DATABASE MANAGEMENT
FINAL PROJECT

GROUP-7

MUSTAFA ARAS
05170000098

BUSE ARGUS
05180000064

SERCAN BAYRAM
05170000038

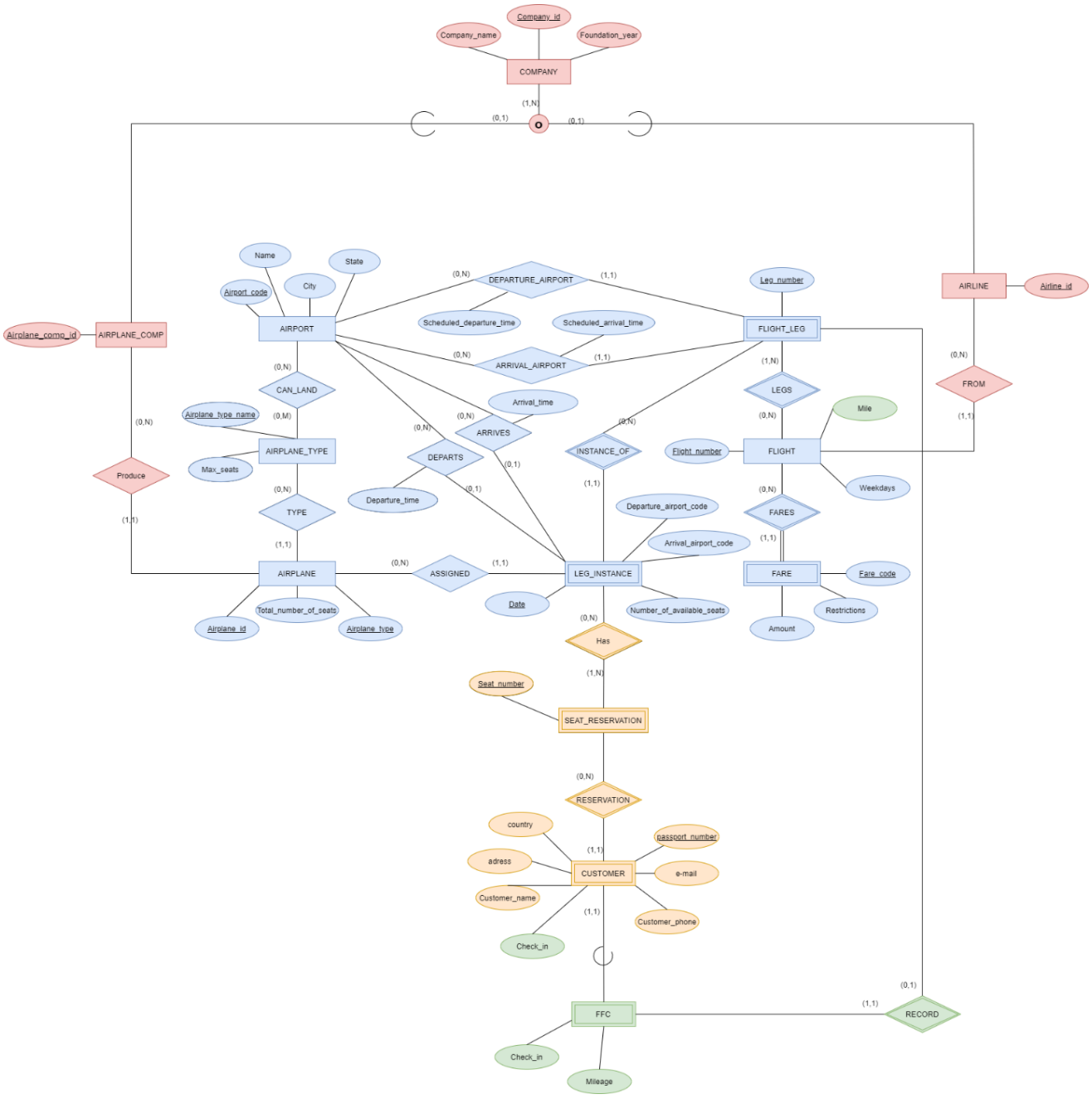
****Grupta bulunan 4.kişi olan Ahmet Öcal hiçbir çalışmaya katılmamıştır**

İçindekiler

Platform

Final projesi için kullanılan platform Microsoft SQL Server 2019'dur.

Tasarım Değişikliği



Draw.io dosyası:

https://drive.google.com/file/d/1cRnP1C4A6vqlGo1rHI21tWIjkCTNK3TT/view?usp=sharing

Vizede tasarlanan FFC varlığı ve ona bağlı olarak gerçekleştirilen mileage atamasının hatalı olduğu final için yapılan tasarımlarda belirlenmiştir. Yenilenen tasarımın diyagramı drive linki ve fotoğraf olarak üst tarafta belirtilmiştir.

Daha önce Check_in işlemi için tanımlanan ONLINE ve PHYSICALLY varlıkları Customer varlığında bulunan Check_in adlı bir attribute'e indirgenmiştir. Bu sayede gereksiz olarak kullanılan ONLINE ve PHYSICALLY varlıkları kaldırılmıştır.

Daha önce REGULAR ve COMMON olarak uçuş sıklığının belirtilmesi için oluşturulan varlıklar FFC üzerinden bir attribute'e indirgenmiş ve bir trigger yardımıyla ödül sisteminin oluşturulmasında kullanılmıştır. Bu sayede gereksiz olarak kullanılan REGULAR ve COMMON varlıkları kaldırılmıştır.

FFC varlığı ise CUSTOMER ve FLIGHT_LEG varlıklarına Foreign Key'ler yardımıyla bağlanmıştır. Bunlara ek olarak FFC varlığına bir trigger yardımıyla CUSTOMER varlığının her yaptığı Check_in işleminde eleman atanmış, aynı zamanda farklı bir trigger yardımıyla ise mileage bilgileri FLIGHT_LEG varlığından alınıp atanan elemanlara Mileage bilgisi olarak eklenmiştir.

1-) Veri tabanı ve ilişkisel modeli oluşturmak için uygun SQL komutları yazıldı.

COMPANY varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE COMPANY (
Company_id int IDENTITY(1,1),
Company_name VARCHAR(45),
Foundation_year DATE ,
PRIMARY KEY(Company_id)
);
```

AIRPLANE_COMP varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE AIRPLANE_COMP (
Airplane_comp_id int IDENTITY(1,1),
Company_id int NOT NULL,
Airplane_company_name VARCHAR(45),
Foundation_year DATE,
PRIMARY KEY (Airplane_comp_id),
FOREIGN KEY (Company_id) REFERENCES COMPANY
);
```

AIRLINE varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE AIRLINE (
Airline_id int IDENTITY(1,1),
Company_id int NOT NULL,
Airline_company_name VARCHAR(45),
Foundation_year DATE,
PRIMARY KEY (Airline_id),
FOREIGN KEY (Company_id) REFERENCES COMPANY
);
```

AIRPLANE_TYPE varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE AIRPLANE_TYPE (
Airplane_type_name VARCHAR(30) NOT NULL,
Max_seats int,
CONSTRAINT CHK_Airplane_TYPE_Seats CHECK (AIRPLANE_TYPE.Max_seats>1),
PRIMARY KEY (Airplane_type_name)
);
```

AIRPLANE varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE AIRPLANE (
Airplane_id int IDENTITY(1,1),
Airplane_comp_id int NOT NULL,
Total_number_of_seats int,
Airplane_type VARCHAR(30) NOT NULL,

CONSTRAINT CHK_AirplaneSeats CHECK (AIRPLANE.Total_number_of_seats > 1),

PRIMARY KEY (Airplane_id),
FOREIGN KEY (Airplane_comp_id) REFERENCES AIRPLANE_COMP,
FOREIGN KEY (Airplane_type) REFERENCES AIRPLANE_TYPE
);
```

AIRPORT varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE AIRPORT (
Airport_code int IDENTITY(1,1),
AirportName VARCHAR(45),
City VARCHAR(30),
AState VARCHAR(30),
PRIMARY KEY (Airport_code)
);
```

CAN_LAND ilişkisi veri tabanına tablo olarak eklendi.

```
CREATE TABLE CAN_LAND (
Airplane_type VARCHAR(30) NOT NULL,
Airport_code int NOT NULL,
PRIMARY KEY (Airplane_type, Airport_code),
FOREIGN KEY (Airplane_type) REFERENCES AIRPLANE_TYPE,
FOREIGN KEY (Airport_code) REFERENCES AIRPORT
);
```

FLIGHT varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE FLIGHT (
Flight_number int NOT NULL,
Weekdays VARCHAR(15),
Airline_id int NOT NULL,
PRIMARY KEY (Flight_number),
FOREIGN KEY (Airline_id) REFERENCES AIRLINE
);
```

FARE varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE FARE (
Flight_number int NOT NULL,
Fare_code VARCHAR(1) NOT NULL,
Amount int,
Restrictions VARCHAR(15),
PRIMARY KEY (Flight_number, Fare_code),
FOREIGN KEY (Flight_number) REFERENCES FLIGHT
);
```

FLIGHT_LEG varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE FLIGHT_LEG (
Flight_number int NOT NULL,
Leg_number int NOT NULL,
Departure_airport_code int,
Arrival_airport_code int,
Scheduled_departure_time DATETIME,
Scheduled_arrival_time DATETIME,
Mile int,
PRIMARY KEY (Flight_number, Leg_number),
FOREIGN KEY (Flight_number) REFERENCES FLIGHT,
FOREIGN KEY (Departure_airport_code) REFERENCES AIRPORT,
FOREIGN KEY (Arrival_airport_code) REFERENCES AIRPORT
);
```

LEG_INSTANCE varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE LEG_INSTANCE (
Flight_number int NOT NULL,
Leg_number int NOT NULL,
LDate DATETIME NOT NULL,
Number_of_available_seats int,
Airplane_id int NOT NULL,
Departure_airport_code int,
Arrival_airport_code int,
Departure_time DATETIME,
Arrival_time DATETIME,

CONSTRAINT CHK_CurrentTime CHECK(GETDATE()<LDATE),
CONSTRAINT CHK_Time CHECK (Departure_time<Arrival_time),
CONSTRAINT CHK_Seats CHECK (LEG_INSTANCE.Number_of_available_seats>1),
CONSTRAINT CHK_Code CHECK
(LEG_INSTANCE.Departure_airport_code!=LEG_INSTANCE.Arrival_airport_code),

PRIMARY KEY (Flight_number, Leg_number, LDate),
FOREIGN KEY (Flight_number, Leg_number) REFERENCES FLIGHT_LEG(Flight_number, Leg_number),
FOREIGN KEY (Airplane_id) REFERENCES AIRPLANE
);
```

SEAT_RESERVATION varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE SEAT_RESERVATION (
Flight_number INTEGER NOT NULL,
Leg_number INTEGER NOT NULL,
LDate DATETIME NOT NULL,
Seat_number INTEGER NOT NULL,
PRIMARY KEY (Flight_number, Leg_number, LDate, Seat_number),
FOREIGN KEY (Flight_number, Leg_number, LDate) REFERENCES LEG_INSTANCE
);
```

CUSTOMER varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE CUSTOMER (
Passport_number VARCHAR(11) NOT NULL,
Customer_phone VARCHAR(11),
Customer_name VARCHAR(15),
Email VARCHAR(30),
CAddress VARCHAR(45),
Country VARCHAR(30),
Flight_number int NOT NULL,
Leg_number int NOT NULL,
LDate DATETIME NOT NULL,
Seat_number int NOT NULL,
Check_in bit DEFAULT 0,

CONSTRAINT CHK_Passport_number CHECK(Passport_number > 0),

PRIMARY KEY (Passport_number),
FOREIGN KEY (Flight_number, Leg_number, LDate, Seat_number) REFERENCES SEAT_RESERVATION ,
);
```

FFC varlığı veri tabanına tablo olarak eklendi.

```
CREATE TABLE FFC (
Mileage int Not null,
Passport_number VARCHAR(11)  NOT NULL,
Flight_number INTEGER NOT NULL,
Leg_number INTEGER NOT NULL,
Seat_number INTEGER NOT NULL,
LDate DATETIME NOT NULL,
Check_in bit DEFAULT 0,
PRIMARY KEY (Passport_number, Flight_number, Leg_number, Seat_number, LDate),
FOREIGN KEY (Passport_number) REFERENCES CUSTOMER ,
FOREIGN KEY (Flight_number, Leg_number, LDate) REFERENCES LEG_INSTANCE
);
```

2-) Yeni oluşturulan veri tabanı, SQL komutlarıyla dolduruldu.

