



YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRİK-ELEKTRONİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM3041 – VERİTABANI YÖNETİMİ

PROJE RAPORU

Konu : Taksi Durağı Bilgi Sistemi

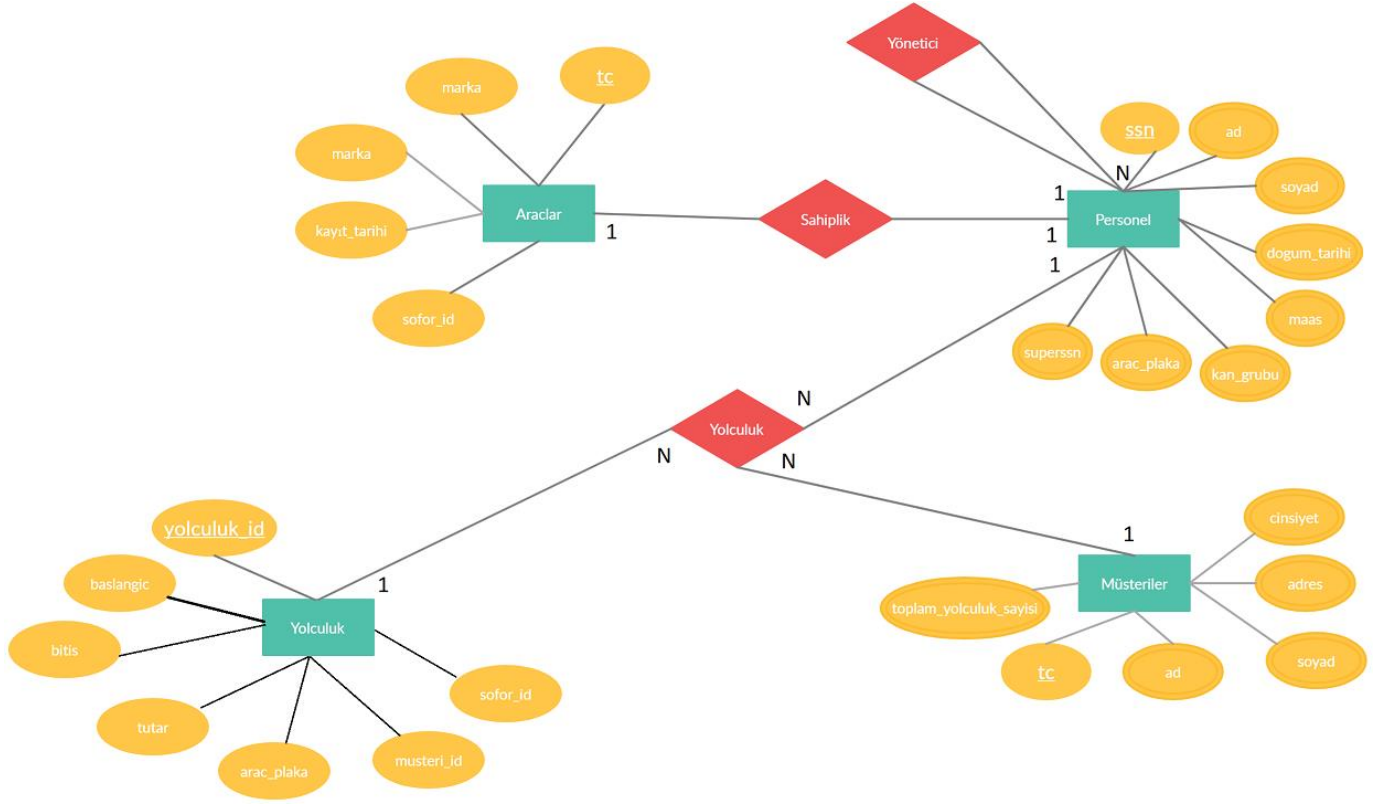
Fatih Mustafa Sağır – 15011704

Eren Tutuş – 17011702

Onur Koçkan – 17011701

Sermed Kerim - 18011606

ER DİYAGRAMI



TABLoların Ekran Görüntüleri

Personel Tablosu

	ssn [PK] character (11)	ad character varying (20)	soyad character varying (20)	dogum_tarihi date	maas integer	kan_grubu character varying (4)	arac_plaka character varying (7)	superssn character (11)
1	11111111111	Ahmet	Demir	1960-02-12	19000	ARh+	34NRB77	[null]
2	22222222222	Ali	Korkmaz	1970-01-15	6000	ORh-	34KLM78	11111111111
3	12313344556	Ömer	Dağlı	1980-01-05	4500	ARh-	34JKL99	11111111111
4	45454545454	Kadir	Şahin	1978-02-10	7000	ORh-	34TYB43	11111111111
5	12121212121	Birkan	Aybek	1988-05-05	8000	ARh+	34NRM99	11111111111
6	35355353667	Ercan	Ertürk	1990-12-10	6670	ORh+	34BVC66	11111111111
7	89898989891	Yusuf	Şen	1966-05-13	8900	ARh+	34YUI06	11111111111
8	18794563321	Taner	Barış	1973-10-09	5600	ORh-	34PLK12	11111111111
9	23156789988	Mehmet	Gül	1987-01-20	5670	ARh-	34CDE45	11111111111
10	44678899012	Mahmut	Bilgin	1969-01-18	9000	ARh+	34BFV48	11111111111








Musteriler Tablosu

	tc [PK] character (11)	ad character varying (20)	soyad character varying (20)	adres character varying (50)	cinsiyet character varying (5)	toplam_yolculuk_sayisi integer
1	1011012223	Nilay	Metin	Fevzi Çakmak Mah. Esenler İ...	Kadın	0
2	45564411223	Ufuk	Sönmez	Kavaklı Mah. Beylikdüzü İsta...	Erkek	0
3	89712345678	Erhan	Şentürk	Gazi Mah. Sultangazi İstanbul	Erkek	1
4	78788999012	Kemal	Aydın	Karadeniz Mah. Gaziosmanp...	Erkek	1
