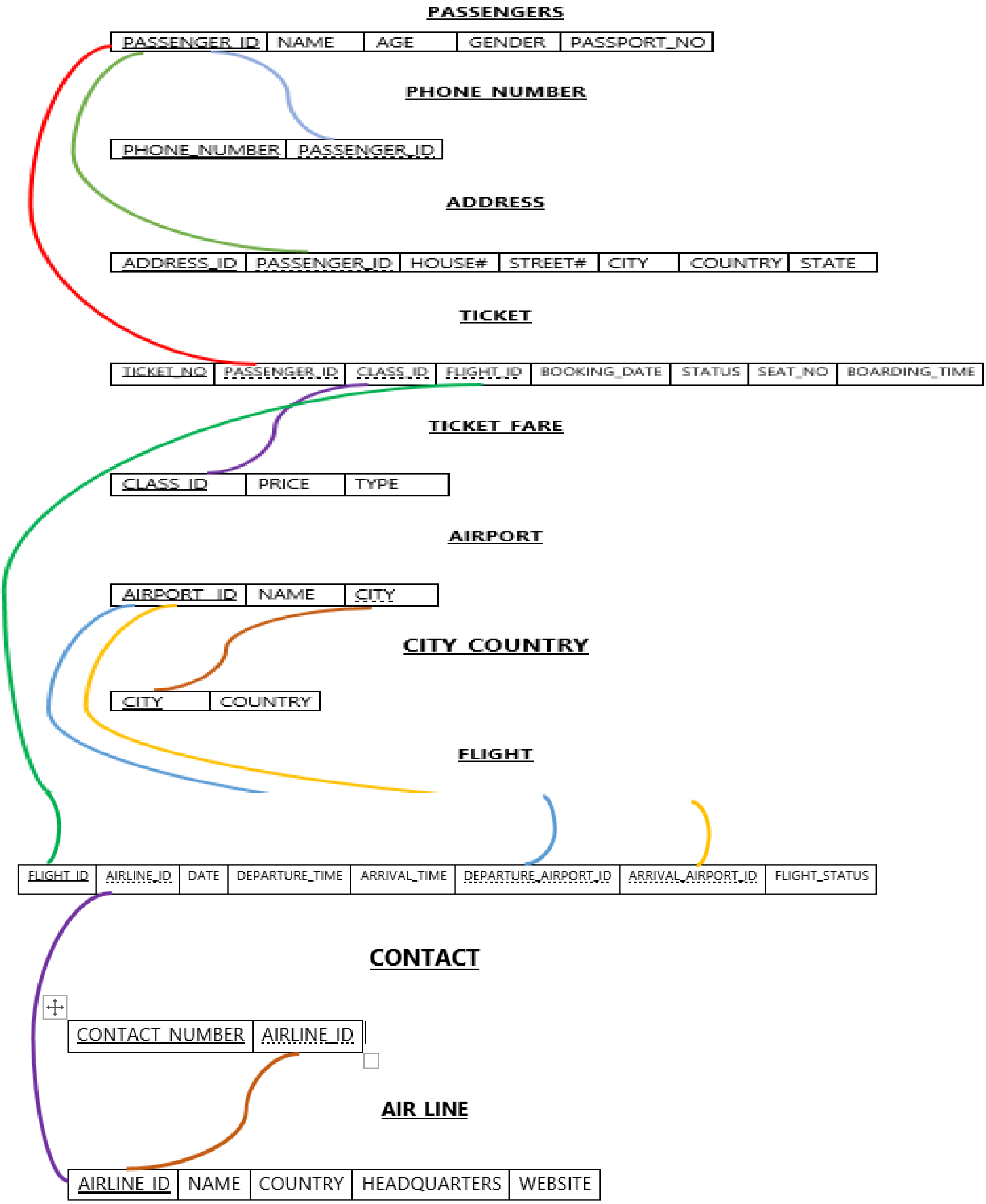
Airline Contact View:

```
CREATE VIEW Airline_Contact AS
SELECT al.Name AS Airline_Name, al.Headquarters AS Headquarters_Location,
       c.Contact_Number AS Contact_Number
FROM Air_Line al
LEFT JOIN Contact c ON al.Airline_Id = c.Airline_Id;
```

8. Relational data model

Our final entities with attributes:

- **Passengers** (Passenger_Id, Name, Age, Gender, Passport_No)
- **Phone_Number** (Phone_Number, Passenger_Id)
- **Address** (Address_Id, Passenger_Id , House#, Street#, City,Country, State)
- **Ticket** (Ticket_No, Passenger_Id, Class_ID, Flight_id, Booking_date, Status, Seat_No, Boarding_Time)
- **Ticket_fare** (Class_ID, Price, Type)
- **Airport** (Airport_ID, Name, City)
- **City_Country** (City, Country)
- **Flight** (Flight_ID, Airline_Id , Date, Departure_Time, Arrival_Time, Departure_Airport_Id, Arrival_Airport_Id, Flight_Status)
- **Air_Line** (Airline_Id, Name, Country, Headquarters, Website)
- **Contact** (Contact_Number, Airline_Id)



