Database Management

2020-2021

Final Project: Database Implementation and Application Development

GROUP-47

HAZIRLAYANLAR;

05180000831-Mustafa Hikmet ÖZTÜRK

05180000830-Gürcan EKER

05180000asdasd939-Enes YAŞARBAŞ

05200001159-İSA TETİK

PART-II DATABASE IMPLEMENTATION

1. Write down the appropriate SQL scripts (DDL statements) for creating the database and its relational model. You can select any of the DBMS you wish.

Bu kısımda tasarlamış olduğumuz diagramın sql tarafında databasenin oluşturulması, tabloların oluşturulması, tablolar arasında ilişkilerin kurulması, attributeların hangi değerleri alacağını unique olup olmadığını ve boş bırakılıp bırakılamayacağını ve primary keylerin hangi alanlar olduğunu sql kodları ile belirlediğimiz kısımdır.

```sql
CREATE DATABASE DB_AIRLINE;

USE [DB_AIRLINE]
GO

CREATE TABLE AIRPORT(
        Airport_Code INT NOT NULL,
        Name VARCHAR(30) Unique,
        City VARCHAR(20),
        State VARCHAR(20),
        PRIMARY KEY(Airport_code)
        );

CREATE TABLE AIRPLANE_TYPE(
        Type_name VARCHAR(50) NOT NULL Unique,
        Max_seat INT,
        Company_id INT NOT NULL
        Primary KEY(Type_name));

Create table CAN_LAND(
        Airport_code int Not null,
        Type_name VARCHAR(50) Not null,
        Primary key(Airport_code,Type_name),
        Foreign key(Airport_code) REFERENCES AIRPORT(Airport_code),
        Foreign key(Type_name) REFERENCES AIRPLANE_TYPE(Type_name));

Create Table COMPANY(
        Company_id int Identity(1,1) Not null ,
        Company_name VARCHAR(30) Unique,
        Year_of_foundation int
        Primary key(Company_id));

Create Table AIRPLANE(
        Airplane_id int Not null,
        Total_no_of_seat int,
        Airplane_type VARCHAR(50) Not null,
        Company_id int Not null
        Primary key(Airplane_id)
        Foreign key(Airplane_type) REFERENCES AIRPLANE_TYPE(Type_name),
        Foreign key(Company_id) REFERENCES COMPANY(Company_id));

Create Table AIRLINE(
        Airline_id int not null Identity(1,1),
        Airline_name VARCHAR(30) Unique,
        Company_id int not null,
        Primary key(Airline_id),
        Foreign key(Company_id) REFERENCES COMPANY(Company_id));

Create Table FLIGHT(
        Flight_number VARCHAR(9) Not null,
        Airline_id int not null,
        Weekdays VARCHAR(20),
        Primary key(Flight_number),
        Foreign key(Airline_id) REFERENCES AIRLINE(Airline_id));

Create Table FARES(
        Flight_number VARCHAR(9) Not null,
        Fare_code BIT not null,
        Amonuts int,
        Restrictions Varchar(30),
        Primary key(Flight_number,Fare_code),
        Foreign key(Flight_number) REFERENCES FLIGHT(Flight_number));

Create Table FLIGHT_LEG(
        Flight_number VARCHAR(9) Not null,
        Leg_number int Identity(1,1) not null,
        Departure_airport_code int not null,
        Arrival_airport_code int not null,
        Scheduled_departure_time time,
        Scheduled_arrival_time time,
        Milleage float,
        Primary key(Flight_number,Leg_number),
        Foreign key(Flight_number) REFERENCES FLIGHT(Flight_number),
        Foreign key(Departure_airport_code) REFERENCES AIRPORT(Airport_Code),
        Foreign key(Arrival_airport_code) REFERENCES AIRPORT(Airport_Code));

Create table LEG_INSTANCE(
        Flight_number VARCHAR(9) Not null,
        Leg_number int  not null,
        Date Date not null,
        Number_of_available_seats int,
        Airplane_id int not null,
        Departure_airport_code int not null,
        Arrival_airport_code int not null,
        Departure_time time,
        Arrival_time time,
        Primary key(Flight_number,Leg_number,Date),
        Foreign key(Flight_number,Leg_number) REFERENCES FLIGHT_LEG(Flight_number,Leg_number),
        Foreign key(Departure_airport_code) REFERENCES AIRPORT(Airport_Code),
        Foreign key(Arrival_airport_code) REFERENCES AIRPORT(Airport_Code),
        Foreign key(Airplane_id) REFERENCES AIRPLANE(Airplane_id)
        );
Create Table CUSTOMER(
        Passport_number VARChar(7) not null,
        Customer_name VARCHAR(30),
        Customer_Phone decimal(10,0) Unique,
        E_mail Varchar(30) Unique,
        Country Varchar(20),
        Adress Varchar(60),
        Primary KEy(Passport_number));

Create Table SEAT_RESERVATION(
        Flight_number VARCHAR(9) Not null,
        Leg_number int not null,
        Date Date not null,
        Seat_number int not null,
        Passport_number VARChar(7) not null,
        Check_in BIT,
        Seat_type VarChar(10),
        Seat_Amount float null,
        Primary key(Flight_number,Leg_number,Date,Seat_number,Passport_number),
        Foreign key(Flight_number,Leg_number,Date) REFERENCES LEG_INSTANCE(Flight_number,Leg_number,Date),
        Foreign key(Passport_number) REFERENCES CUSTOMER(Passport_number)
        );

Create table FFC(
        Ffc_id int Identity(1,1) not null,
        Passport_number varchar(7) not null,
        Total_price float,
        Total_mil float,
        Times_Flew int,
        Mil_point float,
        Segment_point float,
        Segment_level varchar(10),
        Primary key(Passport_number,ffc_id),
        Foreign key(Passport_number) REFERENCES CUSTOMER(Passport_number)
        );

ALTER TABLE AIRPLANE_TYPE
ADD Foreign key(Company_id) references COMPANY(Company_id);
```

2. Populate the database you just created again using SQL script file loaded with sample tuples.

-Verilerin sql tasarında denemesinin daha kolay olabilmesi için verileri tablolara elle girmek yerine sql tarafından insert komudunu kullarak verilerin tüm tablolara tek execute ile eklenmesi sağladık. Bu sayede verilerin girilmesindeki zaman kaybını önlemiş olmaktayız.

```
INSERT INTO COMPANY (Company_name, Year_of_foundation)
VALUES
('Türk Hava Yolları','1933'),('Boeing','1916'),
('Pegasus','1990'),('SunExpress','1989'),
('Airbus','1970'),('Anadolu jet','1993');
```

```sql
INSERT INTO AIRPORT (Airport_Code, Name,City,State)
VALUES
('355','Adnan Menderes Havalimanı','izmir','Acik'),('344','Sabiha Gökçen Havalimanı','istanbul','Acik'),
('568','Dubai Uluslararası Havalimanı','Dubai','Acik'),('533','Tokyo Haneda Havalimanı','Tokyo','Acik'),
('343','İstanbul Havalimanı','istanbul','Acik'),('066','Ankara Esenboğa Havalimanı','Ankara','Acik'),
('116','Yenişehir Havalimanı','Bursa','Acik'),('465','LaGuardia Havalimanı','NewYork','Acik');

INSERT INTO AIRPLANE_TYPE (Type_name, Max_seat,Company_id)
VALUES
('787','270','2'),('777','400','2'),('777X','430','2'),('737','170','2'),('A380','300','5'),
('A300','200','5'),('A321','210','5'),('787X','300','2'),('A350','350','5');

INSERT INTO AIRPLANE VALUES('16','280','787X','2');
INSERT INTO AIRPLANE VALUES('25','335','A350','5');
INSERT INTO AIRPLANE VALUES('11','250','787','2');
INSERT INTO AIRPLANE VALUES('12','380','777','2');
INSERT INTO AIRPLANE VALUES('13','380','777','2');
INSERT INTO AIRPLANE VALUES('14','380','777','2');
INSERT INTO AIRPLANE VALUES('21','285','A380','5');
INSERT INTO AIRPLANE VALUES('22','285','A380','5');
INSERT INTO AIRPLANE VALUES('23','285','A380','5');
INSERT INTO AIRPLANE VALUES('15','150','737','2');
INSERT INTO AIRPLANE VALUES('24','200','A321','5');

INSERT INTO AIRLINE (Airline_name, Company_id)
VALUES
('Türk Hava Yolları','1'),('SunExpress Hava Yolları','3'),
('Pegasus Hava Yolları','4'),('Anadolu Jet hava yolları','6');

INSERT INTO CAN_LAND (Airport_code, Type_name)
VALUES
('355','787'),('355','777'),('355','777X'),('344','A380'),('344','787'),('344','777'),('344','A321'),('343','787'),
('343','777'),('343','737'),('343','A380'),('533','777'),('533','777X'),('568','777'),('568','777X'),('066','777'),('0
66','737'),
('066','787'),('066','A380'),('066','A321');

INSERT INTO FLIGHT (Flight_number, Airline_id, Weekdays)
VALUES
('AI111','1','Haftaiçi'),('AI112','2','Haftaiçi'),('AI113','3','Haftaiçi'),
('AI114','1','Haftasonu'),('DT111','1','Haftaiçi'),('TD111','1','Haftaiçi'),
('DT112','1','Haftasonu'),('IA111','2','Haftaiçi'),('BA111','1','Haftaiçi'),('AB111','2','Haftaiçi');

INSERT INTO FLIGHT_LEG VALUES('AI111','355','344','19:05','21:35','350');
INSERT INTO FLIGHT_LEG VALUES('IA111','344','355','22:05','00:35','350');
INSERT INTO FLIGHT_LEG VALUES('DT112','533','568','19:00','05:55','1350');
INSERT INTO FLIGHT_LEG VALUES('TD111','568','533','08:00','15:55','1350');
INSERT INTO FLIGHT_LEG VALUES('AI114','344','066','16:40','20:50','600');
INSERT INTO FLIGHT_LEG VALUES('AI113','343','066','07:00','13:00','590');
INSERT INTO FLIGHT_LEG VALUES('DT111','344','568','11:00','23:55','2000');

INSERT INTO LEG_INSTANCE VALUES('AI111','1','2021-02-02','800','11','355','344','19:05','21:35');
INSERT INTO LEG_INSTANCE VALUES('IA111','2','2021-02-15','900','12','344','355','22:05','00:35');
INSERT INTO LEG_INSTANCE VALUES('DT112','3','2021-02-05','110','11','533','568','19:00','05:55');
INSERT INTO LEG_INSTANCE VALUES('TD111','4','2021-03-18','111','13','568','533','08:00','15:55');
INSERT INTO LEG_INSTANCE VALUES('AI114','5','2021-02-21','115','14','344','066','16:40','20:50');
INSERT INTO LEG_INSTANCE VALUES('AI113','6','2021-04-25','114','21','343','066','07:00','13:00');
INSERT INTO LEG_INSTANCE VALUES('AI111','1','2021-04-02','800','11','355','344','19:05','21:35');
INSERT INTO LEG_INSTANCE VALUES('DT112','3','2021-03-05','110','11','533','568','19:00','05:55');

INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_Phone,E_mail,Country,Adress)
VALUES
('1111111','Mustafa','5312685323','m.hikmet.ozturk@gmail.com','Türkiye','311 sokak no 11'),
('1111112','Gürcan','5682561524','gurcanrk@gmail.com','İngiltere','598 sokak no 1'),
('1111113','Enes','5362541526','enes@gmail.com','Türkiye','999 sokak no 10'),
('1111114','Isa','4352556859','mısa@gmail.com','Türkiye','656 sokak no 2'),
('1111115','Beyza','1524568956','beyza@gmail.com','Almanya','59 sokak no 5'),
('1111116','Miray','3572695162','miray@gmail.com','Türkiye','13 sokak no 8'),
('1111120','Hikmet','5982685457','hikmet.@gmail.com','Türkiye','268 sokak no 111'),
('1111121','Ahmet','5485796268','ahmet.@gmail.com','İngiltere','847 sokak no 10');

INSERT INTO SEAT_RESERVATION VALUES('DT112','3','2021-03-05','3','1111112',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('IA111','2','2021-02-15','2','1111115',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('TD111','4','2021-03-18','1','1111115',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('AI111','1','2021-04-02','1','1111115',1,'Business',0);
INSERT INTO SEAT_RESERVATION VALUES('AI111','1','2021-04-02','2','1111113',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('AI111','1','2021-04-02','3','1111116',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('DT112','3','2021-03-05','1','1111116',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('DT112','3','2021-03-05','4','1111113',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('DT112','3','2021-03-05','6','1111114',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('DT112','3','2021-03-05','7','1111115',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('IA111','2','2021-02-15','1','1111114',0,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('IA111','2','2021-02-15','3','1111116',1,'Business',0);
INSERT INTO SEAT_RESERVATION VALUES('IA111','2','2021-02-15','4','1111111',1,'Business',0);
INSERT INTO SEAT_RESERVATION VALUES('DT112','3','2021-03-05','4','1111113',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('TD111','4','2021-03-18','10','1111111',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('IA111','2','2021-02-15','11','1111111',1,'Economy',0);
INSERT INTO SEAT_RESERVATION VALUES('IA111','2','2021-02-15','5','1111121',1,'Business',0);
```

