



# DATABASE MANAGEMENT FINAL PROJESİ

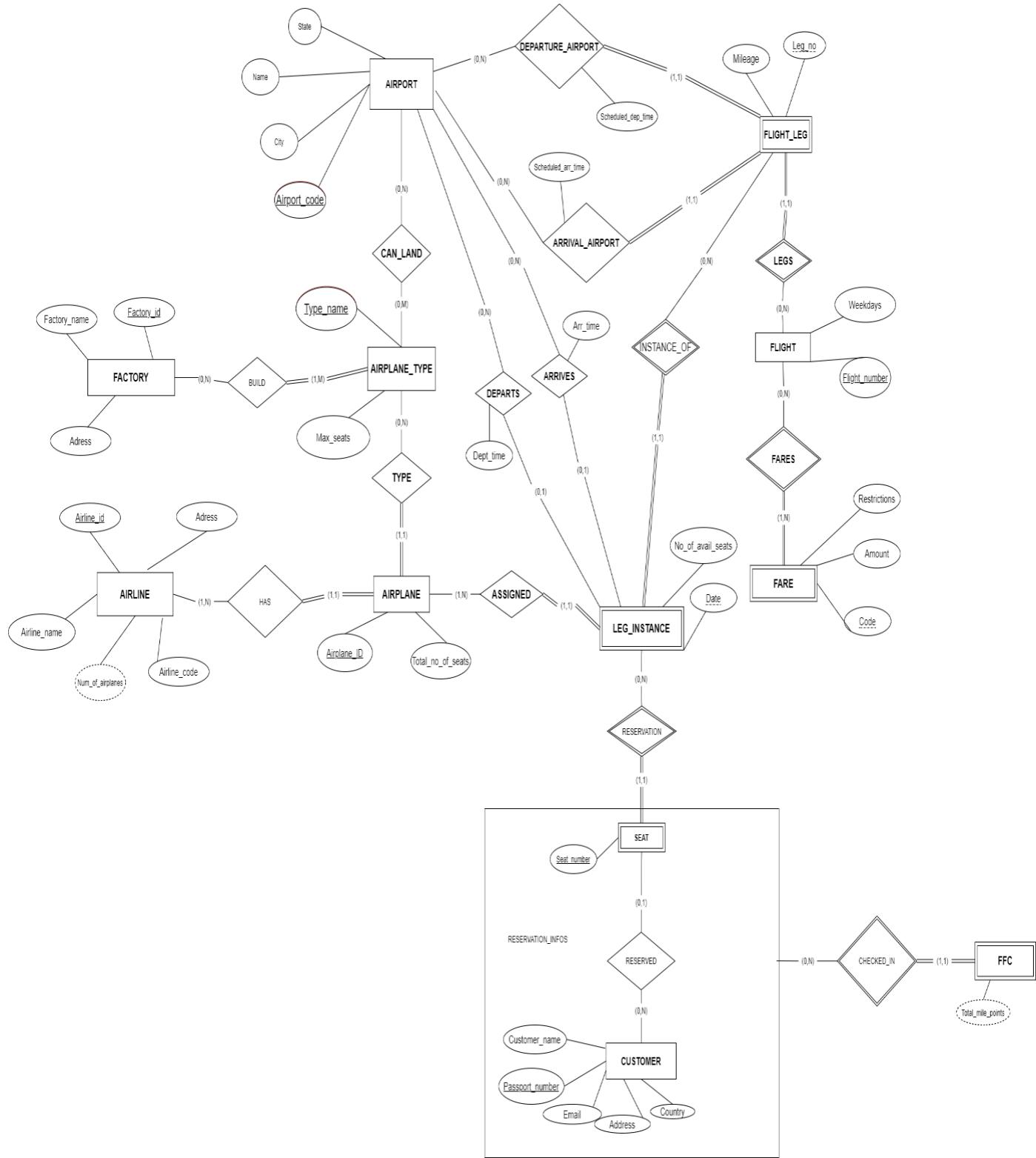
**05150000693 – RAMAZAN TAŞ**

**05160000538 – SABRİ GÜLSEVEN**

**05170000063 – ÖZGE BÜYÜKAŞIK**

**05170000065 – KEREM ERKINSOY**

# EER Diagramı



## 1. Veritabanı Oluşumu

Projenin gerçekleştirmesi için öncelikle program aracılığıyla bir veritabanı kuruyoruz. Bu kurulumu MySQL tabanlı MySQL Workbench üzerinden gerçekleştirdik.

```
CREATE DATABASE flight_control_system_db;
USE flight_control_system_db;
```

## 2. Veritabanı Oluşumu

### 2.1 Tablo Oluşumları

AIRPORT:

```
CREATE TABLE AIRPORT (
    Airport_code VARCHAR(3) NOT NULL,
    Airport_name VARCHAR(30) NOT NULL,
    City VARCHAR(20) NOT NULL,
    State VARCHAR(20) NOT NULL ,
    PRIMARY KEY (Airport_code)
);
```

|   | Field        | Type        | Null | Key | Default | Extra |
|---|--------------|-------------|------|-----|---------|-------|
| ▶ | Airport_code | varchar(3)  | NO   | PRI | NULL    |       |
|   | Airport_name | varchar(30) | NO   |     | NULL    |       |
|   | City         | varchar(20) | NO   |     | NULL    |       |
|   | State        | varchar(20) | NO   |     | NULL    |       |

## AIRLINE:

```
CREATE TABLE AIRLINE(
    Airline_id INT unsigned NOT NULL,
    Airline_name VARCHAR(20) NOT NULL,
    Address VARCHAR(50) NOT NULL,
    Number_of_airplanes INT UNSIGNED NOT NULL DEFAULT 0,
    Airline_code VARCHAR(3) NOT NULL UNIQUE ,
    PRIMARY KEY (Airline_id)
);
```

|   | Field               | Type         | Null | Key | Default | Extra |
|---|---------------------|--------------|------|-----|---------|-------|
| ▶ | Airline_id          | int unsigned | NO   | PRI | NULL    |       |
|   | Airline_name        | varchar(20)  | NO   |     | NULL    |       |
|   | Address             | varchar(50)  | NO   |     | NULL    |       |
|   | Number_of_airplanes | int unsigned | NO   |     | 0       |       |
|   | Airline_code        | varchar(3)   | NO   | UNI | NULL    |       |

## FACTORY

```
CREATE TABLE FACTORY(
    Factory_id INT UNSIGNED NOT NULL ,
    Factory_name VARCHAR(20) NOT NULL UNIQUE,
    Address VARCHAR(50) NOT NULL,
    PRIMARY KEY (Factory_id)
);
```

|   | Field        | Type         | Null | Key | Default | Extra |
|---|--------------|--------------|------|-----|---------|-------|
| ▶ | Factory_id   | int unsigned | NO   | PRI | NULL    |       |
|   | Factory_name | varchar(20)  | NO   | UNI | NULL    |       |
|   | Address      | varchar(50)  | NO   |     | NULL    |       |

## AIRPLANE\_TYPE

```
CREATE TABLE AIRPLANE_TYPE (
    Airplane_type_name VARCHAR(30) NOT NULL,
    Max_seats INT UNSIGNED NOT NULL,
    Company_name VARCHAR (20) NOT NULL ,
    PRIMARY KEY (Airplane_type_name),
    FOREIGN KEY (Company_name) REFERENCES FACTORY (Factory_name) ON DELETE CASCADE ON UPDATE CASCADE,
    CHECK (Max_seats <= 615 AND Max_seats >= 1)
);
```

|   | Field              | Type         | Null | Key | Default | Extra |
|---|--------------------|--------------|------|-----|---------|-------|
| ▶ | Airplane_type_name | varchar(30)  | NO   | PRI | NULL    |       |
|   | Max_seats          | int unsigned | NO   |     | NULL    |       |
|   | Company_name       | varchar(20)  | NO   | MUL | NULL    |       |