5	34555566774	Ayşe	Arslan	Aşık Veysel Mah. Ataşehir İs...	Kadın	2
6	56557789012	Onur	Demir	Davutpaşa Mah. Esenler İsta...	Erkek	1
7	5436671112	Esmâ	Çınar	Soğanlı Mah. Bahçelievler İs...	Kadın	1
8	10987611111	Burak	Sezgin	Bağlarbaşı Mah. Gaziosman...	Erkek	1
9	12343215555	Orhan	Çetin	Atatürk Mah. Ataşehir İstanb...	Erkek	1
10	1112223334	Fatma	Çetin	Karlıtepe Mah. Gaziosmanp...	Kadın	2

Araclar Tablosu

	plaka_no [PK] character varying (7)	marka character varying (20)	model character varying (20)	kayit_tarihi date	sofor_id character (11)
1	34NRB77	Fiat	Egea	2005-12-05	11111111111
2	34KLM78	Fiat	Egea	2008-05-05	22222222222
3	34JKL99	Fiat	Egea	2010-12-22	12313344556
4	34TYB43	Fiat	Doblo	2011-01-14	45454545454
5	34NRM99	Fiat	Doblo	2009-02-25	12121212121
6	34BVC66	Fiat	Egea	2001-01-07	35355353667
7	34YUI09	Fiat	Doblo	2013-10-02	89898989891
8	34PLK12	Fiat	Egea	2011-12-21	18794563321
9	34CDE45	Fiat	Egea	2007-12-03	23156789988
10	34BFV48	Fiat	Doblo	2005-10-13	44678899012

Yolculuk Tablosu

	 yolculuk_id [PK] integer	 baslangic character varying (20)	 bitis character varying (20)	 tutar integer	 arac_plaka character varying (7)	 musteri_id character (11)	 sofor_id character (11)
1	12	Ataşehir	Kadıköy	50	34KLM78	34555566774	22222222222
2	13	Sultangazi	Maslak	67	34BVC66	89712345678	35355353667
3	14	Gaziosmanpaşa	Zeytinburnu	100	34BVC66	78788999012	35355353667
4	15	Kadıköy	Pendik	150	34JKL99	34555566774	12313344556
5	18	Taksim	Kabataş	78	34KLM78	56557789012	22222222222
6	19	Zeytinburnu	Fatih	30	34TYB43	1112223334	45454545454
7	20	Tuzla	Pendik	80	34PLK12	5436671112	18794563321
8	21	Bakırköy	Zeytinburnu	45	34BFV48	10987611111	44678899012
9	22	Bahçelievler	Şirinevler	24	34CDE45	12343215555	23156789988
10	23	Maslak	Taksim	155	34NRM99	1112223334	12121212121