3. Write down 5 triggers. Triggers should be meaningful.

```
CREATE TRIGGER [dbo].[available_seat]
on [dbo].[LEG_INSTANCE]
after insert
as
begin
        declare @flight_n VarChar(7) declare @leg_n int
        declare @Date date declare @arr int
        declare @dep int declare @arrt time
        declare @dept time declare @available int
        declare @plane_id int declare @total_seat int

        select @flight_n=Flight_number
,@leg_n=Leg_number,@Date=Date,@available=Number_of_available_seats,@arr=Arrival_airport_code,
        @dep=Departure_airport_code,@arrt=Arrival_time,@dept=Departure_time,@plane_id=Airplane_id from inserted
        select @total_seat=AIRPLANE.Total_no_of_seat from AIRPLANE where AIRPLANE.Airplane_id =@plane_id
        BEGIN
        IF (@available>@total_seat)begin
                UPDATE LEG_INSTANCE  SET Number_of_available_seats = @total_seat
                        FROM LEG_INSTANCE  WHERE Flight_number = @flight_n and Leg_number=@leg_n and Date=@date
        END END  end
```

3.1)Bu triggerin amacı eğer leg instance tablosuna uçuşun uçağının yolcu kapasitesinden fazla satılabilir koltuk sayısı girildiğinde bu duruma izin vermez ve uçağınalabileceği max yolcu sayısına göre leg_instance tablosundaki Number_of_available_seats alınını max alabileceği total_seat e göre update işlemini gerçekleştirir.

```
CREATE TRIGGER [dbo].[available_seat_count]
on [dbo].[SEAT_RESERVATION]
after insert as begin
        declare @flight_n VarChar(7)
        declare @leg_n int
        declare @Date date
        declare @available int
        select @flight_n=Flight_number ,@leg_n=Leg_number,@Date=Date from inserted
        select @available=Number_of_available_seats from LEG_INSTANCE where Flight_number =@flight_n
        and leg_number=@leg_n
        and Date=@Date
        BEGIN
        IF(@available>0) begin
                UPDATE LEG_INSTANCE  SET Number_of_available_seats = (Number_of_available_seats-1)
                                FROM LEG_INSTANCE  WHERE Flight_number = @flight_n and Leg_number=@leg_n and
Date=@date
                END
        END
        IF(@available<=0) BEGIN
                print 'uçuş için koltuk kalmadı rezerve edilemez'
        END     end
```

3.2)Bu trigger sayesinde Seat_rezervation yapmak isteyen bir customer eğer o uçuş için Available_seat kalmadıysa bu durumu engeller. Eğer Available bbir seat var ise reservation'a izin verir ve o uçuş için available_seat 'i bir azatır.

```
CREATE TRIGGER [dbo].[FightDateControl] ON [dbo].[SEAT_RESERVATION]
AFTER INSERT
AS
declare @flightnumber varchar(9) declare @bugun date
declare @s_no int declare @leg_no int
declare @flightDate date
select @flightnumber=Flight_number,@leg_no=Leg_number,@flightDate=Date,@s_no=Seat_number from inserted
select @bugun=GETDATE()
BEGIN
IF (@bugun>@flightDate)
```

```
begin
    print 'uçuş tarihi geçmiş rezerve edilemez'
    ROLLBACK TRANSACTION
        Delete from SEAT_RESERVATION where SEAT_RESERVATION.Flight_number=@flightnumber and 
SEAT_RESERVATION.Leg_number=@leg_no and SEAT_RESERVATION.Date=@flightDate and SEAT_RESERVATION.Seat_number=@s_no
        End
```

3.3)Bu trigger sayesinde bir uçuşa rezervation yapmak isteyen bir customer eğer uçuşun tarihi geçmiş ise buna izin vermez ve bir mesaj ile bilgilendirir.

```
CREATE TRIGGER [dbo].[seat_control]
on [dbo].[SEAT_RESERVATION]
instead of insert as begin
        declare @flight_n VarChar(7) declare @leg_n int declare @Date date
        declare @seatnumber int declare @pass varchar(7) declare@Check BIT
        declare @type VarChar(10) declare @price float
        select @flight_n=Flight_number ,@leg_n=Leg_number,@Date=Date,  @seatnumber=Seat_number ,@pass=Passport_number,
@Check=Check_in,@type=Seat_type,@price=Seat_Amount from inserted
        IF EXISTS (SELECT * FROM SEAT_RESERVATION WHERE @flight_n=Flight_number and @leg_n=Leg_number and 
@seatnumber=Seat_number and @Date=Date)
        BEGIN
        print('bu koltuk dolu')
        END
        else
        INSERT INTO SEAT_RESERVATION VALUES(@flight_n,@leg_n,@Date,@seatnumber,@pass,@Check,@type,@price);
```

3.4)Bu trigger sayesinde bir uçuşa seat_rezervation yapmak isteyen bir customer eğer o seat için başka bir customer o seat için rezervation yapmış ise bu duruma izin vermez ve bu koltuk dolu diye bir mesaj ile bilgilendirir.

```
CREATE TRIGGER [dbo].[company_control1]
on [dbo].[AIRLINE]
after insert
as
begin
        declare @compid int
        declare @lineid int
        declare @linename varchar(30)
        select @compid=Company_id,@lineid=Airline_id, @linename=Airline_name From inserted
        IF  EXISTS(select * from AIRLINE as al, AIRPLANE as ap, AIRPLANE_TYPE as apt where al.Company_id=ap.Company_id 
and al.Company_id=apt.Company_id)
        Begin
                        ROLLBACK TRANSACTION
                        print 'Bu company airline companysi olamaz.';
        end
end
CREATE TRIGGER [dbo].[company_control2]
on [dbo].[AIRPLANE]
after insert
as
begin
        declare @compid int
        declare @lineid int
        declare @linename varchar(30)
        IF  EXISTS(select * from AIRLINE as al, AIRPLANE as ap where al.Company_id=ap.Company_id )
        Begin
                        ROLLBACK TRANSACTION
                        print 'Bu company airplane companysi olamaz.';

        end
END
CREATE TRIGGER [dbo].[company_control3]
on [dbo].[AIRPLANE_TYPE]
after insert
as
begin
        declare @compid int
        declare @lineid int
        declare @linename varchar(30)
        IF  EXISTS(select * from AIRLINE as al, AIRPLANE_TYPE as apt where al.Company_id=apt.Company_id )
        Begin
                        ROLLBACK TRANSACTION
                        print 'Bu companynın bir airplane_typesi olamaz.';
        end     END
```

3.5)Bu 3 trigger sayesinde airline tablosuna bir kayıt eklerken eğer eklenen company_id airplane veya airplane_type tablosunda var ise bu duruma izin vermez. Ayn şekilde Airplane ve Airplane_Type tablosuna bir kayıt eklenmek istendiğinde Company_id Airline tablosundada var ise bu durumada izin vermez ve hata mesajı ile bilgilendirme yapar.

```
CREATE TRIGGER [dbo].[Add_fares]
   ON  [dbo].[FLIGHT_LEG]
AFTER INSERT AS  BEGIN
        declare @Milleage int declare @fNumber varchar(9)
        declare @airline varchar(30) declare @week varchar(20)
        select @fNumber=Flight_number from inserted
        select @Milleage=Milleage from FLIGHT_LEG  where @fNumber=Flight_number
        select @week=FLIGHT.Weekdays from FLIGHT,FLIGHT_LEG where FLIGHT.Flight_number=@fNumber
        select @airline=AIRLINE.Airline_name from AIRLINE,FLIGHT where AIRLINE.Airline_id=FLIGHT.Airline_id and 
FLIGHT.Flight_number=@fNumber
        BEgin
        IF(@week='Haftaiçi') begin
                IF(@airline='Türk Hava Yolları') begin
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,0,@Milleage*0.75,'Economy')
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,1,@Milleage*2,'Business')
                end
                        ELSE IF(@airline='SunExpress Hava Yolları') begin
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,0,@Milleage*0.70,'Economy')
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,1,@Milleage*1.80,'Business')
                end
                        ELSE IF(@airline='Pegasus Hava Yolları') begin
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,0,@Milleage*0.72,'Economy')
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,1,@Milleage*1.90,'Business')
                end
        END
        ELSE IF(@week='Haftasonu') begin
                IF(@airline='Türk Hava Yolları') begin
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,0,@Milleage*0.80,'Economy')
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,1,@Milleage*2.10,'Business')
                end
                        ELSE IF(@airline='SunExpress Hava Yolları') begin
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,0,@Milleage*0.75,'Economy')
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,1,@Milleage*1.90,'Business')
                end
                        ELSE IF(@airline='Pegasus Hava Yolları') begin
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,0,@Milleage*0.77,'Economy')
                        Insert into FARES(Flight_number,Fare_code,Amonuts,Restrictions) 
Values(@fNumber,1,@Milleage*2,'Business')
                end
        END end
```

3.6)Bu trigger sayesinden FLİGHT_leg tablosuna bir uçuş ayağı eklendiğinde o uçuşun hangi havayolu şirketi tarafından hafta içi mi hafta sonu mu gerçekleştirdiğine bağlı olarak Fares tablosuna otomatik business ve economy fiyatlarını Milleage değerine göre otomatik doldurma işlemini gerçekleştirir.

```
CREATE TRIGGER [dbo].[price_ffc_control]
on [dbo].[SEAT_RESERVATION]
after insert
as
begin
        declare @pass VarChar(7) declare @f_no Varchar(9) declare @dates date
        declare @leg_no int declare @seat_no int declare @cek bit
        declare @koltuk varchar(10) declare @mill float declare @price float
        declare @times float declare @level Varchar(10)
        select 
@f_no=Flight_number,@leg_no=Leg_number,@dates=Date,@seat_no=Seat_number,@pass=Passport_number,@cek=Check_in,@koltuk=Se
at_type from inserted
        select @price=fa.Amonuts from fares as fa,SEAT_RESERVATION as s,LEG_INSTANCE as leg,FLIGHT_LEG as fl, FLIGHT 
as f
        where @f_no=leg.Flight_number and @leg_no=leg.Leg_number and @dates=leg.Date
```

```
    and leg.Flight_number=fl.Flight_number and leg.Leg_number=fl.Leg_number
    and fl.Flight_number=f.Flight_number
    and f.Flight_number=fa.Flight_number
    and @koltuk=fa.Restrictions
    Update SEAT_RESERVATION  set Seat_Amount=@price From SEAT_RESERVATION as s where s.Flight_number=@f_no and 
s.Leg_number=@leg_no and s.Date=@dates and s.Seat_number=@seat_no
        Select @level='Start', @mill=SUM(fl.Milleage), @times=COUNT(fl.Milleage), @price=SUM(s.Seat_Amount)
        from SEAT_RESERVATION as s,LEG_INSTANCE as leg,FLIGHT_LEG as fl
        where s.Check_in=1
        and s.Date=leg.Date and s.Flight_number = leg.Flight_number and s.Leg_number = leg.Leg_number and 
leg.Flight_number = fl.Flight_number
        and leg.Leg_number = fl.Leg_number and s.Passport_number=@pass
        GROUP BY s.Passport_number

    IF EXISTS(Select * FROM FFC WHERE FFC.Passport_number=@pass)
    BEGIN
    UPDATE FFC SET FFC.total_mil = @mill, FFC.Times_Flew=@times ,FFC.Total_price=@price, 
FFC.Mil_point=((((@mill*0.1)+(@price*0.1))*(@times/2))*0.3),
    FFC.Segment_point=(((@mill*0.1)+(@price*0.1))*(@times/2))
        FROM FFC WHERE FFC.Passport_number = @pass
    END
    ELSE IF(@cek=0) begin
        print 'Check_in yapılmamıstır.';
    end
    Else IF NOT EXISTS(Select * FROM FFC WHERE FFC.Passport_number=@pass)
    Insert into FFC(Passport_number,Total_price,Total_mil,Times_Flew,Mil_point,Segment_point,Segment_level)
    Values(@pass,@price,@mill,@times,((((@mill*0.1)+(@price*0.1))*(@times/2))*0.3),(((@mill*0.1)+(@price*0.1))*(@t
imes/2)),@level)
END
```