## AIRPLANE:

```
CREATE TABLE AIRPLANE (
    Airplane_id VARCHAR(6) NOT NULL,
    Total_number_of_seats INT UNSIGNED NOT NULL,
    Airplane_type VARCHAR(20) NOT NULL,
    Airline_code VARCHAR(3) ,
    PRIMARY KEY (Airplane_id),
    FOREIGN KEY (Airplane_type) REFERENCES AIRPLANE_TYPE (Airplane_type_name) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Airline_code) REFERENCES AIRLINE (Airline_code) ON DELETE SET NULL ON UPDATE CASCADE
);
```

|   | Field                 | Type         | Null | Key | Default | Extra |
|---|-----------------------|--------------|------|-----|---------|-------|
| ▶ | Airplane_id           | varchar(6)   | NO   | PRI | NULL    |       |
|   | Total_number_of_seats | int unsigned | NO   |     | NULL    |       |
|   | Airplane_type         | varchar(20)  | NO   | MUL | NULL    |       |
|   | Airline_code          | varchar(3)   | YES  | MUL | NULL    |       |

## CAN\_LAND:

```

CREATE TABLE CAN_LAND (
    Airplane_type_name VARCHAR(30) NOT NULL,
    Airport_code VARCHAR(3) NOT NULL,
    PRIMARY KEY (Airplane_type_name,Airport_code),
    FOREIGN KEY (Airplane_type_name) REFERENCES AIRPLANE_TYPE (Airplane_type_name) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Airport_code) REFERENCES AIRPORT (Airport_code) ON DELETE CASCADE ON UPDATE CASCADE
);

```

|   | Field              | Type        | Null | Key | Default | Extra |
|---|--------------------|-------------|------|-----|---------|-------|
| ▶ | Airplane_type_name | varchar(30) | NO   | PRI | NULL    |       |
|   | Airport_code       | varchar(3)  | NO   | PRI | NULL    |       |

## FLIGHT:

```

CREATE TABLE FLIGHT (
    Flight_number VARCHAR(6) NOT NULL,
    Airline_code VARCHAR(3) NOT NULL,
    Weekdays VARCHAR(20) NOT NULL,
    PRIMARY KEY (Flight_number),
    FOREIGN KEY (Airline_code) REFERENCES AIRLINE (Airline_code) ON DELETE CASCADE ON UPDATE CASCADE
);

```

|   | Field         | Type        | Null | Key | Default | Extra |
|---|---------------|-------------|------|-----|---------|-------|
| ▶ | Flight_number | varchar(6)  | NO   | PRI | NULL    |       |
|   | Airline_code  | varchar(3)  | NO   | MUL | NULL    |       |
|   | Weekdays      | varchar(20) | NO   |     | NULL    |       |

## FLIGHT\_LEG:

```

CREATE TABLE FLIGHT_LEG(
    Flight_number VARCHAR(6) NOT NULL,
    Leg_number INT UNSIGNED NOT NULL,
    Departure_airport_code VARCHAR(3) NOT NULL,
    Arrival_airport_code VARCHAR(3) NOT NULL,
    Scheduled_departure_time DATETIME NOT NULL,
    Scheduled_arrive_time DATETIME NOT NULL,
    Mileage INT UNSIGNED NOT NULL ,
    PRIMARY KEY (Flight_number, Leg_number),
    FOREIGN KEY (Flight_number) REFERENCES FLIGHT (Flight_number) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Arrival_airport_code) REFERENCES AIRPORT (Airport_code) ON DELETE RESTRICT ON UPDATE RESTRICT,
    FOREIGN KEY (Departure_airport_code) REFERENCES AIRPORT (Airport_code) ON DELETE RESTRICT ON UPDATE RESTRICT
);

```

|   | Field                    | Type         | Null | Key | Default | Extra |
|---|--------------------------|--------------|------|-----|---------|-------|
| ▶ | Flight_number            | varchar(6)   | NO   | PRI | NULL    |       |
|   | Leg_number               | int unsigned | NO   | PRI | NULL    |       |
|   | Departure_airport_code   | varchar(3)   | NO   | MUL | NULL    |       |
|   | Arrival_airport_code     | varchar(3)   | NO   | MUL | NULL    |       |
|   | Scheduled_departure_time | datetime     | NO   |     | NULL    |       |
|   | Scheduled_arrive_time    | datetime     | NO   |     | NULL    |       |
|   | Mileage                  | int unsigned | NO   |     | NULL    |       |

## LEG\_INSTANCE:

```
CREATE TABLE LEG_INSTANCE(
    Flight_number VARCHAR(6) NOT NULL,
    Leg_number INT UNSIGNED NOT NULL,
    Flight_date DATE NOT NULL,
    Departure_airport_code VARCHAR(3) NOT NULL DEFAULT "",
    Arrival_airport_code VARCHAR(3) NOT NULL DEFAULT "",
    Departure_time DATETIME NOT NULL,
    Arrive_time DATETIME NOT NULL,
    Airplane_id VARCHAR(6) NOT NULL,
    Number_of_available_seats INT UNSIGNED NOT NULL DEFAULT 1,
    PRIMARY KEY (Flight_number, Leg_number, Flight_date),
    FOREIGN KEY (Flight_number, Leg_number) REFERENCES FLIGHT_LEG (Flight_number, Leg_number) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Arrival_airport_code) REFERENCES AIRPORT (Airport_code) ON UPDATE RESTRICT ON DELETE RESTRICT ,
    FOREIGN KEY (Departure_airport_code) REFERENCES AIRPORT (Airport_code) ON UPDATE RESTRICT ON DELETE RESTRICT ,
    FOREIGN KEY (Airplane_id) REFERENCES AIRPLANE (Airplane_id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

|   | Field                     | Type         | Null | Key | Default | Extra |
|---|---------------------------|--------------|------|-----|---------|-------|
| ▶ | Flight_number             | varchar(6)   | NO   | PRI | NULL    |       |
|   | Leg_number                | int unsigned | NO   | PRI | NULL    |       |
|   | Flight_date               | date         | NO   | PRI | NULL    |       |
|   | Departure_airport_code    | varchar(3)   | NO   | MUL |         |       |
|   | Arrival_airport_code      | varchar(3)   | NO   | MUL |         |       |
|   | Departure_time            | datetime     | NO   |     | NULL    |       |
|   | Arrive_time               | datetime     | NO   |     | NULL    |       |
|   | Airplane_id               | varchar(6)   | NO   | MUL | NULL    |       |
|   | Number_of_available_seats | int unsigned | NO   |     | 1       |       |

## FARE:

```
CREATE TABLE FARE (
    Flight_number VARCHAR(6) NOT NULL,
    Fare_code VARCHAR(1) NOT NULL,
    Amount DOUBLE NOT NULL,
    Restrictions VARCHAR(100) NOT NULL,
    PRIMARY KEY (Flight_number, Fare_code),
    FOREIGN KEY (Flight_number) REFERENCES FLIGHT(Flight_number) ON DELETE CASCADE ON UPDATE CASCADE
);
```

|   | Field         | Type         | Null | Key | Default | Extra |
|---|---------------|--------------|------|-----|---------|-------|
| ▶ | Flight_number | varchar(6)   | NO   | PRI | NULL    |       |
|   | Fare_code     | varchar(1)   | NO   | PRI | NULL    |       |
|   | Amount        | double       | NO   |     | NULL    |       |
|   | Restrictions  | varchar(100) | NO   |     | NULL    |       |

## CUSTOMER:

