SAKARYA
ÜNİVERSİTESİ

Bilgisayar ve Bilişim Bilimleri Fakültesi

Bilgisayar Mühendisliği

Bölümü

Öğrenciler:

G211210069 Hüseyin Akbal huseyin.akbal@ogr.sakarya.edu.tr

G221210039 Dilara Çetin dilara.cetin2@ogr.sakarya.edu.tr

Ders: Veritabanı Yönetim Sistemleri

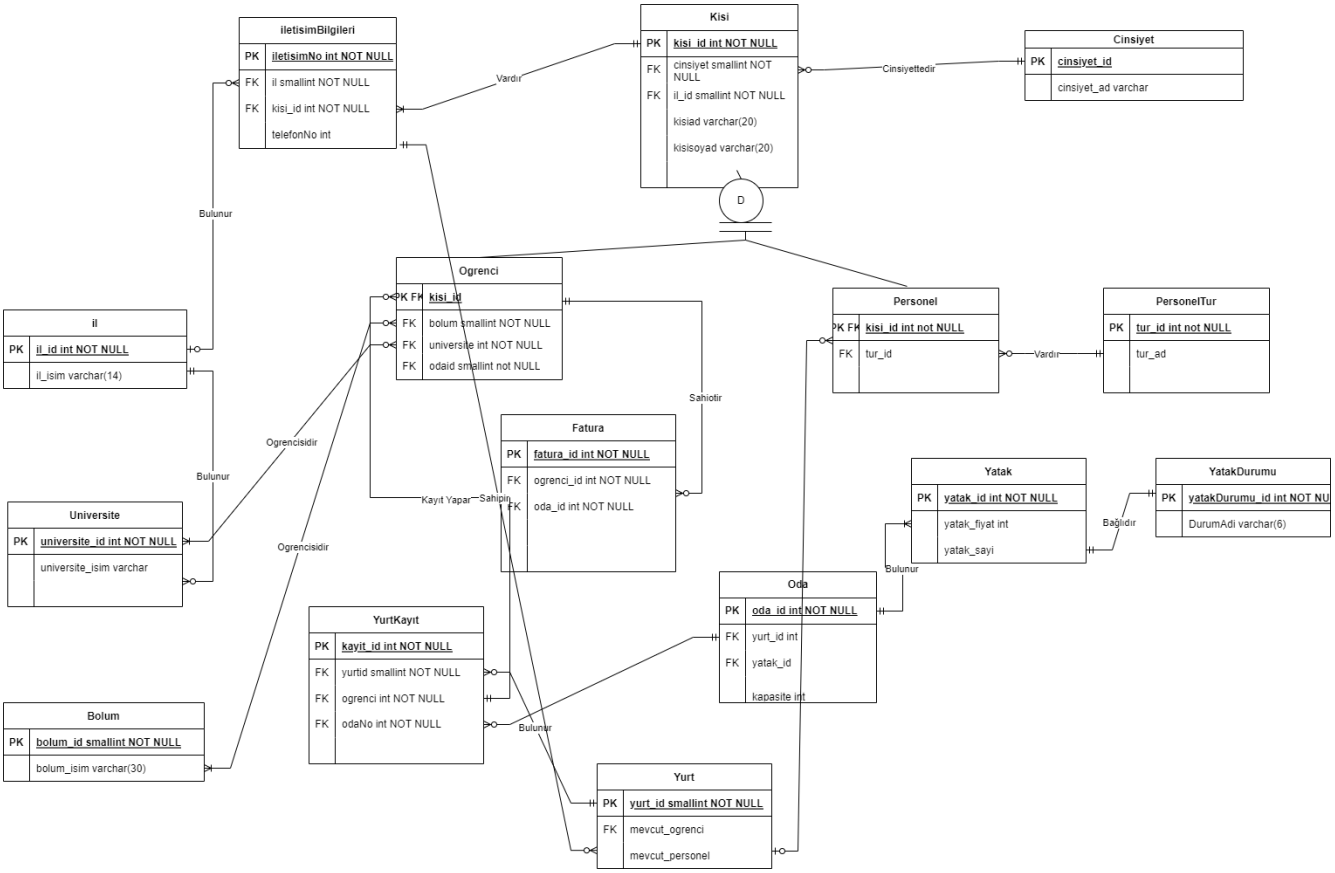
Şube: 2. Öğretim B Grubu

Dersi Veren: İsmail Öztel

Problemin Tanımı

Projemizde bir Yurt Kayıt Sistemi yaptık. Bu bir yurdun gündelik kayıt ekleme silme işlemlerinin kolaylaştırılması ve bilgiye daha kolay bir şekilde ulaşılması için yapılmış bir sistem. Bu kayıt sisteminde her öğrenci ve her personel üzerinden arama, ekleme, güncelleme, silme işlemleri yapılabiliyor. Aynı zamanda öğrencinin bulunduğu oda, öğrencinin kayıt fatura bilgilerine de erişilebiliyor. Projemizde toplam 13 adet saklı yordam/işlev kullandık ve toplam 4 adet tetikleyici kullandık. Bir masaüstü uygulaması yaptık ve programlama dili olarak C#'ı tercih ettik.

Veri Bağıntı Diyagramı



İlişkisel Şema

- Kisi(kisi_id: int, kisiad: varchar, kisisoyad: varchar, cinsiyet: smallint, il_id: smallint)
- Ogrenci(kisi_id: int, bolum: smallint, universite: int, odaid: smallint)
- Personel(kisi_id: int, tur_id: int)
- PersonelTur(tur_id: int, tur_ad: varchar)
- Cinsiyet(cinsiyet_id: int, cinsiyet_ad: varchar)
- İletisimBilgileri(iletisimNo: int, il: smallint, kisi_id: int, telefonNo: int)

- İl(il_id: int, il_isim: varchar)
- Universite(universite_id: int, il: int, universite_isim: varchar)
- Bolum(bolum_id: smallint, bolum_isim: varchar)
- Fatura(fatura_id: int, ogrenci_id: int, oda_id: int)
- YurtKayit(kayit_id: int, yurt_id: smallint, ogrenci: int, odaNo: int)
- Oda(oda_id: int, kapasite: int)
- Yatak(yatak_id: int, yatak_fiyat: int)
- YatakDurumu(yatakDurumu: int, durumAdi: varchar)Yurt(yurt_id: smallint, iletişim: int, kapasite: int)
- Yurt(yurt_id: smallint, mevcut_ogrenci: varchar, mevcut_personel: varchar)

İş Kuralları

- Her öğrenci en az bir bölümde okur.
- Bir bölüm çok sayıda öğrenci tarafından okunabilir.
- Her kişinin bir cinsiyeti vardır.
- Bir cinsiyetten çok sayıda kişi olabilir.
- Bir kişinin çok sayıda iletişim bilgisi olabilir.
- Bir iletişim bilgisi bir kişiye aittir.
- Her öğrenci en fazla bir yurtta bulunur.
- Bir yurtta çok sayıda öğrenci bulunabilir.
- Her öğrenci bir üniversitenin öğrencisidir.
- Bir üniversitenin çok sayıda öğrencisi olabilir.
- Her yurdun bir iletişim bilgisi vardır.
- Bir iletişim bilgisi bir yurda ait olmayabilir.
- Bir iletişim bilgisinde il bilgisi bulunmayabilir.
- Bir il çok sayıda iletişim bilgisinde bulunabilir.
- Her üniversite bir ilde bulunur.
- Bir ilde çok sayıda üniversite bulunabilir.
- Bir öğrenci bir yurda kayıt yapar.
- Bir yurtta çok sayıda öğrenci olabilir
- Bir yurda kayıtta bir oda bilgisi bulunur.
- Bir odaya ait birden çok kayıt bulunabilir.
- Bir yurda ait çok sayıda kayıt bilgisi bulunabilir.
- Bir kayıt bilgisi bir yurda aittir.
- Bir öğrencinin birden çok faturası olabilir.
- Bir fatura bir öğrenciye aittir.
- Bir odada çok sayıda yatak bulunabilir.
- Bir yatak bir odada bulunur.
- Bir yatağın bir uygunluk durumu vardır.
- Bir yatak durumu birden fazla yatağa ait olabilir
- Bir yurtta çok sayıda personel olabilir
- Bir personel bir yurtta çalışabilir.