COMPANY tablosu INSERT INTO komutuyla dolduruldu.

```
INSERT INTO COMPANY VALUES('Air France KLM', '2004-05-05');
INSERT INTO COMPANY VALUES('Turkish Airlines', '1933-05-20');
INSERT INTO COMPANY VALUES('American Airlines Group', '2013-12-09');
```

AIRPLANE_COMP tablosu INSERT INTO komutuyla dolduruldu.

```
INSERT INTO AIRPLANE_COMP VALUES(1, 'Safran', '2005');
INSERT INTO AIRPLANE_COMP VALUES(3, 'Boeing', '1916-07-15');
INSERT INTO AIRPLANE_COMP VALUES(3, 'Lockheed Martin', '1995-03-15');
```

AIRLINE tablosu INSERT INTO komutuyla dolduruldu.

```
INSERT INTO AIRLINE VALUES(1, 'Transavia France', '2006');
INSERT INTO AIRLINE VALUES(2, 'SunExpress', '1989');
INSERT INTO AIRLINE VALUES(2, 'AnadoluJet', '2008-04-23');
```

AIRPLANE_TYPE tablosu INSERT INTO komutuyla dolduruldu.

```
INSERT INTO AIRPLANE_TYPE VALUES('AN-22', 290);
INSERT INTO AIRPLANE_TYPE VALUES('Boeing-737 Max 8', 210);
INSERT INTO AIRPLANE_TYPE VALUES('Boeing-787-9 DreamLiner', 296);
```

AIRPLANE tablosu INSERT INTO komutuyla dolduruldu.

```
INSERT INTO AIRPLANE VALUES(1, 290, 'AN-22');
INSERT INTO AIRPLANE VALUES(1, 290, 'AN-22');
INSERT INTO AIRPLANE VALUES(2, 210, 'Boeing-737 Max 8');
INSERT INTO AIRPLANE VALUES(2, 296, 'Boeing-787-9 DreamLiner');
```

AIRPORT tablosu INSERT INTO komutuyla dolduruldu.

```
INSERT INTO AIRPORT VALUES('Erzurum Airport', 'Erzurum', 'Turkey');
INSERT INTO AIRPORT VALUES('İzmir Adnan Menderes Airport', 'İzmir', 'Turkey');
INSERT INTO AIRPORT VALUES('Bursa Yenişehir Airport', 'Bursa', 'Turkey');
INSERT INTO AIRPORT VALUES('Ankara Esenboğa Airport', 'Ankara', 'Turkey');
INSERT INTO AIRPORT VALUES('İstanbul Sabiha Gökçen Airport', 'İstanbul', 'Turkey');
```

CAN_LAND tablosu INSERT INTO komutuyla dolduruldu.

```
INSERT INTO CAN_LAND VALUES('AN-22', 1);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 1);
INSERT INTO CAN_LAND VALUES('AN-22', 2);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 2);
INSERT INTO CAN_LAND VALUES('AN-22', 3);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 3);
INSERT INTO CAN_LAND VALUES('AN-22', 4);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 4);
INSERT INTO CAN_LAND VALUES('AN-22', 5);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 5);
INSERT INTO CAN_LAND VALUES('Boeing-787-9 DreamLiner', 5);
```

FLIGHT tablosu INSERT INTO komutuyla doldurulduldu.

```
INSERT INTO FLIGHT VALUES(100, 'Tuesday', 1);
INSERT INTO FLIGHT VALUES(101, 'Tuesday', 2);
INSERT INTO FLIGHT VALUES(102, 'Tuesday', 3);
INSERT INTO FLIGHT VALUES(103, 'Friday', 1);
INSERT INTO FLIGHT VALUES(104, 'Saturday', 2);
INSERT INTO FLIGHT VALUES(105, 'Sunday', 3);
```

FARE tablosu INSERT INTO komutuyla doldurulduldu.

```
INSERT INTO FARE VALUES(100, 'F', 100, 'Alcohol');
INSERT INTO FARE VALUES(100, 'J', 80 , 'Alcohol-free');

INSERT INTO FARE VALUES(101, 'F', 60, 'Alcohol');
INSERT INTO FARE VALUES(101, 'J' , 40, 'Alcohol-free');

INSERT INTO FARE VALUES(102, 'F', 120, 'Alcohol');
INSERT INTO FARE VALUES(102, 'J' , 70, 'Alcohol-free');

INSERT INTO FARE VALUES(103, 'F', 160, 'Alcohol');
INSERT INTO FARE VALUES(103, 'J' , 120, 'Alcohol-free');

INSERT INTO FARE VALUES(104, 'F', 70, 'Alcohol');
INSERT INTO FARE VALUES(104, 'J' , 50, 'Alcohol-free');

INSERT INTO FARE VALUES(105, 'F', 160, 'Alcohol');
INSERT INTO FARE VALUES(105, 'J' , 120, 'Alcohol-free');
```

FLIGHT_LEG tablosu INSERT INTO komutuyla doldurulduldu.

```
INSERT INTO FLIGHT_LEG VALUES(100, 1, 4, 5, '2021-02-16 07:30', '2021-02-16 08:35', 70);

INSERT INTO FLIGHT_LEG VALUES(101, 1, 2, 5, '2021-02-16 04:40', '2021-02-16 05:45', 80);

INSERT INTO FLIGHT_LEG VALUES(102, 1, 1, 3, '2021-02-16 11:45', '2021-02-16 13:40', 60);

INSERT INTO FLIGHT_LEG VALUES(103, 1, 1, 4, '2021-02-19 13:20', '2021-02-19 14:25', 90);
INSERT INTO FLIGHT_LEG VALUES(103, 2, 4, 5, '2021-02-19 15:25', '2021-02-19 16:50', 100);

INSERT INTO FLIGHT_LEG VALUES(104, 1, 4, 3, '2021-02-20 15:30', '2021-02-20 16:30', 40);

INSERT INTO FLIGHT_LEG VALUES(105, 1, 5, 4, '2021-02-21 17:00', '2021-02-21 18:15', 50);
INSERT INTO FLIGHT_LEG VALUES(105, 2, 4, 1, '2021-02-21 19:00', '2021-02-21 20:05', 20);
```

LEG_INSTANCE tablosu INSERT INTO komutuyla doldurulduldu.

```
INSERT INTO LEG_INSTANCE VALUES(100, 1, '2021-02-16', 290, 1, 4, 5, '2021-02-16 07:30',
'2021-02-16 08:35');
INSERT INTO LEG_INSTANCE VALUES(101, 1, '2021-02-16', 290, 2, 2, 5, '2021-02-16 04:40',
'2021-02-16 05:45');
INSERT INTO LEG_INSTANCE VALUES(102, 1, '2021-02-16', 210, 3, 1, 3, '2021-02-16 11:45',
'2021-02-16 13:40');
```

```
INSERT INTO LEG_INSTANCE VALUES(103, 1, '2021-02-19', 296, 4, 1, 4, '2021-02-19 13:20',
'2021-02-19 14:25');
INSERT INTO LEG_INSTANCE VALUES(103, 2, '2021-02-19', 296, 4, 4, 5, '2021-02-19 15:25',
'2021-02-19 16:50');

INSERT INTO LEG_INSTANCE VALUES(104, 1, '2021-02-20', 290, 1, 4, 3, '2021-02-20 15:30',
'2021-02-20 16:30');

INSERT INTO LEG_INSTANCE VALUES(105, 1, '2021-02-21', 210, 3, 5, 4, '2021-02-21 17:00',
'2021-02-21 18:15');
INSERT INTO LEG_INSTANCE VALUES(105, 2, '2021-02-21', 210, 3, 4, 1, '2021-02-21 19:00',
'2021-02-21 20:05');
```

SEAT_RESERVATION tablosu INSERT INTO komutuyla doldurulduldu.

```
INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 10);
INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 11);
INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 12);

INSERT INTO SEAT_RESERVATION VALUES(101, 1, '2021-02-16', 13);
INSERT INTO SEAT_RESERVATION VALUES(101, 1, '2021-02-16', 14);

INSERT INTO SEAT_RESERVATION VALUES(102, 1, '2021-02-16', 15);

INSERT INTO SEAT_RESERVATION VALUES(103, 1, '2021-02-19', 16);
INSERT INTO SEAT_RESERVATION VALUES(103, 1, '2021-02-19', 17);
INSERT INTO SEAT_RESERVATION VALUES(103, 1, '2021-02-19', 18);
INSERT INTO SEAT_RESERVATION VALUES(103, 2, '2021-02-19', 19);

INSERT INTO SEAT_RESERVATION VALUES(104, 1, '2021-02-20', 20);

INSERT INTO SEAT_RESERVATION VALUES(105, 1, '2021-02-21', 10);
INSERT INTO SEAT_RESERVATION VALUES(105, 1, '2021-02-21', 11);
INSERT INTO SEAT_RESERVATION VALUES(105, 2, '2021-02-21', 12);
```

CUSTOMER tablosu INSERT INTO komutuyla doldurulduldu.

```
INSERT INTO CUSTOMER VALUES(1234567, 55555555555, 'ali', 'ali@gmail.com', 'Ali adres',
'Ali country', 100, 1, '2021-02-16', 10, 1);
INSERT INTO CUSTOMER VALUES(2345678, 55555555555, 'veli','veli@gmail.com', 'Veli adres',
'veli country', 100, 1, '2021-02-16', 11, 1);
INSERT INTO CUSTOMER VALUES(3456789, 55555555555, 'ayşe','ayşe@gmail.com', 'ayşe adres',
'ayşe country', 100, 1, '2021-02-16', 12, 1);
INSERT INTO CUSTOMER VALUES(4567890, 55555555555, 'ahmet','ahmet@gmail.com','ahmet adres',
'ahmet country',101, 1, '2021-02-16', 13, 1);
INSERT INTO CUSTOMER VALUES(5678901, 55555555555, 'zeki','zeki@gmail.com', 'zeki adres',
'zeki country', 101, 1, '2021-02-16', 14, 1);
INSERT INTO CUSTOMER VALUES(6789012, 55555555555, 'erdem','erdem@gmail.com','erdem adres',
'erdem country',102, 1, '2021-02-16', 15, 1);
INSERT INTO CUSTOMER VALUES(7890123, 55555555555, 'mehmet','mehmet@gmail.com', 'mehmet
adres', 'mehmet country',103, 1, '2021-02-19', 16, 1);
INSERT INTO CUSTOMER VALUES(1231235, 55555555555, 'mustafa','mustafa@gmail.com','mustafa
adres', 'mustafa country',103, 1, '2021-02-19', 17, 1);
INSERT INTO CUSTOMER VALUES(7777777, 55555555555, 'serdar','serdar@gmail.com', 'serdar
adres', 'serdar country', 103, 1, '2021-02-19', 18, 1 );
INSERT INTO CUSTOMER VALUES(1651255, 55555555555, 'buse','buse@gmail.com', 'buse adres',
'buse country',103, 2, '2021-02-19', 19, 1);
```

```
INSERT INTO CUSTOMER VALUES(9012345, 55555555555, 'deniz','deniz@gmail.com', 'deniz
adres', 'deniz country',104, 1, '2021-02-20', 20, 1);
INSERT INTO CUSTOMER VALUES(8901234, 55555555555, 'oğuz','oğuz@gmail.com', 'oğuz adres',
'oğuz country',105, 1, '2021-02-21', 10, 1);
INSERT INTO CUSTOMER VALUES(7512355, 55555555555, 'selda','selda@gmail.com', 'selda
adres', 'selda country',105, 1, '2021-02-21', 11, 1);
INSERT INTO CUSTOMER VALUES(8676777, 55555555555, 'sonat','sonat@gmail.com', 'sonat
adres', 'sonat country',105, 2, '2021-02-21', 12, 1);
```

3-) 9 tane anlamlı trigger oluşturuldu.