3.7)Bu trigger sayesinde Seat_reservation yapılır ise Fares tablosu aracılığı ile Seat_reservation tablosuna uçuşun fiyatı çekilir. Bunun sebebi mil_pointleri olan customerların uçuşları üzerinde istediği zaman indirim alabilmesidir. Aynı zamanda seat yapıldığında ve check_in yapıldıysa FFC tablosuna o customerin bilgilerini kaç defa uçtuğunu, toplam kaç para harcadığını, indirim alabileceği mil_point tl bazında ve segmentleri bulunmaktadır. Eğer bir customer ffc tablosunda var ise üzerinde güncelleme yapmaktadır.

```
CREATE TRIGGER [dbo].[seat_update]
on [dbo].[SEAT_RESERVATION]
after update
as
begin
    declare @pass VarChar(7) declare @f_no Varchar(9) declare @dates date declare @leg_no int
    declare @seat_no int declare @cek bit declare @koltuk varchar(10) declare @mill float
    declare @price float declare @times float declare @level Varchar(10)
    select 
@f_no=Flight_number,@leg_no=Leg_number,@dates=Date,@seat_no=Seat_number,@pass=Passport_number,@cek=Check_in,@koltuk=Se
at_type from inserted
        Select @level='Start', @mill=SUM(fl.Milleage),
        @times=COUNT(fl.Milleage), @price=SUM(s.Seat_Amount)
        from SEAT_RESERVATION as s,LEG_INSTANCE as leg,FLIGHT_LEG as fl
        where @cek=1
        and s.Date=leg.Date
        and s.Flight_number = leg.Flight_number and s.Leg_number = leg.Leg_number
        and leg.Flight_number = fl.Flight_number and leg.Leg_number = fl.Leg_number
        and s.Passport_number=@pass
        GROUP BY s.Passport_number
    IF EXISTS(Select * FROM FFC WHERE FFC.Passport_number=@pass and @cek=1 )
    BEGIN
    UPDATE FFC SET FFC.total_mil = @mill, FFC.Times_Flew=@times ,FFC.Total_price=@price, 
FFC.Mil_point=((((@mill*0.1)+(@price*0.1))*(@times/2))*0.3),FFC.Segment_point=(((@mill*0.1)+(@price*0.1))*(@times/2))
        FROM FFC WHERE FFC.Passport_number = @pass
    END
    ELSE IF(@cek=0) begin
        print 'Check_in yapılmamıstır.';
    end
    Else IF NOT EXISTS(Select * FROM FFC WHERE FFC.Passport_number=@pass and @cek=1)
    Insert into FFC(Passport_number,Total_price,Total_mil,Times_Flew,Mil_point,Segment_point,Segment_level)
    Values(@pass,@price,@mill,@times,((((@mill*0.1)+(@price*0.1))*(@times/2))*0.3),(((@mill*0.1)+(@price*0.1))*(@t
imes/2)),@level)
END
```

3.8)Bu trigger eğer bir customer seat_reservation yapmış fakat check in yapmamış ise daha sonradan check_in yaptığında yine ffc tablosunda gerekli kayıtların güncellemesi veya eklenmesi için gerekli olan triggerdir.

```
CREATE TRIGGER [dbo].[segment_control]
```

```
on [dbo].[FFC]
after insert,update
as
begin
        declare @pass VarChar(7)
        declare @point float
        select @pass=Passport_number,@point=Segment_point from inserted
        IF(@point>300)BEGIN
                update FFC SET FFC.Segment_level='BRONZ' where FFC.Passport_number=@pass
        END
        IF(@point>800)BEGIN
                update FFC SET FFC.Segment_level='SILVER' where FFC.Passport_number=@pass
        END
        IF(@point>1500)BEGIN
                update FFC SET FFC.Segment_level='GOLD' where FFC.Passport_number=@pass
        END
        IF(@point>5000)BEGIN
                update FFC SET FFC.Segment_level='PLATINUM' where FFC.Passport_number=@pass
        END
        IF(@point>15000)BEGIN
                update FFC SET FFC.Segment_level='DIAMOND' where FFC.Passport_number=@pass
        END

End
```

3.9)Son Trigger ise FFC tablosundaki Segment_ponint değerine göre segment level'ini belirlemek için gerekli triggerdir.

4. Write down 5 check constraints. Check constraints should be meaningful.

```
USE [DB_AIRLINE]
GO
--Flight leglerdeki milleage değeri negatif bir değer girilemez kontrolünü yaptıran constraint
1.
Alter table FLIGHT_LEG
add constraint chc_Milleage
check(Milleage>0)
2.
--LEG_INSTANCE Number_of_available_seats değeri negatif değer alamaz
Alter table LEG_INSTANCE
add constraint chc_ava_seat
check(Number_of_available_seats>0)
3.
--AIRPLANE deki total seat alanı 0 den büyük olmalı
Alter table AIRPLANE
add constraint chc_Seat
check(AIRPLANE.Total_no_of_seat>0)
4.
--AIRPLANE_TYPE daki seat alanı 0 dan büyük olmalı
Alter table AIRPLANE_TYPE
add constraint chc_Seat_max
check(Max_seat>0)
5.
--flight_leg de arrival_airport_code ile departure_airport_code eşit olmamalı
Alter table FLIGHT_LEG
add constraint chc_arr_dep
check(Departure_airport_code!=Arrival_airport_code)
--flight_leg de arrival_airport_code ile departure_airport_code eşit olmamalı
Alter table LEG_INSTANCE
add constraint chc_dep_arr
check(Departure_airport_code!=Arrival_airport_code)
6.
--COMPANY tablosundaki name alanı unique olmalı
Alter table COMPANY
add constraint chc_name
unique(Company_name)
7.
--Customer country alanını boş bıraktığı zaman Türkiye olarak otomatik doldurmaktadır.
Alter table Customer
add constraint chc_country
default ('Türkiye') For Country
```

Bu constraintler sayesinde veri tutarsızlığı bir önlemeyi amaçladık;

1. Constraint sayesinde Flight_leg tablosundaki milleage attribute lerine negatif bir değer girilemiyor.
2. Constraint sayesinde LEG_ınstancedeki NUmber_of_available_seat attribute lerine negatif değer girilemiyor.

3. Constraint sayesinde AIRPlane tablosundaki total_no_of_seat alanına negatif değer girilemiyor.
4. Constraint sayesinde AIRPLANE_TYPE tablosundaki max_seat alanına negatif değer girilemiyor.
5. Constraint sayesinde Flight_leg deki kalkış ve varış airport codeları aynı olmamalıdır.
6. Constraint sayesinde company tablosundaki name alanı unique olmalıdır.
7. Constraint sayesinde customer country alanını boş bırakırsa Türkiye olarak otomatik doldurmaktadır.

Bunların dışında Tabloları oluştururkende birden fazla constraintlerden yararlandık. Bunlar sayesinde verilerin daha tutarlı olmasını sağladık.

5. Write down the following SQL statements:

5.a) INSERT-DELETE-UPDATE

```
--CUSTOMER
INSERT INTO CUSTOMER (Passport_number,Customer_name,Customer_Phone,E_mail,Country,Adress)
VALUES
('1111117','Betül','5184765948','betul@gmail.com','Türkiye','1 sokak no 1'),
('1111118','Ayse','5289631578','Ayse@gmail.com','Ispanya','3 sokak no 9');

DELETE FROM CUSTOMER WHERE Customer_name='Hikmet';

Update CUSTOMER
SET Customer_name='Mustafa Hikmet'
WHERE Customer_name='Mustafa'
--COMPANY
INSERT INTO COMPANY (Company_name, Year_of_foundation)
VALUES
('FLY','2021');

Update COMPANY
SET Year_of_foundation='2022'
WHERE Year_of_foundation='2021'

DELETE FROM COMPANY Where Year_of_foundation>2021
--SEat_RESERVATİON
INSERT INTO SEAT_RESERVATION VALUES('AI111','1','2021-04-02','8','1111112',0,'Business',0);

UPDATE  SEAT_RESERVATION_
set Check_in = 1
WHERE Flight_number='AI111'
and Leg_number='1'
and Date='2021-04-02'
and Passport_number='1111112'

DELETE FROM SEAT_RESERVATION WHERE Check_in=0 and Date<GETDATE();
--FARES
INSERT INTO FARES (Flight_number, Amonuts,Restrictions)
VALUES
('IA111','400','Economy');

DELETE FROM FARES WHERE Amonuts<0 or Amonuts>100000;

UPDATE  FARES_
SET     Amonuts = Amonuts *1.2_
WHERE Flight_number IN (SELECT FLIGHT.Flight_number_FROM FLIGHT WHERE Weekdays='Haftasonu')
--FLİGHT_LEG
INSERT INTO FLIGHT_LEG
(Flight_number,Departure_airport_code,Arrival_airport_code,Scheduled_departure_time,Scheduled_arrival_time,Milleage)
VALUES
('AB111','355','344','19:05','21:35','100');

DELETE FROM FLIGHT_LEG WHERE Milleage<101; DELETE FROM FARES WHERE FARES.Flight_number='AB111'

UPDATE FLIGHT_LEG
SET Milleage=Milleage*1.120
WHERE Departure_airport_code IN(SELECT AIRPORT.Airport_Code_FROM AIRPORT WHERE AIRPORT.City='Tokyo')
--LEG_INSTANCE
INSERT INTO LEG_INSTANCE
(Flight_number,Leg_number,Date,Number_of_available_seats,Airplane_id,Departure_airport_code,Arrival_airport_code,Depar
ture_time,Arrival_time)
VALUES
('AI113','6','2021-04-25','114','21','343','066','07:00','13:00');

Update LEG_INSTANCE  SET Date='2021-01-20' Where Date='2021-04-25';

DELETE FROM LEG_INSTANCE WHERE DATE<GetDate()
```

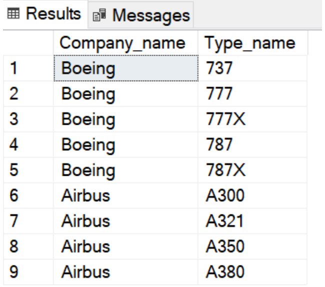
Bu kısımda 5 farklı tablo için insert-delete ve update sorgularını örnekledik.

Kullandığımız tablolar; CUSTOMER, COMPANY, SEAT_RESETVATİON, FARES, FLİGHT_LEG, LEG_İNSTANCE

5.b)min 2, min 3, min 4 tablo kullanarak SELECT komutunun örneklenmesi

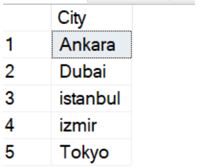
5.b.1)Min2

```
--Uçak şirketinin uçaklarının isimleri
SELECT COMPANY.Company_name,AIRPLANE_TYPE.Type_name
FROM AIRPLANE_TYPE,COMPANY
WHERE AIRPLANE_TYPE.Company_id=COMPANY.Company_id
GROUP BY COMPANY.Company_name,AIRPLANE_TYPE.Type_name
```



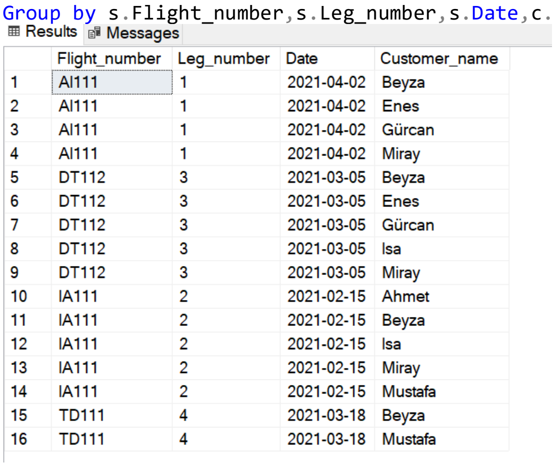
	Company_name	Type_name
1	Boeing	737
2	Boeing	777
3	Boeing	777X
4	Boeing	787
5	Boeing	787X
6	Airbus	A300
7	Airbus	A321
8	Airbus	A350
9	Airbus	A380

```
--uçak tipi 777 olan uçak hangi şehirlere uçmaktadır.
SELECT DISTINCT AIRPORT.City
FROM AIRPORT,CAN_LAND
WHERE CAN_LAND.Type_name='777'
AND CAN_LAND.Airport_code=AIRPORT.Airport_Code
```



	City
1	Ankara
2	Dubai
3	istanbul
4	izmir
5	Tokyo

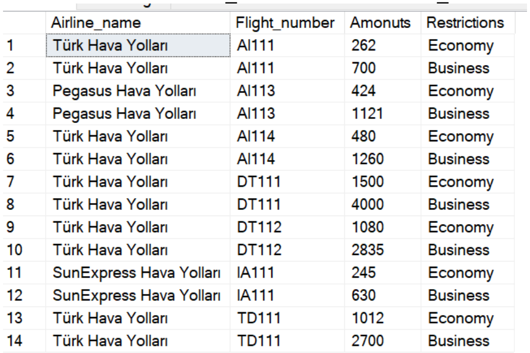
```
--seat yapan yolcuların isimleri,hangi seat'e rezervasyon yaptıkları
SELECT s.Flight_number,s.Leg_number,s.Date,c.Customer_name
FROM CUSTOMER as c,SEAT_RESERVATION AS s
WHERE s.Passport_number=c.Passport_number
Group by s.Flight_number,s.Leg_number,s.Date,c.Customer_name
```



	Flight_number	Leg_number	Date	Customer_name
1	AI111	1	2021-04-02	Beyza
2	AI111	1	2021-04-02	Enes
3	AI111	1	2021-04-02	Gürcan
4	AI111	1	2021-04-02	Miray
5	DT112	3	2021-03-05	Beyza
6	DT112	3	2021-03-05	Enes
7	DT112	3	2021-03-05	Gürcan
8	DT112	3	2021-03-05	Isa
9	DT112	3	2021-03-05	Miray
10	IA111	2	2021-02-15	Ahmet
11	IA111	2	2021-02-15	Beyza
12	IA111	2	2021-02-15	Isa
13	IA111	2	2021-02-15	Miray
14	IA111	2	2021-02-15	Mustafa
15	TD111	4	2021-03-18	Beyza
16	TD111	4	2021-03-18	Mustafa