- Kişinin numarası, adı, soyadı, cinsiyeti, yaiadığı il vardır.
- Öğrenci ve personel; kişiden kalıtım alır.
- Öğrenci; kişinin numarasını, okuduğu bölümün numarasını, ve üniversitenin numarasını ve kaldığı odayı tutar.
- Personel, kişinin numarasını ve personelin türünü tutar.
- İlin numarası ve adı vardır.
- İletişim bilgilerinde iletişim no, telefon no, adres, il, ve kişinin numarası saklanır.
- Üniversitenin numarası ve adı vardır.
- Bölümün numarası ve adı vardır.
- Yurdun numarası, mevcut öğrenci ve personel sayısı vardır.
- Oda; odanın numarasını, , yurt numarasını tutar.
- Yurt kaydında kayıt numarası, yurt numarası, öğrenci numarası, oda numarası vardır.
- Yatağın yatak numarası, ve yatağın fiyatı vardır.
- Yatak durumu, yatak durumu numarasını ve durumun adını tutar.
- Faturada; fatura numarası, öğrenci numarası, oda numarası saklanır.
- Cinsiyetin numarası ve adı vardır.

Uygulama Kodları

SQL Kodları:

Table: public.bolum

```
DROP TABLE IF EXISTS public.bolum;

CREATE TABLE IF NOT EXISTS public.bolum
(
    bolum_id smallint NOT NULL,
    bolum_isim character varying(50) COLLATE pg_catalog."default",
    CONSTRAINT "BOLUM_pkey" PRIMARY KEY (bolum_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.bolum
    OWNER to postgres;
```

Table: public.cinsiyet

```
DROP TABLE IF EXISTS public.cinsiyet;

CREATE TABLE IF NOT EXISTS public.cinsiyet
(
    cinsiyet_id smallint NOT NULL,
    cinsiyet_ad character varying(5) COLLATE pg_catalog."default",
    CONSTRAINT "CINSIYET_pkey" PRIMARY KEY (cinsiyet_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.cinsiyet
    OWNER to postgres;
```

Table: public.fatura

```
DROP TABLE IF EXISTS public.fatura;

CREATE TABLE IF NOT EXISTS public.fatura
(
    fatura_id smallint NOT NULL,
    ogrenci_id smallint NOT NULL,
    oda_id smallint,
    fatura_tarihi date NOT NULL,
    CONSTRAINT fatura_pkey PRIMARY KEY (fatura_id),
    CONSTRAINT oda_fatura_fkey FOREIGN KEY (oda_id)
        REFERENCES public.oda (oda_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT ogrenci_fatura_fkey FOREIGN KEY (ogrenci_id)
        REFERENCES public.ogrenci (kisi_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.fatura
    OWNER to postgres;
```

Table: public.il

```
DROP TABLE IF EXISTS public.il;

CREATE TABLE IF NOT EXISTS public.il
(
    il_id smallint NOT NULL,
    il_isim character varying(13) COLLATE pg_catalog."default",
    CONSTRAINT "IL_pkey" PRIMARY KEY (il_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.il
    OWNER to postgres;
```