3.1-) CustomerUpdate Trigger'ı

Check-in yapmış bir yolcunun artık uçuşla ilgili bilgilerinin değiştirilmemesini sağlayan bir trigger oluşturuldu, yolcunun uçuş dışındaki isim, e-mail, telefon numarası gibi bilgilerinin değiştirilmesine izin verilmiştir.

```
CREATE TRIGGER CustomerUpdate ON CUSTOMER
AFTER UPDATE
AS
BEGIN
    IF (EXISTS(SELECT * FROM inserted, deleted WHERE deleted.Check_in = 1 AND
                            (deleted.Flight_number != inserted.Flight_number OR
                            deleted.LDate != inserted.LDate OR
                            deleted.Leg_number != inserted.Leg_number OR
                            deleted.Seat_number != inserted.Seat_number OR
                            deleted.Country != inserted.Country OR
                            deleted.Passport_number != inserted.Passport_number OR
                            deleted.CAddress != inserted.CAddress OR
                            deleted.Check_in != inserted.Check_in )))
    BEGIN
        RAISERROR ('Check in yapmış yolcunun uçuş bilgileri değiştirilemez!',1,1)
        ROLLBACK TRANSACTION
    END
END
```

3.2-) CHK_MaxSeat Trigger

Yeni bir uçak verisi eklenmek istediğinde koltuk sayısı, girilen uçak tipinin koltuk sayısı ile eşleşmediği zaman bu ekleme işlemini engelleyen bir trigger oluşturuldu.

```
CREATE TRIGGER CHK_MaxSeat ON AIRPLANE
AFTER INSERT
AS
BEGIN
	SET NOCOUNT ON;

	IF EXISTS(SELECT * FROM inserted, AIRPLANE_TYPE WHERE
inserted.Total_number_of_seats = AIRPLANE_TYPE.Max_seats)
	BEGIN
		WAITFOR DELAY '00:00:00'
	END
	ELSE
	BEGIN
		RAISERROR ('Üretilen uçak tipinin koltuk sayısıyla koltuk sayınız
eşleşmediğinden üretilemez!',1,1)
	 ROLLBACK TRANSACTION
	END
END
```

3.3-) seatUpdate Trigger

Uçuş için yeni bir rezervasyon oluşturulduğunda uçuştaki boş koltuk sayısının azaltılmasını sağlayan bir trigger oluşturuldu.

```
CREATE TRIGGER seatUpdate ON Customer
AFTER INSERT
AS
BEGIN
	SET NOCOUNT ON;

	UPDATE LEG_INSTANCE
	SET LEG_INSTANCE.Number_of_available_seats += -1
	From inserted
	WHERE LEG_INSTANCE.Flight_number = inserted.Flight_number AND
LEG_INSTANCE.Leg_number = inserted.Leg_number AND LEG_INSTANCE.LDate = inserted.LDate


END
```

3.4-) seatUpdateOnDELETE Trigger

Uçuş için bir rezervasyon iptal edildiğinde uçuştaki boş koltuk sayısının arttırılmasını sağlayan bir trigger oluşturuldu.

```
CREATE TRIGGER seatUpdateOnDELETE ON Customer
AFTER Delete
AS
BEGIN
	SET NOCOUNT ON;

	UPDATE LEG_INSTANCE
	SET LEG_INSTANCE.Number_of_available_seats += 1
	From deleted
	WHERE LEG_INSTANCE.Flight_number = deleted.Flight_number AND
LEG_INSTANCE.Leg_number = deleted.Leg_number AND LEG_INSTANCE.LDate = deleted.LDate
END
```

3.5-) mileUpdate Trigger

FFC olan müşteriler güncellendiğinde eğer check_in bilgisi değiştiyse ve bu değer 1 ise FFC tablosunda bulunan mileage bilgisini FLIGHT_LEG tablosundan çekilen Mile değerine göre güncelleyen trigger oluşturuldu.

```
CREATE TRIGGER mileUpdate ON FFC
AFTER UPDATE
AS
BEGIN

	SET NOCOUNT ON;

	DECLARE @check_in bit
	SELECT @check_in = Check_in from inserted

	DECLARE @oldCheck_in bit
	SELECT @oldCheck_in = Check_in from deleted

	IF(@check_in = 1 AND @oldCheck_in != @check_in)
	BEGIN
		UPDATE FFC
		SET FFC.Mileage += (SELECT DISTINCT Mile FROM FLIGHT_LEG, FFC
		WHERE FFC.Flight_number = FLIGHT_LEG.Flight_number AND FFC.Leg_number =
FLIGHT_LEG.Leg_number)
		FROM inserted
		INNER JOIN LEG_INSTANCE ON (inserted.Flight_number =
LEG_INSTANCE.Flight_number AND inserted.Leg_number = LEG_INSTANCE.Leg_number AND
inserted.LDate = LEG_INSTANCE.LDate)
		INNER JOIN FLIGHT_LEG ON (LEG_INSTANCE.Flight_number =
FLIGHT_LEG.Flight_number AND LEG_INSTANCE.Leg_number = FLIGHT_LEG.Leg_number)
		WHERE(FFC.Passport_number = inserted.Passport_number AND FFC.Flight_number =
inserted.Flight_number AND FFC.Leg_number = inserted.Leg_number AND FFC.Seat_number =
inserted.Seat_number AND FFC.LDate = inserted.LDate AND inserted.Check_in = FFC.Check_in)
	END
END
```

3.6-) mileUpdateWhenInsert Trigger

FFC varlığının yaptığı yeni uçuşlarda mil bilgisinin güncellenmesini sağlayan trigger oluşturuldu.

```
CREATE TRIGGER mileUpdateWhenInsert ON FFC
AFTER INSERT
AS
BEGIN
	SET NOCOUNT ON;


	UPDATE FFC
	SET FFC.Mileage += (SELECT DISTINCT Mile FROM FLIGHT_LEG, FFC
					WHERE FFC.Flight_number = FLIGHT_LEG.Flight_number AND
FFC.Leg_number = FLIGHT_LEG.Leg_number)
	From inserted
	WHERE FFC.Passport_number = inserted.Passport_number AND FFC.Flight_number =
inserted.Flight_number AND FFC.Leg_number = inserted.Leg_number AND FFC.Seat_number =
inserted.Seat_number AND FFC.LDate = inserted.LDate AND inserted.Check_in = FFC.Check_in
AND FFC.Check_in = 1

END
```

3.7-) custToFFC Trigger

Customer tablosuna insert ve update durumunda girilen veya güncellen yolcunun check_in değeri 0 değilse ve bu uçuş FFC tablosunda yoksa, FFC'ye mileage olarak flight_legden alınan toplam mile bilgisi ve uçuş bilgileri eklenir.

```
ALTER TRIGGER custToFFC ON CUSTOMER
AFTER INSERT,UPDATE
AS
BEGIN
SET NOCOUNT ON;
	IF NOT EXISTS(SELECT * from inserted where inserted.Check_in = 0)
	BEGIN
		INSERT INTO FFC
		SELECT (SELECT SUM(Mile)
				FROM FLIGHT_LEG, inserted
				WHERE FLIGHT_LEG.Flight_number = inserted.Flight_number And
FLIGHT_LEG.Leg_number = inserted.Leg_number
				GROUP BY inserted.Passport_number) , inserted.Passport_number,
inserted.Flight_number, inserted.Leg_number, inserted.Seat_number, inserted.LDate,
inserted.Check_in, NULL
		FROM inserted
		WHERE NOT EXISTS(SELECT * from FFC WHERE FFC.Flight_number =
inserted.Flight_number AND FFC.Leg_number = inserted.Leg_number AND FFC.LDate =
inserted.LDate AND FFC.Seat_number = inserted.Seat_number AND FFC.Check_in =
inserted.Check_in AND FFC.Passport_number = inserted.Passport_number )
	END
END
```

3.8-) customerSegmentatiton Trigger

FFC'lerin mil üzerinden Premium,gold ve common olarak sınıflandırılmasını sağlayan bir trigger yazıldı.

```
CREATE TRIGGER customerSegmentatiton ON FFC
AFTER UPDATE
AS
BEGIN

	SET NOCOUNT ON;
	UPDATE FFC
	SET FFC.Segmentation = CASE
				WHEN  FFC.Mileage > 1000 THEN 'Premium Customer'
				WHEN  FFC.Mileage > 500 THEN 'Gold Customer'
				  ELSE 'Common Customer'
			END
END
```

3.9-) giveReward Trigger

FFC'lerin statütü değişimlerinde (Terfi alma, terfi düşme gibi) özel olarak karşılanması için bir trigger oluşturuldu.

```
CREATE TRIGGER giveReward ON FFC
AFTER UPDATE
AS
BEGIN
	SET NOCOUNT ON;

	IF EXISTS(SELECT * FROM inserted, deleted WHERE inserted.Passport_number =
deleted.Passport_number AND inserted.Segmentation != deleted.Segmentation)
	BEGIN
		SELECT inserted.Passport_number, inserted.Segmentation newSegmentation,
deleted.Segmentation oldSegmentation,
			CASE
				WHEN inserted.segmentation = 'Premium Customer' THEN
'Tebrikler Premium üye olmaya hak kazandınız. Kazandığınız ekstra avantajları öğrenmek
için internet sitemizi ziyaret ediniz.'
				WHEN inserted.segmentation = 'Gold Customer' THEN
'Tebrikler Gold üye olmaya hak kazandınız. Kazandığınız ekstra avantajları öğrenmek için
internet sitemizi ziyaret ediniz.'
				ELSE 'Gold üye olmaya çok yakınsınız bizi tercih etmeye
devam edin.'
			END AS 'Reward'
		FROM inserted,deleted
		WHERE inserted.Passport_number = deleted.Passport_number AND
inserted.Segmentation != deleted.Segmentation
		ORDER BY inserted.Passport_number ASC
	END
END
```

4-) 14 tane anlamlı check-constraint oluşturuldu.

4.1-) Fare_Amount check-constraint

Uçuş ücretinin 0'dan büyük olmasını sağlayan check-constraint oluşturuldu.

```
ALTER TABLE FARE
ADD CONSTRAINT Fare_Amount CHECK(Amount > 0);
```

4.2-) CHK_Fare_code check-constaint

Fare_code kısmına 'F' ve 'J' harflerinden başka bir harfin yazılmasını engelleyen check-constraint oluşturuldu.

```
ALTER TABLE FARE
ADD CONSTRAINT CHK_Fare_code CHECK(Fare_code = 'J' or Fare_code = 'F');
```

4.3-) CHK_Mileage check-constraint

FFC'de bulunan bir yolcunun mil değerinin 0'dan büyük olmasını sağlayan check-constraint oluşturuldu.

```
ALTER TABLE FFC
ADD CONSTRAINT CHK_Mileage CHECK(Mileage > 0);
```

4.4-) CHK_Flight_Leg_Airport_code check-constraint

Bir uçuş ayağında kalkış ve iniş havaalanlarının kodlarının aynı olmamasını sağlayan check-constraint oluşturuldu.

```
ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK_Flight_Leg_Airport_code
CHECK(FLIGHT_LEG.Departure_airport_code!=FLIGHT_LEG.Arrival_airport_code);
```

4.5-) CHK_Flight_Leg_Time check-constraint

Bir uçuş ayağında iniş saatinin kalkış saatinden önce olmamasını sağlayan check-constraint oluşturuldu.

```
ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK_Flight_Leg_Time CHECK(Scheduled_departure_time  <
Scheduled_arrival_time );
```

4.6-) CHK_Flight_LEG_Mileage check-constraint

Uçuş ayağında tanımlanan mil değerinin 0 olamamasını sağlayan check-constraint oluşturuldu.

```
ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK_Flight_LEG_Mileage CHECK(Mile > 0);
```

4.7-) CHK__Flight_LEG_CurrentTime check-constraint

Bir uçuşun kalkış saatinin rezervasyon yapılan saatten önce olmasını engelleyen bir check-constraint oluşturuldu.

```
ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK__Flight_LEG_CurrentTime CHECK(GETDATE() < Scheduled_departure_time);
```

4.8-) CHK_Airplane_TYPE_Seats check-constraint

AIRPLANE_TYPE tablosunun içinde bir uçak tipindeki maksimum koltuk sayısının 1'den fazla olmasını sağlayan check-constraint oluşturuldu.

```
CONSTRAINT CHK_Airplane_TYPE_Seats CHECK (AIRPLANE_TYPE.Max_seats>1),
```

4.9-) CHK_AirplaneSeats check-constraint

AIRPLANE tablosunun içinde bir uçaktaki toplam koltuk sayısının 1'den fazla olmasını sağlayan check-constraint yazıldı.

```
CONSTRAINT CHK_AirplaneSeats CHECK (AIRPLANE.Total_number_of_seats > 1)
```

4.10-) CHK_CurrentTime check-constraint

LEG_INSTANCE tablosunun içinde bir uçuş ayağının tarihinin günümüzden önce olmasını engelleyen bir check-constraint oluşturuldu.

```
CONSTRAINT CHK_CurrentTime CHECK(GETDATE()<LDATE)
```

4.11-) CHK_Time check-constraint

LEG_INSTANCE tablosunda iniş saatinin kalkış saatinden önce olmamasını sağlayan check-constraint oluşturuldu.

```
CONSTRAINT CHK_Time CHECK (Departure_time<Arrival_time)
```

4.12-) CHK_ Seats check-constraint

LEG_INSTANCE tablosunda koltuk sayısının 1'den büyük olmasını sağlayan check-constraint oluşturuldu.

```
CONSTRAINT CHK_Seats CHECK (LEG_INSTANCE.Number_of_available_seats>1)
```

4.13-) CHK_Code check-constraint

LEG_INSTANCE tablosunda iniş havaalanı ile kalkış havaalanının kodlarının aynı olmamasını sağlayan bir check-constraint oluşturuldu.

```
CONSTRAINT CHK_Code CHECK
(LEG_INSTANCE.Departure_airport_code!=LEG_INSTANCE.Arrival_airport_code)
```

4.14-) CHK_ Passport_number check-constraint

CUSTOMER tablosunda pasaport numarasının 0'dan büyük olmasını sağlayan bir check-constraint oluşturuldu.

```
CONSTRAINT CHK_Passport_number CHECK(Passport_number > 0),
```

5-) Aşağıda istenen SQL ifadeleri yazıldı.