KOD BLOKLARI

1) Oluşturacağınız veritabanı en az 4 tablo içermelidir. Her tabloda en az 10 kayıt bulunmalıdır.

```
create database TaksiDuragi_BilgiSistemi

-- PERSONEL TABLOSU
create table personel(
    ssn char(11) not null primary key,
    ad varchar(20) not null,
    soyad varchar(20) not null,
    dogum_tarihi date not null,
    maas int not null,
    kan_grubu varchar(4) not null,
    arac_plaka varchar(7) not null,
    superssn char(11),
    constraint fk_ssn foreign key(superssn) references personel(ssn) on
delete cascade
);

-- ARAÇLAR TABLOSU
create table araclar(
    plaka_no varchar(7) not null primary key,
    marka varchar(20) not null,
    model varchar(20) not null,
    kayit_tarihi date not null,
    sofor_id char(11) not null,
    constraint fk_sofor foreign key(sofor_id) references personel(ssn) on
delete cascade
);

-- MUSTERILER TABLOSU
create table musteriler(
    tc char(11) not null primary key,
    ad varchar(20) not null,
    soyad varchar(20) not null,
    adres varchar(50) not null,
    cinsiyet varchar(5) not null,
    toplam_yolculuk_sayisi integer not null
);

-- YOLCULUK TABLOSU
create table yolculuk(
    yolculuk_id int not null primary key,
    baslangic varchar(20) not null,
    bitis varchar(20) not null,
    tutar int not null,
    arac_plaka varchar(7) not null,
    musteri_id char(11) not null,
    sofor_id char(11) not null,
    constraint fk_plaka foreign key(arac_plaka) references
araclar(plaka_no) on delete cascade,
    constraint fk_musteri foreign key(musteri_id) references
musteriler(tc) on delete cascade,
    constraint fk_sofor foreign key(sofor_id) references personel(ssn) on
delete cascade
)
```

2) Tablolarınızda primary key ve foreign key kısıtlarını kullanmalısınız.

Personel Tablosu için Primary key kısıtı

```
ssn char(11) not null primary key
```

Personel Tablosu için Foreign key kısıtı

```
constraint fk_ssn foreign key(superssn) references personel(ssn) on  
delete cascade
```

Araclar Tablosu için Primary key kısıtı

```
plaka_no varchar(7) not null primary key
```

Araclar Tablosu için Foreign key kısıtı

```
constraint fk_sofor foreign key(sofor_id) references personel(ssn) on  
delete cascade
```

Musteriler Tablosu için Primary key kısıtı

```
tc char(11) not null primary key
```

Yolculuk Tablosu için Primary key kısıtı

```
yolculuk_id int not null primary key
```

Yolculuk Tablosu için Foreign key kısıtı

```
constraint fk_plaka foreign key(arac_plaka) references araclar(plaka_no)  
on delete cascade,
```

```
constraint fk_musteri foreign key(musteri_id) references musteriler(tc)  
on delete cascade,
```

```
constraint fk_sofor foreign key(sofor_id) references personel(ssn) on  
delete cascade
```

3) En az bir tabloda silme kısıtı ve sayı kısıtı olmalıdır

Silme Kısıtı

Personel Tablosu için silme kısıtı

```
constraint fk_ssn foreign key(superssn) references personel(ssn) on delete cascade
```

Personel tablosunda çalışanlar için tanımlanmış superssn yabancı anahtarı ile superssn'e sahip kişi tablodan silindiğinde bu yöneticinin altında çalışan diğer personeller de silinir.

Araclar Tablosu için silme kısıtı

```
constraint fk_sofor foreign key(sofor_id) references personel(ssn) on delete cascade
```

Yolculuk Tablosu için silme kısıtı

```
constraint fk_plaka foreign key(arac_plaka) references araclar(plaka_no) on delete cascade,
```

```
constraint fk_musteri foreign key(musteri_id) references musteriler(tc) on delete cascade,
```

```
constraint fk_sofor foreign key(sofor_id) references personel(ssn) on delete cascade
```

Sayı Kısıtı

```
ALTER TABLE personel ADD CONSTRAINT check_salary CHECK (maas < 19000)
```

Çalışanların yöneticinin maaşından daha fazla maaşa sahip olmamaları sağlanır.

4) Arayüzden en az birer tane insert, update ve delete işlemi gerçekleştirilebilmelidir.

INSERT

```
INSERT INTO araclar(plaka_no, marka, model, kayit_tarihi, sofor_id) " +  
"VALUES( '" + plaka + "', '" + marka + "', '" + model + "', '" +  
kayit_tarihi + "', '" + sofor + "' )" "
```

DELETE

```
DELETE FROM musteriler where tc = '" + tc + "'"
```

UPDATE

```
UPDATE personel SET maas = '" + maas + "' " + "WHERE SSN = '" + tc + "'"
```