Table: public.iletisim_bilgileri

```
DROP TABLE IF EXISTS public.iletisim_bilgileri;

CREATE TABLE IF NOT EXISTS public.iletisim_bilgileri
(
    iletisim_id smallint NOT NULL,
    telefon_no bigint,
    il_no smallint NOT NULL,
    kisi_no smallint NOT NULL,
    CONSTRAINT iletisim_bilgileri_pkey PRIMARY KEY (iletisim_id),
    CONSTRAINT il_iletisim_fkey FOREIGN KEY (il_no)
        REFERENCES public.il (il_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
```

```
    CONSTRAINT kisi_iletisim_fkey FOREIGN KEY (kisi_no)
        REFERENCES public.kisi (kisi_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.iletisim_bilgileri
    OWNER to postgres;

Table: public.kisi

DROP TABLE IF EXISTS public.kisi;

CREATE TABLE IF NOT EXISTS public.kisi
(
    kisi_id smallint NOT NULL,
    kisi_ad character varying(20) COLLATE pg_catalog."default" NOT NULL,
    kisi_soyad character varying(20) COLLATE pg_catalog."default" NOT NULL,
    kisi_tur character varying(20) COLLATE pg_catalog."default",
    kisi_cinsiyet smallint NOT NULL,
    CONSTRAINT kisi_pkey PRIMARY KEY (kisi_id),
    CONSTRAINT "cinsiyet-kisi" FOREIGN KEY (kisi_cinsiyet)
        REFERENCES public.cinsiyet (cinsiyet_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.kisi
    OWNER to postgres;

Table: public.oda

DROP TABLE IF EXISTS public.oda;

CREATE TABLE IF NOT EXISTS public.oda
(
    oda_id smallint NOT NULL,
    yurt_id smallint NOT NULL,
    kat_no smallint,
    kapasite smallint,
    CONSTRAINT oda_pkey PRIMARY KEY (oda_id),
    CONSTRAINT yurt_oda_fkey FOREIGN KEY (yurt_id)
        REFERENCES public.yurt (yurt_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.oda
    OWNER to postgres;

Table: public.ogrenci
```

```
CREATE TABLE “Kisi”.”Ogrenci”
(
    Inherited from table public.kisi: kisi_id smallint NOT NULL,
    Inherited from table public.kisi: kisi_ad character varying(20) COLLATE pg_catalog."default" NOT NULL,
    Inherited from table public.kisi: kisi_soyad character varying(20) COLLATE pg_catalog."default" NOT NULL,
    Inherited from table public.kisi: kisi_tur character varying(20) COLLATE pg_catalog."default",
    Inherited from table public.kisi: kisi_cinsiyet smallint NOT NULL,
    bolum_id smallint NOT NULL,
    universite_id smallint NOT NULL,
    CONSTRAINT ogrenci_pkey PRIMARY KEY (kisi_id),
    CONSTRAINT bolum_ogrenci_fkey FOREIGN KEY (bolum_id)
        REFERENCES public.bolum (bolum_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT universite_ogrenci_fkey FOREIGN KEY (universite_id)
        REFERENCES public.universite (universite_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)


TABLESPACE pg_default;

CREATE TABLE “Kisi”.”Personel”
(
    Inherited from table public.kisi: kisi_id smallint NOT NULL,
    Inherited from table public.kisi: kisi_ad character varying(20) COLLATE pg_catalog."default" NOT NULL,
    Inherited from table public.kisi: kisi_soyad character varying(20) COLLATE pg_catalog."default" NOT NULL,
    Inherited from table public.kisi: kisi_tur character varying(20) COLLATE pg_catalog."default",
    Inherited from table public.kisi: kisi_cinsiyet smallint NOT NULL,
    CONSTRAINT personel_pkey PRIMARY KEY (kisi_id)
)

TABLESPACE pg_default;


Table: public.universite

DROP TABLE IF EXISTS public.universite;

CREATE TABLE IF NOT EXISTS public.universite
(
    universite_id smallint NOT NULL,
    universite_ad character varying COLLATE pg_catalog."default",
    CONSTRAINT universite_pkey PRIMARY KEY (universite_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.universite
    OWNER to postgres;


Table: public.yatak

DROP TABLE IF EXISTS public.yatak;
```

```
CREATE TABLE IF NOT EXISTS public.yatak
(
    yatak_id smallint NOT NULL,
    oda_id smallint NOT NULL,
    yatak_durumu smallint NOT NULL,
    CONSTRAINT yatak_pkey PRIMARY KEY (yatak_id),
    CONSTRAINT oda_yatak_fkey FOREIGN KEY (oda_id)
        REFERENCES public.oda (oda_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT "yatakDurumu_yatak_fkey" FOREIGN KEY (yatak_durumu)
        REFERENCES public.yatak_durumu (durum_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.yatak
    OWNER to postgres;

Table: public.yatak_durumu

DROP TABLE IF EXISTS public.yatak_durumu;

CREATE TABLE IF NOT EXISTS public.yatak_durumu
(
    durum_id smallint NOT NULL,
    durum_ad character varying(5) COLLATE pg_catalog."default",
    CONSTRAINT yatak_durumu_pkey PRIMARY KEY (durum_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.yatak_durumu
    OWNER to postgres;

Table: public.yurt

DROP TABLE IF EXISTS public.yurt;

CREATE TABLE IF NOT EXISTS public.yurt
(
    yurt_id smallint NOT NULL,
    yurt_iletisim smallint NOT NULL,
    yurt_kapasite smallint,
    CONSTRAINT yurt_pkey PRIMARY KEY (yurt_id),
    CONSTRAINT iletisim_yurt_fkey FOREIGN KEY (yurt_iletisim)
        REFERENCES public.iletisim_bilgileri (iletisim_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.yurt
```

```
    OWNER to postgres;

Table: public.yurt_kayit
DROP TABLE IF EXISTS public.yurt_kayit;

CREATE TABLE IF NOT EXISTS public.yurt_kayit
(
    kayit_id smallint NOT NULL,
    yurt_id smallint,
    ogrenci_id smallint NOT NULL,
    "kayıt_tarih" smallint,
    oda_no smallint NOT NULL,
    CONSTRAINT yurt_kayit_pkey PRIMARY KEY (kayit_id),
    CONSTRAINT "oda_yurtKayit_fkey" FOREIGN KEY (oda_no)
        REFERENCES public.oda (oda_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT "ogrenci_yurtKayit_fkey" FOREIGN KEY (ogrenci_id)
        REFERENCES public.ogrenci (kisi_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT "yurt_yurtKayit_fkey" FOREIGN KEY (yurt_id)
        REFERENCES public.yurt (yurt_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.yurt_kayit
    OWNER to postgres;

insert into il (il_isim) VALUES('Adana')
insert into il (il_isim) VALUES('Adıyaman')
insert into il (il_isim) VALUES('Afyon')
insert into il (il_isim) VALUES('Ağrı')
insert into il (il_isim) VALUES('Amasya')
insert into il (il_isim) VALUES('Ankara')
insert into il (il_isim) VALUES('Antalya')
insert into il (il_isim) VALUES('Artvin')
insert into il (il_isim) VALUES('Aydın')
insert into il (il_isim) VALUES('Balıkesir')
insert into il (il_isim) VALUES('Bilecik')
insert into il (il_isim) VALUES('Bingöl')
insert into il (il_isim) VALUES('Bitlis')
insert into il (il_isim) VALUES('Bolu')
insert into il (il_isim) VALUES('Burdur')
insert into il (il_isim) VALUES('Bursa')
insert into il (il_isim) VALUES('Çanakkale')
insert into il (il_isim) VALUES('Çankırı')
insert into il (il_isim) VALUES('Çorum')
insert into il (il_isim) VALUES('Denizli')
insert into il (il_isim) VALUES('Diyarbakır')
insert into il (il_isim) VALUES('Edirne')
insert into il (il_isim) VALUES('Elazığ')
insert into il (il_isim) VALUES('Erzincan')
insert into il (il_isim) VALUES('Erzurum')
```

```
insert into il (il_isim) VALUES('Eskişehir')
insert into il (il_isim) VALUES('Gaziantep')
insert into il (il_isim) VALUES('Giresun')
insert into il (il_isim) VALUES('Gümüşhane')
insert into il (il_isim) VALUES('Hakkari')
insert into il (il_isim) VALUES('Hatay')
insert into il (il_isim) VALUES('Isparta')
insert into il (il_isim) VALUES('Mersin')
insert into il (il_isim) VALUES('İstanbul')
insert into il (il_isim) VALUES('İzmir')
insert into il (il_isim) VALUES('Kars')
insert into il (il_isim) VALUES('Kastamonu')
insert into il (il_isim) VALUES('Kayseri')
insert into il (il_isim) VALUES('Kırklareli')
insert into il (il_isim) VALUES('Kırşehir')
insert into il (il_isim) VALUES('Kocaeli')
insert into il (il_isim) VALUES('Konya')
insert into il (il_isim) VALUES('Kütahya')
insert into il (il_isim) VALUES('Malatya')
insert into il (il_isim) VALUES('Manisa')
insert into il (il_isim) VALUES('Kahramanmaraş')
insert into il (il_isim) VALUES('Mardin')
insert into il (il_isim) VALUES('Muğla')
insert into il (il_isim) VALUES('Muş')
insert into il (il_isim) VALUES('Nevşehir')
insert into il (il_isim) VALUES('Niğde')