5.b.2)Min3

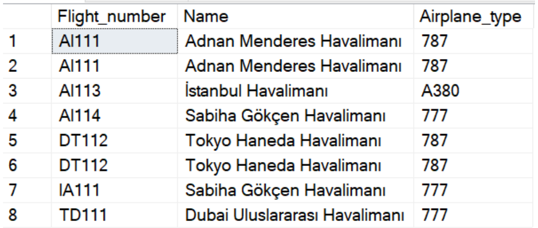
```
--Uçusun  hangi havayolu şireketi tarafından kaç paraya düzenlendiğini gösteren sorgu
Select AIRLINE.Airline_name,FLIGHT.Flight_number,Fares.Amonuts,Fares.Restrictions
From AIRLINE,FLIGHT, FARES
Where Fares.Flight_number=FLIGHT.Flight_number
and AIRLINE.Airline_id=FLIGHT.Airline_id
```



	Airline_name	Flight_number	Amonuts	Restrictions
1	Türk Hava Yolları	AI111	262	Economy
2	Türk Hava Yolları	AI111	700	Business
3	Pegasus Hava Yolları	AI113	424	Economy
4	Pegasus Hava Yolları	AI113	1121	Business
5	Türk Hava Yolları	AI114	480	Economy
6	Türk Hava Yolları	AI114	1260	Business
7	Türk Hava Yolları	DT111	1500	Economy
8	Türk Hava Yolları	DT111	4000	Business
9	Türk Hava Yolları	DT112	1080	Economy
10	Türk Hava Yolları	DT112	2835	Business
11	SunExpress Hava Yolları	IA111	245	Economy
12	SunExpress Hava Yolları	IA111	630	Business
13	Türk Hava Yolları	TD111	1012	Economy
14	Türk Hava Yolları	TD111	2700	Business

```
-- leg_intanceta satıla bilir koltyuk olan uçuşların uçuş numaları hangi havalimanından ve hangi uçaktipi ile
kalktığını gösteren sorgu
SELECT LEG_INSTANCE.Flight_number,AIRPORT.Name,AIRPLANE.Airplane_type
FROm AIRPORT ,LEG_INSTANCE, AIRPLANE
Where LEG_INSTANCE.Number_of_available_seats>3
```

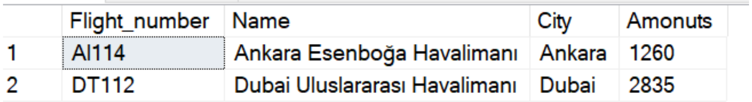
```
and LEG_INSTANCE.Departure_airport_code=AIRPORT.Airport_Code
        and AIRPLANE.Airplane_id=LEG_INSTANCE.Airplane_id
```



	Flight_number	Name	Airplane_type
1	AI111	Adnan Menderes Havalimanı	787
2	AI111	Adnan Menderes Havalimanı	787
3	AI113	İstanbul Havalimanı	A380
4	AI114	Sabiha Gökçen Havalimanı	777
5	DT112	Tokyo Haneda Havalimanı	787
6	DT112	Tokyo Haneda Havalimanı	787
7	IA111	Sabiha Gökçen Havalimanı	777
8	TD111	Dubai Uluslararası Havalimanı	777

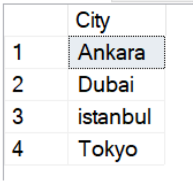
5.b.3)Min 4

```
--Hafta sonu ve busines olan uçuşların uçuş numaraları, hangi şehre ve kaç para ile uçtuklarını  gösteren sorgu
Select FARES.Flight_number, AIRPORT.Name,AIRPORT.City,FARES.Amonuts
from FARES ,FLIGHT_LEG,FLIGHT,AIRPORT
where FLIGHT_LEG.Flight_number IN (Select FLIGHT.Flight_number from FLIGHT where Weekdays='Haftasonu')
And FLIGHT_LEG.Flight_number=FLIGHT.Flight_number
And FLIGHT.Flight_number=FARES.Flight_number
And FLIGHT_LEG.Arrival_airport_code=AIRPORT.Airport_Code
and FARES.Restrictions='Business'
```



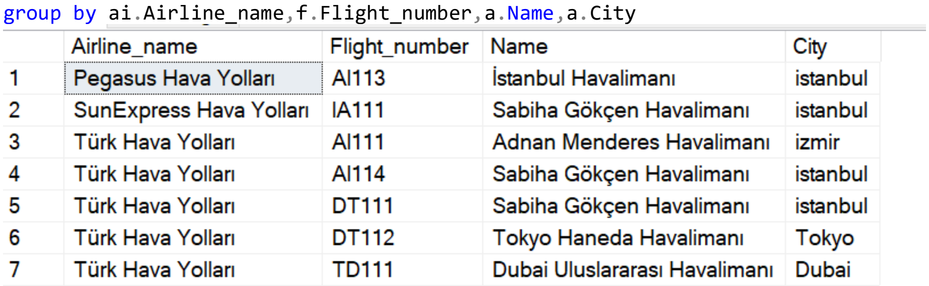
	Flight_number	Name	City	Amonuts
1	AI114	Ankara Esenboğa Havalimanı	Ankara	1260
2	DT112	Dubai Uluslararası Havalimanı	Dubai	2835

```
--Türk hava yolları hangi sehirlere uçus yapmaktadır.
Select DISTINCT AIRPORT.City
FROM AIRPORT,FLIGHT,AIRLINE,FLIGHT_LEG
WHERE AIRLINE.Airline_name='Türk Hava Yolları'
And AIRLINE.Airline_id=FLIGHT.Airline_id
AND FLIGHT.Flight_number=FLIGHT_LEG.Flight_number
AND FLIGHT_LEG.Arrival_airport_code=AIRPORT.Airport_Code
```



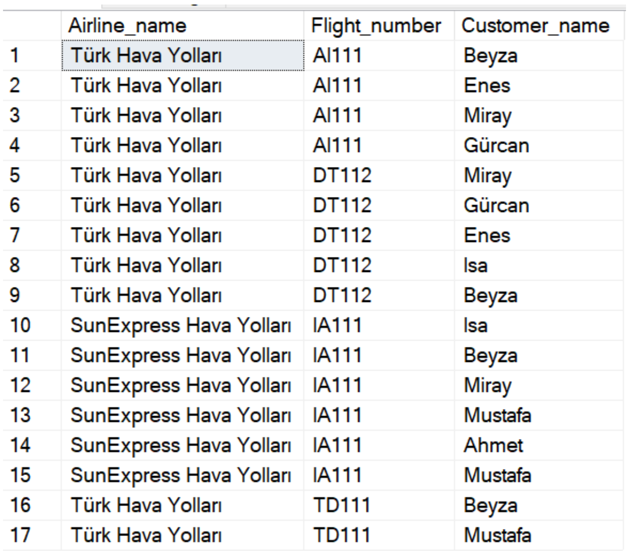
	City
1	Ankara
2	Dubai
3	istanbul
4	Tokyo

```
--havayolu şirketlerinin hangi şehirlerden,hangi uçuş numarası ile ve hangi havalimanllarından kalktığını gösteren sorgu
SELECT ai.Airline_name,f.Flight_number,a.Name,a.City
FROM AIRLINE as ai,FLIGHT as f,FLIGHT_LEG as fl,AIRPORT as a
Where ai.Airline_id=f.Airline_id
and f.Flight_number=fl.Flight_number
and fl.Departure_airport_code=a.Airport_Code
group by ai.Airline_name,f.Flight_number,a.Name,a.City
```



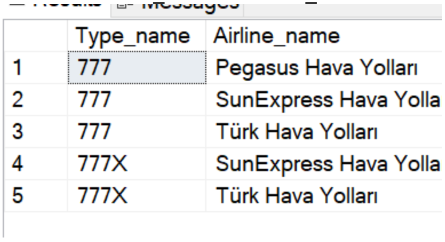
	Airline_name	Flight_number	Name	City
1	Pegasus Hava Yolları	AI113	İstanbul Havalimanı	istanbul
2	SunExpress Hava Yolları	IA111	Sabiha Gökçen Havalimanı	istanbul
3	Türk Hava Yolları	AI111	Adnan Menderes Havalimanı	izmir
4	Türk Hava Yolları	AI114	Sabiha Gökçen Havalimanı	istanbul
5	Türk Hava Yolları	DT111	Sabiha Gökçen Havalimanı	istanbul
6	Türk Hava Yolları	DT112	Tokyo Haneda Havalimanı	Tokyo
7	Türk Hava Yolları	TD111	Dubai Uluslararası Havalimanı	Dubai

```
--rezerve edilen uçuşlar hangi uçuş şirketi tarafından düzenlenmektedır ,hangi uçuş ile uçmakta  ve uçan kişiler kimlerdir.
Select al.Airline_name,seat.Flight_number,cus.Customer_name
FROM AIRLINE as al, FLIGHT as f, FLIGHT_LEG as fl, LEG_INSTANCE as leg, SEAT_RESERVATION as seat,CUSTOMER as cus
WHERE al.Airline_id=f.Airline_id
and f.Flight_number=fl.Flight_number
and fl.Flight_number=leg.Flight_number
and fl.Leg_number=leg.Leg_number
and leg.Flight_number=seat.Flight_number
and leg.Leg_number=seat.Leg_number
and leg.Date=seat.Date
and seat.Passport_number=cus.Passport_number
```



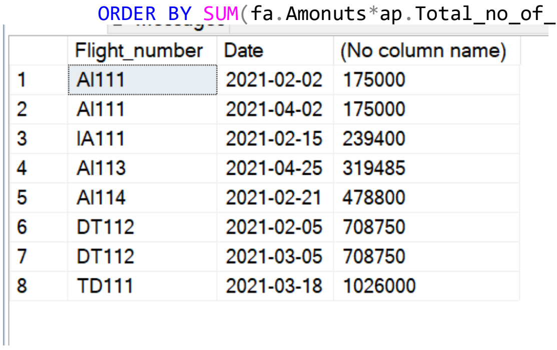
	Airline_name	Flight_number	Customer_name
1	Türk Hava Yolları	AI111	Beyza
2	Türk Hava Yolları	AI111	Enes
3	Türk Hava Yolları	AI111	Miray
4	Türk Hava Yolları	AI111	Gürcan
5	Türk Hava Yolları	DT112	Miray
6	Türk Hava Yolları	DT112	Gürcan
7	Türk Hava Yolları	DT112	Enes
8	Türk Hava Yolları	DT112	Isa
9	Türk Hava Yolları	DT112	Beyza
10	SunExpress Hava Yolları	IA111	Isa
11	SunExpress Hava Yolları	IA111	Beyza
12	SunExpress Hava Yolları	IA111	Miray
13	SunExpress Hava Yolları	IA111	Mustafa
14	SunExpress Hava Yolları	IA111	Ahmet
15	SunExpress Hava Yolları	IA111	Mustafa
16	Türk Hava Yolları	TD111	Beyza
17	Türk Hava Yolları	TD111	Mustafa

```
--uçak kapasitesi 300 den fazla olan uçakların hangi havayolu şirketi tarafından düzenlenen uçuşlarda
uçmuştur.
Select DISTINCT aırt.Type_name,a.Airline_name
From AIRPLANE_TYPE as aırt, CAN_LAND as can,AIRPORT as por,FLIGHT as f,FLIGHT_LEG as fl,AIRLINE as a
Where aırt.Max_seat>300
and aırt.Type_name = can.Type_name
and can.Airport_code=por.Airport_Code
and por.Airport_Code=fl.Arrival_airport_code
and fl.Flight_number=f.Flight_number
and f.Airline_id=a.Airline_id
```



	Type_name	Airline_name
1	777	Pegasus Hava Yolları
2	777	SunExpress Hava Yolla
3	777	Türk Hava Yolları
4	777X	SunExpress Hava Yolla
5	777X	Türk Hava Yolları

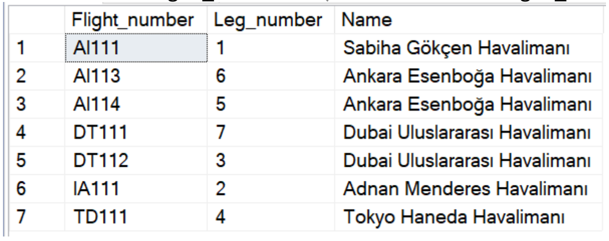
```
--Bir uçuş için tüm biletleri business olarak satan bir havayolu şirketi o uçuştan toplam kaç para kazanır.
Select  leg.Flight_number,leg.Date,SUM(fa.Amonuts*ap.Total_no_of_seat)
From FARES as fa,FLIGHT as f,FLIGHT_LEG as fl, LEG_INSTANCE as leg,AIRPLANE as ap
WHERE fa.Flight_number=f.Flight_number
and f.Flight_number=fl.Flight_number
and fl.Leg_number=leg.Leg_number
and leg.Airplane_id=ap.Airplane_id
and fa.Restrictions='Business'
GROUP BY leg.Leg_number,leg.Flight_number,leg.Date
        ORDER BY SUM(fa.Amonuts*ap.Total_no_of_seat) asc
```



