

SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
VERİ TABANI VE YÖNETİM SİSTEMLERİ ÖDEVİ

Grup üyeleri adı:

Ramazan Umut Tatar	B241210906
Ubey Yarba	B231210574
Muhammed Ömer Karson	B231210054

E-posta adresleri:

ramazan.tatar@ogr.sakarya.edu.tr
ubey.yarba@ogr.sakarya.edu.tr
muhammed.karson@ogr.sakarya.edu.tr

PROBLEMİN KISA TANITIMI:

Uçak bileti Rezervasyon sistemi yaptık.

Bu projenin amacı kullanıcılara uçak bileti rezervasyon sistemi sağlamak.

İŞ KURALLARI:

1-)Kullanıcılar tablosunda her bir kullanıcının ad, soyad, telefon ve e-posta bilgileri yer alır.

2-)Müşteriler tablosu, Kullanıcılar tablosundan türetilerek müşterilerin doğum tarihi bilgisine sahiptir.

3-)Hostesler tablosu, kullanıcı bilgilerine ek olarak uçuş, maaş ve doğum tarihi bilgilerini içerir.

4-)Havalimanları tablosu, her bir havalimanının ID'si, adı ve bulunduğu şehir bilgilerini tutar.

5-)Uçaklar tablosu, her bir uçağın model, kapasite ve ilişkili olduğu havayolu bilgilerini içerir.

6-)Havayolları tablosu, her havayolunun adı, merkezi ülke bilgisi ve havayoluna ait ID numarasını barındırır.

7-)Uçuşlar tablosu, uçuşların kalkış ve varış saatleri, uçuşa ait uçak ve rota bilgilerini içerir.

8-)Rezervasyonlar tablosu, her bir rezervasyonun tarih, kullanıcı bilgileri ve uçuş bilgilerini içerir.

9-)Biletler tablosu, her biletin koltuk numarası, fiyatı ve rezervasyon bilgilerini saklar.

10-)Ödemeler tablosu, her bir ödemenin tarihi, tutarı ve ödeme türünü içerir.

11-)Koltuklar tablosu, uçak içindeki koltukların numarası, rezervasyon durumu ve uçak bilgilerini içerir.

12-)Bagajlar tablosu, her bagajın türü, ağırlığı ve ilişkilendirildiği rezervasyon bilgilerini içerir.

13-)Rotalar tablosu, uçuş rotalarıyla ilgili kalkış ve varış havalimanları, mesafe gibi bilgileri tutar.

14-)Yemek servisi tablosu, uçuşlarda sunulan yemeklerin türünü ve uçuş bilgilerini içerir.

- 15-)Uçuş kapıları tablosu, her uçuş için atanan kapı numarası ve uçuş bilgilerini içerir.
- 16-) Bir müşteri yalnızca bir kullanıcı kaydına bağlıdır ve bir kullanıcı hem müşteri hem de hostes olabilir.
- 17-) Bir hostes yalnızca bir kullanıcı kaydına bağlıdır ve bir kullanıcı hem hostes hem de müşteri olabilir.
- 18-)Bir rota, bir kalkış ve bir varış havalimanına bağlıdır ve bir havalimanı birden fazla rota için kalkış veya varış noktası olabilir.
- 19-)Bir uçak yalnızca bir havayoluna bağlıdır, bir havayolu birden fazla uçağa sahip olabilir.
- 20-)Bir uçuş yalnızca bir rotaya bağlıdır, bir rota birden fazla uçuş için kullanılabilir.
- 21-)Bir uçuş yalnızca bir uçak ile yapılabilir, bir uçak birden fazla uçuшта kullanılabilir.
- 22-)Bir uçuş birden fazla yemek servisine sahip olabilir, bir yemek servisi yalnızca bir uçuşa bağlıdır.
- 23-)Bir rezervasyon yalnızca bir kullanıcıya (müşteri veya hostes) bağlıdır, bir kullanıcı birden fazla rezervasyon yapabilir.
- 24-)Bir rezervasyon yalnızca bir uçuşa bağlıdır, bir uçuş birden fazla rezervasyon içerebilir.
- 25-)Bir bilet yalnızca bir rezervasyona bağlıdır, bir rezervasyon birden fazla bileti kapsayabilir.
- 26-)Bir koltuk yalnızca bir uçağa aittir, bir uçak birden fazla koltuğa sahip olabilir.
- 27-)Bir bilet bir koltuğa atanabilir, her koltuk yalnızca bir biletle ilişkilendirilebilir.
- 28-)Bir ödeme yalnızca bir rezervasyona bağlıdır, bir rezervasyon birden fazla ödeme içerebilir.
- 29-)Bir bagaj yalnızca bir rezervasyona bağlıdır, bir rezervasyon birden fazla bagaj içerebilir.
- 30-)Bir uçuş yalnızca bir uçuş kapısından yapılır, bir uçuş kapısında birden fazla uçuş yapılabilir

