



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK ELEKTRONİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**

**Veritabanı Yönetimi Projesi**  
**TRAFİK CEZA BİLGİ SİSTEMİ**

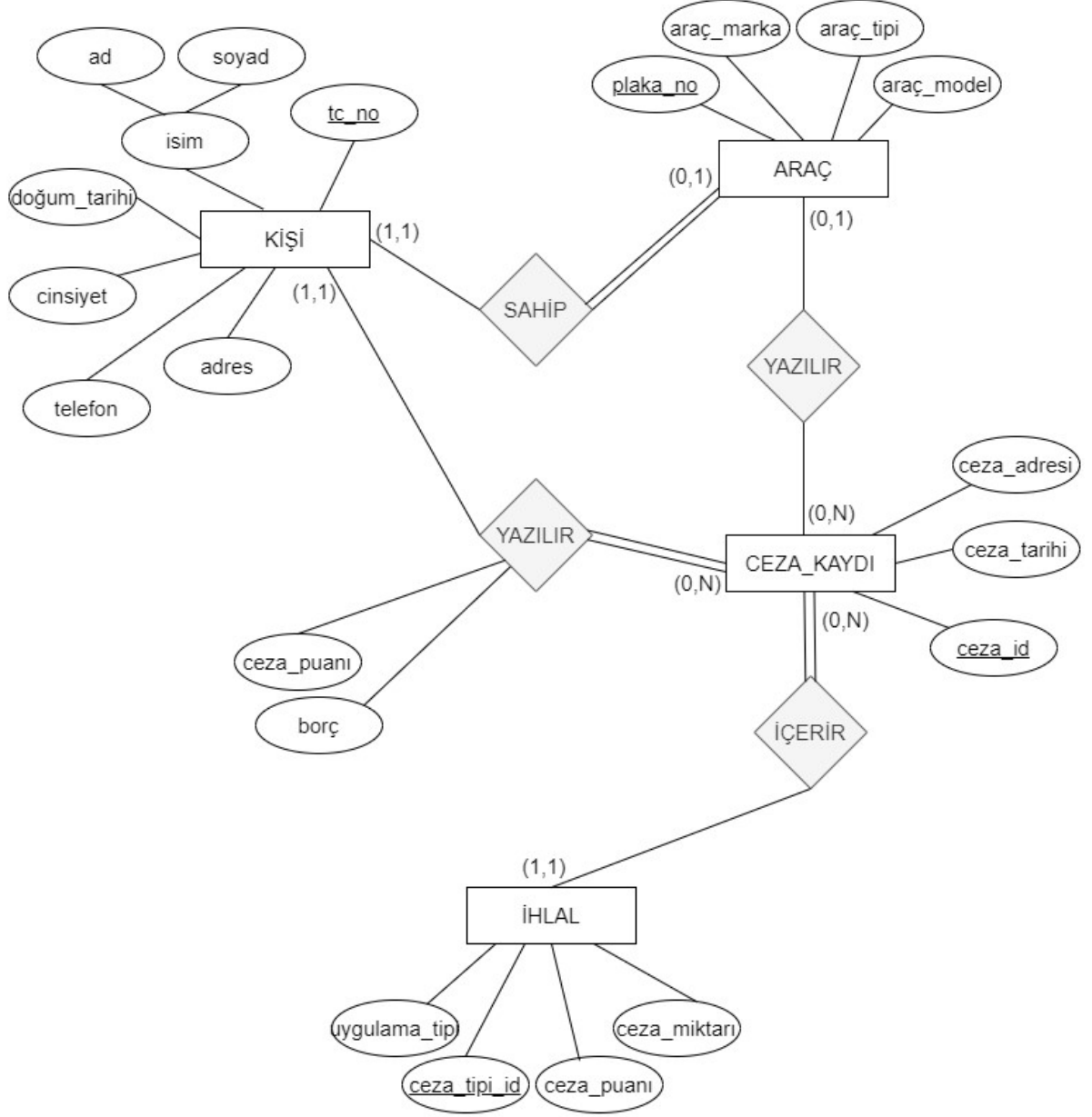
**Betül Özmaral 18011614**

**Fatma Tanrıkulu 16011085**

**Ayşe Hilal Doğan 17011907**

**Zeynep Çolak 20011607**

## ER DİYAGRAMI



ER diyagramı trafik ceza bilgi sistemini ve kısıtları daha detaylı anlatmak amacıyla min-max notasyonu ile verilmiştir. Bu notasyon, dersin referans kitabının (Fundamentals of Database Systems, Elmasri, Navathe, 7th edition) 84-85. sayfalarında yer almaktadır.