	Flight_number	Date	(No column name)
1	AI111	2021-02-02	175000
2	AI111	2021-04-02	175000
3	IA111	2021-02-15	239400
4	AI113	2021-04-25	319485
5	AI114	2021-02-21	478800
6	DT112	2021-02-05	708750
7	DT112	2021-03-05	708750
8	TD111	2021-03-18	1026000

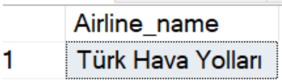
5.c) nested and/or correlated nested queries

```
--fiyatı 400 den yüksek olan uçuşların nereye uçtukları
Select FLIGHT_LEG.Flight_number,FLIGHT_LEG.Leg_number,AIRPORT.Name
FROM AIRPORT,FLIGHT_LEG,FLIGHT WHERE AIRPORT.Airport_Code=FLIGHT_LEG.Arrival_airport_code
and FLIGHT_LEG.Flight_number=FLIGHT.Flight_number
and FLIGHT.Flight_number IN(Select FARES.Flight_number From FARES Where FARES.Amonuts>'400')
```



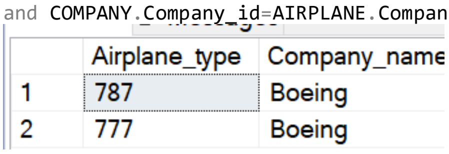
	Flight_number	Leg_number	Name
1	AI111	1	Sabiha Gökçen Havalimanı
2	AI113	6	Ankara Esenboğa Havalimanı
3	AI114	5	Ankara Esenboğa Havalimanı
4	DT111	7	Dubai Uluslararası Havalimanı
5	DT112	3	Dubai Uluslararası Havalimanı
6	IA111	2	Adnan Menderes Havalimanı
7	TD111	4	Tokyo Haneda Havalimanı

```
--1000mill den fazla uçuşlar yapan uçuş şirketlerinin isimleri
Select distinct AIRLINE.Airline_name
From AIRLINE,FLIGHT
Where AIRLINE.Airline_id=FLIGHT.Airline_id
and FLIGHT.Flight_number IN(Select FLIGHT_LEG.Flight_number FROm FLIGHT_LEG where FLIGHT_LEG.Milleage>'1000')
```



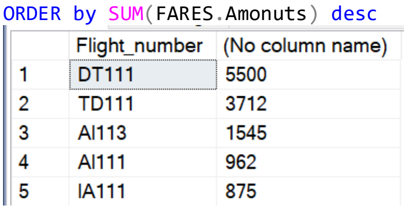
	Airline_name
1	Türk Hava Yolları

```
--%50sın den fazla seat_reservetion yapılmış olan uçuşların hangi uçak ile seyahet ettiklerini ve hangi uçak
şirketi tarafından yapıldığını gösteren sorgu
Select AIRPLANE.Airplane_type,COMPANY.Company_name
FROM AIRPLANE,COMPANY
WHERE AIRPLANE.Airplane_id IN(Select LEG_INSTANCE.Airplane_id FROM LEG_INSTANCE,AIRPLANE WHERE
(AIRPLANE.Total_no_of_seat*0.5)<LEG_INSTANCE.Number_of_available_seats and
LEG_INSTANCE.Airplane_id=AIRPLANE.Airplane_id)
and COMPANY.Company_id=AIRPLANE.Company_id
```



	Airplane_type	Company_name
1	787	Boeing
2	777	Boeing

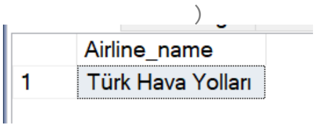
```
--busines ve economy toplam fiyatlarının haftaiçi uçuşların max fiyatlıya göre sıralanması
Select FARES.Flight_number,SUM(FARES.Amonuts)
FROm FARES
WHERE FARES.Flight_number IN (Select FLIGHT.Flight_number FROM FLIGHT WHERE FLIGHT.Weekdays='Haftaiçi')
Group by FARES.Flight_number
ORDER by SUM(FARES.Amonuts) desc
```



	Flight_number	(No column name)
1	DT111	5500
2	TD111	3712
3	AI113	1545
4	AI111	962
5	IA111	875

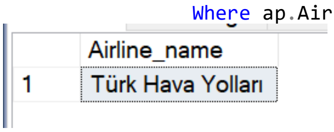
5.d)EXİSTS ve NOT EXİSTS kullanımı

```
--uçuş tarihi geçmiş olan uçuşların hangi havayolu şirketine bağlı olduğunu ve hangi uçuş olduğunu gösteren
sorgu(EXİST)
Select AIRLINE.Airline_name
From AIRLINE
Where  EXISTS(select * from FLIGHT as f,FLIGHT_LEG as fl,LEG_INSTANCE as leg Where GETDATE()>leg.Date
                and leg.Leg_number=fl.Leg_number
                and leg.Flight_number=fl.Flight_number
                and fl.Flight_number=f.Flight_number
                and f.Airline_id=AIRLINE.Airline_id
                )
```



	Airline_name
1	Türk Hava Yolları

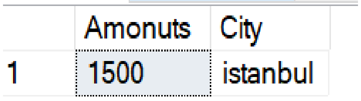
```
--Leg instancede uçuşu olmayan ve can_landı olamayan uçakların id ve typeları(NOT EXISTS)
Select ap.Airplane_id,ap.Airplane_type
From AIRPLANE as ap
Where NOT EXISTS(
                select *
                from LEG_INSTANCE as leg
                Where leg.Airplane_id=ap.Airplane_id)
        AND
                NOT EXISTS(
                select *
                from AIRPLANE_TYPE as apt, CAN_LAND
                Where ap.Airplane_type=CAN_LAND.Type_name)
```



	Airline_name
1	Türk Hava Yolları

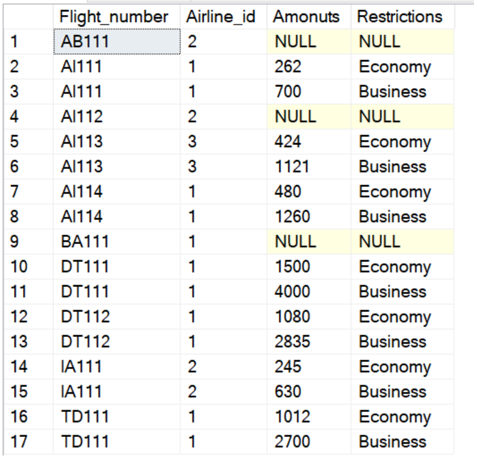
5.e)JOİNLERİN KULLANIMI

```
        --Haftaiçi economy clasında dubaiye gitmek isteyen biri hangi şehirden kaç para ile gider.
Select Fares.Amonuts,AIRPORT.City
From FARES,FLIGHT_LEG  INNER JOIN AIRPORT ON AIRPORT.Airport_Code=FLIGHT_LEG.Departure_airport_code
WHERE FARES.Flight_number IN (Select FLIGHT.Flight_number from FLIGHT where Weekdays='Haftaiçi')
and FLIGHT_LEG.Flight_number=FARES.Flight_number
and FLIGHT_LEG.Arrival_airport_code IN (Select AIRPORT.Airport_Code  from AIRPORT where City='Dubai')
and FARES.Restrictions='economy';
```



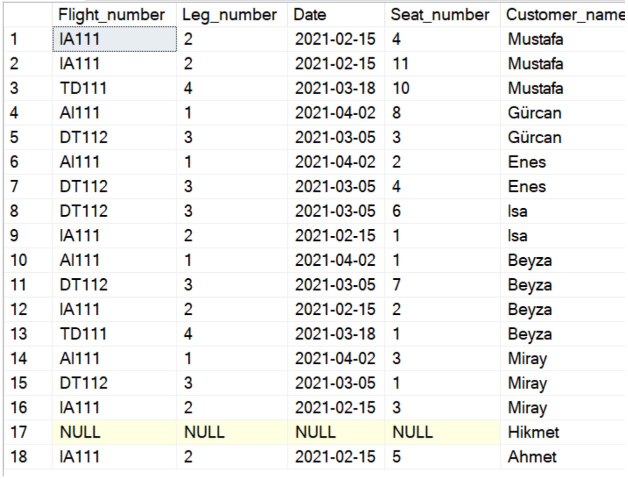
	Amonuts	City
1	1500	istanbul

```
--Left Outer join kullanımı uçuşları olup amountları belli olmayan uçuşları görüntülemekteyiz.(LEFT OUTER
JOİN)
Select FLIGHT.Flight_number,Airline_id,Amonuts,Restrictions
From FLIGHT LEFT OUTER JOIN FARES
ON FLIGHT.Flight_number=FARES.Flight_number
```



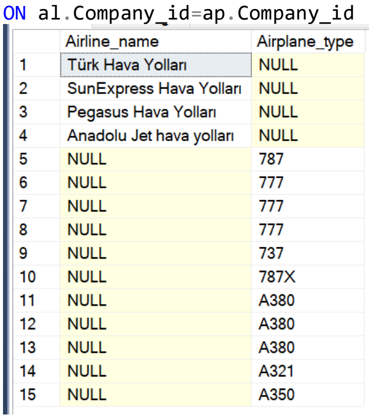
	Flight_number	Airline_id	Amonuts	Restrictions
1	AB111	2	NULL	NULL
2	AI111	1	262	Economy
3	AI111	1	700	Business
4	AI112	2	NULL	NULL
5	AI113	3	424	Economy
6	AI113	3	1121	Business
7	AI114	1	480	Economy
8	AI114	1	1260	Business
9	BA111	1	NULL	NULL
10	DT111	1	1500	Economy
11	DT111	1	4000	Business
12	DT112	1	1080	Economy
13	DT112	1	2835	Business
14	IA111	2	245	Economy
15	IA111	2	630	Business
16	TD111	1	1012	Economy
17	TD111	1	2700	Business

```
--RIGHT OUTER JOIN kullanımı seat reservation yapan customerlerin hangi customer hhangi seat_reservationları
yaptığı ve hangi customerların seat_reservation yapmadığını göstermek için kullandım(RİGHT OUTER JOİN)
Select sr.Flight_number,sr.Leg_number,sr.Date,sr.Seat_number,cus.Customer_name
From SEAT_RESERVATION as sr RIGHT OUTER JOIN CUSTOMER as cus
ON sr.Passport_number=cus.Passport_number
```



	Flight_number	Leg_number	Date	Seat_number	Customer_name
1	IA111	2	2021-02-15	4	Mustafa
2	IA111	2	2021-02-15	11	Mustafa
3	TD111	4	2021-03-18	10	Mustafa
4	AI111	1	2021-04-02	8	Gürcan
5	DT112	3	2021-03-05	3	Gürcan
6	AI111	1	2021-04-02	2	Enes
7	DT112	3	2021-03-05	4	Enes
8	DT112	3	2021-03-05	6	Isa
9	IA111	2	2021-02-15	1	Isa
10	AI111	1	2021-04-02	1	Beyza
11	DT112	3	2021-03-05	7	Beyza
12	IA111	2	2021-02-15	2	Beyza
13	TD111	4	2021-03-18	1	Beyza
14	AI111	1	2021-04-02	3	Miray
15	DT112	3	2021-03-05	1	Miray
16	IA111	2	2021-02-15	3	Miray
17	NULL	NULL	NULL	NULL	Hikmet
18	IA111	2	2021-02-15	5	Ahmet

```
--Full Outer join bir airline şirketi aynı zamanda airplane şirketi olamıyacağı için Full outer join ile
airline in airplane typede NULL airplane type airline de NULL değerini çekiyor.(FULL OUTER JOİN)
Select al.Airline_name,ap.Airplane_type
From AIRLINE as al FULL OUTER JOIN AIRPLANE as ap
ON al.Company_id=ap.Company_id
```

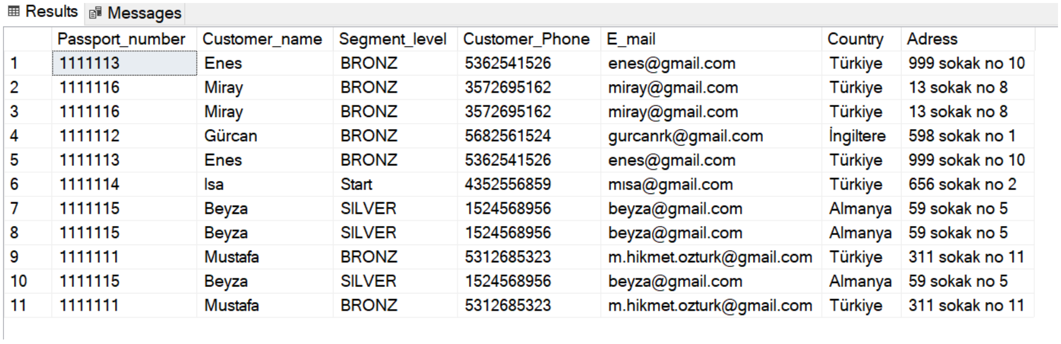


	Airline_name	Airplane_type
1	Türk Hava Yolları	NULL
2	SunExpress Hava Yolları	NULL
3	Pegasus Hava Yolları	NULL
4	Anadolu Jet hava yolları	NULL
5	NULL	787
6	NULL	777
7	NULL	777
8	NULL	777
9	NULL	737
10	NULL	787X
11	NULL	A380
12	NULL	A380
13	NULL	A380
14	NULL	A321
15	NULL	A350

6. Create 5 views that are reasonable

View kısmı koddaki kalabalık görüntünün kaybolmasını sağlamaktadır. Bir select sorgusunu view içerisine alarak daha görüntülemek istediğinizde select from view_name yaptığınızda view içerisine gömülü olan select komutlarını çalıştıracaktır.