insert into il (il_isim) VALUES('Ordu')
insert into il (il_isim) VALUES('Rize')
insert into il (il_isim) VALUES('Sakarya')
insert into il (il_isim) VALUES('Samsun')
insert into il (il_isim)VALUES('Siirt')
insert into il (il_isim) VALUES('Sinop')
insert into il (il_isim) VALUES('Sivas')
insert into il (il_isim)VALUES('Tekirdağ')
insert into il (il_isim) VALUES('Tokat')
insert into il (il_isim) VALUES('Trabzon')
insert into il (il_isim) VALUES('Tunceli')
insert into il (il_isim) VALUES('Şanlıurfa')
insert into il (il_isim) VALUES('Uşak')
insert into il (il_isim) VALUES('Van')
insert into il (il_isim) VALUES('Yozgat')
insert into il (il_isim) VALUES('Zonguldak')
insert into il (il_isim) VALUES('Aksaray')
insert into il (il_isim) VALUES('Bayburt')
insert into il (il_isim) VALUES('Karaman')
insert into il (il_isim) VALUES('Kırıkkale')
insert into il (il_isim) VALUES('Batman')
insert into il (il_isim) VALUES('Şırnak')
insert into il (il_isim) VALUES('Bartın')
insert into il (il_isim) VALUES('Ardahan')
insert into il (il_isim) VALUES('Iğdır')
insert into il (il_isim) VALUES('Yalova')
insert into il (il_isim) VALUES('Karabük')
insert into il (il_isim) VALUES('Kilis')
insert into il (il_isim) VALUES('Osmaniye')
insert into il (il_isim) VALUES('Düzce')
```

```
insert into bolum (bolum_id,bolum_isim) VALUES(1,'Bilgisayar Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(2,'Endüstri Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(3,'Elektrik Elektronik Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(4,'Inşaat Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(5,'Gıda Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(6,'Makine Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(7,'Metalurji Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(8,'Kimya Mühendisliği')
insert into bolum (bolum_id,bolum_isim) VALUES(9,'Hukuk')
insert into bolum (bolum_id,bolum_isim) VALUES(10,'Psikoloji')
insert into bolum (bolum_id,bolum_isim) VALUES(11,'Işletme')
insert into bolum (bolum_id,bolum_isim) VALUES(12,'Sosyoloji')
insert into bolum (bolum_id,bolum_isim) VALUES(13,'Ekonomi')
insert into bolum (bolum_id,bolum_isim) VALUES(14,'Tarih')
insert into bolum (bolum_id,bolum_isim) VALUES(15,'Türk Dili ve Edebiyatı')
insert into bolum (bolum_id,bolum_isim) VALUES(16,'Ingiliz Dili ve Edebiyatı')
insert into bolum (bolum_id,bolum_isim) VALUES(17,'Uluslararası Ilişkiler')
insert into bolum (bolum_id,bolum_isim) VALUES(18,'Tıp')
insert into bolum (bolum_id,bolum_isim) VALUES(19,'Diş Hekimliği')
insert into bolum (bolum_id,bolum_isim) VALUES(20,'Hemşirelik')
insert into bolum (bolum_id,bolum_isim) VALUES(21,'Fizyoterapi ve Rehabilitasyon')
insert into bolum (bolum_id,bolum_isim) VALUES(22,'Fizik')
insert into bolum (bolum_id,bolum_isim) VALUES(23,'Kimya')
insert into bolum (bolum_id,bolum_isim) VALUES(24,'Biyoloji')
insert into bolum (bolum_id,bolum_isim) VALUES(25,'Matematik')
insert into bolum (bolum_id,bolum_isim) VALUES(26,'Gazetecilik')
insert into bolum (bolum_id,bolum_isim) VALUES(27,'Radyo-Televizyon')
insert into bolum (bolum_id,bolum_isim) VALUES(28,'Medya ve Iletişim')

insert into cinsiyet (cinsiyet_id,cinsiyet_ad) values (1,'Erkek')
insert into cinsiyet (cinsiyet_id,cinsiyet_ad) values (2,'Kadın')

insert into universite (universite_id,universite_ad) values (1,'Sakarya Üniversitesi')
insert into universite (universite_id,universite_ad) values (2,'Sakarya Uygulamalı Bilimler Üniversitesi')

insert into yatak_durumu (durum_id,durum_ad) values (1,'Boş')
insert into yatak_durumu (durum_id,durum_ad) values (2,'Dolu')

insert into il (il_isim) values ('Adana')


Create or replace function ogrenciArti()
   returns TRIGGER
   AS
   $$
 begin
 update yurt set mevcut_ogrenci = mevcut_ogrenci+1;
 return new;
 end;
 $$
 LANGUAGE plpgsql;

create trigger artiOgrenci
after INSERT
on ogrenci
for EACH ROW
execute PROCEDURE ogrenciArti();
```

```
Create or replace function ogrenciCikti()
   returns TRIGGER
   AS
   $$
 begin
 update yurt set mevcut_ogrenci = mevcut_ogrenci-1;
 return new;
 end;
 $$
 LANGUAGE plpgsql;

create trigger ciktiOgrenci
after DELETE
on ogrenci
for EACH ROW
execute PROCEDURE ogrenciCikti();



Create VIEW ogrenciListe
AS
Select ogrenci.kisi_id AS "TC",ogrenci.kisi_ad AS "AD",ogrenci.kisi_soyad AS "SOYAD",cinsiyet.cinsiyet_ad AS
"CINSIYET",universite.universite_ad AS "UNIVERSITE",bolum.bolum_isim AS "BOLUM",oda.oda_id as
"ODA",il.il_isim AS "MEMLEKET"
from ogrenci
inner JOIN cinsiyet
on
ogrenci.kisi_cinsiyet = cinsiyet.cinsiyet_id
inner JOIN universite
ON
ogrenci.universite_id = universite.universite_id
inner JOIN bolum
on
ogrenci.bolum_id = bolum.bolum_id
inner join oda
on
ogrenci.oda_id = oda.oda_id
inner join il
on
ogrenci.il_id = il.il_id



Create procedure ogrenci_guncelle (p1 text,p2 text,p3 BIGINT,p4 INT,p5 INT,p6 smallint,p7 smallint,p8
smallint)
language SQL
as
$$
update ogrenci set kisi_ad = p1 , kisi_soyad = p2 , bolum_id = p5 , universite_id = p4,il_id = p6 ,oda_id = p7
where kisi_id = p3;
call oda_guncelle(p7,p8);
$$

Create procedure ogrencii_sil (p1 BIGINT,p2 smallint)
language SQL
```

```
as
$$
delete from ogrenci where kisi_id=p1;
call oda_kayitsil(p2)
$$

CREATE  VIEW personelListe
AS
Select personel.kisi_id AS "TC",personel.kisi_ad AS "AD",personel.kisi_soyad AS "SOYAD",cinsiyet.cinsiyet_ad
AS "CINSIYET",personel_tur.tur_ad AS "MESLEK",il.il_isim AS "MEMLEKET"
from personel
inner join cinsiyet
on
personel.kisi_cinsiyet = cinsiyet.cinsiyet_id
inner join personel_tur
ON
personel.personel_tur = personel_tur.tur_id
inner JOIN il
ON
personel.il_id = il.il_id


Create procedure personel_ekle (p1 BIGINT,p2 text,p3 text,p4 int,p5 int,p6 INT)
language SQL
as
$$
insert into personel (kisi_id,kisi_ad,kisi_soyad,kisi_cinsiyet,personel_tur,il_id) values(p1,p2,p3,p4,p5,p6);
$$


Create procedure personel_sil (p1 BIGINT)
language SQL
as
$$
delete from personel where kisi_id=p1;
$$


Create procedure personel_guncelle (p1 BIGINT,p2 text,p3 text,p4 INT,p5 INT)
language SQL
as
$$
update personel set kisi_ad = p2 , kisi_soyad = p3 , personel_tur = p4 , il_id = p5 where kisi_id = p1;
$$


Create or replace function personell_cikar()
   returns TRIGGER
   AS
   $$
 begin
 update yurt set mevcut_personel = mevcut_personel-1;
 return new;
 end;
 $$
 LANGUAGE plpgsql;
```

```
create trigger personellcikart
after DELETE
on personel
for EACH ROW
execute PROCEDURE personell_cikar();


Create or replace function personell_arttir()
   returns TRIGGER
   AS
   $$
 begin
 update yurt set mevcut_personel = mevcut_personel+1;
 return new;
 end;
 $$
 LANGUAGE plpgsql;


create trigger personellarttir
after INSERT
on personel
for EACH ROW
execute PROCEDURE personell_arttir();


CREATE  VIEW yurtListe
AS
Select yurt.mevcut_ogrenci AS "MEVCUT OGRENCI",yurt.mevcut_personel AS "MEVCUT PERSONEL"
from yurt

Create VIEW faturaListe
AS
select ogrenci.kisi_ad as "AD" , ogrenci.kisi_soyad AS "SOYAD",yatak.yatak_fiyat AS "UCRET"
from fatura
inner join ogrenci
on
fatura.ogrenci_id = ogrenci.kisi_id
inner join yatak
on
fatura.oda_id = yatak.yatak_id

Create procedure fatura_ekle (p1 BIGINT,p2 SMALLINT)
language SQL
as
$$
insert into fatura (ogrenci_id,oda_id) values(p1,p2);
$$


Create procedure ogrenci_ekle (p1 text,p2 text,p3 bigINT,p4 int,p5 int,p6 INT,p7 SMALLINT,p8 SMALLINT,p9
SMALLINT)
language SQL
as
$$
insert into ogrenci (kisi_ad,kisi_soyad,kisi_id,kisi_cinsiyet,universite_id,bolum_id,il_id,oda_id)
values(p1,p2,p3,p4,p5,p6,p7,p8);
```

```
call fatura_ekle(p3,p9);
call oda_guncelle(p8,p9)
$$

Create procedure oda_guncelle (p1 smallint,p2 smallint)
language SQL
as
$$
update oda set yurt_id = 1 , yatak_id = p2 , kapasite = kapasite +1  where oda_id = p1;
$$

Create procedure oda_kayitsil (p1 smallint)
language SQL
as
$$
update oda set yurt_id = 1 , kapasite = kapasite - 1  where oda_id = p1;
$$

CREATE  view odaListe
AS
SELECT oda.oda_id AS "ODA NUMARASI",oda.yatak_id AS "ODA KAPASITESI",oda.kapasite AS "ODA MEVCUDU"
from oda
```