İLİŞKİSEL METİN

kullanıcılar(PK kullanıcı_ID,ad,soyad,cinsiyet,telefon,email)

odemeler(PK odeme_ID,FK rezervasyon_ID,odeme_tarihi,tutar,odeme_tipi)

havalimanlari(PK havalimani_ID,ad,sehir)

rotalar(PK rota_ID,FK kalkis_havalimani_ID,FK varis_havalimani_ID,mesafe)

havayollari(PK havayolu_ID,ad,ulke)

bagajlar(PK bagaj_ID,FK rezervasyon_ID,bagaj_turu,agirlik)

rezervasyonlar(PK rezervasyon_ID,FK kullanıcı_ID,FK ucus_ID,rezervasyon_tarihi)

biletler(PK bilet_ID,FK rezervasyon_ID,koltuk_no,fiyat)

ucaklar(PK ucak_ID,model,kapasite,FK havayolu_ID)

ucuslar(PK ucus_ID,kalkis_saati,varis_saati,FK rota_ID,FK ucak_ID)

koltuklar(PK koltuk_ID,FK ucak_ID,koltuk_no,FK rezervasyon_ID)

ucuskapilari(PK kapi_ID,FK ucus_ID,kapi_no)

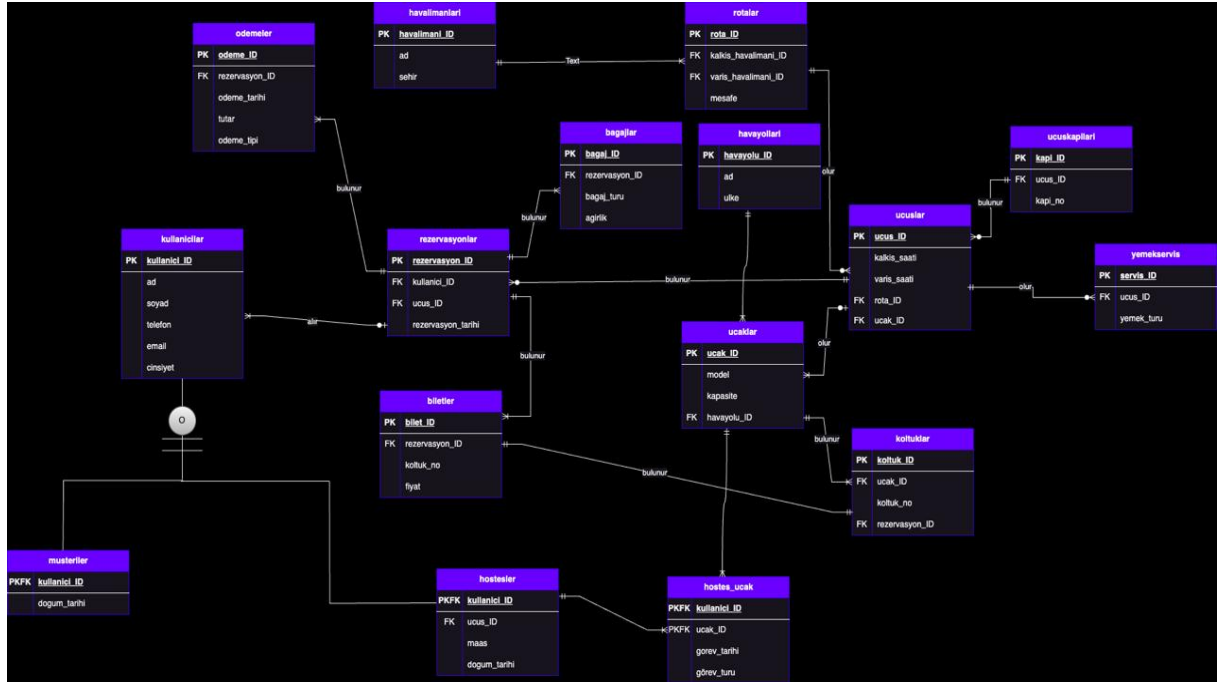
yemekservis(PK servis_ID,FK ucus_ID,yemek_turu)