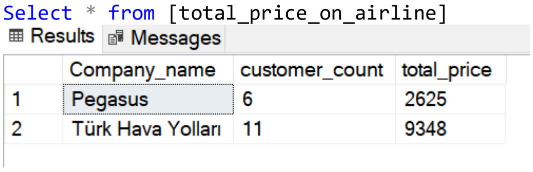
```
create view [dbo].[customer_segment]
as
SELECT        dbo.Economy.Passport_number, dbo.CUSTOMER.Customer_name, dbo.FFC.Segment_level,
dbo.CUSTOMER.Customer_Phone, dbo.CUSTOMER.E_mail, dbo.CUSTOMER.Country, dbo.CUSTOMER.Adress
FROM            dbo.Economy INNER JOIN
                         dbo.FFC ON dbo.Economy.Passport_number = dbo.FFC.Passport_number INNER JOIN
                         dbo.CUSTOMER ON dbo.FFC.Passport_number = dbo.CUSTOMER.Passport_number
GO
Select * from [customer_segment]
```



Results | Messages

	Passport_number	Customer_name	Segment_level	Customer_Phone	E_mail	Country	Adress
1	1111113	Enes	BRONZ	5362541526	enes@gmail.com	Türkiye	999 sokak no 10
2	1111116	Miray	BRONZ	3572695162	miray@gmail.com	Türkiye	13 sokak no 8
3	1111116	Miray	BRONZ	3572695162	miray@gmail.com	Türkiye	13 sokak no 8
4	1111112	Gürcan	BRONZ	5682561524	gurcanrk@gmail.com	İngiltere	598 sokak no 1
5	1111113	Enes	BRONZ	5362541526	enes@gmail.com	Türkiye	999 sokak no 10
6	1111114	Isa	Start	4352556859	mısa@gmail.com	Türkiye	656 sokak no 2
7	1111115	Beyza	SILVER	1524568956	beyza@gmail.com	Almanya	59 sokak no 5
8	1111115	Beyza	SILVER	1524568956	beyza@gmail.com	Almanya	59 sokak no 5
9	1111111	Mustafa	BRONZ	5312685323	m.hikmet.ozturk@gmail.com	Türkiye	311 sokak no 11
10	1111115	Beyza	SILVER	1524568956	beyza@gmail.com	Almanya	59 sokak no 5
11	1111111	Mustafa	BRONZ	5312685323	m.hikmet.ozturk@gmail.com	Türkiye	311 sokak no 11

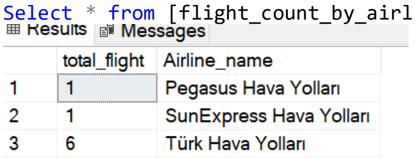
```
create view [dbo].[total_price_on_airline]
as
SELECT         dbo.COMPANY.Company_name, COUNT(dbo.SEAT_RESERVATION.Passport_number) AS customer_count,
SUM(dbo.SEAT_RESERVATION.Seat_Amount) AS total_price
FROM             dbo.COMPANY INNER JOIN
                          dbo.AIRLINE ON dbo.COMPANY.Company_id = dbo.AIRLINE.Company_id INNER JOIN
                          dbo.FLIGHT ON dbo.AIRLINE.Airline_id = dbo.FLIGHT.Airline_id INNER JOIN
                          dbo.SEAT_RESERVATION ON dbo.FLIGHT.Flight_number = dbo.SEAT_RESERVATION.Flight_number
GROUP BY dbo.COMPANY.Company_name
GO
Select * from [total_price_on_airline]
```



Results | Messages

	Company_name	customer_count	total_price
1	Pegasus	6	2625
2	Türk Hava Yolları	11	9348

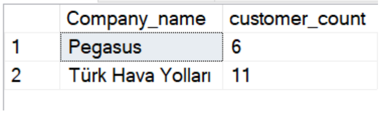
```
create view [dbo].[flight_count_by_airline]
as
SELECT         COUNT(dbo.LEG_INSTANCE.Flight_number) AS total_flight, dbo.AIRLINE.Airline_name
FROM             dbo.FLIGHT INNER JOIN
                          dbo.AIRLINE ON dbo.FLIGHT.Airline_id = dbo.AIRLINE.Airline_id INNER JOIN
                          dbo.FLIGHT_LEG ON dbo.FLIGHT.Flight_number = dbo.FLIGHT_LEG.Flight_number INNER JOIN
                          dbo.LEG_INSTANCE ON dbo.FLIGHT_LEG.Flight_number = dbo.LEG_INSTANCE.Flight_number AND
dbo.FLIGHT_LEG.Leg_number = dbo.LEG_INSTANCE.Leg_number
GROUP BY dbo.AIRLINE.Airline_name
GO
Select * from [flight_count_by_airline]
```



Results | Messages

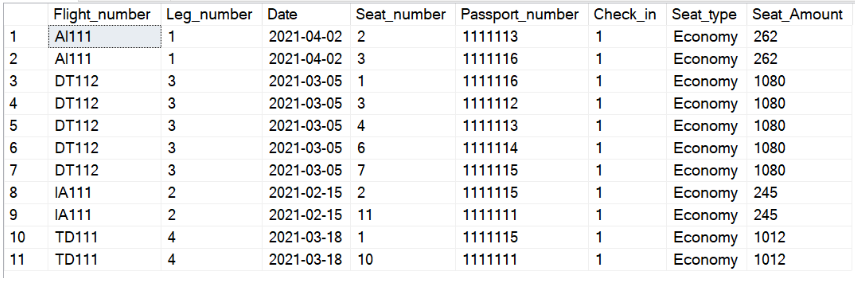
	total_flight	Airline_name
1	1	Pegasus Hava Yolları
2	1	SunExpress Hava Yolları
3	6	Türk Hava Yolları

```
create view [dbo].[total_customer_on_airline]
as
SELECT         dbo.COMPANY.Company_name, COUNT(dbo.SEAT_RESERVATION.Passport_number) AS customer_count
FROM             dbo.COMPANY INNER JOIN
                          dbo.AIRLINE ON dbo.COMPANY.Company_id = dbo.AIRLINE.Company_id INNER JOIN
                          dbo.FLIGHT ON dbo.AIRLINE.Airline_id = dbo.FLIGHT.Airline_id INNER JOIN
                          dbo.SEAT_RESERVATION ON dbo.FLIGHT.Flight_number = dbo.SEAT_RESERVATION.Flight_number
GROUP BY dbo.COMPANY.Company_name
GO
Select * from [total_customer_on_airline]
```



	Company_name	customer_count
1	Pegasus	6
2	Türk Hava Yolları	11

```
Create view [dbo].[Economy]
as
select * from SEAT_RESERVATION Where Check_in =1 and Seat_type='Economy'
GO
Select * from [Economy]
```



	Flight_number	Leg_number	Date	Seat_number	Passport_number	Check_in	Seat_type	Seat_Amount
1	AI111	1	2021-04-02	2	1111113	1	Economy	262
2	AI111	1	2021-04-02	3	1111116	1	Economy	262
3	DT112	3	2021-03-05	1	1111116	1	Economy	1080
4	DT112	3	2021-03-05	3	1111112	1	Economy	1080
5	DT112	3	2021-03-05	4	1111113	1	Economy	1080
6	DT112	3	2021-03-05	6	1111114	1	Economy	1080
7	DT112	3	2021-03-05	7	1111115	1	Economy	1080
8	IA111	2	2021-02-15	2	1111115	1	Economy	245
9	IA111	2	2021-02-15	11	1111111	1	Economy	245
10	TD111	4	2021-03-18	1	1111115	1	Economy	1012
11	TD111	4	2021-03-18	10	1111111	1	Economy	1012

PART-III APPLICATION DEVELOPMENT (20/100 points)

1. Customer segmentation için FFc tablosunda ve seat_reservation tablosunda attributelar ekledik. Bu sayede fazla uçan yolculara ekstra ayrıcalıklar vererek daha fazla uçuş yapmalarını teşvik etmek istedik. Bu ayrıcalıklar mil_pointleri ile uçak fiyatlarında istedikleri zaman indirim almaları hatta yeterince biriktirdiklerinde ücretsiz uçuş yapabilirler. Customer segmentation için genişlettiğpimiz alanlar aşağıdaki tablolarda daha ayrıntılı bir şekilde gözlemlenmektedir.

FFC

Ffc_id	Passport_number	Total_price	Total_mil	Mil_point	Segment_point	Segment_level

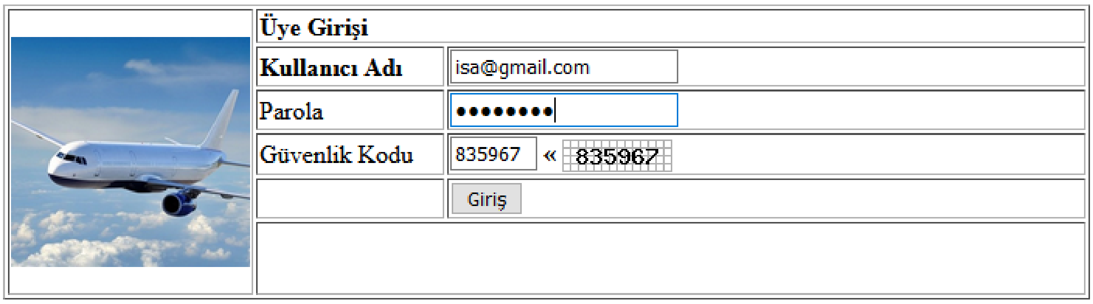
Seat_reservation

Flight_number	Leg_number	Date	Seat_number	Passport_number	Check_in	Seat_type	Seat_Amount

Seat_reservation ve FFc tabloların son halleride bu şekildeydi şimdi Rest api ile tasarladığımız customer segment demosu.

2. REST APİ VE WEB UYGULAMASI

Üyeler mil puanlarını ve diğer uçuş bilgileri için web sayfasındaki kişiler bilgilere ulaşabilmek için **Görsel 1'**deki gibi kullanıcı mail adresini ve parolasını girdikten sonra **Görsel 2'**deki gibi üye bilgileri görüntülenmektedir.