Form Kodları:

Form1.cs:

```
public Form1()
    {
       InitializeComponent();
    }

    private void pictureBox1_Click(object sender, EventArgs e)
    {

    }

    private void panel1_Paint(object sender, PaintEventArgs e)
    {

    }
    NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port =
5432;Database=YurtSistem;user ID=postgres;password=aa");
    private void button1_Click(object sender, EventArgs e)
    {
       Form2 f2 = new Form2();
       f2.ShowDialog();
    }

    private void button2_Click(object sender, EventArgs e)
```

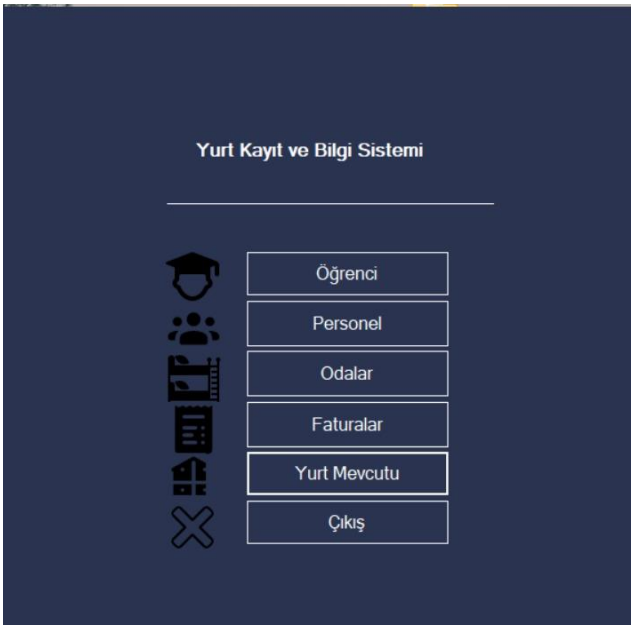
```
    {
        Form3 f3 = new Form3();
        f3.ShowDialog();
    }

    private void button6_Click(object sender, EventArgs e)
    {
        Form4 f4 = new Form4();
        f4.ShowDialog();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        Form5 f5 = new Form5();
        f5.ShowDialog();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Form6 f6 = new Form6();
        f6.ShowDialog();
    }
```

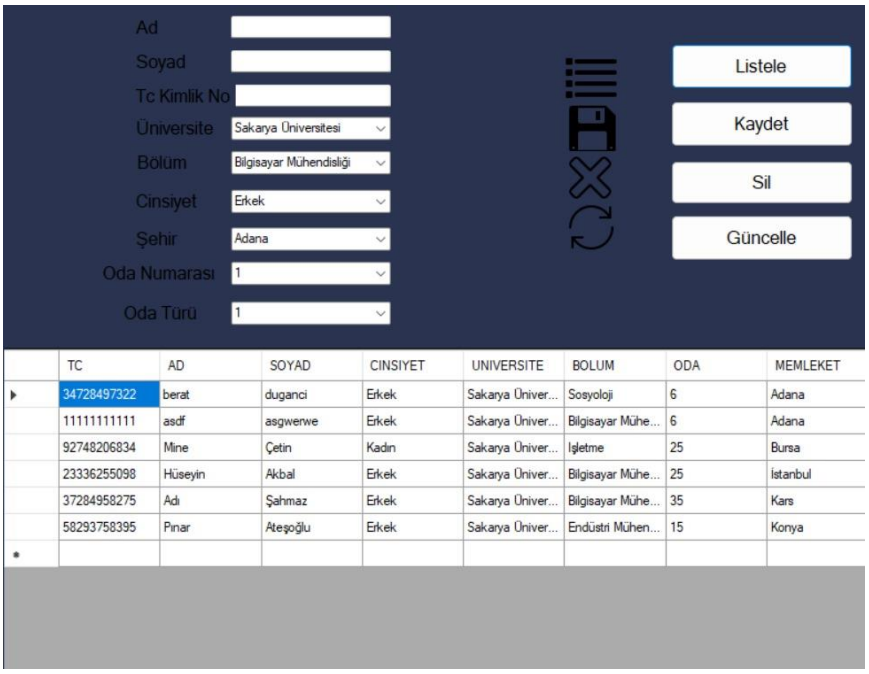
Form2.cs:

```
public Form2()
    {
        InitializeComponent();
    }
    NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port =
5432;Database=DbYurtKayitSistem;user ID=postgres;password=aa");
    private void button3_Click(object sender, EventArgs e)
    {
        baglanti.Open();
        NpgsqlCommand komut3 = new NpgsqlCommand("call ogrencii_sil(@p1,@p2)", baglanti);
        komut3.Parameters.AddWithValue("@p1", long.Parse(textBox3.Text));
        komut3.Parameters.AddWithValue("@p2", comboBox5.SelectedValue);
        komut3.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kayıt başarıyla silindi!","Bilgi",MessageBoxButtons.OK,MessageBoxIcon.Stop);
    }

    private void button1_Click(object sender, EventArgs e)//listele
    {
        string sorgu = "select * from ogrenciListe";
        NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
        DataSet ds = new DataSet();
        da.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];

    }
    //call ogrenci_ekle(kisi_ad, kisi_soyad, kisi_cinsiyet, universite_id, bolum_id) values(@p1, @p2, @p3,
@p4, @p5)
    private void button2_Click(object sender, EventArgs e)//kaydet
    {
        baglanti.Open();
        NpgsqlCommand komut2 = new NpgsqlCommand("call ogrenci_ekle
(@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9)", baglanti);
```

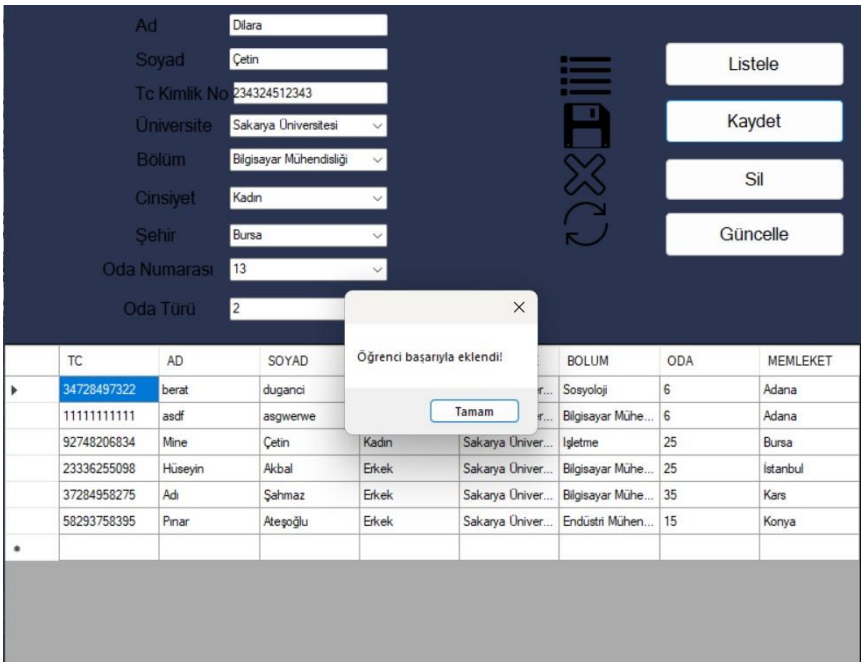
```
    komut2.Parameters.AddWithValue("@p1", textBox1.Text);
    komut2.Parameters.AddWithValue("@p2", textBox2.Text);
    komut2.Parameters.AddWithValue("@p3", long.Parse(textBox3.Text));
    komut2.Parameters.AddWithValue("@p4", comboBox1.SelectedValue);
    komut2.Parameters.AddWithValue("@p5", comboBox2.SelectedValue);
    komut2.Parameters.AddWithValue("@p6", comboBox3.SelectedValue);
    komut2.Parameters.AddWithValue("@p7", comboBox6.SelectedValue);
    komut2.Parameters.AddWithValue("@p8", comboBox4.SelectedValue);
    komut2.Parameters.AddWithValue("@p9", comboBox5.SelectedValue);
    komut2.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("Öğrenci başarıyla eklendi!");
}

private void Form2_Load(object sender, EventArgs e)
{
    baglanti.Open();
    NpgsqlDataAdapter da = new NpgsqlDataAdapter("Select * from bolum", baglanti);
    NpgsqlDataAdapter da1 = new NpgsqlDataAdapter("Select * from cinsiyet", baglanti);
    NpgsqlDataAdapter da2 = new NpgsqlDataAdapter("Select * from universite", baglanti);
    NpgsqlDataAdapter da3 = new NpgsqlDataAdapter("Select * from il", baglanti);
    NpgsqlDataAdapter da4 = new NpgsqlDataAdapter("Select * from oda", baglanti);
    NpgsqlDataAdapter da5 = new NpgsqlDataAdapter("Select * from yatak", baglanti);
    DataTable dt = new DataTable();
    DataTable dt1 = new DataTable();
    DataTable dt2= new DataTable();
    DataTable dt3 = new DataTable();
    DataTable dt4 = new DataTable();
    DataTable dt5 = new DataTable();
    da.Fill(dt);
    da1.Fill(dt1);
    da2.Fill(dt2);
    da3.Fill(dt3);
    da4.Fill(dt4);
    da5.Fill(dt5);
    comboBox3.DisplayMember = "bolum_isim";
    comboBox3.ValueMember = "bolum_id";
    comboBox3.DataSource = dt;
    comboBox1.DisplayMember = "cinsiyet_ad";
    comboBox1.ValueMember = "cinsiyet_id";
    comboBox1.DataSource = dt1;
    comboBox2.DisplayMember = "universite_ad";
    comboBox2.ValueMember = "universite_id";
    comboBox2.DataSource = dt2;
    comboBox4.DisplayMember = "oda_id";
    comboBox4.ValueMember = "oda_id";
    comboBox4.DataSource = dt4;
    comboBox5.DisplayMember = "yatak_id";
    comboBox5.ValueMember = "yatak_id";
    comboBox5.DataSource = dt5;
    comboBox6.DisplayMember = "il_isim";
    comboBox6.ValueMember = "il_id";
    comboBox6.DataSource = dt3;
    baglanti.Close();
}

private void button4_Click(object sender, EventArgs e)
```

```
    {
        baglanti.Open();
        NpgsqlCommand komut2 = new NpgsqlCommand("call ogrenci_guncelle
(@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8)", baglanti);
        komut2.Parameters.AddWithValue("@p3", long.Parse(textBox3.Text));
        komut2.Parameters.AddWithValue("@p1", textBox1.Text);
        komut2.Parameters.AddWithValue("@p2", textBox2.Text);
        komut2.Parameters.AddWithValue("@p4", comboBox2.SelectedValue);
        komut2.Parameters.AddWithValue("@p5", comboBox3.SelectedValue);
        komut2.Parameters.AddWithValue("@p6", comboBox6.SelectedValue);
        komut2.Parameters.AddWithValue("@p7", comboBox4.SelectedValue);
        komut2.Parameters.AddWithValue("@p8", comboBox5.SelectedValue);
        komut2.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Öğrenci başarıyla güncellendi!");
    }
```