PASSENGERS

PASSENGER_ID	NAME	AGE	GENDER	PASSPORT_NO

PHONE NUMBER

PHONE_NUMBER	PASSENGER_ID

ADDRESS

ADDRESS_ID	PASSENGER_ID	HOUSE#	STREET#	CITY	COUNTRY	STATE

TICKET

TICKET_NO	PASSENGER_ID	CLASS_ID	FLIGHT_ID	BOOKING_DATE	STATUS	SEAT_NO	BOARDING_TIME

TICKET FARE

CLASS_ID	PRICE	TYPE

AIRPORT

AIRPORT_ID	NAME	CITY

CITY COUNTRY

CITY	COUNTRY

FLIGHT

FLIGHT_ID	AIRLINE_ID	DATE	DEPARTURE_TIME	ARRIVAL_TIME	DEPARTURE_AIRPORT_ID	ARRIVAL_AIRPORT_ID	FLIGHT_STATUS

CONTACT

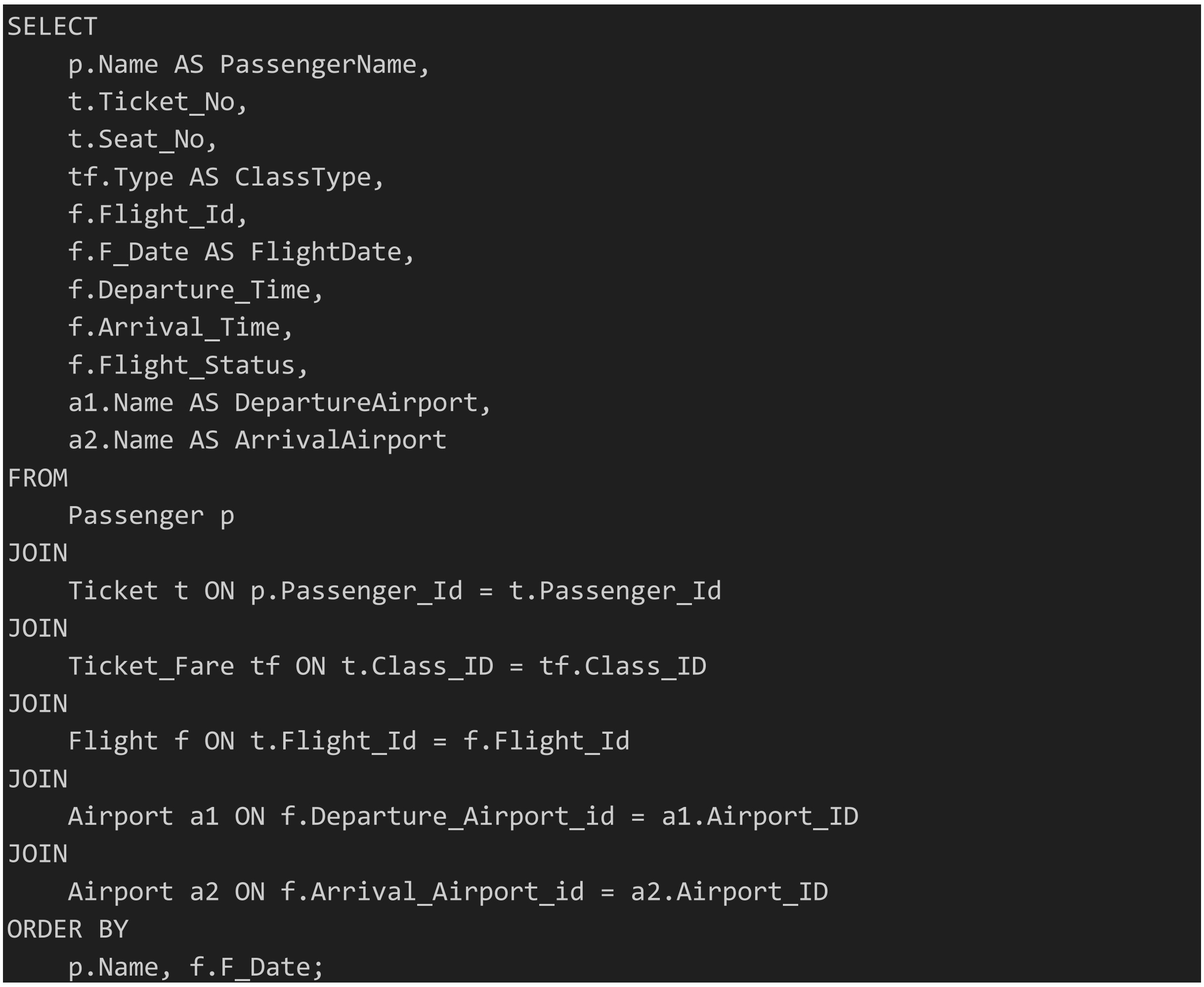
CONTACT_NUMBER	AIRLINE_ID

AIR LINE

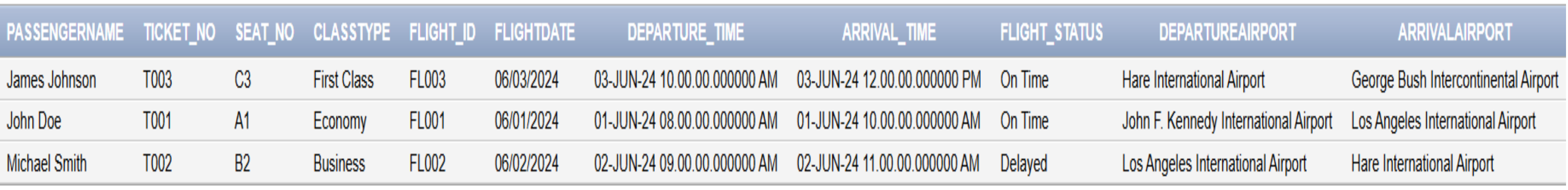
AIRLINE_ID	NAME	COUNTRY	HEADQUARTERS	WEBSITE

9. SELECT statement for reports Report:

Passenger Flight Details

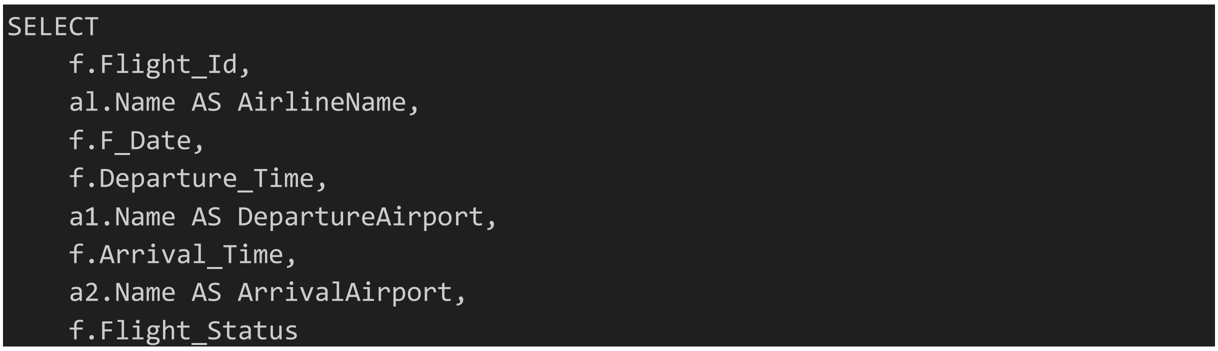


```
SELECT
    p.Name AS PassengerName,
    t.Ticket_No,
    t.Seat_No,
    tf.Type AS ClassType,
    f.Flight_Id,
    f.F_Date AS FlightDate,
    f.Departure_Time,
    f.Arrival_Time,
    f.Flight_Status,
    a1.Name AS DepartureAirport,
    a2.Name AS ArrivalAirport
FROM
    Passenger p
JOIN
    Ticket t ON p.Passenger_Id = t.Passenger_Id
JOIN
    Ticket_Fare tf ON t.Class_ID = tf.Class_ID
JOIN
    Flight f ON t.Flight_Id = f.Flight_Id
JOIN
    Airport a1 ON f.Departure_Airport_id = a1.Airport_ID
JOIN
    Airport a2 ON f.Arrival_Airport_id = a2.Airport_ID
ORDER BY
    p.Name, f.F_Date;
```