5) Arayüzden girilecek bir değere göre ekrana sonuçların listelendiği bir sorgu yazmalısınız.

```
String tc = txtMusteri.getText();

if(Kontrol(conn)) {
    String query = "SELECT * FROM musteriler WHERE tc = '" + tc + "'";
    try {
        person = (DefaultTableModel)table_1.getModel();
        person.setRowCount(0);
        PreparedStatement s = conn.prepareStatement(query);
        ResultSet r = s.executeQuery();
        if(!r.next()) {
            System.out.println("HATA");
        }
        else {
            System.out.println("DOGRU");
            Object [] row =
{r.getString(1),r.getString(2),r.getString(3),r.getString(4),r.getString(5),
r.getString(6)};
            System.out.println(row[1]);
            person.addRow(row);
        }

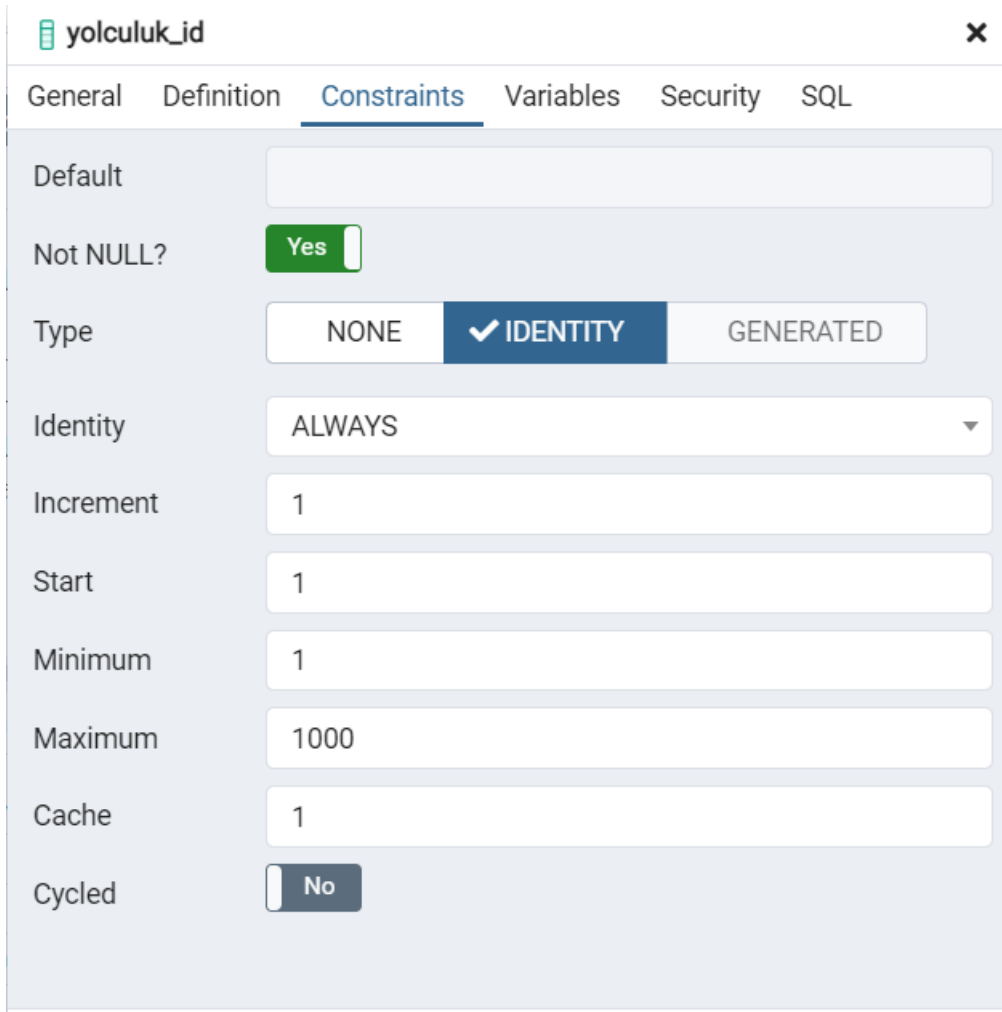
        s.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

6) Arayüzden çağrılan sorgulardan en az biri “view” olarak tanımlanmış olmalıdır.

```
CREATE VIEW soforler AS SELECT ssn FROM personel

CREATE VIEW yolcular AS SELECT tc FROM musteriler
```


7) En az bir adet “sequence” oluşturmali ve arayüzden yapılacak insert sırasında ilgili sütundaki değerlerin otomatik olarak atanmasını sağlamalısınız.



The screenshot shows the 'Constraints' tab for the 'yolculuk_id' sequence in pgAdmin. The 'Identity' checkbox is checked, and the 'Type' is set to 'IDENTITY'. The 'Identity' dropdown is set to 'ALWAYS'. The 'Increment' is 1, 'Start' is 1, 'Minimum' is 1, 'Maximum' is 1000, 'Cache' is 1, and 'Cycled' is set to 'No'.

Yolculuk tablosundaki yolculuk_id birincil anahtarı pgAdmin üzerinden sıralı artıacak şekilde düzenlemiştir.

8) Arayüzden çağrılan sorgulardan en az birinde union veya intersect veya except kullanmış olmalısınız.