```
CREATE TABLE CUSTOMER(
    Passport_number varchar(9) NOT NULL,
    Customer_name VARCHAR (20) NOT NULL,
    Customer_phone BIGINT UNSIGNED NOT NULL,
    Email VARCHAR(35) NOT NULL,
    Address VARCHAR(100) NOT NULL,
    Country VARCHAR(20) NOT NULL,
    PRIMARY KEY (Passport_number)
);
```

|   | Field           | Type         | Null | Key | Default | Extra |
|---|-----------------|--------------|------|-----|---------|-------|
| ▶ | Passport_number | varchar(10)  | NO   | PRI | NULL    |       |
|   | Customer_name   | varchar(30)  | NO   |     | NULL    |       |
|   | Customer_phone  | varchar(20)  | NO   |     | NULL    |       |
|   | Email           | varchar(35)  | NO   |     | NULL    |       |
|   | Address         | varchar(100) | NO   |     | NULL    |       |
|   | Country         | varchar(20)  | NO   |     | NULL    |       |

## SEAT\_RESERVATION:

```
CREATE TABLE SEAT_RESERVATION(
    Flight_number VARCHAR(6) NOT NULL,
    Leg_number INT UNSIGNED NOT NULL,
    Flight_date DATE NOT NULL,
    Seat_number varchar(5) NOT NULL,
    Passport_number varchar(9) NOT NULL,
    PRIMARY KEY (Flight_number,Leg_number,Flight_date,Seat_number),
    FOREIGN KEY (Flight_number,Leg_number,Flight_date) REFERENCES LEG_INSTANCE(Flight_number,Leg_number,Flight_date) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Passport_number) REFERENCES CUSTOMER(Passport_number) ON DELETE CASCADE ON UPDATE CASCADE
);
```

|   | Field           | Type         | Null | Key | Default | Extra |
|---|-----------------|--------------|------|-----|---------|-------|
| ▶ | Flight_number   | varchar(6)   | NO   | PRI | NULL    |       |
|   | Leg_number      | int unsigned | NO   | PRI | NULL    |       |
|   | Flight_date     | date         | NO   | PRI | NULL    |       |
|   | Seat_number     | varchar(5)   | NO   | PRI | NULL    |       |
|   | Passport_number | varchar(9)   | NO   | MUL | NULL    |       |

## CHECKED\_IN:

```

CREATE TABLE CHECKED_IN(
    Flight_number VARCHAR(6) NOT NULL,
    Leg_number INT UNSIGNED NOT NULL,
    Flight_date DATE NOT NULL,
    Seat_number varchar(5) NOT NULL,
    Passport_number varchar(9) NOT NULL,
    Departure_airport_code VARCHAR(3) ,
    Arrival_airport_code VARCHAR(3) ,
    Departure_time DATETIME ,
    Arrival_time DATETIME ,
    Mileage INT UNSIGNED ,
    PRIMARY KEY (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number),
    FOREIGN KEY (Flight_number,Leg_number,Flight_date)
        REFERENCES LEG_INSTANCE(Flight_number,Leg_number,Flight_date) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Passport_number) REFERENCES CUSTOMER(Passport_number) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Departure_airport_code) REFERENCES AIRPORT(Airport_code) ON DELETE CASCADE ON UPDATE CASCADE ,
    FOREIGN KEY (Arrival_airport_code) REFERENCES AIRPORT(Airport_code) ON DELETE CASCADE ON UPDATE CASCADE
);

```

|   | Field                  | Type         | Null | Key | Default | Extra |
|---|------------------------|--------------|------|-----|---------|-------|
| ▶ | Flight_number          | varchar(6)   | NO   | PRI | NULL    |       |
|   | Leg_number             | int unsigned | NO   | PRI | NULL    |       |
|   | Flight_date            | date         | NO   | PRI | NULL    |       |
|   | Seat_number            | varchar(5)   | NO   | PRI | NULL    |       |
|   | Passport_number        | varchar(9)   | NO   | PRI | NULL    |       |
|   | Departure_airport_code | varchar(3)   | YES  | MUL | NULL    |       |
|   | Arrival_airport_code   | varchar(3)   | YES  | MUL | NULL    |       |
|   | Departure_time         | datetime     | YES  |     | NULL    |       |
|   | Arrival_time           | datetime     | YES  |     | NULL    |       |
|   | Mileage                | int unsigned | YES  |     | NULL    |       |

FFC:

```

CREATE TABLE FFC(
    Passport_number varchar(9) NOT NULL,
    Total_mileage_point INT NOT NULL DEFAULT 0,
    PRIMARY KEY (Passport_number),
    FOREIGN KEY (Passport_number)
        REFERENCES CUSTOMER(Passport_number)
);

```

|   | Field               | Type       | Null | Key | Default | Extra |
|---|---------------------|------------|------|-----|---------|-------|
| ▶ | Passport_number     | varchar(9) | NO   | PRI | NULL    |       |
|   | Total_mileage_point | int        | NO   |     | 0       |       |

### 3. TRIGGERS

1.Uçuş tarihi şimdiki zamandan ileri bir tarihte olmalı.

```
DELIMITER $$  
CREATE TRIGGER date_check  
BEFORE INSERT ON LEG_INSTANCE  
FOR EACH ROW  
BEGIN  
IF NEW.Flight_date <= CURDATE() AND DATE(NEW.Departure_time) <= NEW.Flight_date THEN  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid date!';  
END IF;  
END$$  
DELIMITER ;
```

2. Geçerli bir uçuş kaydı için kullanılacak uçağın iki hava yolunda da konaklayabilir olması gerekmektedir.

```
DELIMITER $$  
CREATE TRIGGER can_land_check  
BEFORE INSERT ON LEG_INSTANCE  
FOR EACH ROW  
BEGIN  
DECLARE type_of_airplane VARCHAR(20);  
SELECT Airplane_type  
    INTO type_of_airplane  
    FROM AIRPLANE WHERE Airplane_id = NEW.Airplane_id;  
IF type_of_airplane NOT IN  
(SELECT Airplane_type_name FROM CAN_LAND WHERE Airport_code = NEW.Departure_airport_code) or  
type_of_airplane NOT IN (SELECT Airplane_type_name FROM CAN_LAND WHERE Airport_code = NEW.Arrival_airport_code) THEN  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "The airplane is can't land at these airports!";  
END IF;  
END$$  
DELIMITER ;
```

3. Bir uçağın koltuk sayısı o uçağın tipindeki uçakların koltuk sayılarından büyük olamaz.

```

DELIMITER $$

CREATE TRIGGER airplane_type_max_seat_number_check
BEFORE INSERT ON AIRPLANE
FOR EACH ROW
BEGIN
    DECLARE max_seat INT;
    SELECT Max_seats
    INTO max_seat
    FROM AIRPLANE_TYPE WHERE Airplane_type_name = NEW.Airplane_type;
    IF NEW.Total_number_of_seats > max_seat THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid total seat number!';
    END IF;
END$$
DELIMITER ;

```

4. Müşteri check in yaptığında uçuşun kilometre bilgisini alarak FFC tablosundaki toplam mil puanını arttırır.

```

DELIMITER $$

CREATE TRIGGER update_mileage
AFTER INSERT
ON CHECKED_IN FOR EACH ROW
BEGIN
    DECLARE oldMilage INT;
    SELECT Total_mileage_point
    INTO oldMilage
    FROM FFC WHERE Passport_number = NEW.Passport_number;
    UPDATE FFC SET Total_mileage_point = oldMilage + NEW.Mileage WHERE Passport_number = NEW.Passport_number;
END$$
DELIMITER ;

```

5. Check-in yapan müşterinin uçuş bilgilerini CHECKED\_IN tablosuna ekliyoruz.