musteriler(PKFK kullanıcı_ID,dogum_tarihi)

hostesler(PKFK kullanıcı_ID,FK ucus_ID,maas)

hostes_ucak(PKFK kullanıcı_ID,PKFK ucak_ID,gorev_tarihi,gorev_turu)

VARLIK BAĞINTI MODELİ:



DATABASE:

-- Kullanıcılar Tablosu

```
CREATE TABLE kullanicilar (  
    kullanici_ID SERIAL PRIMARY KEY,  
    ad VARCHAR(100),  
    soyad VARCHAR(100),  
    cinsiyet VARCHAR(10),  
    telefon VARCHAR(25),  
    email VARCHAR(100)  
);
```

-- Müşteriler Tablosu (Kullanıcılarla ilişkili)

```
CREATE TABLE musteriler (  
    kullanici_ID INT PRIMARY KEY,  
    dogum_tarihi DATE,  
    FOREIGN KEY (kullanici_ID) REFERENCES kullanicilar(kullanici_ID)  
);
```

-- Havayolları Tablosu

```
CREATE TABLE havayollari (  
    havayolu_ID SERIAL PRIMARY KEY,  
    ad VARCHAR(100),  
    ulke VARCHAR(50)  
);
```

-- Havalimanları Tablosu

```
CREATE TABLE havalimani (  
    havalimani_ID SERIAL PRIMARY KEY,  
    ad VARCHAR(100),  
    sehir VARCHAR(100)  
);
```

-- Rotalar Tablosu

```
CREATE TABLE rotalar (  
    rota_ID SERIAL PRIMARY KEY,  
    kalkis_havalimani_ID INT,  
    varis_havalimani_ID INT,  
    mesafe DECIMAL(10, 2),  
    FOREIGN KEY (kalkis_havalimani_ID) REFERENCES havalimani(havalimani_ID),  
    FOREIGN KEY (varis_havalimani_ID) REFERENCES havalimani(havalimani_ID)  
);
```

-- Uçaklar Tablosu

```
CREATE TABLE ucaklar (  
    ucak_ID SERIAL PRIMARY KEY,  
    model VARCHAR(100),  
    kapasite INT,  
    havayolu_ID INT,  
    FOREIGN KEY (havayolu_ID) REFERENCES havayollari(havayolu_ID)  
);
```

-- Uçuşlar Tablosu

```
CREATE TABLE ucuslar (  
    ucus_ID SERIAL PRIMARY KEY,  
    kalkis_saati TIMESTAMP,  
    varis_saati TIMESTAMP,  
    rota_ID INT,  
    ucak_ID INT,  
    FOREIGN KEY (rota_ID) REFERENCES rotalar(rota_ID),  
    FOREIGN KEY (ucak_ID) REFERENCES ucaklar(ucak_ID)  
);
```