## VERİTABANI TABLO GÖRÜNÜMLERİ

**Tablo 1-**İhlal tablosu:

	ceza_tipi_id [PK] numeric (3)	ihlal_konusu character varying (50)	uygulama_tipi numeric (1)	ihlal_ceza_puani smallint	ceza_miktari integer
1	536	Alkollu arac kullanimi	1	20	1002
2	449	Trafik isiklarina uymama	2	20	235
3	302	Sehir ici azami hiz asimi	2	10	235
4	434	Arac tescil belgesinin aracta bulunmaması	1	10	92
5	506	Surucu belgesi yetkisi disinda arac kullanma	1	20	1002
6	289	Yayalar icin belirtilen isaretlere uymamak	1	5	108
7	374	Tasit yolunda yurumek	1	5	108
8	672	Muayenesi yapilmamis bir arac kullanmak	1	10	235
9	123	Aracta olmasi gereken isaretleri bulundurmamak	2	5	108
10	741	Sesli isaretlerin izin verilmeyen araclarda olmasi	2	10	1002

**Tablo 2-**Araç tablosu:

	plaka_no [PK] character varying (10)	arac_tc_no numeric (11)	arac_marka character varying (15)	arac_tipi character varying (10)	arac_model character varying (25)
1	34SV5959	23456789076	BMW	SEDAN	520 I
2	34BB9092	34567890298	MERCEDES	SUV	GLA
3	34CS5042	45678923456	AUDI	HATCHBACK	A6
4	34MTK44	34567890234	VOLKSWAGEN	SUV	TIGUAN
5	34MHD99	56789034567	VOLVO	SEDAN	S90
6	34DNU986	56437878965	FIAT	SUV	500X
7	34YSF386	45678923456	RENAULT	SEDAN	MEGANE
8	34SV9041	67890456734	FORD	HATCHBACK	FOCUS
9	34BBF965	34567823456	NISSAN	SUV	QASQHAİ
10	34UK1032	74495261318	FORD	SUV	KUGA

**Tablo 3-**Kişi tablosu:

	tc_no [PK] numeric (11)	isim character varying (15)	soyisim character varying (15)	dogum_tarihi date	adres character varying (50)	telefon numeric (10)	cinsiyet character (5)	ceza_puani smallint	borc integer
1	67890456734	Zeynep	Colak	1992-05-21	Yildiz Mah. Ahu Sok. No:48 ...	5310064621	Kadin	120	0
2	43267890345	Secil	Eker	1988-12-24	Pinar Mah. Gocuk Apt. Kat:8 ...	5485451997	Kadin	20	0
3	34567890298	Ahmet	Yildiz	1974-12-02	Keneddy Cad. No:66/A Bakir...	5438023430	Erkek	0	0
4	45608923456	Hakan	Bilgin	1982-09-04	Kilic Sok. Yilmaz Apt. Kat:1 ...	5392275040	Erkek	0	0
5	45678923406	Dilara	Cicek	1990-03-16	Cigdem Mah. Deniz Sok. No:...	5377359532	Kadin	0	0
6	74495261318	Huseyin	Akin	1988-08-01	Sakarya Mah. No:88 Avcilar/...	4307382419	Erkek	75	235
7	34567823456	Deniz	Turan	1991-02-02	Zafer Mah. Issiz Sok. No:17...	5325480242	Erkek	370	470
8	23456789076	Fatma	Tanrikulu	1993-09-11	Basak Mah. No:27 Basakseh...	5417543010	Kadin	60	1002
9	56789034567	Burak	Acar	1987-11-30	Sarayburnu Mah. No:37 Sey...	5454151918	Erkek	60	1237
10	34567890234	Betul	Ozmaral	1995-06-25	Mimoza Sok. D: 18 Kemer/A...	5365244513	Kadin	170	92
11	45678923456	Ayşe	Dogan	1985-08-18	Altinsehir Mah. Cicek Apt. K...	5527595471	Kadin	110	705
12	56437878965	Emre	Yalcinkaya	1977-07-20	Cagri Sok. Yasemin Apt. No:...	5561791913	Erkek	110	1237
13	23628915743	Yasemin	Yesilsu	1968-04-15	Elmalı Mah. Yesil Apt. Kat:4 ...	5492167673	Kadin	10	108