```

DELIMITER $$

CREATE TRIGGER get_flight_infos
    BEFORE INSERT
    ON CHECKED_IN FOR EACH ROW
BEGIN
    SET NEW.Departure_airport_code = (
        SELECT Departure_airport_code
        from LEG_INSTANCE
        where Flight_number = NEW.Flight_number AND
            Leg_number = NEW.Leg_number AND
            Flight_date = NEW.Flight_date);

    SET NEW.Arrival_airport_code = (
        SELECT Arrival_airport_code
        from LEG_INSTANCE
        where Flight_number = NEW.Flight_number AND
            Leg_number = NEW.Leg_number AND
            Flight_date = NEW.Flight_date);

    SET NEW.Arrival_time = (
        SELECT Arrive_time
        from LEG_INSTANCE
        where Flight_number = NEW.Flight_number AND
            Leg_number = NEW.Leg_number AND
            Flight_date = NEW.Flight_date);
    SET NEW.Departure_time = (
        SELECT Departure_time
        from LEG_INSTANCE
        where Flight_number = NEW.Flight_number AND
            Leg_number = NEW.Leg_number AND
            Flight_date = NEW.Flight_date);
    SET NEW.Mileage = (
        SELECT Mileage
        from FLIGHT_LEG
        where Flight_number = NEW.Flight_number AND
            Leg_number = NEW.Leg_number );
END$$
DELIMITER ;

```

6. Müşteri rezervasyon yaptırmadan önce müsait koltuk durumuna bakılır. Eğer uygun koltuk varsa uygun koltuk sayısı bir azaltılır yoksa hata mesajı verilir.

```
CREATE TRIGGER leg_instance_available_seat_number_check
BEFORE INSERT ON SEAT_RESERVATION
FOR EACH ROW
BEGIN
    DECLARE available_seat INT;
    SELECT Number_of_available_seats
    INTO available_seat
    FROM LEG_INSTANCE |
    WHERE
        Flight_number = NEW.Flight_number AND
        Leg_number = NEW.Leg_number AND
        Flight_date = NEW.Flight_date;
    IF available_seat = 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'There is no available seat for this leg instance!';
    ELSE
        UPDATE LEG_INSTANCE SET Number_of_available_seats = available_seat - 1
        WHERE
            Flight_number = NEW.Flight_number AND
            Leg_number = NEW.Leg_number AND
            Flight_date = NEW.Flight_date;
        END IF;
    END$$
DELIMITER ;
```

7. Yapılan rezervasyon silindiğinde uçuştaki uygun koltuk sayısı 1 artırılmalıdır.

```
DELIMITER $$  
CREATE TRIGGER after_reservation_delete  
AFTER DELETE  
ON SEAT_RESERVATION FOR EACH ROW  
BEGIN  
    DECLARE available_seat INT;  
    SELECT Number_of_available_seats  
    INTO available_seat  
    FROM LEG_INSTANCE  
    WHERE  
        Flight_number = OLD.Flight_number AND  
        Leg_number = OLD.Leg_number AND  
        Flight_date = OLD.Flight_date;  
  
    UPDATE LEG_INSTANCE SET Number_of_available_seats = available_seat + 1  
    WHERE  
        Flight_number = OLD.Flight_number AND  
        Leg_number = OLD.Leg_number AND  
        Flight_date = OLD.Flight_date;  
  
END$$  
DELIMITER ;
```

8. Her müşteri eklendiğinde o müşteriyeye ait pasaport numarasıyla birlikte bir FFC kaydı oluşturulur.

```
DELIMITER $$  
CREATE TRIGGER create_ffc  
AFTER INSERT  
ON CUSTOMER FOR EACH ROW  
BEGIN  
    INSERT INTO FFC (Passport_number) VALUES (NEW.Passport_number);  
END$$  
DELIMITER ;
```

9. Sisteme herhangi bir uçak kaydı eklendiğinde o uçağa ait şirketin uçak sayısı 1 arttırılmaktadır.

```
DELIMITER $$  
CREATE TRIGGER update_airplane_number  
AFTER INSERT  
ON AIRPLANE FOR EACH ROW  
BEGIN  
    DECLARE oldNumber INT;  
    SELECT Number_of_airplanes  
    INTO oldNumber  
    FROM AIRLINE  
    WHERE  
        Airline_code = NEW.Airline_code;  
    UPDATE AIRLINE SET Number_of_airplanes = oldNumber + 1 WHERE Airline_code = NEW.Airline_code;  
END$$  
DELIMITER ;
```

10. Bir şirkete bağlı uçak kaydı silindiğinde o şirketin toplam uçak sayısı 1 azaltılmaktadır.

```
DELIMITER $$  
CREATE TRIGGER update_airplane_number_2  
AFTER DELETE  
ON AIRPLANE FOR EACH ROW  
BEGIN  
    DECLARE oldNumber INT;  
    SELECT Number_of_airplanes  
    INTO oldNumber  
    FROM AIRLINE  
    WHERE  
        Airline_code = OLD.Airline_code;  
    IF oldNumber = 0 THEN  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'There is zero airplane at this airline!';  
    ELSE  
        UPDATE AIRLINE SET Number_of_airplanes = oldNumber - 1 WHERE Airline_code = OLD.Airline_code;  
    END IF;  
END$$  
DELIMITER ;
```

11. LEG\_INSTANCE tablosuna bir kayıt eklenmeden önce o uçuşa ait uygun koltuk sayısına, uçuşu yapacak uçağın toplam koltuk sayısı atanır.

```
DELIMITER $$  
CREATE TRIGGER number_of_seats  
BEFORE INSERT ON LEG_INSTANCE  
FOR EACH ROW  
BEGIN  
    SET NEW.Number_of_available_seats = (SELECT Total_number_of_seats from AIRPLANE where Airplane_id= NEW.Airplane_id);  
END$$  
DELIMITER ;
```

## 4. CHECK CONSTRAINTS

1. Gerçekleştirilecek bir uçuşörneğinde kalkış zamanı varış zamanının ilerisinde olmamalıdır.

```
ALTER TABLE LEG_INSTANCE  
ADD Constraint Date_Check_LI  
CHECK(Departure_time < Arrive_time);
```

2. Planlanan varış zamanı planlanan kalkış zamanından ileri bir tarihte olmalıdır.

```
ALTER TABLE FLIGHT_LEG  
ADD Constraint Date_Check_FL  
CHECK(Scheduled_departure_time < Scheduled_arrive_time );
```

3. Bilet ücreti 0'dan büyük olmalıdır.

```
ALTER TABLE FARE  
ADD Constraint Amount_Check  
CHECK (Amount >= 0);
```

4. "F" First Class, "J" Business Class, "Y" ise Economy Class'ını temsil etmektedir. Bilet kodu bu 3 tipten oluşmalıdır.

```
ALTER TABLE FARE  
ADD Constraint Code_Check  
CHECK (Fare_code = "F" OR Fare_code = "J" OR Fare_code = "Y" );
```

5. Bir uçuşta uygun koltuk sayısı negatif olamaz.

```
ALTER TABLE LEG_INSTANCE  
ADD CONSTRAINT Flight_Seat_Check  
CHECK (Number_of_available_seats >= 0);
```

6. Uçağın kalkış ve varış yerleri aynı olmamalıdır.

```
ALTER TABLE FLIGHT_LEG  
ADD Constraint Airport_Check_FL  
CHECK(Departure_airport_code!=Arrival_airport_code);
```

7. Gerçekleştirilecek bir uçuşta da kalkış ve varış yerleri farklı olmak zorundadır.