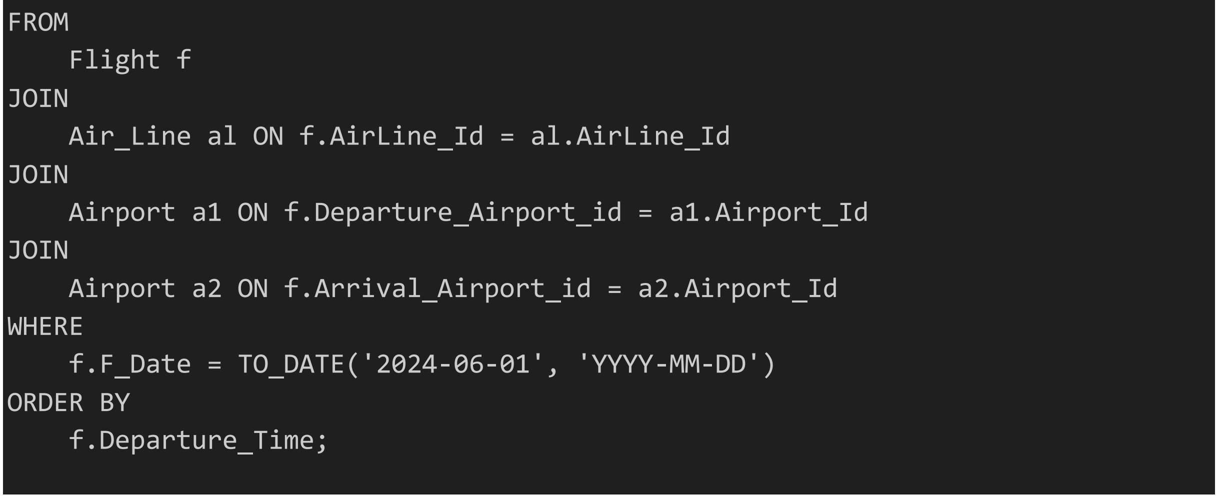


PASSENGERNAME	TICKET_NO	SEAT_NO	CLASSTYPE	FLIGHT_ID	FLIGHTDATE	DEPARTURE_TIME	ARRIVAL_TIME	FLIGHT_STATUS	DEPARTUREAIRPORT	ARRIVALAIRPORT
James Johnson	T003	C3	First Class	FL003	06/03/2024	03-JUN-24 10.00.00.000000 AM	03-JUN-24 12.00.00.000000 PM	On Time	Hare International Airport	George Bush Intercontinental Airport
John Doe	T001	A1	Economy	FL001	06/01/2024	01-JUN-24 08.00.00.000000 AM	01-JUN-24 10.00.00.000000 AM	On Time	John F. Kennedy International Airport	Los Angeles International Airport
Michael Smith	T002	B2	Business	FL002	06/02/2024	02-JUN-24 09.00.00.000000 AM	02-JUN-24 11.00.00.000000 AM	Delayed	Los Angeles International Airport	Hare International Airport

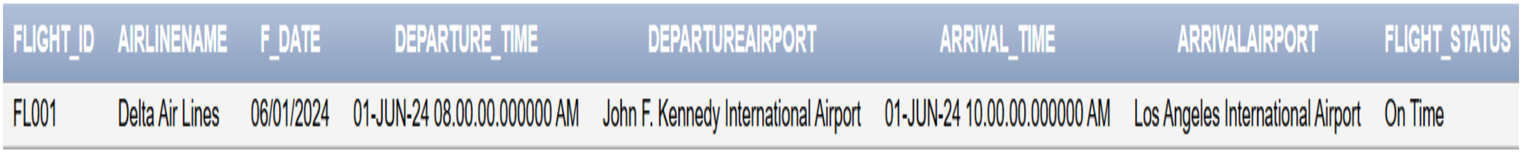
Report: Flight Schedule by Date



```
SELECT
    f.Flight_Id,
    al.Name AS AirlineName,
    f.F_Date,
    f.Departure_Time,
    a1.Name AS DepartureAirport,
    f.Arrival_Time,
    a2.Name AS ArrivalAirport,
    f.Flight_Status
```

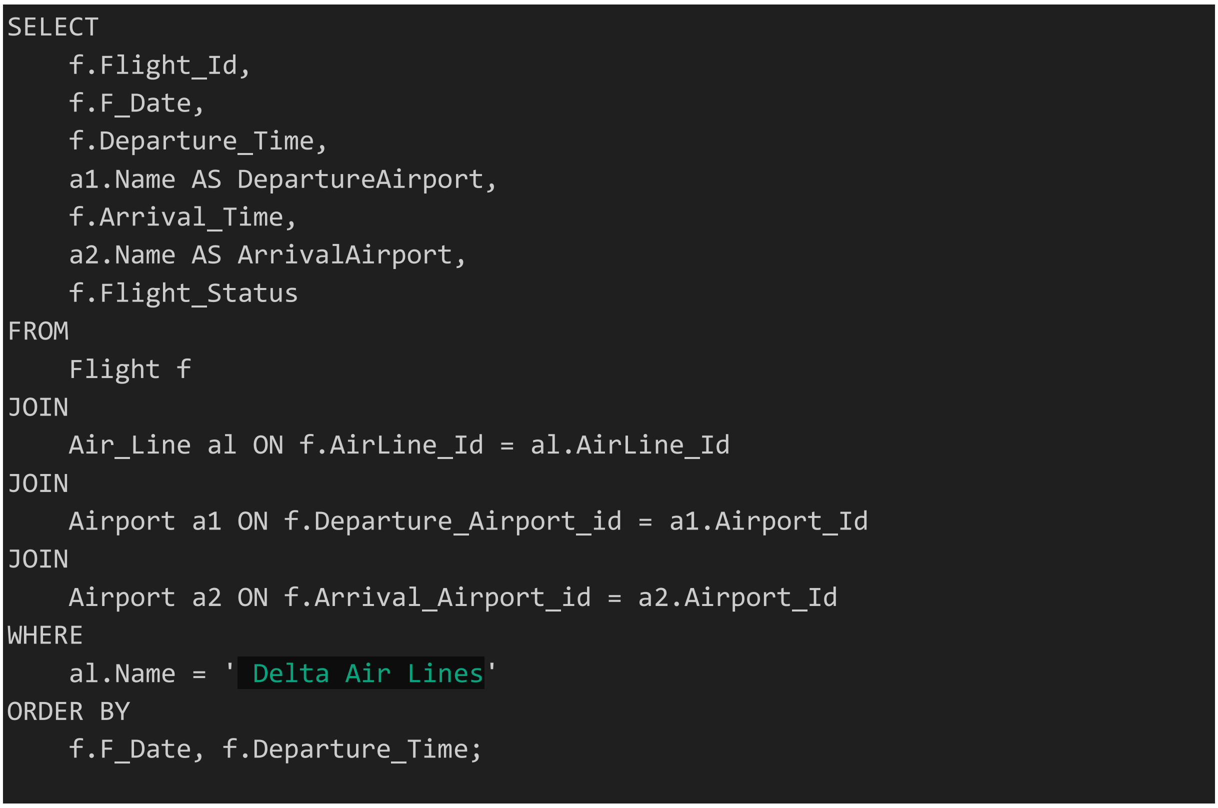


```
FROM
    Flight f
JOIN
    Air_Line al ON f.AirLine_Id = al.AirLine_Id
JOIN
    Airport a1 ON f.Departure_Airport_id = a1.Airport_Id
JOIN
    Airport a2 ON f.Arrival_Airport_id = a2.Airport_Id
WHERE
    f.F_Date = TO_DATE('2024-06-01', 'YYYY-MM-DD')
ORDER BY
    f.Departure_Time;
```

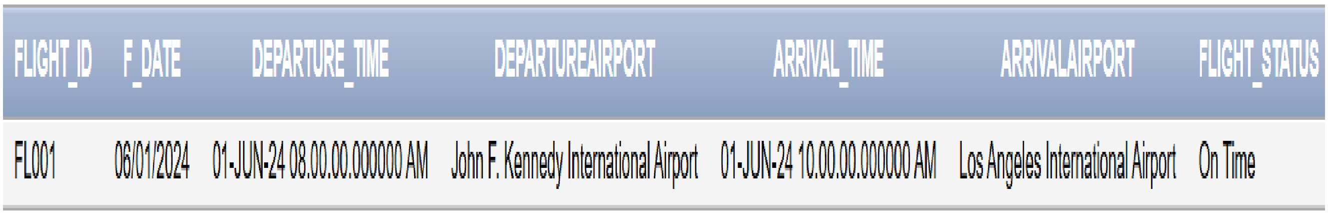


FLIGHT_ID	AIRLINENAME	F_DATE	DEPARTURE_TIME	DEPARTUREAIRPORT	ARRIVAL_TIME	ARRIVALAIRPORT	FLIGHT_STATUS
FL001	Delta Air Lines	06/01/2024	01-JUN-24 08.00.00.000000 AM	John F. Kennedy International Airport	01-JUN-24 10.00.00.000000 AM	Los Angeles International Airport	On Time