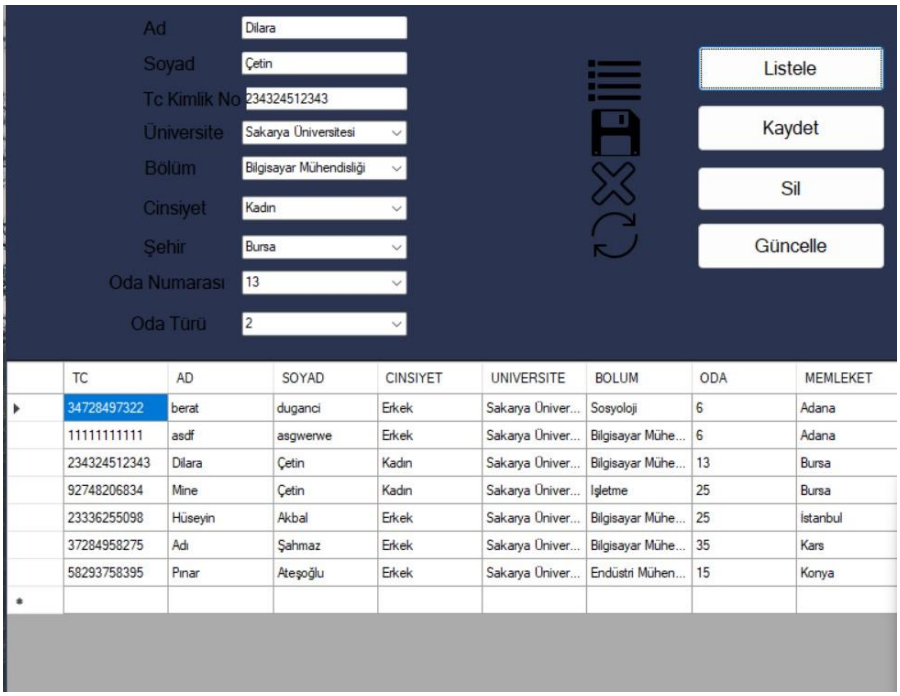




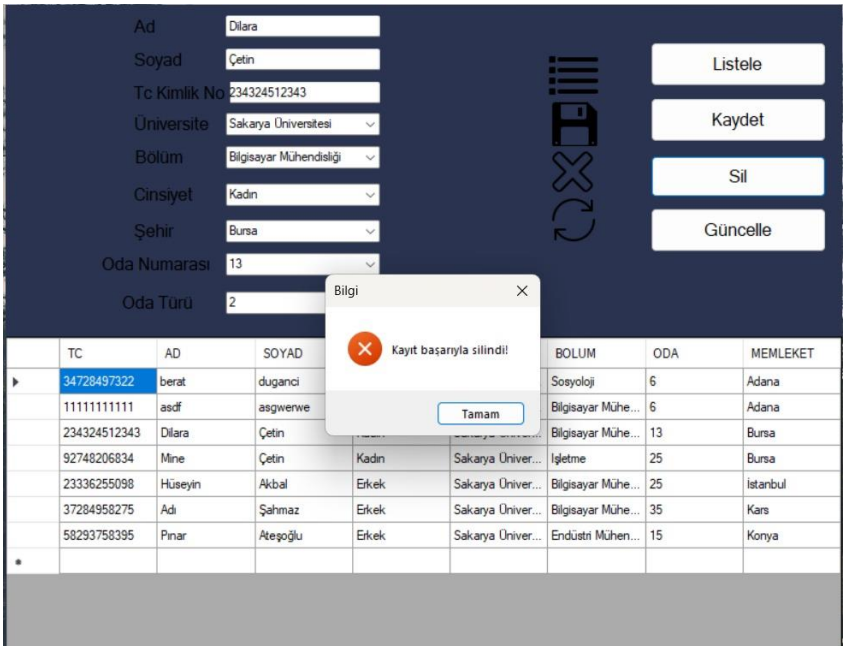
TC	AD	SOYAD	CINSIYET	UNIVERSITE	BOLUM	ODA	MEMLEKET
34728497322	berat	duganci	Erkek	Sakarya Üniver...	Sosyoloji	6	Adana
11111111111	asdf	asgwerwe	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	6	Adana
92748206834	Mine	Çetin	Kadın	Sakarya Üniver...	İşletme	25	Bursa
23336255098	Hüseyin	Akbal	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	25	İstanbul
37284958275	Adı	Şahmaz	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	35	Kars
58293758395	Pınar	Ateşoğlu	Erkek	Sakarya Üniver...	Endüstri Mühen...	15	Konya



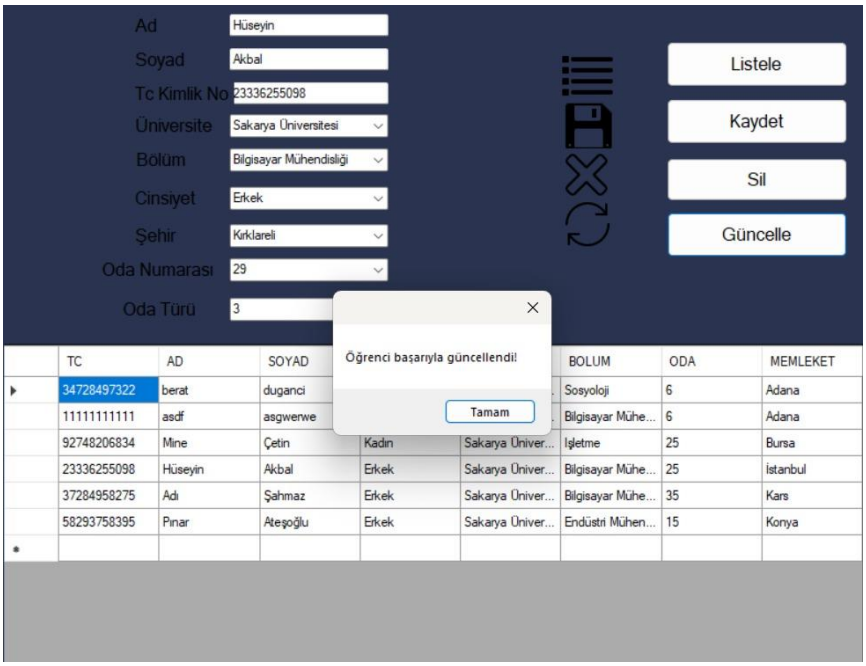
TC	AD	SOYAD			BOLUM	ODA	MEMLEKET
34728497322	berat	duganci			Sosyoloji	6	Adana
11111111111	asdf	asgwerwe			Bilgisayar Mühe...	6	Adana
92748206834	Mine	Çetin	Kadın	Sakarya Üniver...	İşletme	25	Bursa
23336255098	Hüseyin	Akbal	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	25	İstanbul
37284958275	Adı	Şahmaz	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	35	Kars
58293758395	Pınar	Ateşoğlu	Erkek	Sakarya Üniver...	Endüstri Mühen...	15	Konya