-- Rezervasyonlar Tablosu

```
CREATE TABLE rezervasyonlar (  
    rezervasyon_ID SERIAL PRIMARY KEY,  
    kullanıcı_ID INT,  
    ucus_ID INT,  
    rezervasyon_tarihi DATE,  
    FOREIGN KEY (kullanici_ID) REFERENCES kullanicilar(kullanici_ID),  
    FOREIGN KEY (ucus_ID) REFERENCES ucuslar(ucus_ID)  
);
```

-- Koltuklar Tablosu

```
CREATE TABLE koltuklar (  
    koltuk_ID SERIAL PRIMARY KEY,  
    ucak_ID INT,  
    koltuk_no VARCHAR(5),
```

```
rezervasyon_ID INT,  
  
FOREIGN KEY (ucak_ID) REFERENCES ucaklar(ucak_ID),  
  
FOREIGN KEY (rezervasyon_ID) REFERENCES rezervasyonlar(rezervasyon_ID)  
);
```

-- Ödemeler Tablosu

```
CREATE TABLE odemeler (  
  
    odeme_ID SERIAL PRIMARY KEY,  
  
    rezervasyon_ID INT,  
  
    odeme_tarihi DATE,  
  
    tutar DECIMAL(10, 2),  
  
    odeme_tipi VARCHAR(50),  
  
    FOREIGN KEY (rezervasyon_ID) REFERENCES rezervasyonlar(rezervasyon_ID)  
);
```

-- Bagajlar Tablosu

```
CREATE TABLE bagajlar (  
  
    bagaj_ID SERIAL PRIMARY KEY,  
  
    rezervasyon_ID INT,  
  
    bagaj_turu VARCHAR(50),  
  
    agirlik DECIMAL(5, 2),  
  
    FOREIGN KEY (rezervasyon_ID) REFERENCES rezervasyonlar(rezervasyon_ID)  
);
```

-- Biletler Tablosu

```
CREATE TABLE biletler (  

```



```
bilet_ID SERIAL PRIMARY KEY,  
rezervasyon_ID INT,  
koltuk_no VARCHAR(5),  
fiyat DECIMAL(10, 2),  
FOREIGN KEY (rezervasyon_ID) REFERENCES rezervasyonlar(rezervasyon_ID)  
);
```

-- Uçuş Kapıları Tablosu

```
CREATE TABLE ucuskapilari (  
    kapi_ID SERIAL PRIMARY KEY,  
    ucus_ID INT,  
    kapi_no VARCHAR(10),  
    FOREIGN KEY (ucus_ID) REFERENCES ucuslar(ucus_ID)  
);
```

-- Yemek Servis Tablosu

```
CREATE TABLE yemekservis (  
    servis_ID SERIAL PRIMARY KEY,  
    ucus_ID INT,  
    yemek_turu VARCHAR(50),  
    FOREIGN KEY (ucus_ID) REFERENCES ucuslar(ucus_ID)  
);
```

-- Hostesler Tablosu (Kullanıcılarla ilişkili)

```
CREATE TABLE hostesler (  
    kullanıcı_ID INT PRIMARY KEY,
```

```
ucus_ID INT,  
  
maas DECIMAL(10, 2),  
  
dogum_tarihi DATE,  
  
FOREIGN KEY (kullanici_ID) REFERENCES kullanicilar(kullanici_ID),  
  
FOREIGN KEY (ucus_ID) REFERENCES ucuslar(ucus_ID)  
  
);
```