```
ALTER TABLE LEG_INSTANCE  
ADD Constraint Airport_Check_LI  
CHECK(Departure_airport_code!=Arrival_airport_code);
```

8. Müşterinin pasaport numarası 7 ile 9 hane arasında bir değerde olmalıdır.

```
Alter table CUSTOMER  
ADD Constraint Passport_number_check  
CHECK (LENGTH(Passport_number) BETWEEN 7 and 9);
```

## 5. SQL STATEMENTS

Airport,Factory ve Airline tablolarına yeterli veri girişi eklenmiştir.

```
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("SAW","Sabina Gökcen Airport","İstanbul","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("ADB","Adnan Menderes Airport","İzmir","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("USQ","Uşak Airport","Uşak","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("KZR","Zafer Airport ","Kütahya","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("ISL","Atatürk Airport","İstanbul","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("ERZ","Erzurum Airport","Erzurum","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("ESB","Esenboğa Airport","Ankara","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("YEI","Bursa Airport","Bursa","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("DLM","Dalaman Airport","Muğla","Public");  
INSERT INTO AIRPORT(Airport_code,Airport_name,City,State) VALUES("GZT","Gaziantep Airport","Gaziantep","Public");  
  
INSERT INTO FACTORY VALUES (789052719,"Airbus","Toulouse,França");  
INSERT INTO FACTORY VALUES (862795626,"Boeing","Chicago,ABD");  
INSERT INTO FACTORY VALUES (859764289,"Bombardier","Wichita,Kansas,ABD");  
INSERT INTO AIRLINE (Airline_id,Airline_name,Address,Airline_code) VALUES (249871359,"Turkish Airlines","İstanbul,Türkiye","THY");  
INSERT INTO AIRLINE (Airline_id,Airline_name,Address,Airline_code) VALUES (179245063,"Anadolu Jet","Ankara,Türkiye","TK");  
INSERT INTO AIRLINE (Airline_id,Airline_name,Address,Airline_code) VALUES (054795173,"Pegasus","İstanbul,Türkiye","PC");  
INSERT INTO AIRLINE (Airline_id,Airline_name,Address,Airline_code) VALUES (047982417,"Sunexpress","Antalya,Türkiye","XQ");  
INSERT INTO FACTORY VALUES (278614098,"Douglas","Santa Monica, California,ABD");
```

\* Uçak tipleri ve o tiplere ait uçaklar için veri girişi yapılmıştır.

```
INSERT INTO Airplane_type VALUES ("Douglas DC-2","14","Douglas");
INSERT INTO AIRPLANE_TYPE (Airplane_type_name,Max_seats,Company_name) VALUES ("Airbus A320",150,"Airbus");
INSERT INTO AIRPLANE_TYPE (Airplane_type_name,Max_seats,Company_name) VALUES ("Airbus A310",220,"Airbus");
INSERT INTO AIRPLANE_TYPE (Airplane_type_name,Max_seats,Company_name) VALUES ("Boeing 777",301,"Boeing");
INSERT INTO AIRPLANE_TYPE (Airplane_type_name,Max_seats,Company_name) VALUES ("Boeing 767-300",351,"Boeing");
INSERT INTO AIRPLANE_TYPE (Airplane_type_name,Max_seats,Company_name) VALUES ("Learjet 40",7,"Bombardier");

INSERT INTO AIRPLANE VALUES ("N930NN",130,"Airbus A320","THY");
INSERT INTO AIRPLANE VALUES ("N973JM",150,"Airbus A320","PC");
INSERT INTO AIRPLANE VALUES ("N4216S",200,"Airbus A310","XQ");
INSERT INTO AIRPLANE VALUES ("N340LV",220,"Airbus A310","THY");
INSERT INTO AIRPLANE VALUES ("C-GTF0",250,"Boeing 777","XQ");
INSERT INTO AIRPLANE VALUES ("JA8089",300,"Boeing 777","TK");
INSERT INTO AIRPLANE VALUES ("N901CM",310,"Boeing 767-300","THY");
INSERT INTO AIRPLANE VALUES ("N7025U",350,"Boeing 767-300","PC");
INSERT INTO AIRPLANE VALUES ("JA5894",7,"Learjet 40","TK");
```

\* CAN\_LAND tablosuna veri girişi yapıldı.

```
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A320","SAW");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A320","ADB");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A320","USQ");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A320","KZR");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A320","ISL");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A310","SAW");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A310","DLM");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A310","GZT");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A310","ESB");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A310","ISL");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A310","ERZ");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Airbus A310","KZR");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 777","SAW");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 777","KZR");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 777","ESB");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 777","ADB");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 777","ISL");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 777","GZT");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 767-300","KZR");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 767-300","GZT");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 767-300","ISL");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 767-300","ESB");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 767-300","ERZ");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 767-300","ADB");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Boeing 767-300","SAW");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Learjet 40","ISL");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Learjet 40","KZR");
INSERT INTO CAN_LAND (AIRPLANE_TYPE_NAME,AIRPORT_CODE) VALUES ("Learjet 40","GZT");
```

\*Uçuş bilgilerine ait veri girişleri yapılmıştır.

```
INSERT INTO FLIGHT (FLIGHT_NUMBER,AIRLINE_CODE,WEEKDAYS) VALUES ("DF2753","THY","PT CR CM");
INSERT INTO FLIGHT VALUES ("LN3211","PC","SL CM CT PZ");
INSERT INTO FLIGHT VALUES ("GT4638","XO","PT CT");
INSERT INTO FLIGHT VALUES ("KV3323","TK","SL PR CT");
INSERT INTO FLIGHT VALUES ("LX3100","TK","PZ");
INSERT INTO FLIGHT VALUES ("FD5610","PC","PT SL CR PR CM");
INSERT INTO FLIGHT VALUES ("EN4267","PC","CT PZ");
INSERT INTO FLIGHT VALUES ("TK2752","THY","PR CM CT PZ");

INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("DF2753",901,"SAN","ADB",'2021-02-15 14:30:00','2021-02-15 16:00:00',350);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("LN3211",902,"SAM","ESB",'2021-02-16 15:30:00','2021-02-16 16:45',325);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("GT4638",903,"ISL","KZR",'2021-02-17 08:30:00','2021-02-17 09:45',250);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("KV3323",904,"KZR","GZT",'2021-03-17 08:30:00','2021-03-17 10:30:00',195);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("LX3100",905,"ADB","ISL",'2021-02-18 08:30:00','2021-02-18 09:40:00',440);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("FD5610",906,"ERZ","USQ",'2021-03-01 21:30:00','2021-03-01 22:30:00',440);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("EN4267",906,"ADB","USQ",'2021-03-05 21:30:00','2021-03-05 23:15:00',256);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("DF2753",907,"ESB","SAW",'2021-05-01 21:30:00','2021-05-01 22:30:00',221);
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("GT4638",908,"ADB","SAW",'2021-03-15 11:30:00','2021-03-15 13:30:00',221);
```

\*Uçuş örnek kayıtları oluşturuldu.