TC	AD	SOYAD	CINSIYET	UNIVERSITE	BOLUM	ODA	MEMLEKET
34728497322	berat	duganci	Erkek	Sakarya Üniver...	Sosyoloji	6	Adana
11111111111	asdf	asgwerwe	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	6	Adana
234324512343	Dilara	Çetin	Kadın	Sakarya Üniver...	Bilgisayar Mühe...	13	Bursa
92748206834	Mine	Çetin	Kadın	Sakarya Üniver...	İşletme	25	Bursa
23336255098	Hüseyin	Akbal	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	25	İstanbul
37284958275	Adı	Şahmaz	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	35	Kars
58293758395	Pınar	Ateşoğlu	Erkek	Sakarya Üniver...	Endüstri Mühen...	15	Konya



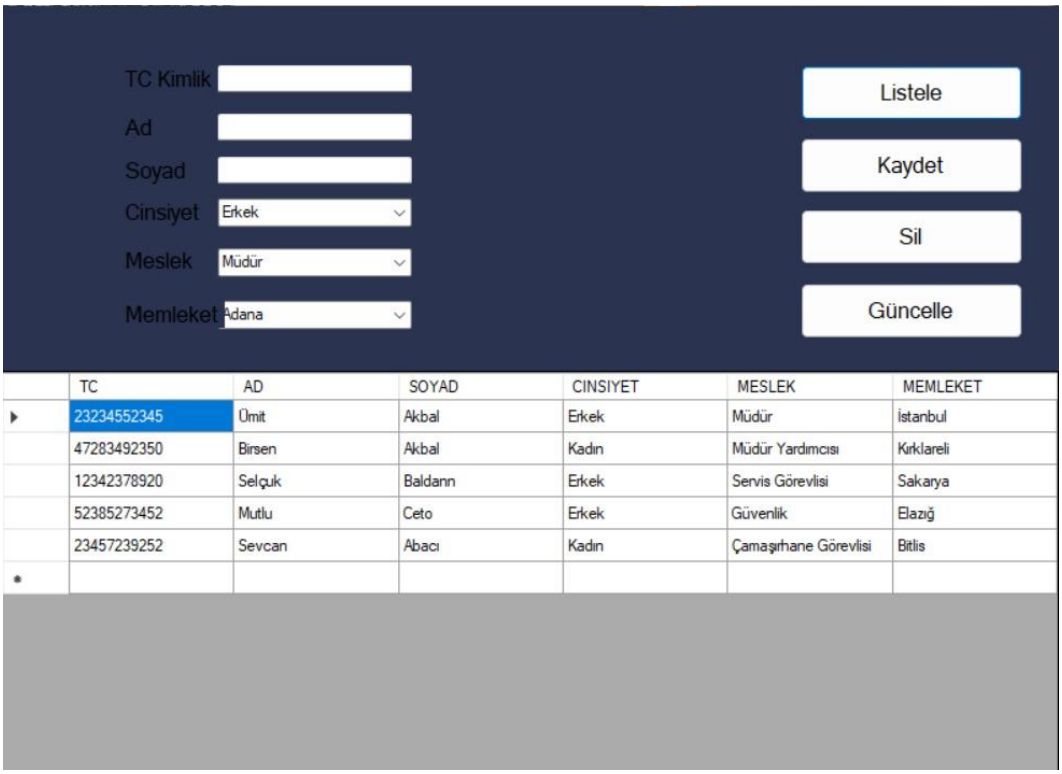
TC	AD	SOYAD			BOLUM	ODA	MEMLEKET
34728497322	berat	duganci			Sosyoloji	6	Adana
11111111111	asdf	asgwerwe			Bilgisayar Mühe...	6	Adana
234324512343	Dilara	Çetin			Bilgisayar Mühe...	13	Bursa
92748206834	Mine	Çetin	Kadın	Sakarya Üniver...	İşletme	25	Bursa
23336255098	Hüseyin	Akbal	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	25	İstanbul
37284958275	Adı	Şahmaz	Erkek	Sakarya Üniver...	Bilgisayar Mühe...	35	Kars
58293758395	Pınar	Ateşoğlu	Erkek	Sakarya Üniver...	Endüstri Mühen...	15	Konya



Form3.cs:

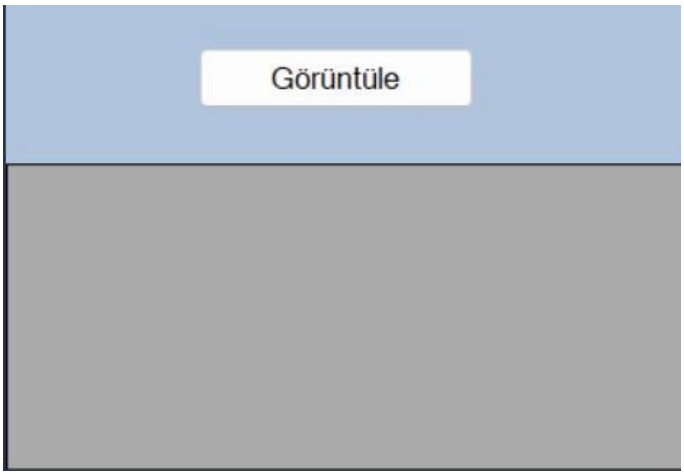
```
public Form3()
    {
      InitializeComponent();
    }
    NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port =
5432;Database=DbYurtKayitSistem;user ID=postgres;password=aa");
    private void button1_Click(object sender, EventArgs e)
    {
      string sorgu = "select * from personelListe";
      NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
      DataSet ds = new DataSet();
      da.Fill(ds);
      dataGridView1.DataSource = ds.Tables[0];
    }

    private void Form3_Load(object sender, EventArgs e)
    {
      baglanti.Open();
      NpgsqlDataAdapter da = new NpgsqlDataAdapter("Select * from personel_tur", baglanti);
      NpgsqlDataAdapter da1 = new NpgsqlDataAdapter("Select * from cinsiyet", baglanti);
      NpgsqlDataAdapter da2 = new NpgsqlDataAdapter("Select * from il", baglanti);
      DataTable dt = new DataTable();
      DataTable dt1 = new DataTable();
      DataTable dt2 = new DataTable();
      da.Fill(dt);
      da1.Fill(dt1);
      da2.Fill(dt2);
      comboBox3.DisplayMember = "il_isim";
      comboBox3.ValueMember = "il_id";
      comboBox3.DataSource = dt2;
      comboBox1.DisplayMember = "cinsiyet_ad";
      comboBox1.ValueMember = "cinsiyet_id";
```

```
            comboBox1.DataSource = dt1;
            comboBox2.DisplayMember = "tur_ad";
            comboBox2.ValueMember = "tur_id";
            comboBox2.DataSource = dt;
            baglanti.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            NpgsqlCommand komut2 = new NpgsqlCommand("call personel_ekle (@p1,@p2,@p3,@p4,@p5,@p6)",
baglanti);
            komut2.Parameters.AddWithValue("@p1", long.Parse(textBox1.Text));
            komut2.Parameters.AddWithValue("@p2", textBox2.Text);
            komut2.Parameters.AddWithValue("@p3", textBox3.Text);
            komut2.Parameters.AddWithValue("@p4", comboBox1.SelectedValue);
            komut2.Parameters.AddWithValue("@p5", comboBox2.SelectedValue);
            komut2.Parameters.AddWithValue("@p6", comboBox3.SelectedValue);
            komut2.ExecuteNonQuery();
            baglanti.Close();
            MessageBox.Show("Personel başarıyla eklendi!");
        }

        private void button3_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            NpgsqlCommand komut3 = new NpgsqlCommand("call personel_sil(@p1)", baglanti);
            komut3.Parameters.AddWithValue("@p1", long.Parse(textBox1.Text));
            komut3.ExecuteNonQuery();
            baglanti.Close();
            MessageBox.Show("Kayıt başarıyla silindi!", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Stop);
        }

        private void button4_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            NpgsqlCommand komut2 = new NpgsqlCommand("call personel_guncelle (@p1,@p2,@p3,@p4,@p5)",
baglanti);
            komut2.Parameters.AddWithValue("@p1", long.Parse(textBox1.Text));
            komut2.Parameters.AddWithValue("@p2", textBox2.Text);
            komut2.Parameters.AddWithValue("@p3", textBox3.Text);
            komut2.Parameters.AddWithValue("@p4", comboBox2.SelectedValue);
            komut2.Parameters.AddWithValue("@p5", comboBox3.SelectedValue);
            komut2.ExecuteNonQuery();
            baglanti.Close();
            MessageBox.Show("Personel başarıyla güncellendi!");
        }
```