-- Hostes-Uçak İlişkisi Tablosu

```
CREATE TABLE hostes_ucak (  
  
    kullanici_ID INT,  
  
    ucak_ID INT,  
  
    gorev_tarihi DATE,  
  
    gorev_turu VARCHAR(50),  
  
    PRIMARY KEY (kullanici_ID, ucak_ID),  
  
    FOREIGN KEY (kullanici_ID) REFERENCES kullanicilar(kullanici_ID),  
  
    FOREIGN KEY (ucak_ID) REFERENCES ucaklar(ucak_ID)  
  
);
```

```
INSERT INTO kullanicilar (ad, soyad, telefon, email,cinsiyet) VALUES  
  
('Ahmet', 'Kaya', '5551234567', 'ahmet.kaya@mail.com','Erkek'),  
  
('Ayşe', 'Yılmaz', '5552345678', 'ayse.yilmaz@mail.com','Erkek'),  
  
('Mehmet', 'Çelik', '5553456789', 'mehmet.celik@mail.com','Erkek'),  
  
('Fatma', 'Demir', '5554567890', 'fatma.demir@mail.com','Kadın'),  
  
('Ali', 'Öztürk', '5555678901', 'ali.ozturk@mail.com','Erkek');
```

```
INSERT INTO musteriler (kullanici_ID, dogum_tarihi) VALUES
```

(1, '1990-01-01'),
(2, '1985-02-15'),
(3, '1988-03-30'),
(4, '1992-04-10'),
(5, '1980-05-20');

INSERT INTO havayollari (ad, ulke) VALUES

('Türk Hava Yolları', 'Türkiye'),
('Lufthansa', 'Almanya'),
('Emirates', 'Birleşik Arap Emirlikleri'),
('British Airways', 'İngiltere'),
('Air France', 'Fransa');

INSERT INTO havalimani (ad, sehir) VALUES

('İstanbul Havalimanı', 'İstanbul'),
('Frankfurt Havalimanı', 'Frankfurt'),
('Dubai Havalimanı', 'Dubai'),
('Heathrow Havalimanı', 'Londra'),
('Charles de Gaulle Havalimanı', 'Paris');

INSERT INTO rotalar (kalkis_havalimani_ID, varis_havalimani_ID, mesafe) VALUES

(1, 2, 2000.50),
(2, 3, 3000.75),
(3, 4, 4000.20),
(4, 5, 1500.30),
(5, 1, 2500.80);

INSERT INTO ucaklar (model, kapasite, havayolu_ID) VALUES

('Airbus A320', 180, 1),

('Boeing 747', 400, 2),

('Airbus A380', 500, 3),

('Boeing 777', 350, 4),

('Airbus A350', 300, 5);

INSERT INTO ucuslar (kalkis_saati, varis_saati, rota_ID, ucak_ID) VALUES

('2024-12-25 08:00:00', '2024-12-25 12:00:00', 1, 1),

('2024-12-26 10:00:00', '2024-12-26 16:00:00', 2, 2),

('2024-12-27 09:30:00', '2024-12-27 15:30:00', 3, 3),

('2024-12-28 14:00:00', '2024-12-28 20:00:00', 4, 4),

('2024-12-29 07:00:00', '2024-12-29 11:00:00', 5, 5);

INSERT INTO rezervasyonlar (kullanici_ID, ucus_ID, rezervasyon_tarihi) VALUES

(1, 1, '2024-12-20'),

(2, 2, '2024-12-21'),

(3, 3, '2024-12-22'),

(4, 4, '2024-12-23'),

(5, 5, '2024-12-24');

INSERT INTO koltuklar (ucak_ID, koltuk_no, rezervasyon_ID) VALUES

(1, '1A', 1),

(2, '2B', 2),

(3, '3C', 3),

(4, '4D', 4),
(5, '5E', 5);

INSERT INTO odemeler (rezervasyon_ID, odeme_tarihi, tutar, odeme_tipi) VALUES
(1, '2024-12-20', 100.00, 'Kredi Kartı'),
(2, '2024-12-21', 150.00, 'Nakit'),
(3, '2024-12-22', 200.00, 'Kredi Kartı'),
(4, '2024-12-23', 250.00, 'Banka Havalesi'),
(5, '2024-12-24', 300.00, 'Kredi Kartı');