Report: Flights by Airline

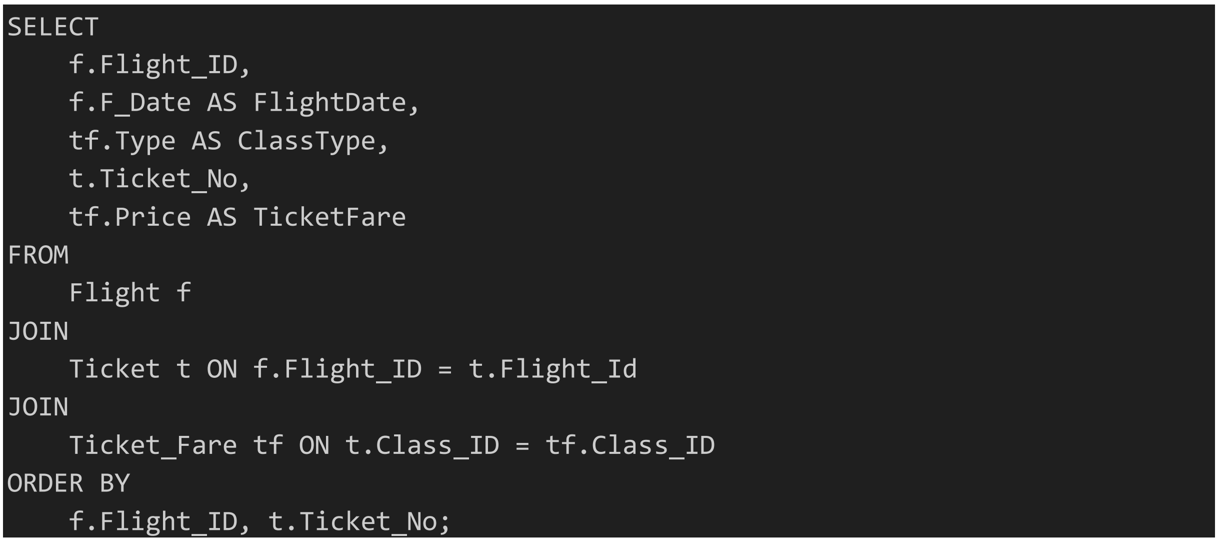


```
SELECT
    f.Flight_Id,
    f.F_Date,
    f.Departure_Time,
    a1.Name AS DepartureAirport,
    f.Arrival_Time,
    a2.Name AS ArrivalAirport,
    f.Flight_Status
FROM
    Flight f
JOIN
    Air_Line al ON f.AirLine_Id = al.AirLine_Id
JOIN
    Airport a1 ON f.Departure_Airport_id = a1.Airport_Id
JOIN
    Airport a2 ON f.Arrival_Airport_id = a2.Airport_Id
WHERE
    al.Name = ' Delta Air Lines'
ORDER BY
    f.F_Date, f.Departure_Time;
```

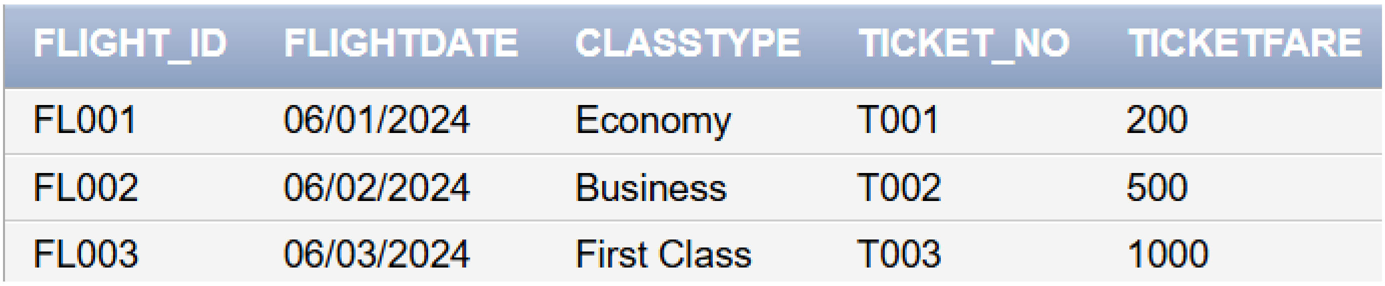


FLIGHT_ID	F_DATE	DEPARTURE_TIME	DEPARTUREAIRPORT	ARRIVAL_TIME	ARRIVALAIRPORT	FLIGHT_STATUS
FL001	06/01/2024	01-JUN-24 08.00.00.000000 AM	John F. Kennedy International Airport	01-JUN-24 10.00.00.000000 AM	Los Angeles International Airport	On Time

Report: Fare Details by Flight

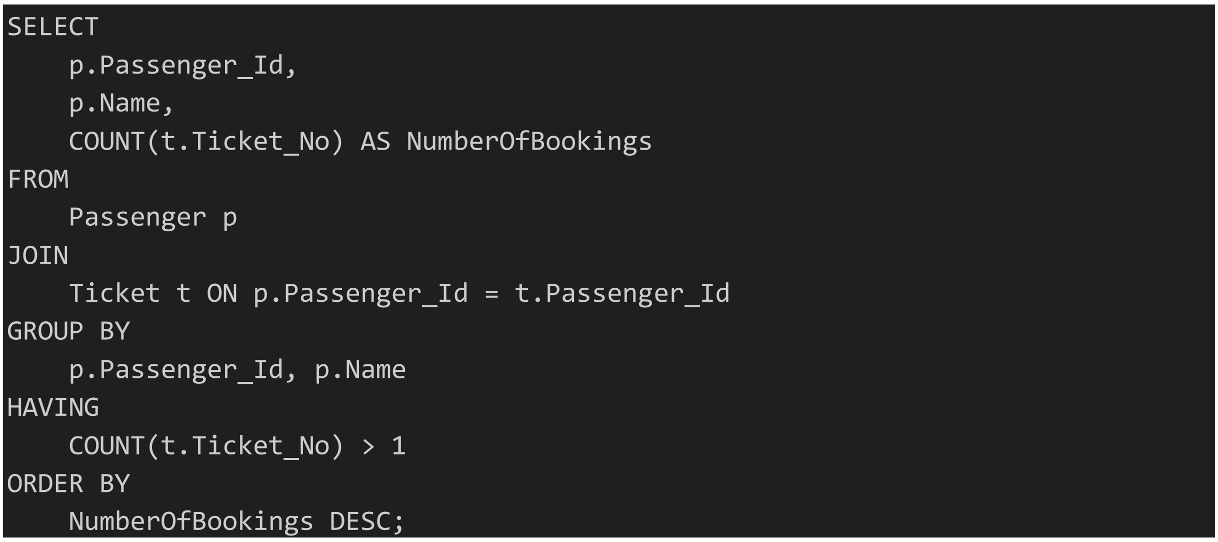


```
SELECT
    f.Flight_ID,
    f.F_Date AS FlightDate,
    tf.Type AS ClassType,
    t.Ticket_No,
    tf.Price AS TicketFare
FROM
    Flight f
JOIN
    Ticket t ON f.Flight_ID = t.Flight_Id
JOIN
    Ticket_Fare tf ON t.Class_ID = tf.Class_ID
ORDER BY
    f.Flight_ID, t.Ticket_No;
```

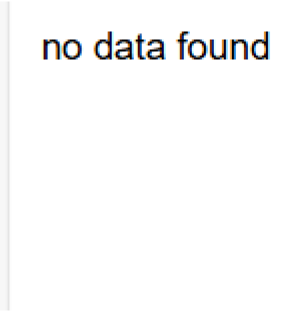


FLIGHT_ID	FLIGHTDATE	CLASSTYPE	TICKET_NO	TICKETFARE
FL001	06/01/2024	Economy	T001	200
FL002	06/02/2024	Business	T002	500
FL003	06/03/2024	First Class	T003	1000