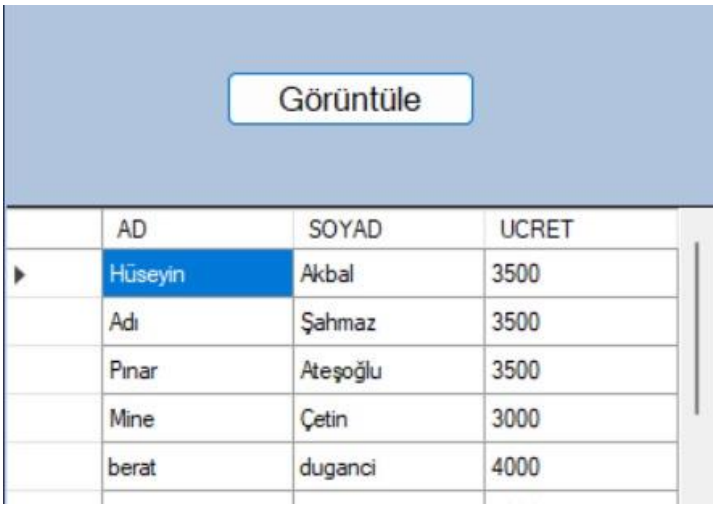
Form4.cs:

```
public Form4()
    {
      InitializeComponent();
    }

    private void Form4_Load(object sender, EventArgs e)
    {

    }
    NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port =
5432;Database=DbYurtKayitSistem;user ID=postgres;password=aa");
    private void button1_Click(object sender, EventArgs e)
    {
      string sorgu = "select * from yurtListe";
      NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
      DataSet ds = new DataSet();
      da.Fill(ds);
      dataGridView1.DataSource = ds.Tables[0];
    }
```



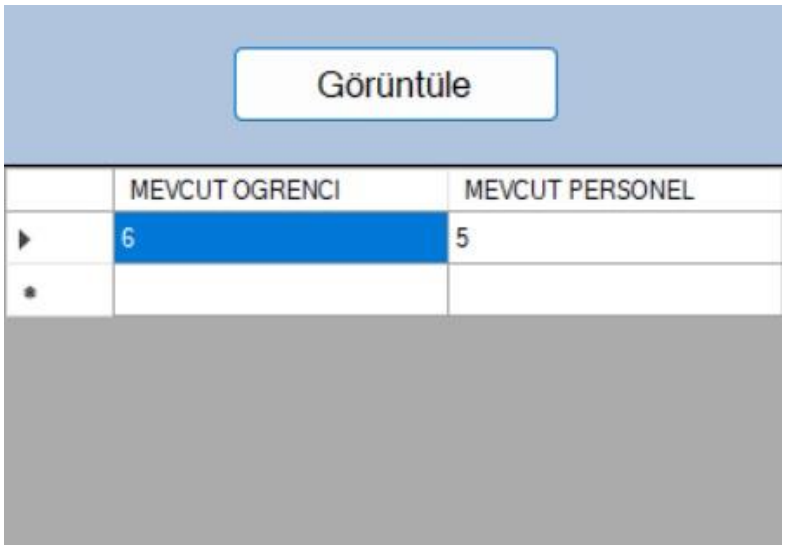
Form5.cs:

```
public Form5()
    {
       InitializeComponent();
    }
    NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port =
5432;Database=DbYurtKayitSistem;user ID=postgres;password=aa");
    private void button1_Click(object sender, EventArgs e)
    {

    }

    private void button2_Click(object sender, EventArgs e)
    {
       string sorgu = "select * from faturaListe";
       NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
       DataSet ds = new DataSet();
       da.Fill(ds);
       dataGridView1.DataSource = ds.Tables[0];
    }
```



Form6.cs:

```
public Form6()
    {
       InitializeComponent();
    }
    NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port =
5432;Database=DbYurtKayitSistem;user ID=postgres;password=aa");
    private void button1_Click(object sender, EventArgs e)
    {
       string sorgu = "select * from odaListe";
       NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
       DataSet ds = new DataSet();
       da.Fill(ds);
       dataGridView1.DataSource = ds.Tables[0];
    }
```



İşlev/Saklı Yordam Fonksiyonları

- ogrenci_ekle: Öğrencinin bilgilerini parametre olarak girmemizi isteyerek bu bilgileri ogrenci tablosuna ekleme yapıyor ve öğrencinin faturasını fatura_ekle fonksiyonuyla oluşturuyor. Aynı zamanda oda_ekle fonksiyonuyla kalacağı odanın öğrenci sayısı arttırılıyor.
- ogrenci_guncelle: Girilen tc bilgisiyle eşleşen ogrencinin bilgilerini parametredekilere göre değiştiriyor:
- personel_ekle: Girilen parametredeki bilgilere göre personel ekliyor.
- personel_sil: Parametredeki tc ile eşleşen personeli siliyor.
- personel_guncelle: Tc ile eşleşen kişinin bilgilerini değiştiriyor.
- oda_guncelle: verilen parametredeki oda numarasıyla eşleşen odanın ogrenci sayısını arttırıyor.
- oda_kayitsil: Fonksiyonu verilen parametredeki oda numarasıyla eşleşen odanın ogrenci sayısını azaltıyor.
- fatura_ekle: Fonksiyonu ogrencinin kaç kişilik odada kaldığına göre fiyatını görüntülemek için fatura ekliyor.

Triggerlar

- ogrenciarti: Fonksiyonu trigger için oluşturduğumuz bir prosedür yurdun mevcut_ogrenci sayısını arttırıyor.
- ogrencicikti: Fonksiyonu trigger için oluşturduğumuz bir prosedür yurdun mevcut_ogrenci sayısını azaltıyor.
- personel_arttir: Fonksiyonu trigger için oluşturduğumuz bir prosedür yurdun mevcut_personel sayısını arttırıyor.
- personel_cikar: Fonksiyonu trigger için oluşturduğumuz bir prosedür yurdun mevcut_personel sayısını azaltıyor.
- Artıogrenci:Bu trigger ogrenci eklendiğinde yurt tablosu içerisinde tutuğumuz mevcut_ogrenci sayısını 1 arttırıyor.
- ogrenciCikti:Bu trigger ogrenci silindiğinde yurt tablosu içerisinde tuttuğumuz mevcut_ogrenci sayısını 1 azaltıyor.
- personelCikart:Bu trigger personel çıkartıldığında yurt tablosu içerisinde tuttuğumuz mevcut_personel sayısını 1 azaltıyor.
- Personell_arttir:Bu trigger personel eklendiğinde yurt tablosu içerisinde tuttuğuuz mevcut_personel sayısını 1 arttırıyor.13