INSERT INTO bagajlar (rezervasyon_ID, bagaj_turu, agirlik) VALUES
(1, 'El Bagajı', 10.00),
(2, 'Kabin Bagajı', 15.00),
(3, 'Kabin Bagajı', 20.00),
(4, 'Check-in Bagajı', 25.00),
(5, 'El Bagajı', 12.00);

INSERT INTO biletler (rezervasyon_ID, koltuk_no, fiyat) VALUES
(1, '1A', 150.00),
(2, '2B', 175.00),
(3, '3C', 200.00),
(4, '4D', 250.00),
(5, '5E', 300.00);

INSERT INTO ucuskapilari (ucus_ID, kapi_no) VALUES
(1, 'A1'),

(2, 'B2'),

(3, 'C3'),

(4, 'D4'),

(5, 'E5');

INSERT INTO yemekservis (ucus_ID, yemek_turu) VALUES

(1, 'Ekspres Yiyecek'),

(2, 'Öğle Yemeği'),

(3, 'Akşam Yemeği'),

(4, 'Kahvaltı'),

(5, 'Akşam Yemeği');

INSERT INTO hostesler (kullanici_ID, ucus_ID, maas,dogum_tarihi) VALUES

(1, 1, 6000.00,'1990-01-01'),

(2, 2, 7500.00,'1985-02-15'),

(3, 3, 5200.00,'1988-03-30'),

(4, 4, 7800.00,'1992-04-10'),

(5, 5, 7000.00,'1980-05-20');

INSERT INTO hostes_ucak (kullanici_ID, ucak_ID, gorev_tarihi, gorev_turu) VALUES

(1, 1, '2024-12-25', 'Yemek Servisi'),

(2, 2, '2024-12-26', 'Yardım'),

(3, 3, '2024-12-27', 'Yemek Servisi'),

(4, 4, '2024-12-28', 'Yardım'),

(5, 5, '2024-12-29', 'Yemek Servisi');

FONKSİYONLAR:

1-) BAGAJ SAYISINI VEREN FONKSİYON

-- FUNCTION: public.toplam_bagaj_sayisi()

--TOPLAM BAGAJ SAYISINI VERİR

-- DROP FUNCTION IF EXISTS public.toplam_bagaj_sayisi();

```
CREATE OR REPLACE FUNCTION public.toplam_bagaj_sayisi(  
    )
```

RETURNS integer

LANGUAGE 'plpgsql'

COST 100

VOLATILE PARALLEL UNSAFE

AS \$BODY\$

BEGIN

RETURN (SELECT COUNT(*) FROM bagajlar);

END;

\$BODY\$;

ALTER FUNCTION public.toplam_bagaj_sayisi()

OWNER TO postgres;

2-)TOPLAM HAVAYOLU SAYISINI VEREN FONKSİYON

-- FUNCTION: public.toplam_havayolu_sayisi()

--TOPLAM HAVAYOLU SAYISINI VERİR

-- DROP FUNCTION IF EXISTS public.toplam_havayolu_sayisi();

```
CREATE OR REPLACE FUNCTION public.toplam_havayolu_sayisi(  
    )  
  
    RETURNS integer  
  
    LANGUAGE 'plpgsql'  
  
    COST 100  
  
    VOLATILE PARALLEL UNSAFE  
  
AS $BODY$  
  
BEGIN  
  
    RETURN (SELECT COUNT(*) FROM havayollari);  
  
END;  
  
$BODY$;
```

```
ALTER FUNCTION public.toplam_havayolu_sayisi()  
  
    OWNER TO postgres;
```

3-)TOPLAM KULLANICI SAYISI FONKSİYONU

```
-- FUNCTION: public.toplam_kullanici_sayisi()  
  
--TOPLAM KULLANICI SAYISINI VERİR  
  
-- DROP FUNCTION IF EXISTS public.toplam_kullanici_sayisi();
```

```
CREATE OR REPLACE FUNCTION public.toplam_kullanici_sayisi(  
    )  
  
    RETURNS integer  
  
    LANGUAGE 'plpgsql'  
  
    COST 100
```