Report: Passengers with Multiple Bookings

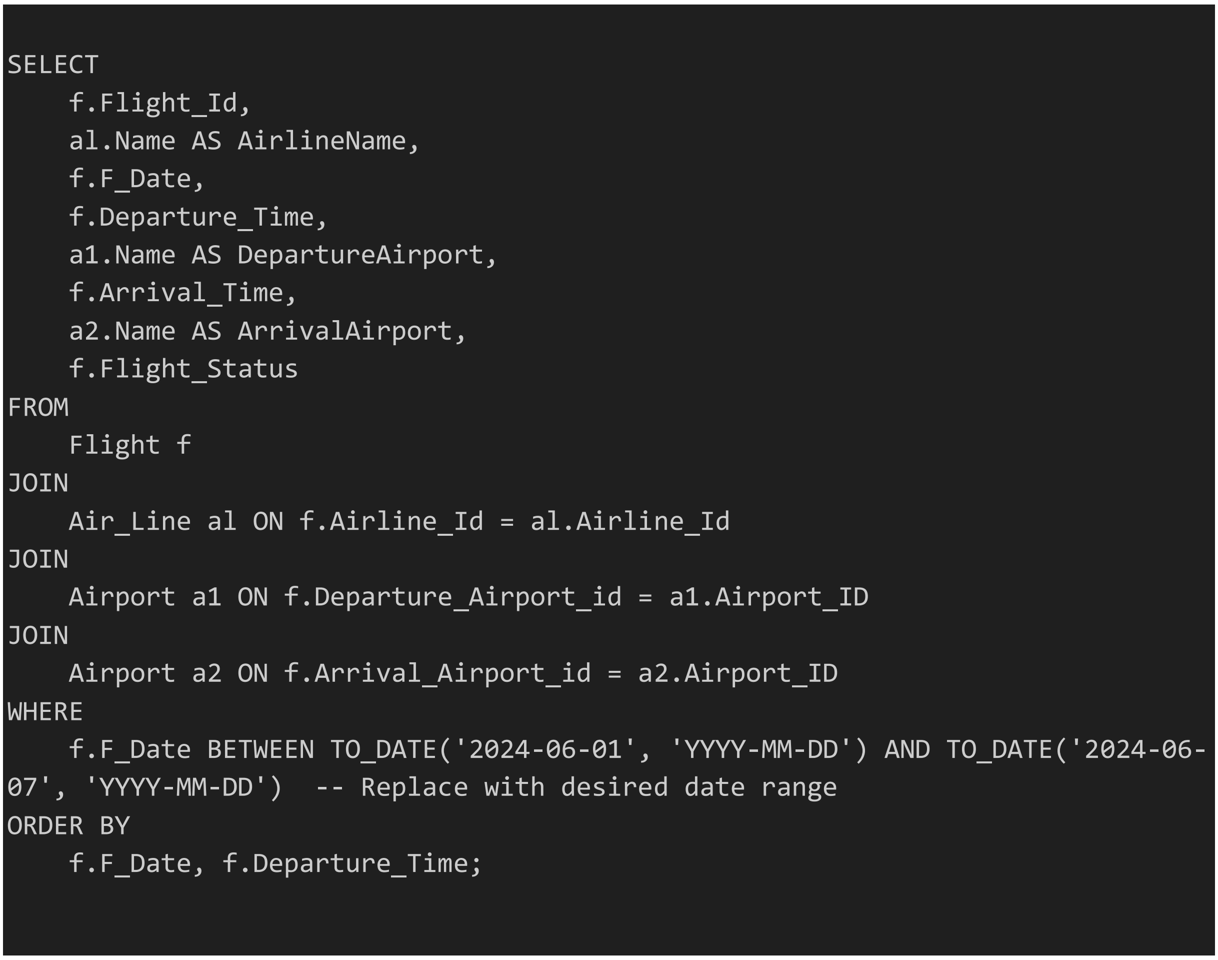


```
SELECT
    p.Passenger_Id,
    p.Name,
    COUNT(t.Ticket_No) AS NumberOfBookings
FROM
    Passenger p
JOIN
    Ticket t ON p.Passenger_Id = t.Passenger_Id
GROUP BY
    p.Passenger_Id, p.Name
HAVING
    COUNT(t.Ticket_No) > 1
ORDER BY
    NumberOfBookings DESC;
```

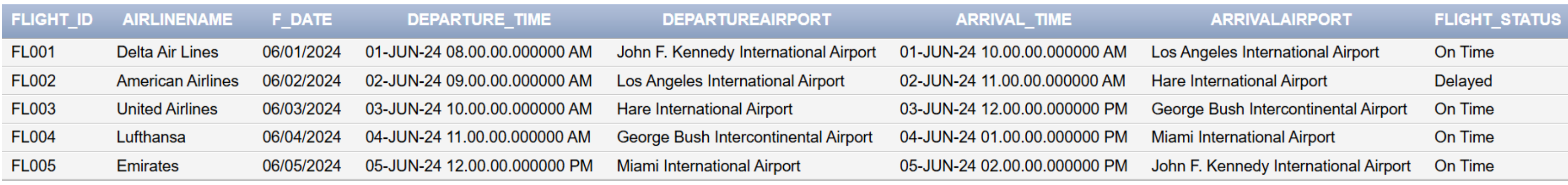


```
no data found
```

Report: Flights by Date Range

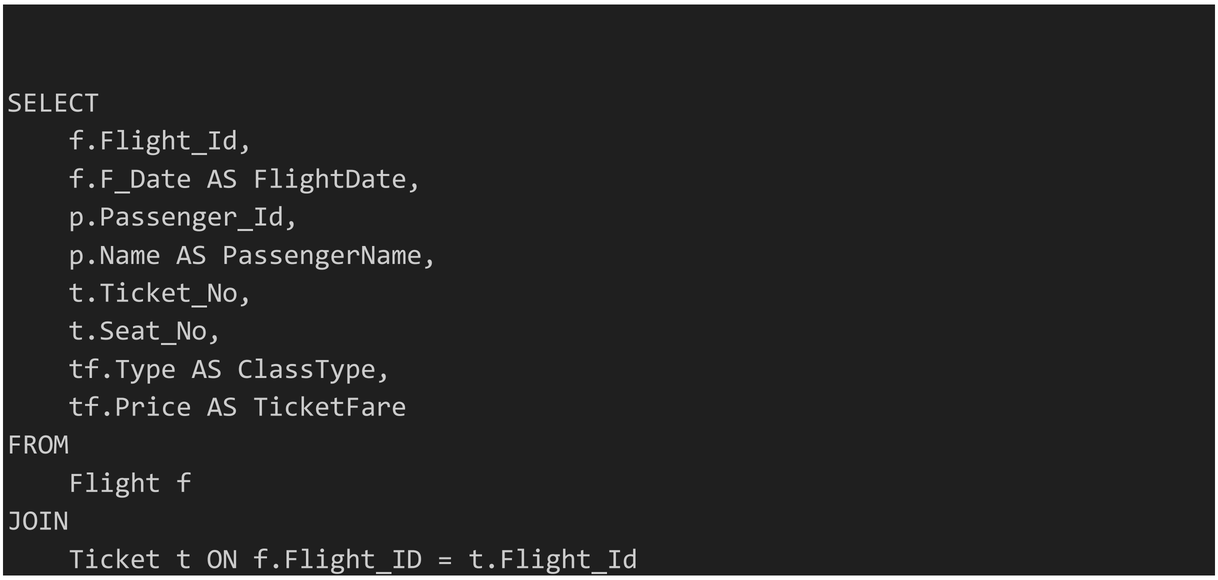


```
SELECT
    f.Flight_Id,
    al.Name AS AirlineName,
    f.F_Date,
    f.Departure_Time,
    a1.Name AS DepartureAirport,
    f.Arrival_Time,
    a2.Name AS ArrivalAirport,
    f.Flight_Status
FROM
    Flight f
JOIN
    Air_Line al ON f.Airline_Id = al.Airline_Id
JOIN
    Airport a1 ON f.Departure_Airport_id = a1.Airport_ID
JOIN
    Airport a2 ON f.Arrival_Airport_id = a2.Airport_ID
WHERE
    f.F_Date BETWEEN TO_DATE('2024-06-01', 'YYYY-MM-DD') AND TO_DATE('2024-06-
07', 'YYYY-MM-DD')  -- Replace with desired date range
ORDER BY
    f.F_Date, f.Departure_Time;
```