5.a. 5 farklı tablo için INSERT,UPDATE ve DELETE ifadeleri yazıldı.

Pasaport numarası '86776777' olan bir müşterinin silinmesi

```
DELETE FROM CUSTOMER
WHERE Passport_number = '86776777'
```

Koltuk numarası '20' olan bir kaydın silinmesi

```
DELETE FROM SEAT_RESERVATION
WHERE Seat_number=20
```

Ülkesi 'selda country' olan bir FFC kaydının silinmesi

```
DELETE FROM FFC
WHERE FFC.Passport_number IN (SELECT FFC.Passport_number
                                                    FROM FFC,CUSTOMER
                                                    WHERE FFC.Passport_number =
Customer.Passport_number AND Customer.Country = 'selda country' )
```

Fiyatı 50 dolardan küçük olan kayıtların silinmesi

```
DELETE FROM FARE
WHERE Amount < 50
```

Uçuş tarihi '2021-02-21 00:00:00.000' olan uçuş ayağının silinmesi

```
DELETE FROM LEG_INSTANCE
WHERE LEG_INSTANCE.LDate = '2021-02-21 00:00:00.000'
```

Bilgileri verilen yolcunun check-in bilgisinin güncellenmesi

```
UPDATE CUSTOMER
SET Check_in = 1
WHERE Passport_number = '7777777' AND Flight_number = 103 AND Leg_number = 1 AND LDate =
'2021-02-19'
```

FFC'lerin mil değerinin güncellenmesi

```
UPDATE FFC
SET Mileage += 500
```

Tüm uçuş ayaklarının mil değerlerinin güncellenmesi (Bundan sonra yapılacak uçuşlarda değerlendirilecektir)

```
UPDATE FLIGHT_LEG
SET Mile += 20
```

Belirli havayollarına inecek olan uçuşların fiyat bilgilerinin güncellenmesi

```
UPDATE FARE
SET FARE.Amount += 100
FROM Fare, AIRPORT, FLIGHT_LEG
WHERE Fare.Flight_number = FLIGHT_LEG.Flight_number AND FLIGHT_LEG.Arrival_airport_code =
1
```

Uçak şirketinin bağlı olduğu şirketin güncellenmesi

```
UPDATE AIRPLANE_COMP
SET Company_id = 2
FROM COMPANY, AIRPLANE_COMP
WHERE Airplane_comp_id = 2
```

5.b. Oluşturulan veritabanı için 10 tane SELECT deyimi yazıldı. 3 tanesinde minimum 2 tablo, 4 tanesinde minimum 3 tablo, 3 tanesinde de minimum 4 tablo kullanıldı.

```
Minimum 2 tablo kullanılanlar:
--------------------------------------------------------------------------------
```

Bir uçağın tipi, id'si ve maximum koltuk sayısı gösterilir

```
SELECT atype.Airplane_type_name, a.Airplane_id, atype.Max_seats
FROM AIRPLANE_TYPE AS atype, AIRPLANE AS A
WHERE atype.Airplane_type_name = a.Airplane_type
```

Bir uçağın inebileceği havaalanlarını gösterir.

```
SELECT a.Airport_code, a.AirportName, cl.Airplane_type
FROM CAN_LAND AS cl, AIRPORT as a
WHERE cl.Airport_code = a.Airport_code
ORDER BY a.Airport_code
```

Uçuşun kalkış zamanı ve müsait koltuk sayısını gösterir.

```
SELECT fl.Flight_number, li.Number_of_available_seats, li.Departure_time
FROM FLIGHT_LEG fl, LEG_INSTANCE li
WHERE fl.Leg_number = li.Leg_number AND fl.Flight_number = li.Flight_number
ORDER BY li.Departure_time AS
```

Minimum 3 tablo kullanılanlar:

Bir müşterinin check-in bilgisi ve uçuş bilgileri gösterilir.

```
SELECT c.Passport_number, sr.Flight_number, c.Seat_number, fl.Scheduled_arrival_time,
CASE
       WHEN c.Check_in = 0 THEN 'not checked'
       WHEN c.Check_in = 1 THEN 'checked'
       ELSE 'unknown'
END AS 'check-in'
FROM CUSTOMER AS c, SEAT_RESERVATION  AS sr, FLIGHT_LEG AS fl
WHERE c.Flight_number = sr.Flight_number AND c.LDate = sr.LDate AND c.Leg_number =
sr.Leg_number AND c.Seat_number = sr.Seat_number
AND c.Leg_number = fl.Leg_number AND c.Flight_number = fl.Flight_number
```

En erken uçuşu olan 5 yolcu gösterilir.

```
SELECT TOP 5 c.Customer_name, li.Departure_airport_code, li.Departure_time
FROM CUSTOMER AS c, SEAT_RESERVATION AS sr, LEG_INSTANCE AS li
WHERE c.Flight_number = sr.Flight_number AND c.LDate = sr.LDate AND c.Leg_number =
sr.Leg_number AND c.Seat_number = sr.Seat_number
AND sr.Flight_number = li.Flight_number AND sr.Leg_number = li.Leg_number AND sr.LDate =
li.LDate
ORDER BY li.Departure_time ASC
```

Bir uçuşu organize eden havayolu şirketiyle, u şirketin bağlı olduğu şirketi gösterilir.

```
SELECT FLIGHT.Flight_number,  AIRLINE.Airline_company_name AIRLINE,Company.Company_name
COMPANY
FROM COMPANY, AIRLINE, FLIGHT
WHERE COMPANY.Company_id = AIRLINE.Company_id AND AIRLINE.Airline_id = FLIGHT.Airline_id
ORDER BY FLIGHT.Flight_number ASC
```

Bir Airline şirketinin organize ettiği uçuşlar ve uçuşla ilgili bazı bilgiler gösterilir.

```
SELECT DISTINCT AIRLINE.Airline_company_name,FLIGHT.Flight_number, FARE.Amount,
FARE.Restrictions
FROM FARE, FLIGHT, AIRLINE
WHERE AIRLINE.Airline_id = FLIGHT.Airline_id AND FLIGHT.Flight_number = FARE.Flight_number
```

Minimum 4 tablo kullanılanlar:

Bir uçuşta check-in yapmış yolcular gösterilir.

```
SELECT FLIGHT.Flight_number, CUSTOMER.Passport_number, CUSTOMER.Customer_name
from CUSTOMER, FLIGHT, FLIGHT_LEG, LEG_INSTANCE
WHERE CUSTOMER.Flight_number = LEG_INSTANCE.Flight_number AND CUSTOMER.Leg_number =
LEG_INSTANCE.Leg_number and CUSTOMER.LDate = LEG_INSTANCE.LDate AND CUSTOMER.Check_in = 1
AND LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND LEG_INSTANCE.Leg_number =
FLIGHT_LEG.Leg_number
AND FLIGHT.Flight_number = FLIGHT_LEG.Flight_number
ORDER BY FLIGHT.Flight_number ASC
```

Bir uçuşta dolu olan koltuklar ve uçuşun yapıldığı uçak tipi gösterilir.

```
SELECT LEG_INSTANCE.Flight_number, SEAT_RESERVATION.Seat_number
,AIRPLANE_TYPE.Airplane_type_name
FROM SEAT_RESERVATION, LEG_INSTANCE, AIRPLANE_TYPE, AIRPLANE
WHERE SEAT_RESERVATION.Flight_number = LEG_INSTANCE.Flight_number AND
SEAT_RESERVATION.Leg_number = LEG_INSTANCE.Leg_number AND SEAT_RESERVATION.LDate =
LEG_INSTANCE.LDate
AND LEG_INSTANCE.Airplane_id = AIRPLANE.Airplane_id
AND AIRPLANE.Airplane_type = AIRPLANE_TYPE.Airplane_type_name
```

Bir uçuşun hangi gün yapıldığı, hangi havaalanına iniş yaptığı ve uçak tipi gösterilir.

```
SELECT FLIGHT.Flight_number, FLIGHT.Weekdays, AIRPORT.AirportName Arrival_Airport,
AIRPLANE.Airplane_type
FROM FLIGHT, FLIGHT_LEG, LEG_INSTANCE, AIRPORT, AIRPLANE
WHERE FLIGHT.Flight_number = FLIGHT_LEG.Flight_number
AND FLIGHT_LEG.Flight_number = LEG_INSTANCE.Flight_number AND FLIGHT_LEG.Leg_number =
LEG_INSTANCE.Leg_number
AND LEG_INSTANCE.Arrival_airport_code = AIRPORT.Airport_code AND LEG_INSTANCE.Airplane_id
= AIRPLANE.Airplane_id
```

5.c. 4 tane NESTED/CORRELATED NESTED QUERY ifadesi yazıldı.

Fiyatı en yüksek olan uçuş/uçuşların uçuş bilgileri gösterildi.

```
SELECT
FARE.Amount,FARE.Fare_code,FARE.Flight_number,FARE.Restrictions,LEG_INSTANCE.Airplane_id,L
EG_INSTANCE.Arrival_airport_code,LEG_INSTANCE.Departure_airport_code,LEG_INSTANCE.Number_o
f_available_seats,LEG_INSTANCE.LDate,LEG_INSTANCE.Arrival_time,LEG_INSTANCE.Departure_time
FROM FARE,LEG_INSTANCE
WHERE FARE.Flight_number=LEG_INSTANCE.Flight_number AND FARE.Amount IN
                (SELECT DISTINCT MAX(FARE.Amount)
                 FROM FARE )
ORDER BY FARE.Amount ASC
```

102 numaralı uçuşun fiyat ve havaalanı bilgileri gösterildi.

```
SELECT
FLIGHT_LEG.Flight_number,FLIGHT_LEG.Departure_airport_code,FLIGHT_LEG.Arrival_airport_code
,FARE.Fare_code,FARE.Amount
FROM FARE,FLIGHT_LEG
WHERE FARE.Flight_number=FLIGHT_LEG.Flight_number AND
FARE.Flight_number in
(SELECT Flight_number FROM FLIGHT_LEG WHERE Flight_number=102)
```

Fiyatı ortalama fiyattan yüksek ve alkollü olan uçuşa ait fiyat ve bu müşterilerin bilgileri gösterildi.

```
SELECT Customer.Customer_name,
CUSTOMER.Email,CUSTOMER.Customer_phone,FARE.Amount,CUSTOMER.Flight_number
 FROM FARE,CUSTOMER,FLIGHT_LEG
 WHERE FARE.Amount >
                (SELECT AVG(FARE.Amount) FROM FARE )
                AND CUSTOMER.Flight_number=FLIGHT_LEG.Flight_number AND
CUSTOMER.Leg_number= FLIGHT_LEG.Leg_number AND FLIGHT_LEG.Flight_number=FARE.Flight_number
AND FARE.Restrictions='Alcohol'
```

Fiyatı 80 dolardan büyük olan uçuşlara ait bilgiler gösterildi.

```
SELECT  DISTINCT FARE.Amount,FARE.Fare_code,FARE.Restrictions,FARE.Flight_number, (Select
DISTINCT airport1.AirportName from AIRPORT AS airport1 where airport1.Airport_code =
LEG_INSTANCE.Departure_airport_code) Departure_Airport, (Select DISTINCT
airport2.AirportName from AIRPORT AS airport2 where airport2.Airport_code =
LEG_INSTANCE.Arrival_airport_code) Arrival_Airport, LEG_INSTANCE.Departure_time,
LEG_INSTANCE.Arrival_time, FLIGHT.Weekdays
FROM LEG_INSTANCE
INNER JOIN FLIGHT_LEG ON LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND
LEG_INSTANCE.Leg_number = FLIGHT_LEG.Leg_number
INNER JOIN FLIGHT ON FLIGHT_LEG.Flight_number = FLIGHT.Flight_number
INNER JOIN FARE ON FARE.Flight_number=LEG_INSTANCE.Flight_number
WHERE FARE.Amount IN (SELECT DISTINCT FARE.Amount FROM FARE
                      WHERE FARE.Amount>80)
ORDER BY FARE.Amount ASC
```

5.d. 2 tane EXISTS/NOT EXISTS ifadesi yazıldı.