```
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,
Departure_time,Arrive_time,Airplane_id)
VALUES ("GT4638",908,"2021-03-15","ADB","SAW",'2021-03-15 11:30:00','2021-03-15 13:30:00',"N930NN");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,
Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2753",907,"2021-05-01","ESB","SAW",'2021-02-17 23:30:00','2021-02-18 02:00:00',"N901CM");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("GT4638",903,"2021-02-17","ISL","KZR",'2021-02-17 08:30:00','2021-02-17 09:45',"N340LV");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2753",901,"2021-02-15","SAM","ADB",'2021-02-15 14:30:00','2021-02-15 16:00:00',"JA8089");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2753",901,"2021-02-16","SAM","ADB",'2021-02-16 14:30:00','2021-02-16 16:00:00',"JA8089");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2753",901,"2021-02-17","SAM","ADB",'2021-02-17 14:30:00','2021-02-17 16:00:00',"JA8089");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2753",901,"2021-02-18","SAM","ADB",'2021-02-18 14:30:00','2021-02-18 16:00:00',"JA8089");
```

```
INSERT INTO FLIGHT (FLIGHT_NUMBER,AIRLINE_CODE,WEEKDAYS) VALUES ("DF2755","PC","PT CR CM");
INSERT INTO FLIGHT_LEG (Flight_number,Leg_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrive_time,Mileage)
VALUES ("DF2755",901,"ADB","SAW",'2021-02-15 14:30:00','2021-02-15 16:00:00',350);
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2755",901,"2021-02-15","ADB","SAW",'2021-02-15 14:30:00','2021-02-15 16:00:00',"JA8089");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2755",901,"2021-02-17","ADB","SAW",'2021-02-17 14:30:00','2021-02-17 16:00:00',"JA8089");
INSERT INTO LEG_INSTANCE (Flight_number,Leg_number,Flight_date,Departure_airport_code,Arrival_airport_code,Departure_time,Arrive_time,Airplane_id)
VALUES ("DF2755",901,"2021-02-19","ADB","SAW",'2021-02-19 14:30:00','2021-02-19 16:00:00',"JA8089");
```

## \* Biletlere dair bilgiler girildi.

```
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("DF2753","F",750,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("DF2753","J",550,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("DF2753","Y",300,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("LN3211","F",625,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("LN3211","Y",205,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("GT4638","J",450,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("KV3323","J",425,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("LX3100","Y",250,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("FD5610","Y",175,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("EN4267","J",480,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("TK2752","Y",140,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("LN3211","J",350,"MASK ON");
INSERT INTO FARE (FLIGHT_NUMBER,FARE_CODE,AMOUNT,RESTRICTIONS) VALUES ("GT4638","Y",210,"MASK ON");
```

## \* Müşteri bilgileri girildi.

```
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES("U00005048","Kerem Erkinsoy","5369874123","kerem@gmail.com","Usak","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES("U00006048","İbrahim Kale","5055448210","ibo@gmail.com","Samsun","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES("U00009048","Lokman Hekim","5055446515","loki@gmail.com","Konya","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES("U00007858","Aybars Aydin","5545447810","ibo@gmail.com","Eskisehir","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES("U00001905","Taylan Tufan","5545447810","tt@gmail.com","Alanya","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES("U00001903","Halil Toprak","50519074586","tt@gmail.com","Izmir","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00006071","Özge Buyukasik","905414778681","o.b@g.com","Izmir","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00008041","Ramazan Tas","905414778682","r.t@g.com","Usak","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00004169","Sabri Gulseven","905414778683","s.g@g.com","Bursa","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00007526","Hasan Tahsin","905414778684","h.t@g.com","Izmir","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00006345","Ibrahim Afellay","44404444","i.a@g.com","Amsterdam","Hollanda");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00004978","Gorkem Aktas","905414778686","g.a@g.com","Ankara","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00009987","Emre Karagun","5414778687","e.k@g.com","Istanbul","Turkiye");
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_phone,Email,Address,Country)
VALUES ("U00009988","Cristiano Pele","2245671052","c.p@g.com","Lizbon","Portekiz");
```

## \* Koltuk rezervasyon kayıtları girildi.

```
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("DF2753",901,"2021-02-15","15A","U00005048");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","19C","U00006071");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","20C","U00001903");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","21C","U00001905");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","22C","U00007858");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("DF2753",901,"2021-02-15","22C","U00006048");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("DF2753",901,"2021-02-15","22D","U00009048");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("DF2753",907,"2021-05-01","15D","U00003026");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","25C","U00008046");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","20C","U00004560");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","21C","U00003026");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","22C","U00004026");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("DF2753",901,"2021-02-15","10B","U00007067");
INSERT INTO SEAT_RESERVATION (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("DF2753",901,"2021-02-15","20A","U00004605");
```

## \* Check-in yapan müşterilerin bilgileri girildi.

```
INSERT INTO checked_in (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","19C","U00006071");
INSERT INTO checked_in (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("DF2753",907,"2021-05-01","15D","U00003026");
INSERT INTO checked_in (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","25C","U00008046");
INSERT INTO checked_in (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","20C","U00004560");
INSERT INTO checked_in (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","21C","U00003026");
INSERT INTO checked_in (Flight_number,Leg_number,Flight_date,Seat_number,Passport_number)
VALUES ("GT4638",903,"2021-02-17","22C","U00004026");
```

\* DF2753 uçuşundaki bilet kodu Ekonomi sınıfından olan biletlerin ücretine 20 birim zam uygulanmıştır.

**UPDATE FARE**

```
SET Amount=Amount+20  
WHERE Flight_number="DF2753" AND FARE_CODE="Y";
```

\* İstanbul – İzmir arası mesafenin uçuş bilgisine göre

```
SELECT AIRPORT1.City AS Arrival_Airport_City,  
AIRPORT2.City AS Departure_Airport_City,Mileage  
FROM AIRPORT AS AIRPORT1, AIRPORT AS AIRPORT2,FLIGHT_LEG  
WHERE AIRPORT1.City="İstanbul" AND AIRPORT2.City="İzmir"  
AND AIRPORT1.Airport_code=FLIGHT_LEG.Departure_airport_code  
AND AIRPORT2.Airport_code=FLIGHT_LEG.Arrival_airport_code;
```

\* Çıktısı

|   | Arrival_Airport_City | Departure_Airport_City | Mileage |
|---|----------------------|------------------------|---------|
| ▶ | İstanbul             | İzmir                  | 350     |

\* Planlanan tarihte gerçekleşen uçuşların listelenmesi

```
SELECT FLIGHT_LEG.Flight_number,Departure_time,Arrive_time,Scheduled_departure_time,Scheduled_arrive_time  
FROM FLIGHT_LEG,LEG_INSTANCE  
WHERE FLIGHT_LEG.Scheduled_departure_time=LEG_INSTANCE.Departure_time  
AND FLIGHT_LEG.Scheduled_arrive_time=LEG_INSTANCE.Arrive_time;
```

Çıktısı:

|   | Flight_number | Departure_time      | Arrive_time         | Scheduled_departure_time | Scheduled_arrive_time |
|---|---------------|---------------------|---------------------|--------------------------|-----------------------|
| ▶ | DF2755        | 2021-02-15 14:30:00 | 2021-02-15 16:00:00 | 2021-02-15 14:30:00      | 2021-02-15 16:00:00   |
|   | DF2753        | 2021-02-15 14:30:00 | 2021-02-15 16:00:00 | 2021-02-15 14:30:00      | 2021-02-15 16:00:00   |
|   | DF2755        | 2021-02-15 14:30:00 | 2021-02-15 16:00:00 | 2021-02-15 14:30:00      | 2021-02-15 16:00:00   |
|   | DF2753        | 2021-02-15 14:30:00 | 2021-02-15 16:00:00 | 2021-02-15 14:30:00      | 2021-02-15 16:00:00   |
|   | GT4638        | 2021-02-17 08:30:00 | 2021-02-17 09:45:00 | 2021-02-17 08:30:00      | 2021-02-17 09:45:00   |
|   | GT4638        | 2021-03-15 11:30:00 | 2021-03-15 13:30:00 | 2021-03-15 11:30:00      | 2021-03-15 13:30:00   |

\* Havayollarında konumlanabilecek uçakların max kapasitelerine göre sıralanması:

```
SELECT can_land.Airport_code,airplane_type.airplane_type_name, airplane_type.Max_seats
FROM can_land, airplane_type
WHERE can_land.Airplane_type_name = airplane_type.Airplane_type_name
ORDER BY Max_seats DESC;
```

\* Çıktısı:

|   | Airport_code | airplane_type_name | Max_seats |
|---|--------------|--------------------|-----------|
| ▶ | SAW          | Boeing 767-300     | 351       |
|   | KZR          | Boeing 767-300     | 351       |
|   | ISL          | Boeing 767-300     | 351       |
|   | GZT          | Boeing 767-300     | 351       |
|   | ESB          | Boeing 767-300     | 351       |
|   | ERZ          | Boeing 767-300     | 351       |
|   | ADB          | Boeing 767-300     | 351       |
|   | SAW          | Boeing 777         | 301       |
|   | KZR          | Boeing 777         | 301       |
|   | ISL          | Boeing 777         | 301       |
|   | GZT          | Boeing 777         | 301       |
|   | ESB          | Boeing 777         | 301       |
|   | ADB          | Boeing 777         | 301       |
|   | SAW          | Airbus A310        | 220       |
|   | KZR          | Airbus A310        | 220       |
|   | ISL          | Airbus A310        | 220       |
|   | GZT          | Airbus A310        | 220       |
|   | ESB          | Airbus A310        | 220       |
|   | ERZ          | Airbus A310        | 220       |
|   | DLM          | Airbus A310        | 220       |
|   | USQ          | Airbus A320        | 150       |
|   | SAW          | Airbus A320        | 150       |
|   | KZR          | Airbus A320        | 150       |
|   | ISL          | Airbus A320        | 150       |
|   | ADB          | Airbus A320        | 150       |

\* Uçuş ücretinin 200 birimden az olduğu uçuşlar ve bunu gerçekleştiren hava yolu şirketleri.

```
SELECT Airline_code,Flight_number
FROM FLIGHT
WHERE Flight_number IN (
    SELECT Flight_number |
    FROM FARE
    WHERE Amount < 200);
```

|   | Airline_code | Flight_number |
|---|--------------|---------------|
| ▶ | PC           | FD5610        |
|   | THY          | TK2752        |
| * | NULL         | NULL          |

\* İstanbul'a gidecek olan yolcuların isimleri

```
SELECT Customer_name  
FROM CUSTOMER , CHECKED_IN, AIRPORT  
WHERE CUSTOMER.Passport_number = CHECKED_IN.Passport_number  
AND CHECKED_IN.Arrival_airport_code =AIRPORT.Airport_code  
AND AIRPORT.City = "Istanbul" ;
```

Çıktısı:

| Customer_name |
|---------------|
| EMRE DURSUN   |

\* Gerçekleşecek uçuşların kapasitesi 300'ün üzerinde olanlarının birkaç uçuş bilgisi listelenmiştir.

```
SELECT Flight_number ,Leg_number,Flight_date  
FROM LEG_INSTANCE  
WHERE Airplane_id IN(  
SELECT Airplane_id  
FROM AIRPLANE  
WHERE Total_number_of_seats > 300);
```

Çıktısı:

|   | Flight_number | Leg_number | Flight_date |
|---|---------------|------------|-------------|
| ▶ | DF2753        | 907        | 2021-05-01  |
| * | NULL          | NULL       | NULL        |

\* CHECK-IN yapmayan müşteriler

```
SELECT *  
FROM SEAT_RESERVATION AS A  
WHERE NOT EXISTS (  
SELECT Flight_number ,Leg_number ,Flight_date ,Seat_number Passport_number  
FROM CHECKED_IN AS B  
WHERE A.Flight_number = B.Flight_number  
AND A.Leg_number = B.Leg_number  
AND A.Flight_date =B.Flight_date  
AND A.Seat_number=B.Seat_number  
AND A.Passport_number=B.Passport_number);
```

Çıktısı:

| Flight_number | Leg_number | Flight_date | Seat_number | Passport_number |
|---------------|------------|-------------|-------------|-----------------|
| DF2753        | 901        | 2021-02-15  | 15A         | U00001903       |
| GT4638        | 903        | 2021-02-17  | 20C         | U00001903       |
| GT4638        | 903        | 2021-02-17  | 21C         | U00001905       |
| GT4638        | 903        | 2021-02-17  | 31C         | U00003026       |
| GT4638        | 903        | 2021-02-17  | 32C         | U00004026       |
| DF2753        | 901        | 2021-02-15  | 20A         | U00004605       |
| DF2753        | 901        | 2021-02-15  | 28C         | U00006048       |
| GT4638        | 903        | 2021-02-17  | 19C         | U00006071       |
| DF2753        | 901        | 2021-02-15  | 10B         | U00007067       |
| GT4638        | 903        | 2021-02-17  | 22C         | U00007858       |
| DF2753        | 901        | 2021-02-15  | 22D         | U00009048       |

\*\*

```
SELECT Flight_number, Fare_code, Amount
FROM FARE
WHERE Flight_number IN (
    SELECT Flight_number
    FROM LEG_INSTANCE WHERE Flight_date = CURDATE() + 1
);
```

DELETE SORGULARI:

```
DELETE FROM AIRPLANE WHERE Airplane_type="Boeing 777";
DELETE FROM FLIGHT WHERE Weekdays="Pt" AND Flight_number="DF2753";
DELETE FROM CUSTOMER WHERE Passport_number="U00004560";
DELETE FROM CAN_LAND WHERE Airplane_type_name = "Boeing 777";
```

\*\*

```
UPDATE LEG_INSTANCE
SET Airplane_id="N930NA "
WHERE LEG_INSTANCE.Airplane_id=AIRPLANE.Airplane_id
AND AIRPLANE.Airplane_type = "Boeing 777" AND AIRPLANE.Airline_code="TK" ;
UPDATE FARE
SET Amount=Amount+20
WHERE Flight_number="DF2753" AND FARE_CODE="Y";
```

Çıktısı:

\* Hangi havalimanına hangi uçak ve hangi havayolu şirketinden inecek olanları gösterir.

```
CREATE VIEW WHICH_CAN_LAND AS
SELECT *
FROM CAN_LAND
RIGHT JOIN AIRPLANE ON CAN_LAND.Airplane_type_name=AIRPLANE.Airplane_type;
```

\* Bu view'ın amacı kullanıcının uçuş araması yaparken en yakın tarihli uçuşları daha önceden görebilmesini sağlamak. Bu view için departure airport ve arrival airport condition'larıyla yazılacak sorgular iki havalimanı arasındaki uçuşları en yakın tarihten en uzak tarihe sıralamaktadır.