**Tablo 4-Ceza kaydı tablosu:**

	ceza_id [PK] integer	ceza_tipi numeric (3)	ceza_tarihi date	ceza_adres character varying (50)	uyg_kisi_tc numeric (11)	uyg_plaka_no character varying (10)
1	89	449	2020-10-27	Cumhuriyet Sok. Seyhan/Ad...	34567823456	34UK1032
2	90	302	2017-07-16	Yasemin Sok. Sariyer/Istanbul	56789034567	34YSF386
3	91	302	2019-01-01	Buyukhendek Cad. Beyoglu/I...	34567823456	34CS5042
4	92	536	2019-03-09	Mercan Sok. Eminonu/Istan...	23456789076	[null]
5	93	672	2017-02-10	Bahariye Sok. Kadikoy/Istan...	56437878965	[null]
6	94	536	2018-05-11	Basak Mah. Basaksehir/Ista...	56789034567	[null]
7	95	434	2020-09-13	Saklicesme Sok. Kemer/Ant...	34567890234	[null]
8	96	672	2018-01-10	Polat Sok. Kecioren/Ankara	45678923456	[null]
9	97	536	2019-04-05	Camkusu Sok. Edremit/ Bali...	56437878965	[null]
10	98	289	2020-08-15	E5 Karayolu Bahcelievler/Ist...	23628915743	[null]

## SİSTEM İÇİN GERÇEKLENEN MADDELER

**1. Oluşturacağınız veritabanı en az 4 tablo içermelidir,**

**2. Tablolarınızda primary key ve foreign key kısıtlarını kullanmalısınız.**

```
create table kisi(  
    tc_no numeric(11) not null primary key  
    ...  
create table ihlal(  
    ceza_tipi_id numeric(3) not null primary key  
    ...  
create table arac(  
    plaka_no varchar(10) not null primary key,  
    foreign key(arac_tc_no) references kisi(tc_no)  
    ...  
create table ceza_kaydi(  
    ceza_id int primary key,  
    ...  
    foreign key(uyg_plaka_no) references arac(plaka_no),  
    foreign key(uyg_kisi_tc) references kisi(tc_no)  
    ...
```

**3. En az bir tabloda silme kısıtı ve sayı kısıtı olmalıdır.**

```
create table ceza_kaydi(  
    ...  
    constraint fk_ceza_kaydi foreign key(ceza_tipi) references ihlal(ceza_tipi_id) on  
    delete cascade  
    ...  
create table ihlal(  
    uygulama_tipi numeric(1) not null,  
    check(uygulama_tipi in (1,2))
```

...

```
create table kisi(  
    dogum_tarihi date,  
    check (dogum_tarihi>'01/01/1920')
```

...

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

- **insert işlemi “ceza\_ekle” ekranında çalışır.**

--Javada kullanılan sorgu

```
String query1 = " insert into ceza_kaydi ( ceza_tipi, ceza_tarihi, ceza_adres,  
uyg_kisi_tc,uyg_plaka_no)" + " values ( ?, ?, ?, ?,?)" ;
```

- **update işlemi “ceza silme” ekranında çalışır.**

--Bir butonun çalıştırdığı sorgu

```
String cezano = JOptionPane.showInputDialog("Silinecek cezanın id'sini girin");
```

```
String query3 = "UPDATE ceza_kaydi SET uyg_plaka_no = NULL WHERE ceza_id = ' " +  
cezano + " ' " ;
```

--Başka bir butonun çalıştırdığı sorgu

```
String cezano = JOptionPane.showInputDialog("Silinecek cezanın id'sini girin");
```

```
String query3 = "UPDATE ceza_kaydi SET uyg_kisi_tc= NULL WHERE ceza_id = ' " +  
cezano + " ' " ;
```

- **delete işlemi update triggerının içinde çalışmaktadır.**

```
IF (new.uyg_plaka_no IS NULL) AND (new.uyg_kisi_tc IS NULL) THEN
```

```
    RAISE NOTICE 'Ceza kaydi siliniyor';
```

```
    DELETE FROM ceza_kaydi
```

```
    WHERE ceza_id= new.ceza_id;
```

```
END IF;
```

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

- **Ceza görüntüleme ve silme ekranında girilen tc ile tc'ye ait cezalar ekrandaki JTable'da gösterilmektedir.**

--Javada kullanılan sorgu

```
String query2 = "SELECT ceza_id,ceza_tipi,ceza_tarihi,ceza_adres FROM ceza_kaydi  
WHERE uyg_kisi_tc = " + tc;
```