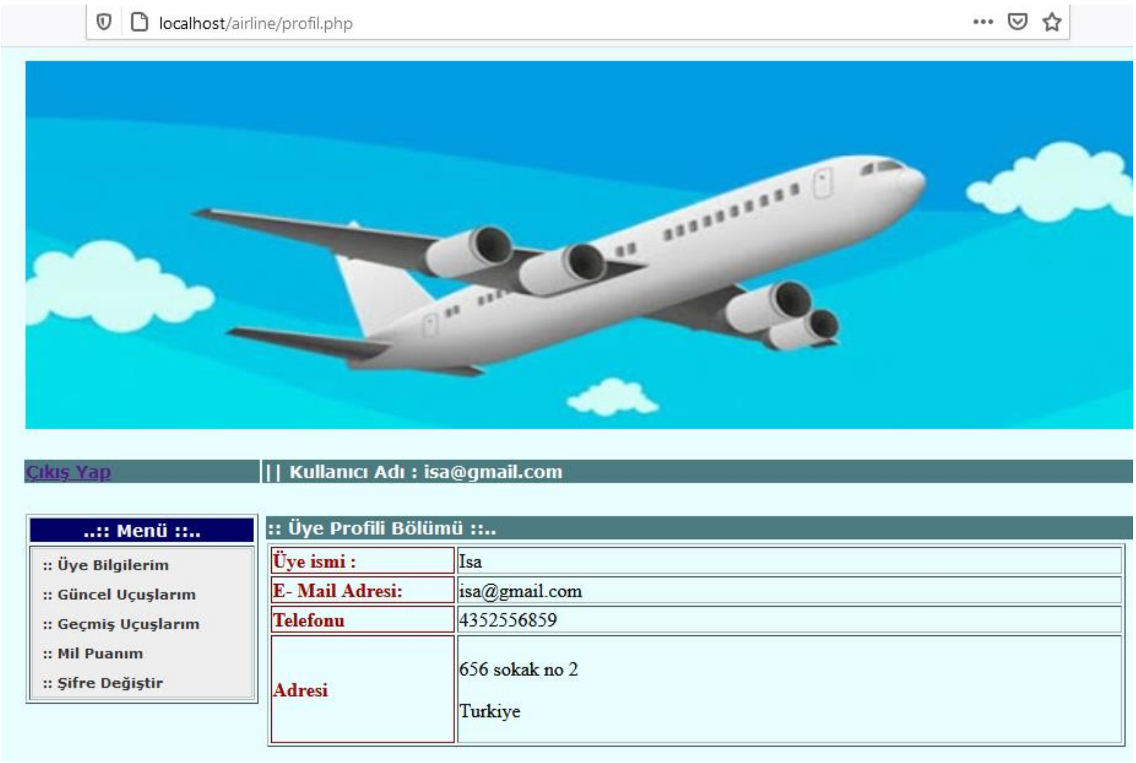
Üye Girişi

Kullanıcı Adı: isa@gmail.com

Parola: ••••••••

Güvenlik Kodu: 835967 « 835967

Görsel 1: Üye girişi bölümü



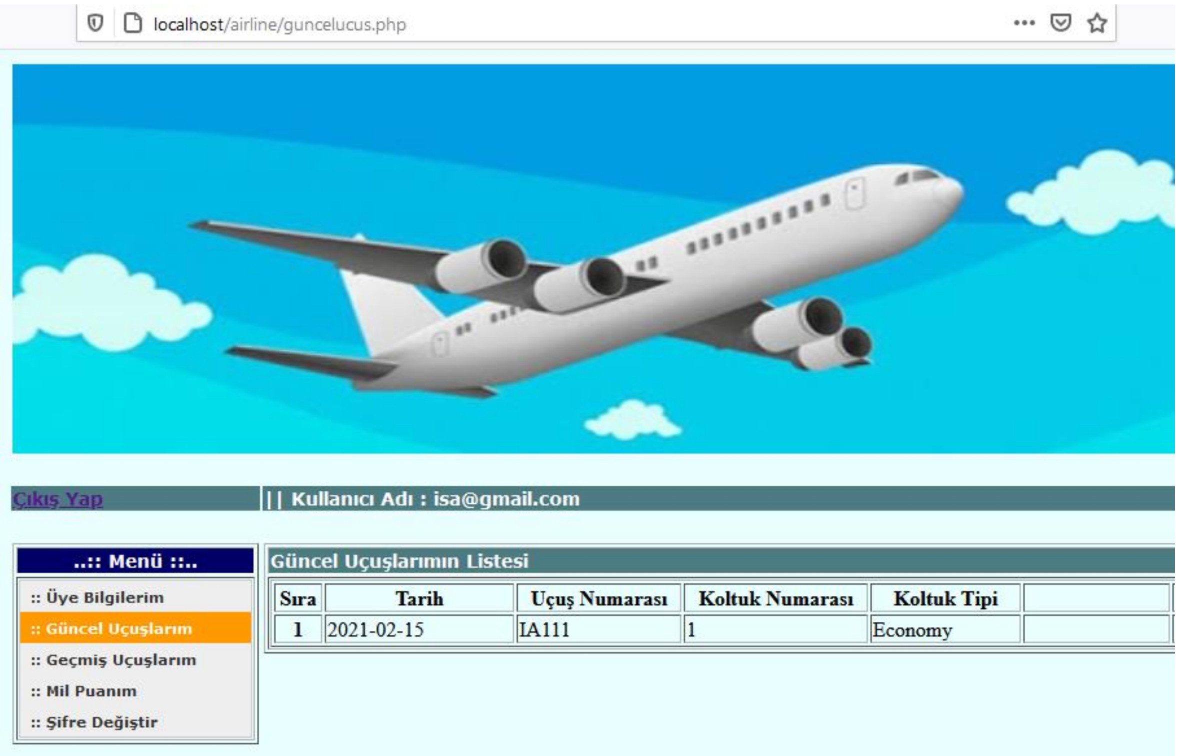
localhost/airline/profil.php

|| Kullanıcı Adı : isa@gmail.com

:: Üye Profili Bölümü ::..

Üye ismi :	Isa
E- Mail Adresi:	isa@gmail.com
Telefonu	4352556859
Adresi	656 sokak no 2 Turkiye

Görsel 2: Üye bilgileri bölümü

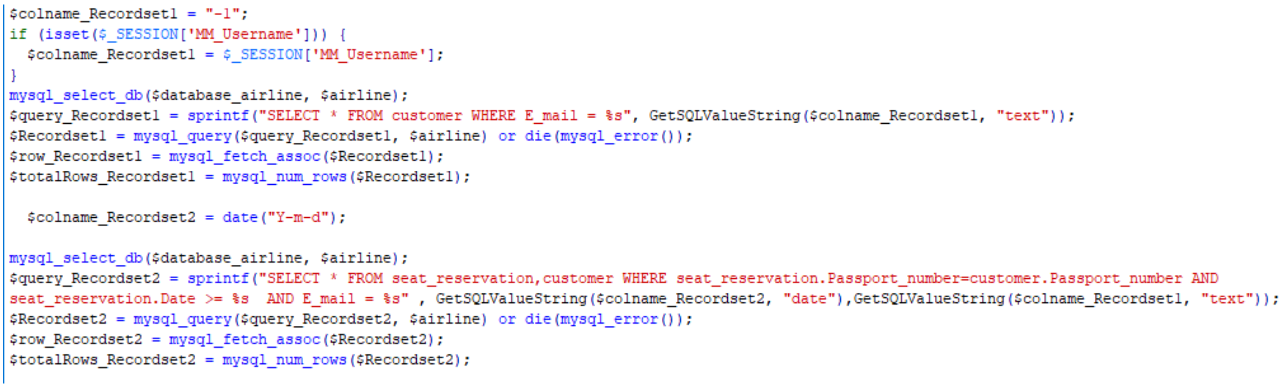


|| Kullanıcı Adı : isa@gmail.com

Güncel Uçuşlarımın Listesi

Sıra	Tarih	Uçuş Numarası	Koltuk Numarası	Koltuk Tipi	
1	2021-02-15	IA111	1	Economy	

Görsel 3: Güncel uçuşlarım bölümü

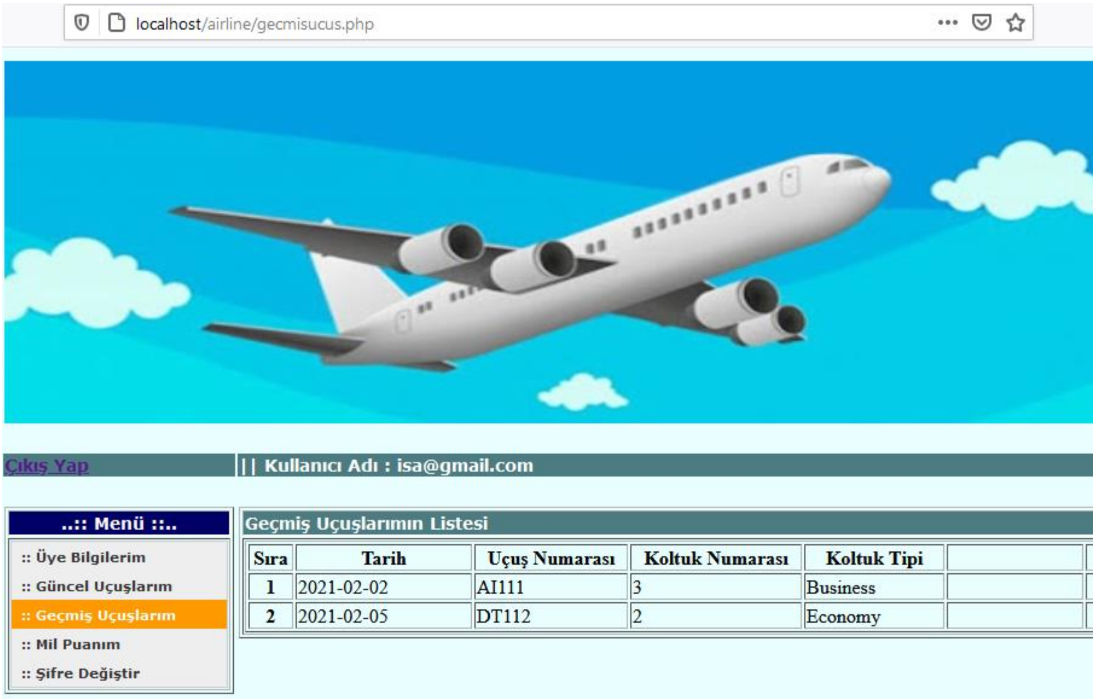


```
$colname_Recordset1 = "-1";
if (isset($_SESSION['MM_Username'])) {
  $colname_Recordset1 = $_SESSION['MM_Username'];
}
mysql_select_db($database_airline, $airline);
$query_Recordset1 = sprintf("SELECT * FROM customer WHERE E_mail = %s", GetSQLValueString($colname_Recordset1, "text"));
$Recordset1 = mysql_query($query_Recordset1, $airline) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);

  $colname_Recordset2 = date("Y-m-d");

mysql_select_db($database_airline, $airline);
$query_Recordset2 = sprintf("SELECT * FROM seat_reservation,customer WHERE seat_reservation.Passport_number=customer.Passport_number AND
seat_reservation.Date >= %s  AND E_mail = %s" , GetSQLValueString($colname_Recordset2, "date"),GetSQLValueString($colname_Recordset1, "text"));
$Recordset2 = mysql_query($query_Recordset2, $airline) or die(mysql_error());
$row_Recordset2 = mysql_fetch_assoc($Recordset2);
$totalRows_Recordset2 = mysql_num_rows($Recordset2);
```

Görsel 4: Güncel uçuş bilgileri için PHP ve SQL sorgu kodları

Görsel 3'te Güncel uçuşlarım bölümü bulunmaktadır. Bu bölümde üyenin günü geçmemiş güncel uçuş bilgileri bulunmaktadır. **Görsel 4**'te bu bilgileri göstermek için gerekli PHP ve SQL sorgu kodları görülmektedir.

$query_Recordset2 = sprintf("SELECT * FROM seat_reservation,customer WHERE seat_reservation.Passport_number=customer.Passport_number AND seat_reservation.Date >= %s AND E_mail = %s" , GetSQLValueString($colname_Recordset2, "date"),GetSQLValueString($colname_Recordset1, "text"));

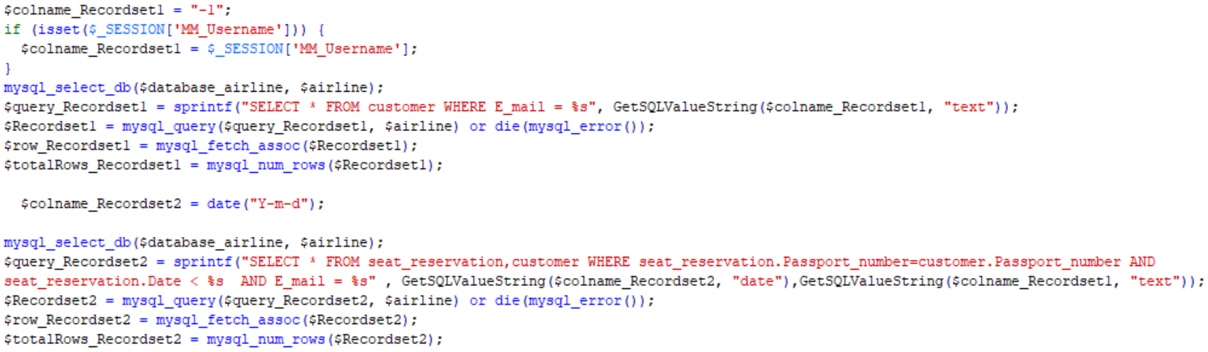


|| Kullanıcı Adı : isa@gmail.com

Geçmiş Uçuşlarımın Listesi

Sıra	Tarih	Uçuş Numarası	Koltuk Numarası	Koltuk Tipi	
1	2021-02-02	AI111	3	Business	
2	2021-02-05	DT112	2	Economy	