```
CREATE VIEW NEAREST_LEG_INSTANCES AS
SELECT *
FROM LEG_INSTANCE
ORDER BY Flight_date;
```

Çıktısı:

| Flight_number | Leg_number | Flight_date | Departure_airport_code | Arrival_airport_code | Departure_time      | Arrive_time         | Airplane_id | Number_of_available_seats |
|---------------|------------|-------------|------------------------|----------------------|---------------------|---------------------|-------------|---------------------------|
| DF2753        | 910        | 2021-02-13  | SAW                    | ADB                  | 2021-02-13 14:30:00 | 2021-02-13 16:00:00 | JA8089      | 300                       |
| LN3211        | 911        | 2021-02-13  | SAW                    | ADB                  | 2021-02-13 14:30:00 | 2021-02-13 16:00:00 | N901CM      | 310                       |
| DF2753        | 901        | 2021-02-15  | SAW                    | ADB                  | 2021-02-15 14:30:00 | 2021-02-15 16:00:00 | JA8089      | 297                       |
| DF2755        | 901        | 2021-02-15  | ADB                    | SAW                  | 2021-02-15 14:30:00 | 2021-02-15 16:00:00 | JA8089      | 300                       |
| DF2753        | 901        | 2021-02-16  | SAW                    | ADB                  | 2021-02-16 14:30:00 | 2021-02-16 16:00:00 | JA8089      | 300                       |
| DF2753        | 901        | 2021-02-17  | SAW                    | ADB                  | 2021-02-17 14:30:00 | 2021-02-17 16:00:00 | JA8089      | 300                       |
| DF2755        | 901        | 2021-02-17  | ADB                    | SAW                  | 2021-02-17 14:30:00 | 2021-02-17 16:00:00 | JA8089      | 300                       |
| GT4638        | 903        | 2021-02-17  | ISL                    | KZR                  | 2021-02-17 08:30:00 | 2021-02-17 09:45:00 | N340LV      | 216                       |
| DF2753        | 901        | 2021-02-18  | SAW                    | ADB                  | 2021-02-18 14:30:00 | 2021-02-18 16:00:00 | JA8089      | 300                       |
| DF2755        | 901        | 2021-02-19  | ADB                    | SAW                  | 2021-02-19 14:30:00 | 2021-02-19 16:00:00 | JA8089      | 300                       |
| GT4638        | 908        | 2021-03-15  | ADB                    | SAW                  | 2021-03-15 11:30:00 | 2021-03-15 13:30:00 | N930NN      | 130                       |
| DF2753        | 907        | 2021-05-01  | ESB                    | SAW                  | 2021-02-17 23:30:00 | 2021-02-18 02:00:00 | N901CM      | 310                       |

\* Customer segmentation view'ı. FFC tablosunda tutulan customerların sahip oldukları toplam mil puanları 1000'in üzerinde olunca customer otomatik olarak customer segmentation view'ına eklenmektedir. Bu view'da bulunan customerlar için bilet fiyatlarında indirimler sağlanacaktır.

```
CREATE VIEW CUSTOMER_SEG AS
SELECT *
FROM FFC
WHERE Total_mileage_point >= 1000;
```

Çıktısı:

| Result Grid |                 |                     |
|-------------|-----------------|---------------------|
|             | Passport_number | Total_mileage_point |
| ▶           | U00003026       | 1042                |

REST API GORSELLERİ:

The screenshot shows the Postman application interface. The top bar has a 'GET' method selected and the URL 'http://localhost:3000/api/flights'. Below the URL, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. On the right side, there are buttons for 'Send', 'Cookies', and 'Save Response'. Under the 'Headers' tab, 'Content-Type' is set to 'application/json'. The main area displays the JSON response in 'Pretty' format:

```
1 {
2   "data": [
3     {
4       "Flight_number": "DF2753",
5       "Airline_code": "THY",
6       "Weekdays": "PT CR CM"
7     },
8     {
9       "Flight_number": "DF2755",
10      "Airline_code": "PC",
11      "Weekdays": "PT CR CM"
12    },
13    {
14      "Flight_number": "EN4267",
15      "Airline_code": "PC",
16      "Weekdays": "CT PZ"
17    },
18    {
19      "Flight_number": "FD5610",
20      "Airline_code": "PC",
21      "Weekdays": "PT SL CR PR CM"
22    },
23    {
24      "Flight_number": "GT4638",
25      "Airline_code": "VIA"
25    }
  ]
}
```

At the bottom of the screen, there are icons for 'Bootcamp', 'Runner', and 'Trash'.

GET http://localhost:3000/api/flights/THY

Send Cookies

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (10) Test Results Status: 200 OK Time: 531 ms Size: 584 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2     "data": [
3         {
4             "Flight_number": "DF2753",
5             "Airline_code": "THY",
6             "Weekdays": "PT CR CM"
7         },
8         {
9             "Flight_number": "TK2752",
10            "Airline_code": "THY",
11            "Weekdays": "PR CM CT PZ"
12        }
13    ]
14 }
```

GET http://localhost:3000/api/airlines

Send Cookies

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (10) Test Results Status: 200 OK Time: 570 ms Size: 952 B Save Response

Pretty Raw Preview Visualize JSON

```
4     {
5         "Airline_id": 47982417,
6         "Airline_name": "Sunexpress",
7         "Address": "Antalya,Türkiye",
8         "Airline_code": "XQ",
9         "Number_of_airplanes": 2
10    },
11    {
12        "Airline_id": 54795173,
13        "Airline_name": "Pegasus",
14        "Address": "İstanbul,Türkiye",
15        "Airline_code": "PC",
16        "Number_of_airplanes": 2
17    },
18    {
19        "Airline_id": 179245063,
20        "Airline_name": "Anadolu Jet",
21        "Address": "Ankara,Türkiye",
22        "Airline_code": "TK",
23        "Number_of_airplanes": 2
24    },
25    {
26        "Airline_id": 249871359,
27        "Airline_name": "Turkish Airlines",
28        "Address": "İstanbul,Türkiye",
29    }
```

GET http://localhost:3000/api/airports

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (10) Test Results Status: 200 OK Time: 563 ms Size: 1.33 KB Save Response

Pretty Raw Preview Visualize JSON

```
2   "data": [
3     {
4       "Airport_code": "ADB",
5       "Airport_name": "Adnan Menderes Airport",
6       "City": "İzmir",
7       "State": "Public"
8     },
9     {
10    "Airport_code": "DLM",
11    "Airport_name": "Dalaman Airport",
12    "City": "Muğla",
13    "State": "Public"
14  },
15  {
16    "Airport_code": "ERZ",
17    "Airport_name": "Erzurum Airport",
18    "City": "Erzurum",
19    "State": "Public"
20  },
21  {
22    "Airport_code": "ESB",
23    "Airport_name": "Esenboğa Airport",
24    "City": "Ankara",
25    "State": "Public"
26  },
27 ]
```

GET http://localhost:3000/api/airports/ADB

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (10) Test Results Status: 200 OK Time: 537 ms Size: 537 B Save Response

Pretty Raw Preview Visualize JSON

```
1   "data": [
2     {
3       "Airport_code": "ADB",
4       "Airport_name": "Adnan Menderes Airport",
5       "City": "İzmir",
6       "State": "Public"
7     }
8   ],
9 ]
```

GET http://localhost:3000/api/leg-instances

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (10) Test Results Status: 200 OK Time: 576 ms Size: 3.77 KB Save Response

Pretty Raw Preview Visualize JSON

```
4     "Flight_number": "DF2753",
5     "Leg_number": 901,
6     "Flight_date": "2021-02-14T21:00:00.000Z",
7     "Departure_airport_code": "SAW",
8     "Arrival_airport_code": "ADB",
9     "Departure_time": "2021-02-15T11:30:00.000Z",
10    "Arrive_time": "2021-02-15T13:00:00.000Z",
11    "Airplane_id": "JA8089",
12    "Number_of_available_seats": 297
13  },
14  {
15    "Flight_number": "DF2753",
16    "Leg_number": 901,
17    "Flight_date": "2021-02-15T21:00:00.000Z",
18    "Departure_airport_code": "SAW",
19    "Arrival_airport_code": "ADB",
20    "Departure_time": "2021-02-16T11:30:00.000Z",
21    "Arrive_time": "2021-02-16T13:00:00.000Z",
22    "Airplane_id": "JA8089",
23    "Number_of_available_seats": 300
24  },
25  {
26    "Flight_number": "DF2753",
27    "Leg_number": 901,
```

Bootcamp Runner Trash