VOLATILE PARALLEL UNSAFE

AS \$BODY\$

BEGIN

RETURN (SELECT COUNT(*) FROM kullanicilar);

END;

\$BODY\$;

ALTER FUNCTION public.toplam_kullanici_sayisi()

OWNER TO postgres;

4-)TOPLAM REZERVASYON SAYISINI VEREN FONKSİYON

-- FUNCTION: public.toplam_rezervasyon_sayisi()

-- DROP FUNCTION IF EXISTS public.toplam_rezervasyon_sayisi();

CREATE OR REPLACE FUNCTION public.toplam_rezervasyon_sayisi(

)

RETURNS integer

LANGUAGE 'plpgsql'

COST 100

VOLATILE PARALLEL UNSAFE

AS \$BODY\$

BEGIN

RETURN (SELECT COUNT(*) FROM rezervasyonlar);

END;

\$BODY\$;

```
ALTER FUNCTION public.toplam_rezervasyon_sayisi()
```

```
OWNER TO postgres;
```

5)--5. Ucus bilgilerini döndürüyor

```
CREATE OR REPLACE FUNCTION public.aramaucus(
```

```
    kalkis_havalimani integer,
```

```
    varis_havalimani integer)
```

```
    RETURNS TABLE(ucus_id integer, kalkis_saati timestamp without time zone, varis_saati  
timestamp without time zone, fiyat numeric)
```

```
    LANGUAGE 'plpgsql'
```

```
    COST 100
```

```
    VOLATILE PARALLEL UNSAFE
```

```
    ROWS 1000
```

```
AS $BODY$
```

```
BEGIN
```

```
-- RETURN QUERY: Sorgunun sonucunu döndürmek için kullanılır.
```

```
RETURN QUERY
```

```
SELECT
```

```
    u.ucus_ID,      -- Uçuş ID'si
```

```
    u.kalkis_saati, -- Kalkış zamanı
```

```
    u.varis_saati,  -- Varış zamanı
```

```
    b.fiyat         -- Uçuş fiyatı
```

```
FROM ucuslar u
```

JOIN rotalar r ON u.rota_ID = r.rota_ID -- Uçuş ile rotaları ilişkilendirir.

JOIN biletler b ON u.ucus_ID = b.rezervasyon_ID -- Uçuş ile biletleri ilişkilendirir.

WHERE

r.kalkis_havalimani_ID = kalkis_havalimani -- Kalkış havalimanını kontrol eder.

AND r.varis_havalimani_ID = varis_havalimani; -- Varış havalimanını kontrol eder.

END;

\$BODY\$;

ALTER FUNCTION public.aramaucus(integer, integer)

OWNER TO postgres;

TRIGGERLAR

1)--Bu tetikleyici, kullanıcılar tablosuna yeni bir kayıt eklenmeden önce, e-posta adresinin benzersiz olup olmadığını kontrol eder.

CREATE OR REPLACE FUNCTION kontrol_email_benzersizligi()

RETURNS TRIGGER AS \$\$

BEGIN

IF EXISTS (SELECT 1 FROM kullanıcılar WHERE email = NEW.email) THEN

RAISE EXCEPTION 'Bu e-posta adresi zaten kayıtlı: %', NEW.email;

END IF;

RETURN NEW;

END;

\$\$ LANGUAGE plpgsql;

```
CREATE TRIGGER email_benzersizligi_trigger
BEFORE INSERT OR UPDATE ON kullanicilar
FOR EACH ROW
EXECUTE FUNCTION kontrol_email_benzersizligi();
```