Görsel 5: Geçmiş uçuşlarım bölümü

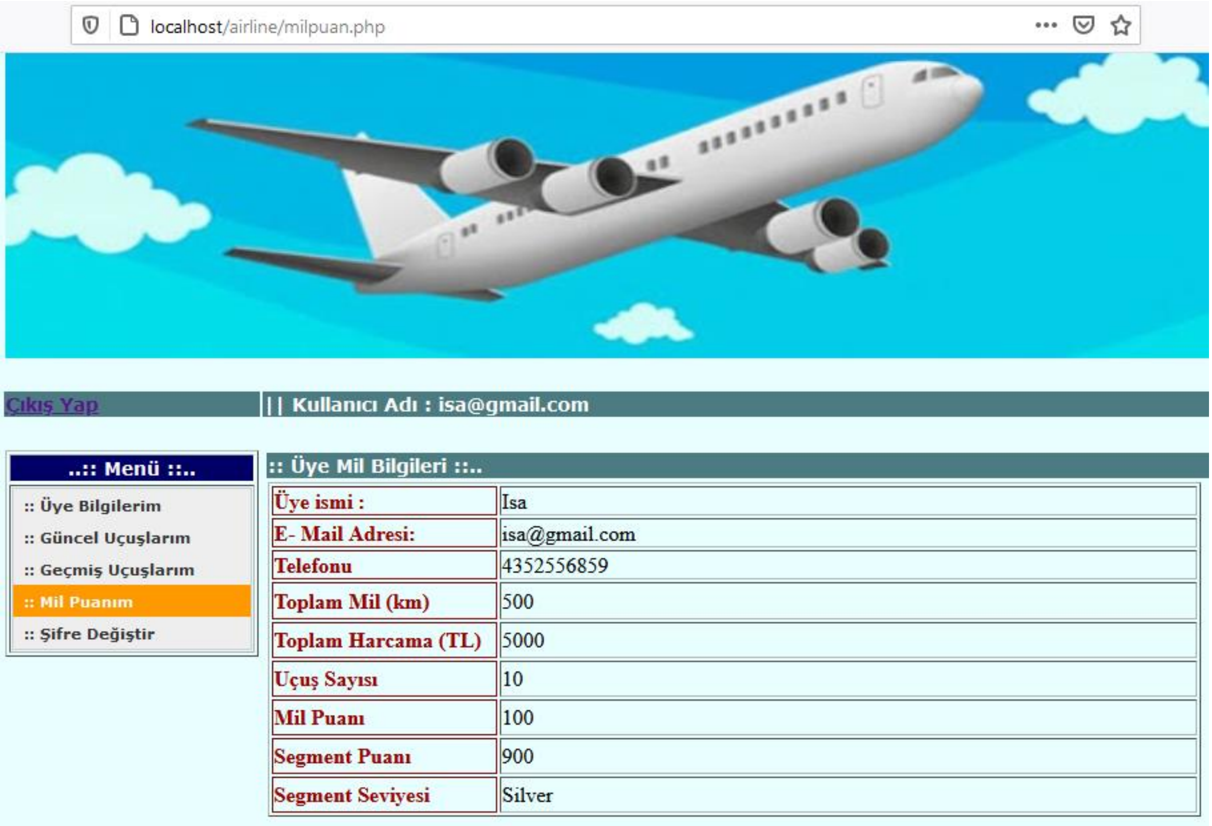


```
$colname_Recordset1 = "-1";
if (isset($_SESSION['MM_Username'])) {
  $colname_Recordset1 = $_SESSION['MM_Username'];
}
mysql_select_db($database_airline, $airline);
$query_Recordset1 = sprintf("SELECT * FROM customer WHERE E_mail = %s", GetSQLValueString($colname_Recordset1, "text"));
$Recordset1 = mysql_query($query_Recordset1, $airline) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);

  $colname_Recordset2 = date("Y-m-d");

mysql_select_db($database_airline, $airline);
$query_Recordset2 = sprintf("SELECT * FROM seat_reservation,customer WHERE seat_reservation.Passport_number=customer.Passport_number AND
seat_reservation.Date < %s  AND E_mail = %s" , GetSQLValueString($colname_Recordset2, "date"),GetSQLValueString($colname_Recordset1, "text"));
$Recordset2 = mysql_query($query_Recordset2, $airline) or die(mysql_error());
$row_Recordset2 = mysql_fetch_assoc($Recordset2);
$totalRows_Recordset2 = mysql_num_rows($Recordset2);
```

Görsel 6: Geçmiş uçuş bilgileri için PHP ve SQL sorgu kodları

Görsel 5'te Geçmiş uçuşlarım bölümü bulunmaktadır. Bu bölümde üyenin günü geçmiş eski uçuş bilgileri bulunmaktadır. **Görsel 6**'da bu bilgileri göstermek için gerekli PHP ve SQL sorgu kodları görülmektedir.

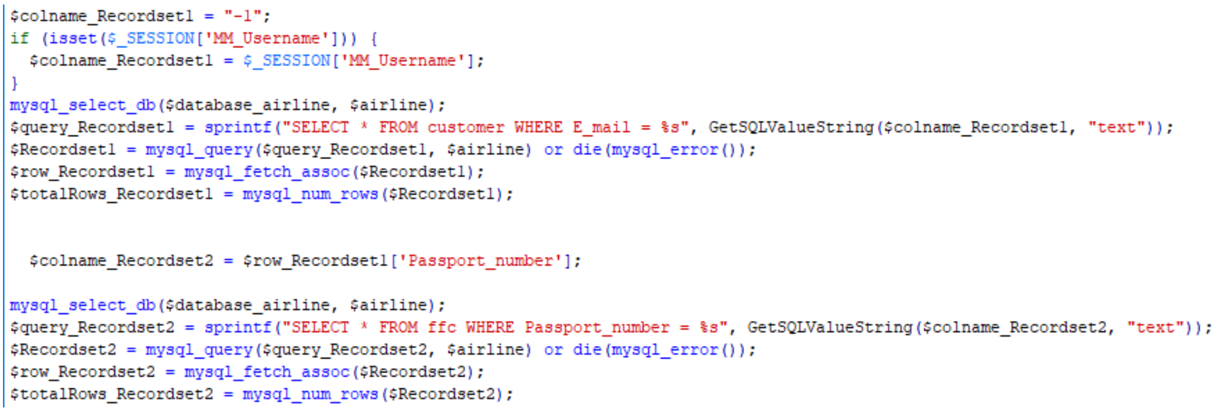
$query_Recordset2 = sprintf("SELECT * FROM seat_reservation,customer WHERE seat_reservation.Passport_number=customer.Passport_number AND seat_reservation.Date < %s AND E_mail = %s" , GetSQLValueString($colname_Recordset2, "date"),GetSQLValueString($colname_Recordset1, "text"));



:: Üye Mil Bilgileri ::..

Üye ismi :	Isa
E- Mail Adresi:	isa@gmail.com
Telefonu	4352556859
Toplam Mil (km)	500
Toplam Harcama (TL)	5000
Uçuş Sayısı	10
Mil Puanı	100
Segment Puanı	900
Segment Seviyesi	Silver

Görsel 7: Üye mil ve puan bilgileri bölümü



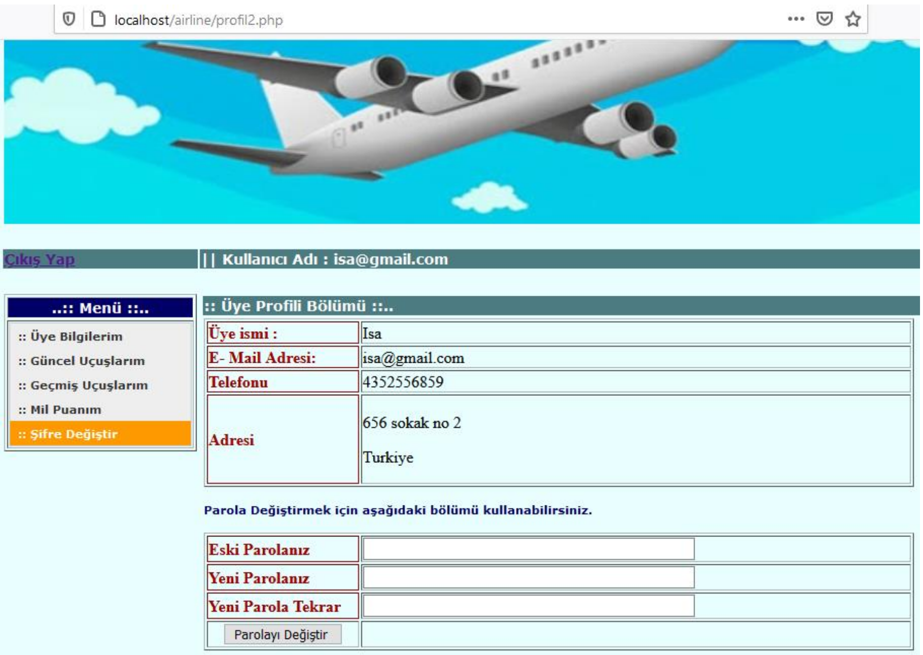
```
$colname_Recordset1 = "-1";
if (isset($_SESSION['MM_Username'])) {
  $colname_Recordset1 = $_SESSION['MM_Username'];
}
mysql_select_db($database_airline, $airline);
$query_Recordset1 = sprintf("SELECT * FROM customer WHERE E_mail = %s", GetSQLValueString($colname_Recordset1, "text"));
$Recordset1 = mysql_query($query_Recordset1, $airline) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);


  $colname_Recordset2 = $row_Recordset1['Passport_number'];

mysql_select_db($database_airline, $airline);
$query_Recordset2 = sprintf("SELECT * FROM ffc WHERE Passport_number = %s", GetSQLValueString($colname_Recordset2, "text"));
$Recordset2 = mysql_query($query_Recordset2, $airline) or die(mysql_error());
$row_Recordset2 = mysql_fetch_assoc($Recordset2);
$totalRows_Recordset2 = mysql_num_rows($Recordset2);
```

Görsel 8: Üye mil ve puan bilgileri için PHP ve SQL sorgu kodları

Görsel 7'de Üye mil ve puan bilgililerini görüntüleyen bölüm bulunmaktadır. Bu bölümde üyenin uçuşlarda kazandığı puanlar ve diğer uçuş bilgileri bulunmaktadır. **Görsel 8**'de bu bilgileri göstermek için gerekli PHP ve SQL sorgu kodları görülmektedir.

```
$colname_Recordset1 = "-1";

if (isset($_SESSION['MM_Username'])) {

  $colname_Recordset1 = $_SESSION['MM_Username'];

}

mysql_select_db($database_airline, $airline);

$query_Recordset1 = sprintf("SELECT * FROM customer WHERE E_mail = %s", GetSQLValueString($colname_Recordset1, "text"));

$Recordset1 = mysql_query($query_Recordset1, $airline) or die(mysql_error());

$row_Recordset1 = mysql_fetch_assoc($Recordset1);
```

```
$totalRows_Recordset1 = mysql_num_rows($Recordset1);



  $colname_Recordset2 = $row_Recordset1['Passport_number'];

mysql_select_db($database_airline, $airline);

$query_Recordset2 = sprintf("SELECT * FROM ffc WHERE Passport_number = %s", GetSQLValueString($colname_Recordset2, "text"));

$Recordset2 = mysql_query($query_Recordset2, $airline) or die(mysql_error());

$row_Recordset2 = mysql_fetch_assoc($Recordset2);

$totalRows_Recordset2 = mysql_num_rows($Recordset2);
```



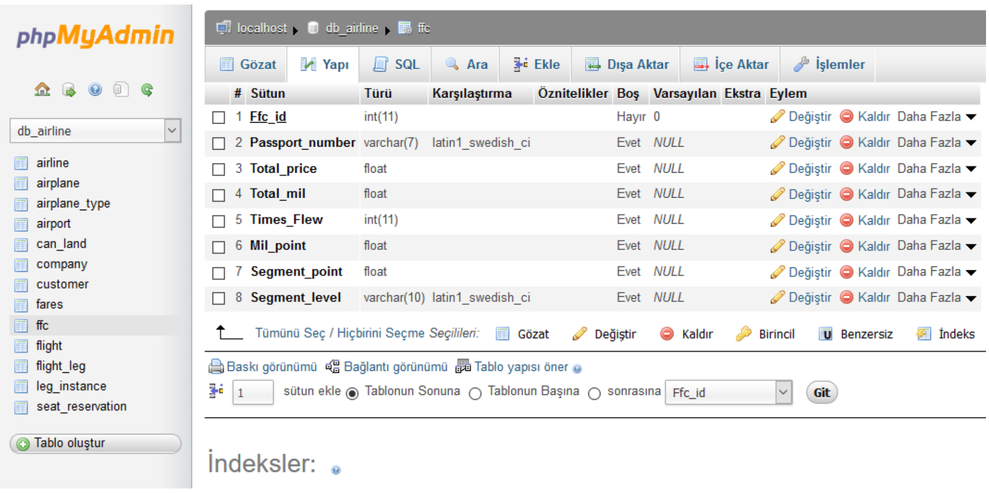
Üye ismi :	Isa
E- Mail Adresi:	isa@gmail.com
Telefonu	4352556859
Adresi	656 sokak no 2 Turkiye

Parola Değiştirmek için aşağıdaki bölümü kullanabilirsiniz.

Eski Parolanız	
Yeni Parolanız	
Yeni Parola Tekrar	
Parolayı Değiştir	

Görsel 9: Üye şifre değiştirme bölümü

Görsel 9'da Üye şifre değiştirme işlemini sağlayan bölüm bulunmaktadır. Bu bölümde üye sisteme giriş parolasını değiştirebilmektedir. **Görsel 10**'da Web sayfasında kullanılan MySQL veritanaı ve tabloları görülmektedir.



	#	Sütun	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan	Ekstra	Eylem
□	1	Ffc_id	int(11)			Hayır	0		Değiştir Kaldır Daha Fazla
□	2	Passport_number	varchar(7)	latin1_swedish_ci		Evet	NULL		Değiştir Kaldır Daha Fazla
□	3	Total_price	float			Evet	NULL		Değiştir Kaldır Daha Fazla
□	4	Total_mil	float			Evet	NULL		Değiştir Kaldır Daha Fazla
□	5	Times_Flew	int(11)			Evet	NULL		Değiştir Kaldır Daha Fazla
□	6	Mil_point	float			Evet	NULL		Değiştir Kaldır Daha Fazla
□	7	Segment_point	float			Evet	NULL		Değiştir Kaldır Daha Fazla
□	8	Segment_level	varchar(10)	latin1_swedish_ci		Evet	NULL		Değiştir Kaldır Daha Fazla

Tümünü Seç / Hiçbirini Seçme Seçilileri: Gözat Değiştir Kaldır Birincil Benzersiz İndeks
Baskı görünümü Bağlantı görünümü Tablo yapısı öner
1 sütun ekle ● Tablonun Sonuna ○ Tablonun Başına ○ sonrasına Ffc_id Git
İndeksler:

Görsel 10: Havayolu MySQL veritabanı ve tabloları