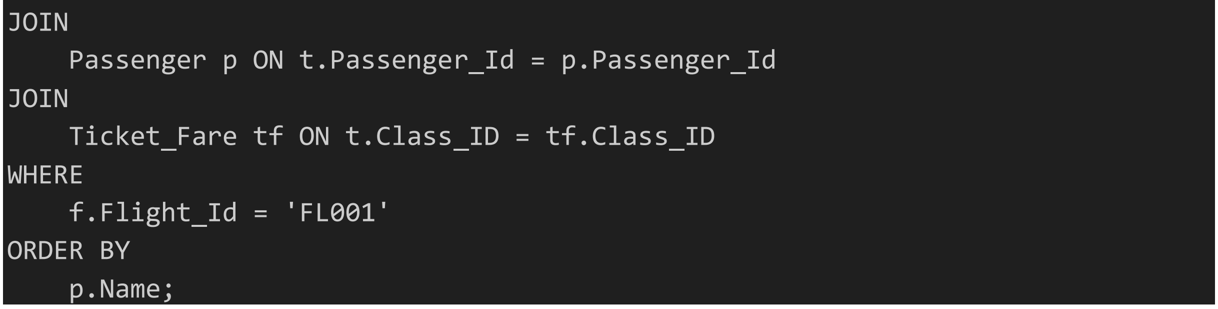


FLIGHT_ID	AIRLINENAME	F_DATE	DEPARTURE_TIME	DEPARTUREAIRPORT	ARRIVAL_TIME	ARRIVALAIRPORT	FLIGHT_STATUS
FL001	Delta Air Lines	06/01/2024	01-JUN-24 08.00.00.000000 AM	John F. Kennedy International Airport	01-JUN-24 10.00.00.000000 AM	Los Angeles International Airport	On Time
FL002	American Airlines	06/02/2024	02-JUN-24 09.00.00.000000 AM	Los Angeles International Airport	02-JUN-24 11.00.00.000000 AM	Hare International Airport	Delayed
FL003	United Airlines	06/03/2024	03-JUN-24 10.00.00.000000 AM	Hare International Airport	03-JUN-24 12.00.00.000000 PM	George Bush Intercontinental Airport	On Time
FL004	Lufthansa	06/04/2024	04-JUN-24 11.00.00.000000 AM	George Bush Intercontinental Airport	04-JUN-24 01.00.00.000000 PM	Miami International Airport	On Time
FL005	Emirates	06/05/2024	05-JUN-24 12.00.00.000000 PM	Miami International Airport	05-JUN-24 02.00.00.000000 PM	John F. Kennedy International Airport	On Time

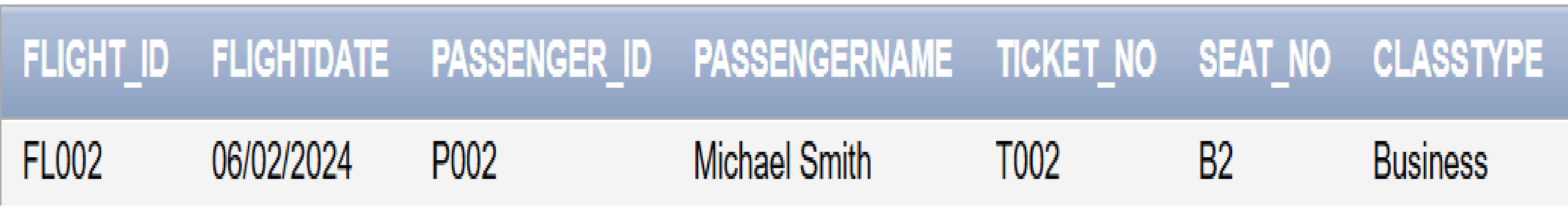
Report: Passengers in particular Flight



```
SELECT
    f.Flight_Id,
    f.F_Date AS FlightDate,
    p.Passenger_Id,
    p.Name AS PassengerName,
    t.Ticket_No,
    t.Seat_No,
    tf.Type AS ClassType,
    tf.Price AS TicketFare
FROM
    Flight f
JOIN
    Ticket t ON f.Flight_ID = t.Flight_Id
```

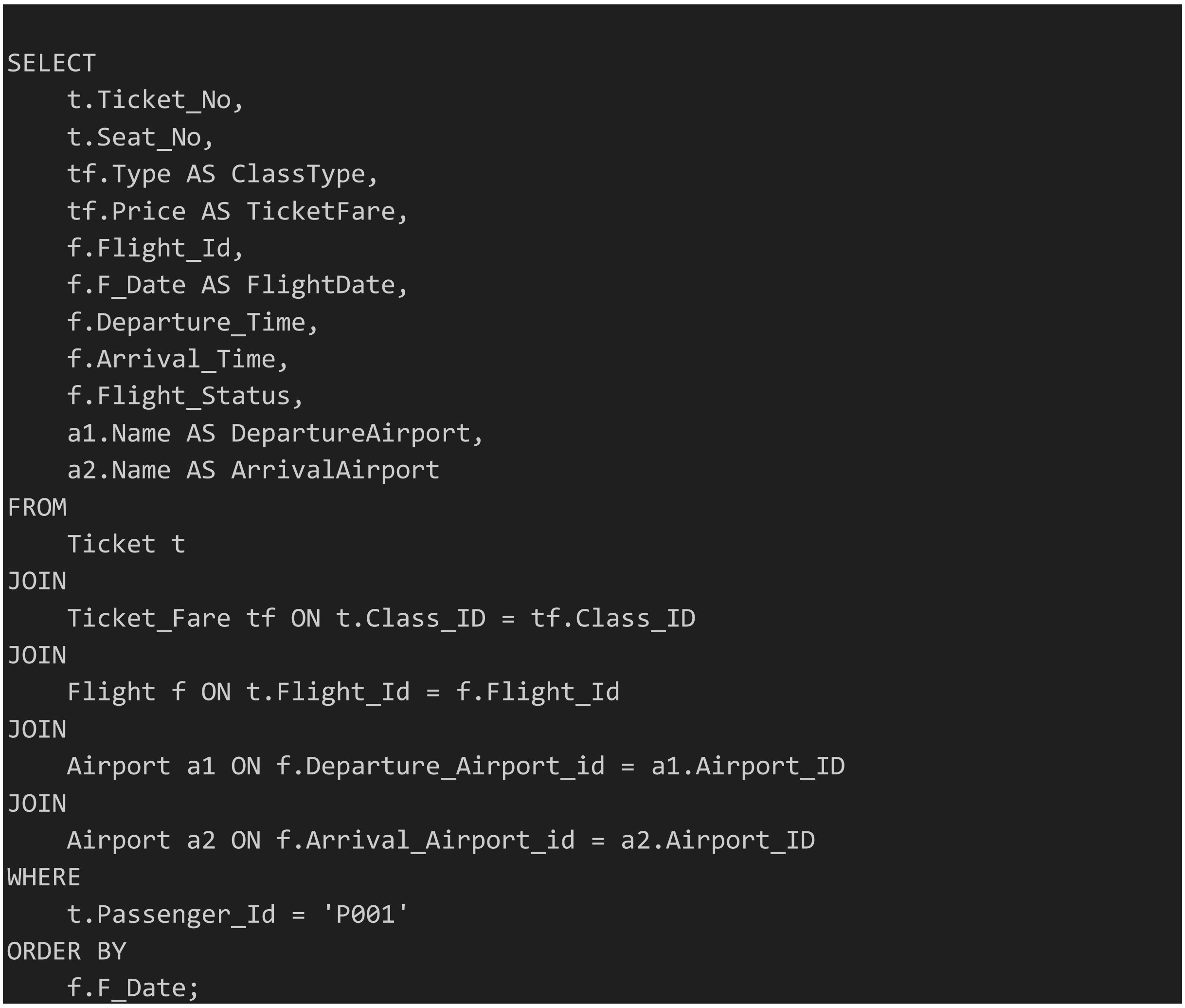


```
JOIN
    Passenger p ON t.Passenger_Id = p.Passenger_Id
JOIN
    Ticket_Fare tf ON t.Class_ID = tf.Class_ID
WHERE
    f.Flight_Id = 'FL001'
ORDER BY
    p.Name;
```