- *Ceza görüntüleme ve silme ekranında girilen plaka ile plakaya yazılmış cezalar ekrandaki JTable’da gösterilmektedir.*

--Javada kullanılan sorgu

```
String query2 = "SELECT ceza_id, ceza_tipi, ceza_tarihi, ceza_adres FROM ceza_kaydi  
WHERE uyg_plaka_no= ' " + plaka + " ' " ;
```

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

- *Ceza ekleme ekranındaki ihlaller ve özellikleri butonuyla kayıtlı cezalar veritabanından view ile çekilip JTable’da gösterilmektedir.*

--Javada kullanılan sorgu:

```
String query = "Select * from tipler";
```

--View için yazılmış SQL sorgusu:

```
CREATE VIEW tipler AS SELECT ceza_tipi_id, ihlal_konusu, uygulama_tipi from ihlal
```

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

- *Arayüzden yapılan ceza kaydı insert’inde ceza\_id kısmına değer verilmemiştir. Insert yaparken sql otomatik sequence ile değeri vermiştir.*

```
create sequence ceza_id_seq  
start 1  
maxvalue 999  
increment by 1  
owned by ceza_kaydi.ceza_id;
```

--Sequence’in ilgili kolona bağlanması:

```
ALTER TABLE ceza_kaydi ALTER COLUMN ceza_id SET DEFAULT  
nextval('ceza_id_seq');
```

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

- *Tanımlı işlemler ekranındaki buton, ceza kaydı olup arabası olmayan kişileri getirmektedir.*

--Javada kullanılan sorgu

```
String query = "select isim, soyisim, adres from kisi, ceza_kaydi where tc_no=uyg_kisi_tc  
except"
```

```
+ " select isim, soyisim, adres from kisi, arac where tc_no=arac_tc_no";
```

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

- *Aggregate fonksiyon RECORD isimli fonksiyonun içinde kullanılmıştır.*

```
SELECT min(current_date - ceza_tarihi) INTO gun from ceza_kaydi,kisi WHERE  
tc=uyg_kisi_tc AND tc=tc_no;
```

- *Having indir isimli fonksiyonda kullanılmıştır.*

```
SELECT e.tc_no FROM kisi e, kisi e2 GROUP BY e.tc_no  
HAVING avg(e2.ceza_puani)>e.ceza_puani) AND (ceza_puani>0)
```

**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.**

**Fonksiyonlar:**

- *İlk fonksiyon arayüzden parametre olarak aldığı değeri, ortalama ceza puanını altında kalan ceza sahiplerine yüzdelik indirimini olarak uyguluyor.*
- *Return değeri olarak kişi sütununda, indirim uyguladığı kişilerin sayısını döndürmektedir.*
- *9 ve 10.maddede de belirtilen fonksiyonlardan biri yazılmış olup aggregate ve having ifadeleri kullanılmıştır.*

```
CREATE OR REPLACE FUNCTION indir(indirim INTEGER)
```

```
RETURNS integer AS $$
```

```
DECLARE
```

```
sayiinteger;
```

```
BEGIN
```

```
WITH rows AS (
```

```
UPDATE kisi SET borc= borc- borc*(indirim/100.0) WHERE tc_no IN
```

```
(SELECT e.tc_no FROM kisi e, kisi e2 GROUP BY e.tc_no
```

```
HAVING avg(e2.ceza_puani)>e.ceza_puani) AND (ceza_puani>0)
```

```
RETURNING 1)
```



```
SELECT count(*) INTO sayi FROM rows;

return sayi;

END;

$$ LANGUAGE 'plpgsql';
```

- *İkinci fonksiyon arayüzden parametre olarak aldığı TC numarasına sahip kullanıcın isim, soyisim, borç ve en son borcunun üzerinden geçen süreyi gün cinsinden bularak ekrana yazdırır.*
- *Fonksiyon içerisinde “record” tanımı yapılmış olup farklı tablolardaki değerleri birleştiren bir yeni tür oluşturulmuştur. Fonksiyon sonucunda oluşturulan yeni tür return edilmektedir.*
- *Fonksiyon madde 10’da istenilenlere uygun hazırlanmıştır.*

```
CREATE TYPE yeni_tur AS (isim VARCHAR(15), soyisim VARCHAR(15), borc
INTEGER, gun int);
```

```
CREATE OR REPLACE FUNCTION RECORD(tc numeric)
```

```
RETURNS yeni_tur AS $$
```

```
DECLARE
```

```
    bilgi yeni_tur;
```

```
    gun int := 0;
```

```
BEGIN
```

```
    SELECT min(current_date - ceza_tarihi) INTO gun from ceza_kaydi, kisi WHERE
tc=uyg_kisi_tc AND tc=tc_no;
```

```
    SELECT isim, soyisim, borc, gun INTO bilgi
```

```
    FROM kisi, ceza_kaydi
```

```
    WHERE tc=uyg_kisi_tc AND tc=tc_no;
```

```
    RETURN bilgi;
```

```
END;
```

```
$$ LANGUAGE 'plpgsql';
```