Alkolsüz ve müsait koltuğa sahip uçuşa ait koltuk ve uçuş bilgileri gösterildi.

```
SELECT LEG_INSTANCE.Flight_number, LEG_INSTANCE.Leg_number,
LEG_INSTANCE.LDate,availableSeat.Seat_number, FARE.Restrictions
FROM LEG_INSTANCE, FARE, availableSeat
WHERE EXISTS(SELECT * FROM availableSeat WHERE LEG_INSTANCE.Flight_number =
availableSeat.Flight_number AND LEG_INSTANCE.Leg_number = availableSeat.Leg_number)
AND FARE.Restrictions = 'Alcohol-free' AND Fare.Flight_number = LEG_INSTANCE.Flight_number
```

Eğer Premium bir yolcu yoksa bunu belirten bir ifade yazıldı.

```
DECLARE @allCustomerCheckedin varchar(30)
SELECT @allCustomerCheckedin = 'Premium yolcu bulunmamaktadır'

SELECT DISTINCT @allCustomerCheckedin
FROM FFC
WHERE NOT EXISTS(Select * From FFC,LEG_INSTANCE Where FFC.Segmentation = 'Premium
Customer' AND LEG_INSTANCE.Flight_number = FFC.Flight_number AND LEG_INSTANCE.Leg_number =
FFC.Leg_number AND LEG_INSTANCE.LDate = FFC.LDate)
```

5.e. 3 tane LEFT/RIGHT/FULL OUTER JOIN ifadesi yazıldı.

Şirketlerin isimleri ve varsa da uçak şirket id'leri gösterildi.

```
SELECT COMPANY.Company_name, AIRPLANE_COMP.Airplane_comp_id
FROM COMPANY
LEFT JOIN AIRPLANE_COMP ON COMPANY.Company_id = AIRPLANE_COMP.Company_id
ORDER BY COMPANY.Company_name
```

Müşterilerin isim, pasaport numarası ve uçuş numarası gösterildi.

```
SELECT CUSTOMER.Customer_name, CUSTOMER.Passport_number, FLIGHT.Flight_number
FROM FLIGHT
RIGHT JOIN CUSTOMER ON CUSTOMER.Flight_number = FLIGHT.Flight_number
ORDER BY CUSTOMER.Customer_name
```

Tüm uçak şirketlerinin isimleri ve havaalanı id'leri gösterildi.

```
SELECT AIRPLANE_COMP.Airplane_company_name, AIRLINE.Airline_id
FROM AIRPLANE_COMP
FULL OUTER JOIN AIRLINE ON AIRPLANE_COMP.Company_id = AIRLINE.Company_id
ORDER BY AIRPLANE_COMP.Airplane_company_name DE
```

6-) 5 tane mantıklı VIEW yazıldı.

Belirli pasaport numarasına sahip bir yolcunun yaptığı uçuşları gösteren bir VIEW oluşturuldu.

```
CREATE VIEW customerFlight AS
SELECT Passport_number, Flight_number, Leg_number, LDate, Seat_number
FROM CUSTOMER
WHERE Passport_number = 1234567
```

Bir şirketin sahip olduğu alt şirketleri gösteren bir VIEW oluşturuldu.

```
CREATE VIEW subCompany AS
SELECT COMPANY.Company_name, AIRPLANE_COMP.Airplane_company_name,
AIRLINE.Airline_company_name
FROM COMPANY
FULL OUTER JOIN AIRPLANE_COMP ON Company.Company_id = AIRPLANE_COMP.Company_id
FULL OUTER JOIN AIRLINE ON Company.Company_id = AIRLINE.Company_id
```

Uçuşun fiyat ve kısıt bilgilerini gösteren bir VIEW oluşturuldu.

```
CREATE VIEW fareInformation AS
SELECT FARE.Flight_number, FARE.Amount AS Fiyat, FARE.Restrictions AS Kısıtlar
FROM FARE
```

Pilot için uçuş bilgilerini gösteren bir VIEW oluşturuldu.

```
CREATE VIEW forPilot AS
SELECT FLIGHT.Flight_number, LEG_INSTANCE.Departure_airport_code, (Select DISTINCT
airport1.AirportName from AIRPORT AS airport1 where airport1.Airport_code =
LEG_INSTANCE.Departure_airport_code) Departure_Airport, LEG_INSTANCE.Arrival_airport_code,
(Select DISTINCT airport2.AirportName from AIRPORT AS airport2 where airport2.Airport_code
= LEG_INSTANCE.Arrival_airport_code) Arrival_Airport, LEG_INSTANCE.Departure_time,
LEG_INSTANCE.Arrival_time, FLIGHT.Weekdays
FROM LEG_INSTANCE
INNER JOIN FLIGHT_LEG ON LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND
LEG_INSTANCE.Leg_number = FLIGHT_LEG.Leg_number
INNER JOIN FLIGHT ON FLIGHT_LEG.Flight_number = FLIGHT.Flight_number
```

Herhangi bir uçuştaki boş koltukları gösteren bir VIEW oluşturuldu.

```
CREATE VIEW availableSeat AS
SELECT FLIGHT_LEG.Flight_number , FLIGHT_LEG.Leg_number, SEAT_RESERVATION.Seat_number
FROM SEAT_RESERVATION
INNER JOIN LEG_INSTANCE ON SEAT_RESERVATION.Flight_number = LEG_INSTANCE.Flight_number AND
SEAT_RESERVATION.Leg_number = LEG_INSTANCE.Leg_number AND SEAT_RESERVATION.LDate =
LEG_INSTANCE.LDate
INNER JOIN FLIGHT_LEG  ON LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND
LEG_INSTANCE.Leg_number = FLIGHT_LEG.Leg_number
WHERE NOT Seat_number IN (SELECT SEAT_RESERVATION.Seat_number FROM
SEAT_RESERVATION,CUSTOMER WHERE SEAT_RESERVATION.Flight_number = CUSTOMER.Flight_number
AND SEAT_RESERVATION.Leg_number = CUSTOMER.Leg_number AND SEAT_RESERVATION.LDate =
CUSTOMER.LDate AND SEAT_RESERVATION.Seat_number = CUSTOMER.Seat_number)
```