2)—Bu tetikleyici, bir kullanıcının aynı uçuşa birden fazla rezervasyon yapmasını engeller.

```
CREATE OR REPLACE FUNCTION kontrol_rezervasyon_tekrari()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM rezervasyonlar
        WHERE kullanıcı_ID = NEW.kullanıcı_ID AND uçuş_ID = NEW.uçuş_ID
    ) THEN
        RAISE EXCEPTION 'Bu kullanıcı için zaten bir rezervasyon mevcut.';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER rezervasyon_tekrari_trigger
BEFORE INSERT ON rezervasyonlar
FOR EACH ROW
EXECUTE FUNCTION kontrol_rezervasyon_tekrari();
```

3) –Bu tetikleyici, hostes maaşının belirli bir alt sınırın (örneğin, 5000) altına düşmesini engeller.

```
CREATE OR REPLACE FUNCTION kontrol_hostes_maasi()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    IF NEW.maas < 5000 THEN
```

```
        RAISE EXCEPTION 'Hostes maaşı 5000''den az olamaz. Girilen: %', NEW.maas;
```

```
    END IF;
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER hostes_maasi_trigger
```

```
BEFORE INSERT OR UPDATE ON hostesler
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION kontrol_hostes_maasi();
```

4) –Bu tetikleyici,uçuş kapısı bilgisi eklenmeden önce, uçuşun kalkış saatinin geçmemiş olması gerektiğini kontrol eder.

```
CREATE OR REPLACE FUNCTION kontrol_ucus_kapisi_zamani()
```

```
RETURNS TRIGGER AS $$
```

```
DECLARE
```

```
    kalkis_saati TIMESTAMP;
```

```
BEGIN
```

```
    SELECT kalkis_saati INTO kalkis_saati FROM ucuslar WHERE ucus_ID = NEW.ucus_ID;
```

```
IF kalkis_saati < NOW() THEN
```

```
    RAISE EXCEPTION 'Bu uçuş için kapı bilgisi artık eklenemez. Kalkış saati geçti: %',  
    kalkis_saati;
```

```
END IF;
```

```
RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER ucus_kapisi_zamani_trigger
```

```
BEFORE INSERT ON ucuskapilari
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION kontrol_ucus_kapisi_zamani();
```

LİSTELEME:

Form1

Kullanıcı Bilgileri

	kullanici_id	ad	soyad
▶	20	fds	dsfs
	22	sada	safsad
	24	sada	safsad
	25	adsas	asda
	19	Ömer	KARSON
	29	Ömer	KARSON
	30	Ömer	KARSON

Kullanıcı ID

Kullanıcı Adı

Kullanıcı Soyadı

Cinsiyet

Telefon NO

Email

Liste

Ekle

Sil

Güncelle

EKLEME:

Form1

Kullanıcı Bilgileri

	kullanici_id	ad	soyad
	24	sada	safsad
	25	adsas	asda
	19	Ömer	KARSON
	29	Ömer	KARSON
	30	Ömer	KARSON
	31	Muhammed	Emre

Kullanıcı ID:

Kullanıcı Adı:

Kullanıcı Soyadı:

Cinsiyet:

Telefon NO:

Email:

SİLME:

Form1

Kullanıcı Bilgileri

	kullanici_id	ad	soyad
	22	sada	safsad
	24	sada	safsad
	25	adsas	asda
	19	Ömer	KARSON
	29	Ömer	KARSON
	31	Muhammed	Emre

Kullanıcı ID:

Kullanıcı Adı:

Kullanıcı Soyadı:

Cinsiyet:

Telefon NO:

Email:

GÜNCELLEME:

Form1

Kullanıcı Bilgileri

	kullanici_id	ad	soyad
	22	sada	safsad
	24	sada	safsad
	25	adsas	asda
	29	Ömer	KARSON
	31	Muhammed	Emre
	19	Ubey	YARBA

Kullanıcı ID:

Kullanıcı Adı:

Kullanıcı Soyadı:

Cinsiyet:

Telefon NO:

Email:

KAYNAK KODU:

https://github.com/RamazanTatar/SQL/blob/main/Ucak_Rezervasyonlari

<https://github.com/Ubey02/Veritaban-Proje>