- *Son fonksiyon arayüzden parametre olarak aldığı integer değeri, ceza puanı 50'nin altında olan ve borcu 0 olan ceza sahiplerine yüzdelik indirimi olarak uygular.*
- *Fonksiyon içerisinde bir "cursor" tanımı yapılmış olup, fonksiyon sonucunda cursor tablosundaki satır sayısı yani kaç tane ceza sahiplerine indirim yapıldığı bilgisi integer olarak return edilir.*
- *Fonksiyon madde 10'da istenilenlere uygun hazırlanmıştır.*

```

CREATE OR REPLACE FUNCTION indir2(indirim int)
RETURNS integer AS $$
DECLARE
    sayi integer;
    kisi_cursor CURSOR FOR SELECT e.tc_no, e.ceza_puani, e.borc
    FROM kisi e
    WHERE (e.ceza_puani < 50) AND (e.borc=0) AND e.ceza_puani <> 0;
BEGIN
    sayi:=0;
    FOR satir IN kisi_cursor LOOP
        sayi:=sayi+1;
        UPDATE kisi
        SET ceza_puani= ceza_puani - indirim
        Where tc_no = satir.tc_no AND ceza_puani > indirim;
    END LOOP;
    Return sayi;

END;
$$ LANGUAGE 'plpgsql';

```

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

- *Arayüzde kullanıcıdan ceza\_id alınarak gerekli kolonda update işlemi gerçekleştirilir. Bunun sonucunda aşağıdaki update\_trigger çalışır.*
- *Arayüzde ceza ekleme ekranıyla ceza\_kaydi tablosuna insert yapılır ve insert\_trigger çalışır.*

**->insert\_trigger :** *Kişi tablosuna ceza eklendiğinde tetiklenir. Arayüzden girilen bir değerle, ceza\_kaydi tablosuna insert yapıldıktan sonra çalışır. Ceza\_kaydi'na bir insert yapıldığında;*

*1- Ceza araç sahibine yazılacaksa ekrana bilgilendirme mesajı verilir ve aracın sahibinin tc si plaka üzerinden araç tablosundan çekilir ve o kişinin ceza puanı ve borcu cezanın türüne göre arttırılır. (ceza puanı ve borcunun ne kadar arttırılacağı bilgisi ihlal tablosundan çekilir.) Eğer aracın sahibinin tcsi ve girilen tc aynı ise, iki kez ceza yazılmaması için bir kontrol yapılır ve ceza ona göre yazılır, ekrana bilgilendirme mesajı verilir.*

*2- Ceza şahsa yazılacaksa, kisi tablosunda girilen tc üzerine ceza yazılır. Ceza puanı ve borcu, cezanın türüne bağlı olarak arttırılır.*

```
CREATE TRIGGER insert_trigger
```

```
AFTER INSERT
```

```
ON ceza_kaydi
```

```
FOR EACH ROW EXECUTE PROCEDURE trig_fonk1();
```

```
CREATE or REPLACE FUNCTION trig_fonk1()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
IF (new.uyg_plaka_no IS NOT NULL) THEN
```

```
RAISE NOTICE 'Araci sahibine ceza yaziliyor';
```

```
UPDATE kisi
```

```
set ceza_puanı = ceza_puanı + (SELECT a.ihlal_ceza_puanı FROM ihlal a WHERE  
new.ceza_tipi = a.ceza_tipi_id),
```

```
borc = borc + (SELECT b.ceza_miktari FROM ihlal b WHERE new.ceza_tipi  
= b.ceza_tipi_id)
```

```
WHERE tc_no IN(SELECT arac_tc_no FROM arac WHERE  
new.uyg_plaka_no = plaka_no);
```

```
END IF;
```

```

        IF (new.uyg_kisi_tc IN (SELECT arac_tc_no FROM arac WHERE
new.uyg_plaka_no = plaka_no) ) THEN

        RAISE NOTICE 'Araci kullanan kisi plaka sahibiyle ayni kisi, plaka_no null yapılıyor
ve plakaya yazılan ceza silinecek ';

        UPDATE ceza_kaydi set uyg_plaka_no=NULL where ceza_id=new.ceza_id ;

        END IF;

        IF (new.uyg_kisi_tc IS NOT NULL) THEN

        RAISE NOTICE 'Sahsa ceza yaziliyor';

        UPDATE kisi

        set ceza_puani = ceza_puani + (SELECT a.ihlal_ceza_puani FROM ihlal a WHERE
new.ceza_tipi = a.ceza_tipi_id),

        borc = borc + (SELECT b.ceza_miktari FROM ihlal b WHERE new.ceza_tipi
= b.ceza_tipi_id)

        WHERE tc_no = new.uyg_kisi_tc;

        END IF;

        RETURN new;

END;

$$ LANGUAGE 'plpgsql';