7-) Yazılan SQL Kodlarının Tamamı

```
CREATE TABLE COMPANY (
Company_id int IDENTITY(1,1),
Company_name VARCHAR(45),
Foundation_year DATE ,
PRIMARY KEY(Company_id)
);

CREATE TABLE AIRPLANE_COMP (
Airplane_comp_id int IDENTITY(1,1),
Company_id int NOT NULL,
Airplane_company_name VARCHAR(45),
Foundation_year DATE,
PRIMARY KEY (Airplane_comp_id),
FOREIGN KEY (Company_id) REFERENCES COMPANY
);

CREATE TABLE AIRLINE (
Airline_id int IDENTITY(1,1),
Company_id int NOT NULL,
Airline_company_name VARCHAR(45),
Foundation_year DATE,
PRIMARY KEY (Airline_id),
FOREIGN KEY (Company_id) REFERENCES COMPANY
);

CREATE TABLE AIRPLANE_TYPE (
Airplane_type_name VARCHAR(30) NOT NULL,
Max_seats int,
CONSTRAINT CHK_Airplane_TYPE_Seats CHECK (AIRPLANE_TYPE.Max_seats>1),
PRIMARY KEY (Airplane_type_name)
);
```

```
CREATE TABLE AIRPLANE (
Airplane_id int IDENTITY(1,1),
Airplane_comp_id int NOT NULL,
Total_number_of_seats int,
Airplane_type VARCHAR(30) NOT NULL,

CONSTRAINT CHK_AirplaneSeats CHECK (AIRPLANE.Total_number_of_seats > 1),

PRIMARY KEY (Airplane_id),
FOREIGN KEY (Airplane_comp_id) REFERENCES AIRPLANE_COMP,
FOREIGN KEY (Airplane_type) REFERENCES AIRPLANE_TYPE
);

CREATE TABLE AIRPORT (
Airport_code int IDENTITY(1,1),
AirportName VARCHAR(45),
City VARCHAR(30),
AState VARCHAR(30),
PRIMARY KEY (Airport_code)
);

CREATE TABLE CAN_LAND (
Airplane_type VARCHAR(30) NOT NULL,
Airport_code int NOT NULL,
PRIMARY KEY (Airplane_type, Airport_code),
FOREIGN KEY (Airplane_type) REFERENCES AIRPLANE_TYPE ,
FOREIGN KEY (Airport_code) REFERENCES AIRPORT
);

CREATE TABLE FLIGHT (
Flight_number int NOT NULL,
Weekdays VARCHAR(15),
Airline_id int NOT NULL,
PRIMARY KEY (Flight_number),
FOREIGN KEY (Airline_id) REFERENCES AIRLINE
```

```
);

CREATE TABLE FARE (
Flight_number int NOT NULL,
Fare_code VARCHAR(1) NOT NULL,
Amount int,
Restrictions VARCHAR(15),
PRIMARY KEY (Flight_number, Fare_code),
FOREIGN KEY (Flight_number) REFERENCES FLIGHT
);

CREATE TABLE FLIGHT_LEG (
Flight_number int NOT NULL,
Leg_number int NOT NULL,
Departure_airport_code int,
Arrival_airport_code int,
Scheduled_departure_time DATETIME,
Scheduled_arrival_time DATETIME,
Mile int,
PRIMARY KEY (Flight_number, Leg_number),
FOREIGN KEY (Flight_number) REFERENCES FLIGHT ,
FOREIGN KEY (Departure_airport_code) REFERENCES AIRPORT,
FOREIGN KEY (Arrival_airport_code) REFERENCES AIRPORT
);

CREATE TABLE LEG_INSTANCE (
Flight_number int NOT NULL,
Leg_number int NOT NULL,
LDate DATETIME NOT NULL,
Number_of_available_seats int,
Airplane_id int NOT NULL,
Departure_airport_code int,
Arrival_airport_code int,
Departure_time DATETIME,
Arrival_time DATETIME,
```

```
CONSTRAINT CHK_CurrentTime CHECK(GETDATE()<LDATE),
CONSTRAINT CHK_Time CHECK (Departure_time<Arrival_time),
CONSTRAINT CHK_Seats CHECK (LEG_INSTANCE.Number_of_available_seats>1),
CONSTRAINT CHK_Code CHECK (LEG_INSTANCE.Departure_airport_code!=LEG_INSTANCE.Arrival_airport_code),

PRIMARY KEY (Flight_number, Leg_number, LDate),
FOREIGN KEY (Flight_number, Leg_number) REFERENCES FLIGHT_LEG(Flight_number, Leg_number) ,
FOREIGN KEY (Airplane_id) REFERENCES AIRPLANE
);

CREATE TABLE SEAT_RESERVATION (
Flight_number INTEGER NOT NULL,
Leg_number INTEGER NOT NULL,
LDate DATETIME NOT NULL,
Seat_number INTEGER NOT NULL,
PRIMARY KEY (Flight_number, Leg_number, LDate, Seat_number),
FOREIGN KEY (Flight_number, Leg_number, LDate) REFERENCES LEG_INSTANCE
);
----------
CREATE TABLE CUSTOMER (
Passport_number VARCHAR(7) NOT NULL,
Customer_phone VARCHAR(11),
Customer_name VARCHAR(15),
Email VARCHAR(30),
CAddress VARCHAR(45),
Country VARCHAR(30),
Flight_number int NOT NULL,
Leg_number int NOT NULL,
LDate DATETIME NOT NULL,
Seat_number int NOT NULL,
Check_in bit DEFAULT 0,

CONSTRAINT CHK_Passport_number CHECK(Passport_number > 0),
```

```
PRIMARY KEY (Passport_number,Flight_number, Leg_number, LDate, Seat_number),
FOREIGN KEY (Flight_number, Leg_number, LDate, Seat_number) REFERENCES SEAT_RESERVATION,
);
---------------
CREATE TABLE FFC (
Mileage int Not null,
Passport_number VARCHAR(7) NOT NULL,
Flight_number INTEGER NOT NULL,
Leg_number INTEGER NOT NULL,
Seat_number INTEGER NOT NULL,
LDate DATETIME NOT NULL,
Check_in bit DEFAULT 0,
Segmentation varchar(20),
PRIMARY KEY (Passport_number, Flight_number, Leg_number, Seat_number, LDate),
FOREIGN KEY (Passport_number,Flight_number, Leg_number, LDate, Seat_number) REFERENCES CUSTOMER  ,
FOREIGN KEY (Flight_number, Leg_number, LDate) REFERENCES LEG_INSTANCE
);
---------------------------------------------------------------------------------------------------
ALTER TABLE FARE
ADD CONSTRAINT Fare_Amount CHECK(Amount > 0);

ALTER TABLE FARE
ADD CONSTRAINT CHK_Fare_code CHECK(Fare_code = 'J' or Fare_code = 'F');

ALTER TABLE FFC
ADD CONSTRAINT CHK_Mileage CHECK(Mileage > 0);

ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK_Flight_Leg_Airport_code CHECK(FLIGHT_LEG.Departure_airport_code!=FLIGHT_LEG.Arrival_airport_code);

ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK_Flight_Leg_Time CHECK(Scheduled_departure_time  < Scheduled_arrival_time );

ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK_Flight_LEG_Mileage CHECK(Mile > 0);
```

```
ALTER TABLE FLIGHT_LEG
ADD CONSTRAINT CHK__Flight_LEG_CurrentTime CHECK(GETDATE() < Scheduled_departure_time);

-----------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------
CREATE TRIGGER CHK_MaxSeat ON AIRPLANE
AFTER INSERT
AS
BEGIN
        SET NOCOUNT ON;

        IF EXISTS(SELECT * FROM inserted, AIRPLANE_TYPE WHERE inserted.Total_number_of_seats = AIRPLANE_TYPE.Max_seats)
        BEGIN
                WAITFOR DELAY '00:00:00'
        END
        ELSE
        BEGIN
                RAISERROR ('Üretilen uçak tipinin koltuk sayısıyla koltuk sayınız eşleşmediğinden üretilemez!',1,1)
    ROLLBACK TRANSACTION
        END
END
-----------------------------------------------------------------------------------------------
--Check in yapmış yolcunun bilgilerinin değiştirilmemesi
CREATE TRIGGER CustomerUpdate ON CUSTOMER
AFTER UPDATE
AS
BEGIN
  IF (EXISTS(SELECT * FROM inserted, deleted WHERE deleted.Check_in = 1 AND
          (deleted.Flight_number != inserted.Flight_number OR
           deleted.LDate != inserted.LDate OR
           deleted.Leg_number != inserted.Leg_number OR
           deleted.Seat_number != inserted.Seat_number OR
                                        deleted.Country != inserted.Country OR
```

```
                                        deleted.Passport_number != inserted.Passport_number OR
                                        deleted.CAddress != inserted.CAddress OR
                                        deleted.Check_in != inserted.Check_in )))
  BEGIN
    RAISERROR ('Check in yapmış yolcunun uçuş bilgileri değiştirilemez!',1,1)
    ROLLBACK TRANSACTION
  END
END
-----------------------------------------------------------------------------------------------
CREATE TRIGGER mileUpdate ON FFC
AFTER UPDATE
AS
BEGIN

        SET NOCOUNT ON;

        DECLARE @check_in bit
        SELECT @check_in = Check_in from inserted

        DECLARE @oldCheck_in bit
        SELECT @oldCheck_in = Check_in from deleted

        IF(@check_in = 1 AND @oldCheck_in != @check_in)
        BEGIN
                UPDATE FFC
                SET FFC.Mileage += (SELECT DISTINCT Mile FROM FLIGHT_LEG, FFC
                WHERE FFC.Flight_number = FLIGHT_LEG.Flight_number AND FFC.Leg_number = FLIGHT_LEG.Leg_number)
                FROM inserted
                INNER JOIN LEG_INSTANCE ON (inserted.Flight_number = LEG_INSTANCE.Flight_number AND inserted.Leg_number =
LEG_INSTANCE.Leg_number AND inserted.LDate = LEG_INSTANCE.LDate)
                INNER JOIN FLIGHT_LEG ON (LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND
LEG_INSTANCE.Leg_number = FLIGHT_LEG.Leg_number)
                WHERE(FFC.Passport_number = inserted.Passport_number AND FFC.Flight_number = inserted.Flight_number AND
FFC.Leg_number = inserted.Leg_number AND FFC.Seat_number = inserted.Seat_number AND FFC.LDate = inserted.LDate AND
inserted.Check_in = FFC.Check_in)
        END
```

```
END

-----------------------------------------------------------------------------------------------

CREATE TRIGGER mileUpdateWhenInsert ON FFC

AFTER INSERT

AS

BEGIN

	SET NOCOUNT ON;


	UPDATE FFC

	SET FFC.Mileage += (SELECT DISTINCT Mile FROM FLIGHT_LEG, inserted

						WHERE inserted.Flight_number = FLIGHT_LEG.Flight_number AND inserted.Leg_number
= FLIGHT_LEG.Leg_number)

	From inserted

	WHERE FFC.Passport_number = inserted.Passport_number AND FFC.Flight_number = inserted.Flight_number AND FFC.Leg_number
= inserted.Leg_number AND FFC.Seat_number = inserted.Seat_number AND FFC.LDate = inserted.LDate AND inserted.Check_in = FFC.Check_in
AND FFC.Check_in = 1


END

-----------------------------------------------------------------------------------------------


CREATE TRIGGER seatUpdate ON Customer

AFTER INSERT

AS

BEGIN

	SET NOCOUNT ON;


	UPDATE LEG_INSTANCE

	SET LEG_INSTANCE.Number_of_available_seats += -1

	From inserted

	WHERE LEG_INSTANCE.Flight_number = inserted.Flight_number AND LEG_INSTANCE.Leg_number = inserted.Leg_number AND
LEG_INSTANCE.LDate = inserted.LDate




END


-----------------------------------------------------------------------------------------------
```

```
CREATE TRIGGER seatUpdateOnDELETE ON Customer

AFTER Delete

AS

BEGIN

	SET NOCOUNT ON;


	UPDATE LEG_INSTANCE

	SET LEG_INSTANCE.Number_of_available_seats += 1

	From deleted

	WHERE LEG_INSTANCE.Flight_number = deleted.Flight_number AND LEG_INSTANCE.Leg_number = deleted.Leg_number AND
LEG_INSTANCE.LDate = deleted.LDate




END

----------------------------------------------------------------------------------------------


ALTER TRIGGER custToFFC ON CUSTOMER

AFTER INSERT,UPDATE

AS

BEGIN

SET NOCOUNT ON;

	IF NOT EXISTS(SELECT * from inserted where inserted.Check_in = 0)

	BEGIN

		INSERT INTO FFC

		SELECT (SELECT SUM(Mile)

				FROM FLIGHT_LEG, inserted

				WHERE FLIGHT_LEG.Flight_number = inserted.Flight_number And FLIGHT_LEG.Leg_number =
inserted.Leg_number

				GROUP BY inserted.Passport_number) , inserted.Passport_number, inserted.Flight_number,
inserted.Leg_number, inserted.Seat_number, inserted.LDate, inserted.Check_in, NULL

		FROM inserted

		WHERE NOT EXISTS(SELECT * from FFC WHERE FFC.Flight_number = inserted.Flight_number AND FFC.Leg_number =
inserted.Leg_number AND FFC.LDate = inserted.LDate AND FFC.Seat_number = inserted.Seat_number AND FFC.Check_in = inserted.Check_in
AND FFC.Passport_number = inserted.Passport_number )

	END

END

----------------------------------------------------------------------------------------------
```

```
CREATE TRIGGER customerSegmentatiton ON FFC
AFTER UPDATE
AS
BEGIN

	SET NOCOUNT ON;
	UPDATE FFC
	SET FFC.Segmentation = CASE
		WHEN  FFC.Mileage > 1000 THEN 'Premium Customer'
		WHEN  FFC.Mileage > 500 THEN 'Gold Customer'
		ELSE 'Common Customer'
	END

END

CREATE TRIGGER giveReward ON FFC
AFTER UPDATE
AS
BEGIN
	SET NOCOUNT ON;

	IF EXISTS(SELECT * FROM inserted, deleted WHERE inserted.Passport_number = deleted.Passport_number AND
inserted.Segmentation != deleted.Segmentation)
	BEGIN
		SELECT inserted.Passport_number, inserted.Segmentation newSegmentation, deleted.Segmentation oldSegmentation,
			CASE
				WHEN inserted.segmentation = 'Premium Customer' THEN 'Tebrikler Premium üye
olmaya hak kazandınız. Kazandığınız ekstra avantajları öğrenmek için internet sitemizi ziyaret ediniz.'
				WHEN inserted.segmentation = 'Gold Customer' THEN 'Tebrikler Gold üye olmaya hak
kazandınız. Kazandığınız ekstra avantajları öğrenmek için internet sitemizi ziyaret ediniz.'
				ELSE 'Gold üye olmaya çok yakınsınız bizi tercih etmeye devam edin.'
			END AS 'Reward'
		FROM inserted,deleted
		WHERE inserted.Passport_number = deleted.Passport_number AND inserted.Segmentation != deleted.Segmentation
		ORDER BY inserted.Passport_number ASC
```

```
        END
END
```

```
INSERT INTO COMPANY VALUES('Air France KLM', '2004-05-05');
INSERT INTO COMPANY VALUES('Turkish Airlines', '1933-05-20');
INSERT INTO COMPANY VALUES('American Airlines Group', '2013-12-09');

INSERT INTO AIRPLANE_COMP VALUES(1, 'Safran', '2005');
INSERT INTO AIRPLANE_COMP VALUES(3, 'Boeing', '1916-07-15');
INSERT INTO AIRPLANE_COMP VALUES(3, 'Lockheed Martin', '1995-03-15');

INSERT INTO AIRLINE VALUES(1, 'Transavia France', '2006');
INSERT INTO AIRLINE VALUES(2, 'SunExpress', '1989');
INSERT INTO AIRLINE VALUES(2, 'AnadoluJet', '2008-04-23');

INSERT INTO AIRPLANE_TYPE VALUES('AN-22', 290);
INSERT INTO AIRPLANE_TYPE VALUES('Boeing-737 Max 8', 210);
INSERT INTO AIRPLANE_TYPE VALUES('Boeing-787-9 DreamLiner', 296);

INSERT INTO AIRPLANE VALUES(1, 290, 'AN-22');
INSERT INTO AIRPLANE VALUES(1, 290, 'AN-22');
INSERT INTO AIRPLANE VALUES(2, 210, 'Boeing-737 Max 8');
INSERT INTO AIRPLANE VALUES(2, 296, 'Boeing-787-9 DreamLiner');

INSERT INTO AIRPORT VALUES('Erzurum Airport', 'Erzurum', 'Turkey');
INSERT INTO AIRPORT VALUES('İzmir Adnan Menderes Airport', 'İzmir', 'Turkey');
INSERT INTO AIRPORT VALUES('Bursa Yenişehir Airport', 'Bursa', 'Turkey');
INSERT INTO AIRPORT VALUES('Ankara Esenboğa Airport', 'Ankara', 'Turkey');
INSERT INTO AIRPORT VALUES('İstanbul Sabiha Gökçen Airport', 'İstanbul', 'Turkey');

INSERT INTO CAN_LAND VALUES('AN-22', 1);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 1);
```

```
INSERT INTO CAN_LAND VALUES('AN-22', 2);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 2);
INSERT INTO CAN_LAND VALUES('AN-22', 3);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 3);
INSERT INTO CAN_LAND VALUES('AN-22', 4);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 4);