```
select ad, soyad
from personel
where maas < 5000

except

select ad, soyad
from personel
where superssn is null
```

Maaşı 5000'den az olan ve yönetici olmayan çalışanlar belirlenmiştir.

9) Sorgularınızın en az biri aggregate fonksiyonlar içermeli, having ifadesi kullanılmalıdır.

```
SELECT ad, soyad, sum(tutar)
FROM personel, yolculuk
WHERE ssn = sofor_id
GROUP BY ad, soyad
HAVING sum(tutar) > 1000
```

En çok kazandıran şoförler belirlenmiştir.

10) Arayüzden girilen değerleri parametre olarak alıp ekrana sonuç döndüren 3 farklı SQL fonksiyonu tanımlamış olmalısınız. Bu fonksiyonlarda en az birer tane “record” ve “cursor” tanımı-kullanımı olmalıdır.

Record(Tür) tanımı yapılarak tanımlanmış fonksiyon

```
CREATE TYPE new_yolculuk AS (id integer, baslangic VARCHAR(20), bitis
varchar(20), cost integer, plaka varchar(7), customer varchar(11), sofor
varchar(11));
CREATE or REPLACE FUNCTION yolculuk_bilgisi(m_tc yolculuk.musteri_id%type)
RETURNS new_yolculuk AS '
DECLARE
bilgi new_yolculuk;
BEGIN
SELECT yolculuk_id, baslangic, bitis, tutar, arac_plaka, muster_i_id,
sofor_id INTO bilgi
FROM yolculuk
WHERE muster_i_id = m_tc;
return bilgi;
END;
' LANGUAGE 'plpgsql';
```

Cursor içeren fonksiyon örneği

```
CREATE or REPLACE FUNCTION toplam_kazanc(sofor_tc personel.ssn%type)
RETURNS integer AS '
DECLARE
kazanc integer;
curs CURSOR FOR SELECT tutar FROM yolculuk WHERE sofor_id = sofor_tc;
BEGIN
kazanc := 0;
FOR satir IN curs LOOP
kazanc := kazanc + satir.tutar;
END LOOP;
RETURN kazanc;
END;
' LANGUAGE 'plpgsql';
```

```

CREATE TYPE customer AS(ad varchar(20), soyad varchar(20), tutar integer,
tercih_sayisi integer);
CREATE or REPLACE FUNCTION encok_terciheden()
RETURNS customer[] AS '
DECLARE
c customer[];
i integer;
curs CURSOR FOR SELECT ad, soyad, sum(tutar), count(*) FROM yolculuk,
musteriler WHERE tc = musteri_id GROUP BY tc ORDER BY count(*) DESC LIMIT
5;
BEGIN
    i := 1;
    FOR satir IN curs LOOP
        i := i + 1;
    END LOOP;
    RETURN c;
END;
' LANGUAGE 'plpgsql';

```

11) 2 adet trigger tanımlamalı ve arayüzden girilecek değerlerle tetiklemelisiniz. Trigger'ın çalıştığına dair arayüze bilgilendirme mesajı döndürülmelidir.

Maaşı güncellenecek olan kişinin yeni maaşının eski maaşının 1.5 katından fazla olmasını engelleyen Trigger

```

CREATE OR REPLACE FUNCTION trig_fonk_maas()
RETURNS TRIGGER AS $$
BEGIN
    IF(new.maas>1.5*old.maas) THEN
        RETURN old;
    ELSE
        RETURN new;
    END IF;
END;
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER trg_maas
BEFORE UPDATE
ON personel
FOR EACH ROW EXECUTE PROCEDURE trig_fonk_maas();

```

Yolculuk tablosuna insert yapıldığı zaman yolculuğu yapan müşterinin toplam yolculuk sayısını 1 artıran Trigger

```

CREATE OR REPLACE FUNCTION trig_yolculuk()
RETURNS TRIGGER AS $$
BEGIN
    update musteriler
    set toplam_yolculuk_sayisi=toplam_yolculuk_sayisi + 1
    where tc=new.musteri_id;
    return new;
END;

$$ LANGUAGE 'plpgsql';
CREATE TRIGGER trg_y
AFTER INSERT
ON yolculuk
FOR EACH ROW EXECUTE PROCEDURE trig_yolculuk();

```