```

**-> update\_trigger :** Kişinin cezası ödendiğinde arayüzde ceza\_kaydi tablosunda uyg\_kisi\_tc null yapılır ve araç sahibi cezasını öderse uyg\_plaka\_no null yapılır. Bu update'ler ile bu trigger çalışır;

1- Arayüzde uyg\_kisi\_tc null yapılıncaya, ekrana bilgilendirme mesajı verilir ve kisi tablosunda kişinin borcu ödediği cezaya göre azaltılır.

2- Arayüzde uyg\_plaka\_no null yapılıncaya, ekrana bilgilendirme mesajı verilir ve plaka sahibinin tc no'su – bu değer plaka vasıtasıyla araç tablosundan bulunur- sonra kişi tablosunda o tc numarasının ödediği cezaya göre borcu azaltılır.

3- Eğer araç sahibi tc ve şahıs tc aynı ise uyg\_plaka\_no null yapılır.

4- Eğer ceza\_kaydi tablosunda hem kişi hem araç sahibi cezasını ödemişse bu satır tablodan silinir.

```
CREATE TRIGGER update_trigger
AFTER UPDATE OF uyg_kisi_tc ,uyg_plaka_no
ON ceza_kaydi
FOR EACH ROW EXECUTE PROCEDURE trig_fonk2();
```

```
CREATE or REPLACE FUNCTION trig_fonk2()
RETURNS TRIGGER AS $$
```

```
DECLARE
```

```
    ihlaltutar ihlal.ceza_miktari%type;
```

```
    borcu kisi.borc%type;
```

```
    plaka_tc arac.arac_tc_no%type;
```

```
BEGIN
```

```
    SELECT b.ceza_miktari INTO ihlaltutar FROM ihlal b WHERE b.ceza_tipi_id =
old.ceza_tipi;
```

```
    IF (old.uyg_kisi_tc IS NOT NULL) AND (new.uyg_kisi_tc IS NULL) THEN
```

```
        SELECT borc INTO borcu FROM kisi WHERE tc_no=old.uyg_kisi_tc;
```

```
        RAISE NOTICE 'Kisinin borcundan azaltiliyor';
```

```
        UPDATE kisi
```

```
        SET borc = borc- ihlaltutar
```

```
        WHERE tc_no = old.uyg_kisi_tc AND (borc-ihlaltutar)>=0;
```

```
    IF (borcu<ihlaltutar) THEN
```

```
        UPDATE kisi
```

```
        SET borc = 0
```

```
        WHERE tc_no = old.uyg_kisi_tc;
```

```
    END IF;
```

```
END IF;
```

```

IF (old.uyg_plaka_no IS NOT NULL) AND (new.uyg_plaka_no IS NULL) THEN

    SELECT arac_tc_no INTO plaka_tc FROM arac WHERE old.uyg_plaka_no =
plaka_no;

    SELECT borc INTO borcu FROM kisi where tc_no =plaka_tc ;

    RAISE NOTICE 'Arac sahibinin borcundan azaltiliyor';

    UPDATE kisi

    SET borc = borc - ihlaltutar

    WHERE tc_no =plaka_tc AND (borc-ihlaltutar)>=0;

    IF (borcu<ihlaltutar) THEN

        UPDATE kisi

        SET borc = 0

        WHERE tc_no =plaka_tc;

    END IF;

END IF;

IF (new.uyg_plaka_no IS NULL) AND (new.uyg_kisi_tc IS NULL) THEN

    RAISE NOTICE 'Ceza kaydi siliniyor';

    DELETE FROM ceza_kaydi

        WHERE ceza_id= new.ceza_id;

    END IF;

    RETURN new;

END;

$$ LANGUAGE 'plpgsql';

```