INSERT INTO CAN_LAND VALUES('AN-22', 5);
INSERT INTO CAN_LAND VALUES('Boeing-737 Max 8', 5);
INSERT INTO CAN_LAND VALUES('Boeing-787-9 DreamLiner', 5);

INSERT INTO FLIGHT VALUES(100, 'Tuesday', 1);
INSERT INTO FLIGHT VALUES(101, 'Tuesday', 2);
INSERT INTO FLIGHT VALUES(102, 'Tuesday', 3);
INSERT INTO FLIGHT VALUES(103, 'Friday', 1);
INSERT INTO FLIGHT VALUES(104, 'Saturday', 2);
INSERT INTO FLIGHT VALUES(105, 'Sunday', 3);

INSERT INTO FARE VALUES(100, 'F', 100, 'Alcohol');
INSERT INTO FARE VALUES(100, 'J', 80 , 'Alcohol-free');

INSERT INTO FARE VALUES(101, 'F', 60, 'Alcohol');
INSERT INTO FARE VALUES(101, 'J' , 40, 'Alcohol-free');

INSERT INTO FARE VALUES(102, 'F', 120, 'Alcohol');
INSERT INTO FARE VALUES(102, 'J' , 70, 'Alcohol-free');

INSERT INTO FARE VALUES(103, 'F', 160, 'Alcohol');
INSERT INTO FARE VALUES(103, 'J' , 120, 'Alcohol-free');

INSERT INTO FARE VALUES(104, 'F', 70, 'Alcohol');
INSERT INTO FARE VALUES(104, 'J' , 50, 'Alcohol-free');

INSERT INTO FARE VALUES(105, 'F', 160, 'Alcohol');
INSERT INTO FARE VALUES(105, 'J' , 120, 'Alcohol-free');
```

```
INSERT INTO FLIGHT_LEG VALUES(100, 1, 4, 5, '2021-02-16 07:30', '2021-02-16 08:35', 70);

INSERT INTO FLIGHT_LEG VALUES(101, 1, 2, 5, '2021-02-16 04:40', '2021-02-16 05:45', 80);

INSERT INTO FLIGHT_LEG VALUES(102, 1, 1, 3, '2021-02-16 11:45', '2021-02-16 13:40', 60);

INSERT INTO FLIGHT_LEG VALUES(103, 1, 1, 4, '2021-02-19 13:20', '2021-02-19 14:25', 90);
INSERT INTO FLIGHT_LEG VALUES(103, 2, 4, 5, '2021-02-19 15:25', '2021-02-19 16:50', 100);

INSERT INTO FLIGHT_LEG VALUES(104, 1, 4, 3, '2021-02-20 15:30', '2021-02-20 16:30', 40);

INSERT INTO FLIGHT_LEG VALUES(105, 1, 5, 4, '2021-02-21 17:00', '2021-02-21 18:15', 50);
INSERT INTO FLIGHT_LEG VALUES(105, 2, 4, 1, '2021-02-21 19:00', '2021-02-21 20:05', 20);

INSERT INTO LEG_INSTANCE VALUES(100, 1, '2021-02-16', 290, 1, 4, 5, '2021-02-16 07:30', '2021-02-16 08:35');
INSERT INTO LEG_INSTANCE VALUES(101, 1, '2021-02-16', 290, 2, 2, 5, '2021-02-16 04:40', '2021-02-16 05:45');
INSERT INTO LEG_INSTANCE VALUES(102, 1, '2021-02-16', 210, 3, 1, 3, '2021-02-16 11:45', '2021-02-16 13:40');

INSERT INTO LEG_INSTANCE VALUES(103, 1, '2021-02-19', 296, 4, 1, 4, '2021-02-19 13:20', '2021-02-19 14:25');
INSERT INTO LEG_INSTANCE VALUES(103, 2, '2021-02-19', 296, 4, 4, 5, '2021-02-19 15:25', '2021-02-19 16:50');

INSERT INTO LEG_INSTANCE VALUES(104, 1, '2021-02-20', 290, 1, 4, 3, '2021-02-20 15:30', '2021-02-20 16:30');

INSERT INTO LEG_INSTANCE VALUES(105, 1, '2021-02-21', 210, 3, 5, 4, '2021-02-21 17:00', '2021-02-21 18:15');
INSERT INTO LEG_INSTANCE VALUES(105, 2, '2021-02-21', 210, 3, 4, 1, '2021-02-21 19:00', '2021-02-21 20:05');

INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 10);
INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 11);
INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 12);

INSERT INTO SEAT_RESERVATION VALUES(101, 1, '2021-02-16', 13);
INSERT INTO SEAT_RESERVATION VALUES(101, 1, '2021-02-16', 14);

INSERT INTO SEAT_RESERVATION VALUES(102, 1, '2021-02-16', 15);
```

```
INSERT INTO SEAT_RESERVATION VALUES(103, 1, '2021-02-19', 16);

INSERT INTO SEAT_RESERVATION VALUES(103, 1, '2021-02-19', 17);

INSERT INTO SEAT_RESERVATION VALUES(103, 1, '2021-02-19', 18);

INSERT INTO SEAT_RESERVATION VALUES(103, 2, '2021-02-19', 19);


INSERT INTO SEAT_RESERVATION VALUES(104, 1, '2021-02-20', 20);


INSERT INTO SEAT_RESERVATION VALUES(105, 1, '2021-02-21', 10);

INSERT INTO SEAT_RESERVATION VALUES(105, 1, '2021-02-21', 11);

INSERT INTO SEAT_RESERVATION VALUES(105, 2, '2021-02-21', 12);


INSERT INTO CUSTOMER VALUES(1234567, 55555555555, 'ali', 'ali@gmail.com', 'Ali adres', 'Ali country', 100, 1, '2021-02-16', 10, 0);

INSERT INTO CUSTOMER VALUES(2345678, 55555555555, 'veli','veli@gmail.com', 'Veli adres', 'veli country', 100, 1, '2021-02-16', 11, 1);

INSERT INTO CUSTOMER VALUES(3456789, 55555555555, 'ayşe','ayşe@gmail.com', 'ayşe adres', 'ayşe country', 100, 1, '2021-02-16', 12, 0);

INSERT INTO CUSTOMER VALUES(4567890, 55555555555, 'ahmet','ahmet@gmail.com','ahmet adres', 'ahmet country',101, 1, '2021-02-16', 13,
1);

INSERT INTO CUSTOMER VALUES(5678901, 55555555555, 'zeki','zeki@gmail.com', 'zeki adres', 'zeki country', 101, 1, '2021-02-16', 14, 0);

INSERT INTO CUSTOMER VALUES(6789012, 55555555555, 'erdem','erdem@gmail.com','erdem adres', 'erdem country',102, 1, '2021-02-16', 15,
1);

INSERT INTO CUSTOMER VALUES(7890123, 55555555555, 'mehmet','mehmet@gmail.com', 'mehmet adres', 'mehmet country',103, 1, '2021-02-
19', 16, 0);

INSERT INTO CUSTOMER VALUES(1231235, 55555555555, 'mustafa','mustafa@gmail.com','mustafa adres', 'mustafa country',103, 1, '2021-02-
19', 17, 1);

INSERT INTO CUSTOMER VALUES(7777777, 55555555555, 'serdar','serdar@gmail.com', 'serdar adres', 'serdar country', 103, 1, '2021-02-19', 18,
0);

INSERT INTO CUSTOMER VALUES(1651255, 55555555555, 'buse','buse@gmail.com', 'buse adres', 'buse country',103, 2, '2021-02-19', 19, 0);

INSERT INTO CUSTOMER VALUES(9012345, 55555555555, 'deniz','deniz@gmail.com', 'deniz adres', 'deniz country',104, 1, '2021-02-20', 20, 1);

INSERT INTO CUSTOMER VALUES(8901234, 55555555555, 'oğuz','oğuz@gmail.com', 'oğuz adres', 'oğuz country',105, 1, '2021-02-21', 10, 1);

INSERT INTO CUSTOMER VALUES(7512355, 55555555555, 'selda','selda@gmail.com', 'selda adres', 'selda country',105, 1, '2021-02-21', 11, 0);

INSERT INTO CUSTOMER VALUES(8676777, 55555555555, 'sonat','sonat@gmail.com', 'sonat adres', 'sonat country',105, 2, '2021-02-21', 12, 1);

-----------------------------------------------------------------------------------------------------

DELETE FROM CUSTOMER

WHERE Passport_number = '86776777'


DELETE FROM SEAT_RESERVATION

WHERE Seat_number=20
```

```
DELETE FROM FFC
WHERE FFC.Passport_number IN (SELECT FFC.Passport_number
                              FROM FFC,CUSTOMER
                              WHERE FFC.Passport_number = Customer.Passport_number AND
Customer.Country = 'selda country'
                              )
DELETE FROM FARE
WHERE Amount < 50

DELETE FROM LEG_INSTANCE
WHERE LEG_INSTANCE.LDate = '2021-02-21 00:00:00.000'
```

```
UPDATE CUSTOMER
SET Check_in = 1
WHERE Passport_number = '7777777' AND Flight_number = 103 AND Leg_number = 1 AND LDate = '2021-02-19'

UPDATE FFC
SET Mileage += 500

UPDATE FLIGHT_LEG
SET Mile += 20

UPDATE FARE
SET FARE.Amount += 100
FROM Fare, AIRPORT, FLIGHT_LEG
WHERE Fare.Flight_number = FLIGHT_LEG.Flight_number AND FLIGHT_LEG.Arrival_airport_code = 1

UPDATE AIRPLANE_COMP
SET Company_id = 2
```

```
FROM COMPANY, AIRPLANE_COMP
WHERE Airplane_comp_id = 2

------------------------------
Select statements with 2 tables
-------------------------------------------------------------------------------------------------
---Bir uçağın tipi,id ve max koltuğu gösterilir------
SELECT atype.Airplane_type_name, a.Airplane_id, atype.Max_seats
FROM AIRPLANE_TYPE AS atype, AIRPLANE AS A
WHERE atype.Airplane_type_name = a.Airplane_type

--------Bir uçak hangi airportlara inebilir---------
SELECT a.Airport_code, a.AirportName, cl.Airplane_type
FROM CAN_LAND AS cl, AIRPORT as a
WHERE cl.Airport_code = a.Airport_code
ORDER BY a.Airport_code
-----------------Uçuşun ne zaman kalkacağı ve uygun koltuk sayısı-------
SELECT fl.Flight_number, li.Number_of_available_seats, li.Departure_time
FROM FLIGHT_LEG fl, LEG_INSTANCE li
WHERE fl.Leg_number = li.Leg_number AND fl.Flight_number = li.Flight_number
ORDER BY li.Departure_time ASC

-------------------------------------------------------------------------------------------------
Select statements with 3 tables
-------------------------------------------------------------------------------------------------
-------------------Bir customerın check-in bilgisi ve uçuş bilgilerini yazdırır-------
SELECT c.Passport_number, sr.Flight_number, c.Seat_number, fl.Scheduled_arrival_time,
CASE
        WHEN c.Check_in = 0 THEN 'not checked'
        WHEN c.Check_in = 1 THEN 'checked'
        ELSE 'unknown'
END AS 'check-in'
FROM CUSTOMER AS c, SEAT_RESERVATION  AS sr, FLIGHT_LEG AS fl
WHERE c.Flight_number = sr.Flight_number AND c.LDate = sr.LDate AND c.Leg_number = sr.Leg_number AND c.Seat_number = sr.Seat_number
AND c.Leg_number = fl.Leg_number AND c.Flight_number = fl.Flight_number
```

```
----------En erken uçuşu olan 5 yolcu----------

SELECT TOP 5 c.Customer_name, li.Departure_airport_code, li.Departure_time

FROM CUSTOMER AS c, SEAT_RESERVATION AS sr, LEG_INSTANCE AS li

WHERE c.Flight_number = sr.Flight_number AND c.LDate = sr.LDate AND c.Leg_number = sr.Leg_number AND c.Seat_number = sr.Seat_number

AND sr.Flight_number = li.Flight_number AND sr.Leg_number = li.Leg_number AND sr.LDate = li.LDate

ORDER BY li.Departure_time ASC

----------Bir uçuşu organize eden havayolu şirketiyle, u şirketin bağlı olduğu şirketi yazdırır---------

SELECT FLIGHT.Flight_number,  AIRLINE.Airline_company_name AIRLINE,Company.Company_name COMPANY

FROM COMPANY, AIRLINE, FLIGHT

WHERE COMPANY.Company_id = AIRLINE.Company_id AND AIRLINE.Airline_id = FLIGHT.Airline_id

ORDER BY FLIGHT.Flight_number ASC

----------Bir Airline şirketinin organize ettiği uçuşlar ve uçuşla ilgili bazı bilgiler----

SELECT DISTINCT AIRLINE.Airline_company_name,FLIGHT.Flight_number, FARE.Amount, FARE.Restrictions

FROM FARE, FLIGHT, AIRLINE

WHERE AIRLINE.Airline_id = FLIGHT.Airline_id AND FLIGHT.Flight_number = FARE.Flight_number

---------------------------------------------------------------------------------------------------

Select statements with 4 tables

---------------------------------------------------------------------------------------------------

----------Bir uçuşta check-in yapmış yolcuların yazdırılması-----

SELECT FLIGHT.Flight_number, CUSTOMER.Passport_number, CUSTOMER.Customer_name

from CUSTOMER, FLIGHT, FLIGHT_LEG, LEG_INSTANCE

WHERE CUSTOMER.Flight_number = LEG_INSTANCE.Flight_number AND CUSTOMER.Leg_number = LEG_INSTANCE.Leg_number and
CUSTOMER.LDate = LEG_INSTANCE.LDate AND CUSTOMER.Check_in = 1

AND LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND LEG_INSTANCE.Leg_number = FLIGHT_LEG.Leg_number

AND FLIGHT.Flight_number = FLIGHT_LEG.Flight_number

ORDER BY FLIGHT.Flight_number ASC

----------Bir uçuşta dolu olan koltuklar ve uçuşun yapıldığı uçak tipi--------

SELECT LEG_INSTANCE.Flight_number, SEAT_RESERVATION.Seat_number ,AIRPLANE_TYPE.Airplane_type_name

FROM SEAT_RESERVATION, LEG_INSTANCE, AIRPLANE_TYPE, AIRPLANE

WHERE SEAT_RESERVATION.Flight_number = LEG_INSTANCE.Flight_number AND SEAT_RESERVATION.Leg_number =
LEG_INSTANCE.Leg_number AND SEAT_RESERVATION.LDate = LEG_INSTANCE.LDate

AND LEG_INSTANCE.Airplane_id = AIRPLANE.Airplane_id

AND AIRPLANE.Airplane_type = AIRPLANE_TYPE.Airplane_type_name

----------Bir uçuşun hangi gün yapıldığı, hangi havaalanına iniş yaptığı ve uçak tipi-------

SELECT FLIGHT.Flight_number, FLIGHT.Weekdays, AIRPORT.AirportName Arrival_Airport, AIRPLANE.Airplane_type
```

```
FROM FLIGHT, FLIGHT_LEG, LEG_INSTANCE, AIRPORT, AIRPLANE

WHERE FLIGHT.Flight_number = FLIGHT_LEG.Flight_number

AND FLIGHT_LEG.Flight_number = LEG_INSTANCE.Flight_number AND FLIGHT_LEG.Leg_number = LEG_INSTANCE.Leg_number

AND LEG_INSTANCE.Arrival_airport_code = AIRPORT.Airport_code AND LEG_INSTANCE.Airplane_id = AIRPLANE.Airplane_id

-----------------------------------------------------------------------------------------------

NESTED QUERIES

-----------------------------------------------------------------------------------------------

--Fiyatı en yüksek olan uçuş/uçuşların uçuş bilgileri

SELECT
FARE.Amount,FARE.Fare_code,FARE.Flight_number,FARE.Restrictions,LEG_INSTANCE.Airplane_id,LEG_INSTANCE.Arrival_airport_code,LEG_INS
TANCE.Departure_airport_code,LEG_INSTANCE.Number_of_available_seats,LEG_INSTANCE.LDate,LEG_INSTANCE.Arrival_time,LEG_INSTANCE.D
eparture_time

FROM FARE,LEG_INSTANCE

WHERE FARE.Flight_number=LEG_INSTANCE.Flight_number AND FARE.Amount IN

        (SELECT DISTINCT MAX(FARE.Amount)

         FROM FARE )

ORDER BY FARE.Amount ASC


--102 NUMARALI UÇUŞUN FİYAT VE HAVAALANI BİLGİLERİ

SELECT FLIGHT_LEG.Flight_number,FLIGHT_LEG.Departure_airport_code,FLIGHT_LEG.Arrival_airport_code,FARE.Fare_code,FARE.Amount

FROM FARE,FLIGHT_LEG

WHERE FARE.Flight_number=FLIGHT_LEG.Flight_number AND

FARE.Flight_number in

(SELECT Flight_number FROM FLIGHT_LEG WHERE Flight_number=102)

----------Lüks uçuş----------

SELECT Customer.Customer_name, CUSTOMER.Email,CUSTOMER.Customer_phone,FARE.Amount,CUSTOMER.Flight_number

 FROM FARE,CUSTOMER,FLIGHT_LEG

 WHERE FARE.Amount >

        (SELECT AVG(FARE.Amount) FROM FARE )

        AND CUSTOMER.Flight_number=FLIGHT_LEG.Flight_number AND CUSTOMER.Leg_number= FLIGHT_LEG.Leg_number AND
FLIGHT_LEG.Flight_number=FARE.Flight_number AND FARE.Restrictions='Alcohol'



---------80'DEN BÜYÜK---------

SELECT  DISTINCT FARE.Amount,FARE.Fare_code,FARE.Restrictions,FARE.Flight_number, (Select DISTINCT airport1.AirportName from AIRPORT
AS airport1 where airport1.Airport_code = LEG_INSTANCE.Departure_airport_code) Departure_Airport, (Select DISTINCT airport2.AirportName
from AIRPORT AS airport2 where airport2.Airport_code = LEG_INSTANCE.Arrival_airport_code) Arrival_Airport, LEG_INSTANCE.Departure_time,
LEG_INSTANCE.Arrival_time, FLIGHT.Weekdays
```

```
FROM LEG_INSTANCE

INNER JOIN FLIGHT_LEG ON LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND LEG_INSTANCE.Leg_number =
FLIGHT_LEG.Leg_number

INNER JOIN FLIGHT ON FLIGHT_LEG.Flight_number = FLIGHT.Flight_number

INNER JOIN FARE ON FARE.Flight_number=LEG_INSTANCE.Flight_number

WHERE FARE.Amount IN (SELECT DISTINCT FARE.Amount FROM FARE

        WHERE FARE.Amount>80)

ORDER BY FARE.Amount ASC
```