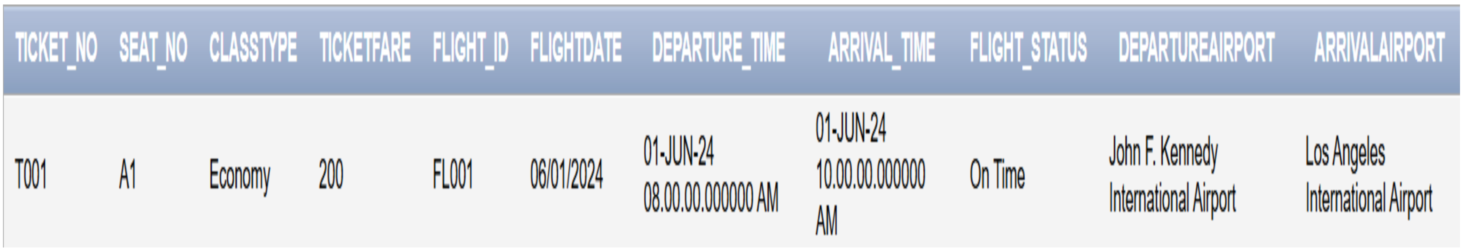


FLIGHT_ID	FLIGHTDATE	PASSENGER_ID	PASSENGERNAME	TICKET_NO	SEAT_NO	CLASSTYPE
FL002	06/02/2024	P002	Michael Smith	T002	B2	Business

Report: Ticket_Details of Passenger

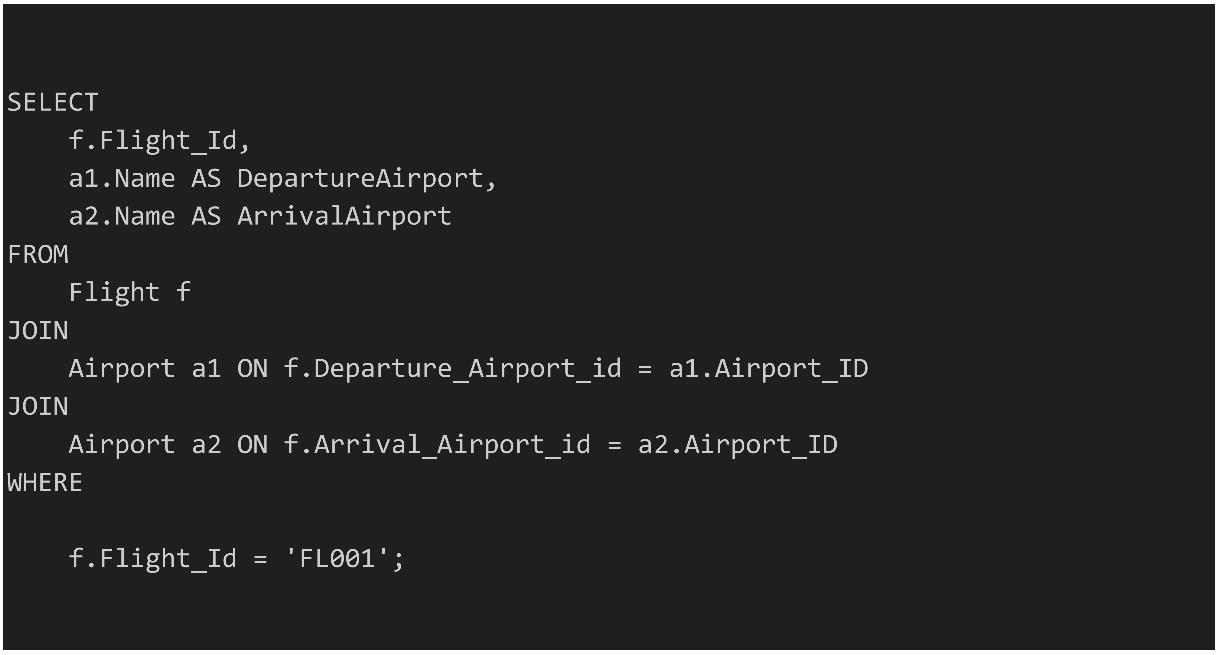


```
SELECT
    t.Ticket_No,
    t.Seat_No,
    tf.Type AS ClassType,
    tf.Price AS TicketFare,
    f.Flight_Id,
    f.F_Date AS FlightDate,
    f.Departure_Time,
    f.Arrival_Time,
    f.Flight_Status,
    a1.Name AS DepartureAirport,
    a2.Name AS ArrivalAirport
FROM
    Ticket t
JOIN
    Ticket_Fare tf ON t.Class_ID = tf.Class_ID
JOIN
    Flight f ON t.Flight_Id = f.Flight_Id
JOIN
    Airport a1 ON f.Departure_Airport_id = a1.Airport_ID
JOIN
    Airport a2 ON f.Arrival_Airport_id = a2.Airport_ID
WHERE
    t.Passenger_Id = 'P001'
ORDER BY
    f.F_Date;
```



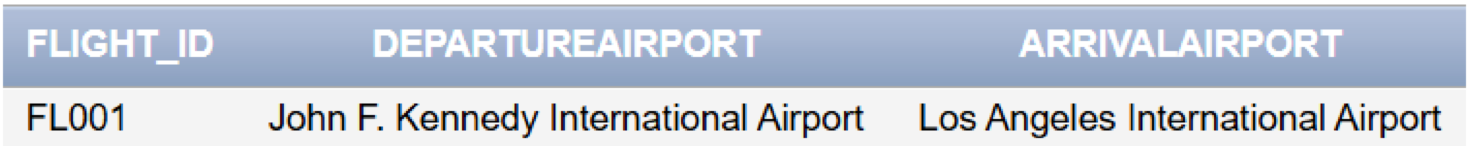
TICKET_NO	SEAT_NO	CLASSTYPE	TICKETFARE	FLIGHT_ID	FLIGHTDATE	DEPARTURE_TIME	ARRIVAL_TIME	FLIGHT_STATUS	DEPARTUREAIRPORT	ARRIVALAIRPORT
T001	A1	Economy	200	FL001	06/01/2024	01-JUN-24 08.00.00.000000 AM	01-JUN-24 10.00.00.000000 AM	On Time	John F. Kennedy International Airport	Los Angeles International Airport