EXISTS

```
SELECT LEG_INSTANCE.Flight_number, LEG_INSTANCE.Leg_number, LEG_INSTANCE.LDate,availableSeat.Seat_number, FARE.Restrictions

FROM LEG_INSTANCE, FARE, availableSeat

WHERE EXISTS(SELECT * FROM availableSeat WHERE LEG_INSTANCE.Flight_number = availableSeat.Flight_number AND
LEG_INSTANCE.Leg_number = availableSeat.Leg_number)

AND FARE.Restrictions = 'Alcohol-free' AND Fare.Flight_number = LEG_INSTANCE.Flight_number



DECLARE @allCustomerCheckedin varchar(30)

SELECT @allCustomerCheckedin = 'Premium yolcu bulunmamaktadır'


SELECT DISTINCT @allCustomerCheckedin

FROM FFC

WHERE NOT EXISTS(Select * From FFC,LEG_INSTANCE Where FFC.Segmentation = 'Premium Customer' AND LEG_INSTANCE.Flight_number =
FFC.Flight_number AND LEG_INSTANCE.Leg_number = FFC.Leg_number AND LEG_INSTANCE.LDate = FFC.LDate)
```

Select statements with JOIN

```
SELECT COMPANY.Company_name, AIRPLANE_COMP.Airplane_comp_id

FROM COMPANY

LEFT JOIN AIRPLANE_COMP ON COMPANY.Company_id = AIRPLANE_COMP.Company_id

ORDER BY COMPANY.Company_name


SELECT CUSTOMER.Customer_name, CUSTOMER.Passport_number, FLIGHT.Flight_number
```

```
FROM FLIGHT
RIGHT JOIN CUSTOMER ON CUSTOMER.Flight_number = FLIGHT.Flight_number
ORDER BY CUSTOMER.Customer_name

SELECT AIRPLANE_COMP.Airplane_company_name, AIRLINE.Airline_id
FROM AIRPLANE_COMP
FULL OUTER JOIN AIRLINE ON AIRPLANE_COMP.Company_id = AIRLINE.Company_id
ORDER BY AIRPLANE_COMP.Airplane_company_name

----------------------------------------------------------------------------------------------------
Views
----------------------------------------------------------------------------------------------------
---------Belirli pasaport nuamrasına ssahip bir yolcunun yaptığu uçuşlar---------
CREATE VIEW customerFlight AS
SELECT Passport_number, Flight_number, Leg_number, LDate, Seat_number
FROM CUSTOMER
WHERE Passport_number = 1234567

SELECT * FROM customerFlight

----------------------------------------------------------------------------------------------------
---------Bir company'nin sahip olduğu tüm şirketler-----------
CREATE VIEW subCompany AS
SELECT COMPANY.Company_name, AIRPLANE_COMP.Airplane_company_name, AIRLINE.Airline_company_name
FROM COMPANY
FULL OUTER JOIN AIRPLANE_COMP ON Company.Company_id = AIRPLANE_COMP.Company_id
FULL OUTER JOIN AIRLINE ON Company.Company_id = AIRLINE.Company_id

SELECT * FROM subCompany

----------------------------------------------------------------------------------------------------
----------Uçuşun fiyat ve kısıt bilgileri----------
CREATE VIEW fareInformation AS
SELECT FARE.Flight_number, FARE.Amount AS Fiyat, FARE.Restrictions AS Kısıtlar
FROM FARE
```

```
SELECT * FROM fareInformation


-----------------------------------------------------------------------------------------------------

----------Pilot için uçuş bilgilerini yazdırır----------

CREATE VIEW forPilot AS

SELECT FLIGHT.Flight_number, LEG_INSTANCE.Departure_airport_code, (Select DISTINCT airport1.AirportName from AIRPORT AS airport1
where airport1.Airport_code = LEG_INSTANCE.Departure_airport_code) Departure_Airport, LEG_INSTANCE.Arrival_airport_code, (Select
DISTINCT airport2.AirportName from AIRPORT AS airport2 where airport2.Airport_code = LEG_INSTANCE.Arrival_airport_code) Arrival_Airport,
LEG_INSTANCE.Departure_time, LEG_INSTANCE.Arrival_time, FLIGHT.Weekdays

FROM LEG_INSTANCE

INNER JOIN FLIGHT_LEG ON LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND LEG_INSTANCE.Leg_number =
FLIGHT_LEG.Leg_number

INNER JOIN FLIGHT ON FLIGHT_LEG.Flight_number = FLIGHT.Flight_number



SELECT * FROM forPilot


-----------------------------------------------------------------------------------------------------

----------Herhangi bir uçuştaki boş koltukları yazdırır---------

CREATE VIEW availableSeat AS

SELECT FLIGHT_LEG.Flight_number , FLIGHT_LEG.Leg_number, SEAT_RESERVATION.Seat_number

FROM SEAT_RESERVATION

INNER JOIN LEG_INSTANCE ON SEAT_RESERVATION.Flight_number = LEG_INSTANCE.Flight_number AND SEAT_RESERVATION.Leg_number =
LEG_INSTANCE.Leg_number AND SEAT_RESERVATION.LDate = LEG_INSTANCE.LDate

INNER JOIN FLIGHT_LEG  ON LEG_INSTANCE.Flight_number = FLIGHT_LEG.Flight_number AND LEG_INSTANCE.Leg_number =
FLIGHT_LEG.Leg_number

WHERE NOT Seat_number IN (SELECT SEAT_RESERVATION.Seat_number FROM SEAT_RESERVATION,CUSTOMER WHERE
SEAT_RESERVATION.Flight_number = CUSTOMER.Flight_number AND SEAT_RESERVATION.Leg_number = CUSTOMER.Leg_number AND
SEAT_RESERVATION.LDate = CUSTOMER.LDate AND SEAT_RESERVATION.Seat_number = CUSTOMER.Seat_number)



INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 24);

INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 25);

INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 26);

INSERT INTO SEAT_RESERVATION VALUES(100, 1, '2021-02-16', 27);

SELECT * FROM availableSeat
```