Report: Flight departure and arrival Airport



```
SELECT
    f.Flight_Id,
    a1.Name AS DepartureAirport,
    a2.Name AS ArrivalAirport
FROM
    Flight f
JOIN
    Airport a1 ON f.Departure_Airport_id = a1.Airport_ID
JOIN
    Airport a2 ON f.Arrival_Airport_id = a2.Airport_ID
WHERE

    f.Flight_Id = 'FL001';
```

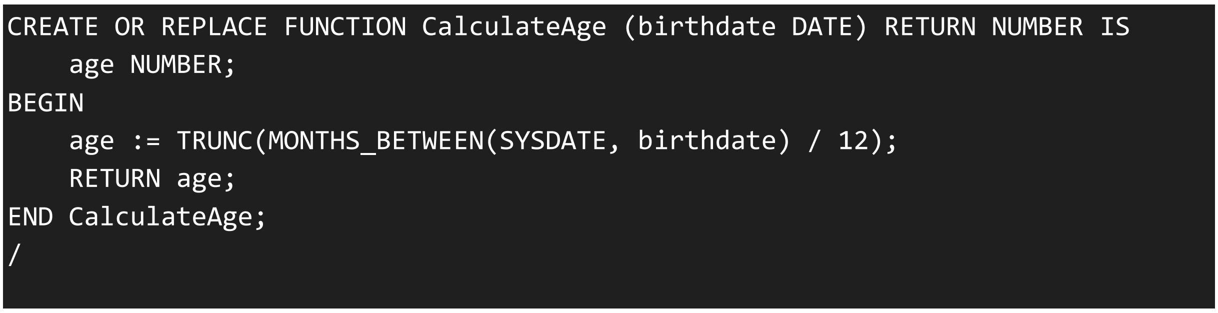


FLIGHT_ID	DEPARTUREAIRPORT	ARRIVALAIRPORT
FL001	John F. Kennedy International Airport	Los Angeles International Airport

10.Demonstration of Functions, Procedures, and Triggers.

Functions

Function to Calculate Age:



```
CREATE OR REPLACE FUNCTION CalculateAge (birthdate DATE) RETURN NUMBER IS
    age NUMBER;
BEGIN
    age := TRUNC(MONTHS_BETWEEN(SYSDATE, birthdate) / 12);
    RETURN age;
END CalculateAge;
/
```

Function to Retrieve Total Number of Passengers for a Flight:

```
CREATE OR REPLACE FUNCTION GetTotalPassengers (FlightID NUMBER) RETURN NUMBER
IS
    totalPassengers NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO totalPassengers
    FROM Ticket
    WHERE Flight_id = FlightID;
    RETURN totalPassengers;
END GetTotalPassengers;
/
```

Function to Retrieve the Total Revenue for a Specific Flight:

```
CREATE OR REPLACE FUNCTION GetFlightRevenue (FlightID NUMBER) RETURN NUMBER IS
    totalRevenue NUMBER;
BEGIN
    SELECT SUM(tf.Price)
    INTO totalRevenue
    FROM Ticket t
    JOIN Ticket_fare tf ON t.Class_ID = tf.Class_ID
    WHERE t.Flight_id = FlightID;

    RETURN totalRevenue;
END GetFlightRevenue;
/
```

Stored Procedures

Procedure to Book a Ticket:

```
CREATE OR REPLACE PROCEDURE BookTicket (
    PassengerId NUMBER,
    ClassId NUMBER,
    FlightId NUMBER,
    BookingDate DATE,
    Status VARCHAR2,
    SeatNo VARCHAR2,
    BoardingTime DATE
) IS
BEGIN
    INSERT INTO Ticket (Passenger_Id, Class_ID, Flight_id, Booking_date,
Status, Seat_No, Boarding_Time)
    VALUES (PassengerId, ClassId, FlightId, BookingDate, Status, SeatNo,
BoardingTime);
```

```
END BookTicket;
/
```

Procedure to Update Flight Status:

```

CREATE OR REPLACE PROCEDURE UpdateFlightStatus (FlightID NUMBER) IS
    CurrentTime DATE := SYSDATE;
BEGIN
    UPDATE Flight
    SET Flight_Status = 'Arrived'
    WHERE Flight_ID = FlightID
    AND Arrival_Time < CurrentTime
    AND Flight_Status <> 'Arrived';
END UpdateFlightStatus;
/
```

Procedure to Cancel a Ticket:

```
CREATE OR REPLACE PROCEDURE CancelTicket (
    TicketNo NUMBER
) IS
BEGIN
    UPDATE Ticket
    SET Status = 'Cancelled'
    WHERE Ticket_No = TicketNo;

END CancelTicket;
```

Database Triggers

Before Delete Trigger

```
CREATE OR REPLACE TRIGGER delflights
BEFORE DELETE ON Flight
FOR EACH ROW
BEGIN
    INSERT INTO Flight_backup (Flight_id, F_Date)
    VALUES (:OLD.Flight_id, :OLD.F_Date);
END;
```

BEFORE INSERT Trigger

```
CREATE OR REPLACE TRIGGER before_insert_flight
BEFORE INSERT ON Flight
FOR EACH ROW
BEGIN
    -- Ensure that flight dates are in the future
    IF :NEW.F_Date < SYSDATE THEN
        RAISE_APPLICATION_ERROR(-20001, 'Flight date must be in the future.');
    END IF;
END;
```

AFTER UPDATE Trigger

```
CREATE OR REPLACE TRIGGER after_update_flight
AFTER UPDATE ON Flight
FOR EACH ROW
BEGIN
    INSERT INTO Flight_Audit (Flight_id, Action, Action_Date)
    VALUES (:NEW.Flight_id, 'UPDATE', SYSDATE);
END;
/
```

PASSENGER AUDIT TRIGGER

```
CREATE OR REPLACE TRIGGER trg_Passenger_Audit
AFTER INSERT OR UPDATE OR DELETE ON Passenger
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        INSERT INTO Passenger_Audit (Passenger_Id, Action, Action_Date)
        VALUES (:NEW.Passenger_Id, 'INSERT', SYSDATE);
    ELSIF UPDATING THEN
        INSERT INTO Passenger_Audit (Passenger_Id, Action, Action_Date)
        VALUES (:OLD.Passenger_Id, 'UPDATE', SYSDATE);
    ELSIF DELETING THEN
        INSERT INTO Passenger_Audit (Passenger_Id, Action, Action_Date)
        VALUES (:OLD.Passenger_Id, 'DELETE', SYSDATE);
    END IF;
END;
```

MAX TICKET LIMIT TRIGGER

```
CREATE OR REPLACE TRIGGER trg_Passenger_Max_Tickets
BEFORE INSERT ON Ticket
FOR EACH ROW
DECLARE
    v_ticket_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_ticket_count
    FROM Ticket
    WHERE Passenger_Id = :NEW.Passenger_Id;

    IF v_ticket_count >= 5 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Maximum number of tickets reached for this passenger.');
    END IF;
END;
/
```