

VERİTABANI YÖNETİM SİSTEMLERİ PROJE ÇALIŞMASI

Hazırlayan:

G181210022

Ecem Amanvermez

ecem.amanvermez@ogr.sakarya.edu.tr

a) Uygulamanın Kısa Tanıtımı

Uygulama bir okul/kurs sistemi bilgi takip uygulamasıdır. Bir okul veya kursun yani bir eğitim kurumunun sahip olduğu birçok özelliği ve varlığı içermekte bunları saklayıp, listeleyip, değiştirip, silbilmektedir.

Öğrenciler, Öğretmenler, Sınavlar, Sınıflar, Şubeler, Dersler, Devamsızlık, Yemekhane, Okul'un sahip olduğu Demirbaşlar gibi birçok varlık bulunmaktadır.

PostgreSql veritabanı yönetim sistemi ve Python programlama dili kullanılarak geliştirilmiştir.

a) İş Kuralları

Bir kullanıcı bir kullanıcı adına sahiptir.

Bir kullanıcı bir şifreye sahiptir.

Bir kullanıcıya ait bir öğrenciler tablosu tanımlanmıştır.

Bir öğrencinin bir id'si vardır.

Bir kullanıcıya ait bir binalar tablosu tanımlanmıştır.

Bir binanın bir id'si vardır.

Bir kullanıcıya ait bir demirbaşlar tablosu tanımlanmıştır.

Bir demirbaşın bir id'si vardır.

Bir kullanıcıya ait bir dersler tablosu tanımlanmıştır.

Bir dersin bir id'si vardır.

Bir kullanıcıya ait bir devamsızlık tablosu tanımlanmıştır.

Bir devamsızlığın bir id'si vardır.

Bir kullanıcı için bir harcamalar tablosu tanımlanmıştır.

Bir harcamanın bir id'si vardır.

Bir kullanıcı için bir öğretmen tablosu vardır.

Bir öğretmenin bir id'si vardır.

Bir kullanıcı için bir sınavlar tablosu vardır.

Bir sınavın bir id'si vardır.

Bir kullanıcı için bir sınıflar tablosu vardır.

Bir sınıfın bir id'si vardır.

Bir kullanıcı için bir şubeler tablosu vardır.

Bir şubenin bir id'si vardır.

Bir kullanıcı için bir yemekhane menü tablosu vardır.

Her menünün bir id'si vardır.

a) İlişkisel Şema

ogrenciler(ogrenciid:int,ogrenciadi:varchar(255),ogrencinumarasi:varchar(255),ogrencisinif:varchar(255),ogrencisoyadi:varchar(255),ogrencisube:varchar(255),kullaniciid:int)

ogretmenler(ogretmenid:int,ogretmenadi:varchar(255),ogretmensoyadi:varchar(255),ogretmenbrans:varchar(255),kullaniciid:int)

kullanici(kullaniciid:int,kadi:varchar(255),sifre:varchar(255))

binalar(binaid:int,binaadi:varchar(255),binakonum:varchar(255),kullaniciid:int)

demirbaslar(demirbasid:int,demirbasadi:varchar(255),kullaniciid:int)

dersler(dersid:int,dersadi:varchar(255),dersbrans:varchar(255),kullaniciid:int)

devamsizlik(devamsizlikid:int,devamsizlikgunu:varchar(10),ogrencino:varchar(255),kullaniciid:int)

harcamalar(harcamaid:int,harcamadi:varchar(255),harcamaneden:varchar(255),kullaniciid:int)

sinavlar(sinavid:int,sinavadi:varchar(255),sinavagirensayisi:varchar(255),sinavsuresi:varchar(255),kullaniciid:int)

siniflar(sinifid:int,sinifsube:varchar(10),sinifadi:varchar(255),kullaniciid:int)

subeler(subeid:int,subeadı:varchar(255),subemevcut:varchar(10),kullaniciid:int)

yemekhanemenu(yemekid:int,yemekadi:varchar(255),yemekfiyat:varchar(10),kullaniciid:int)

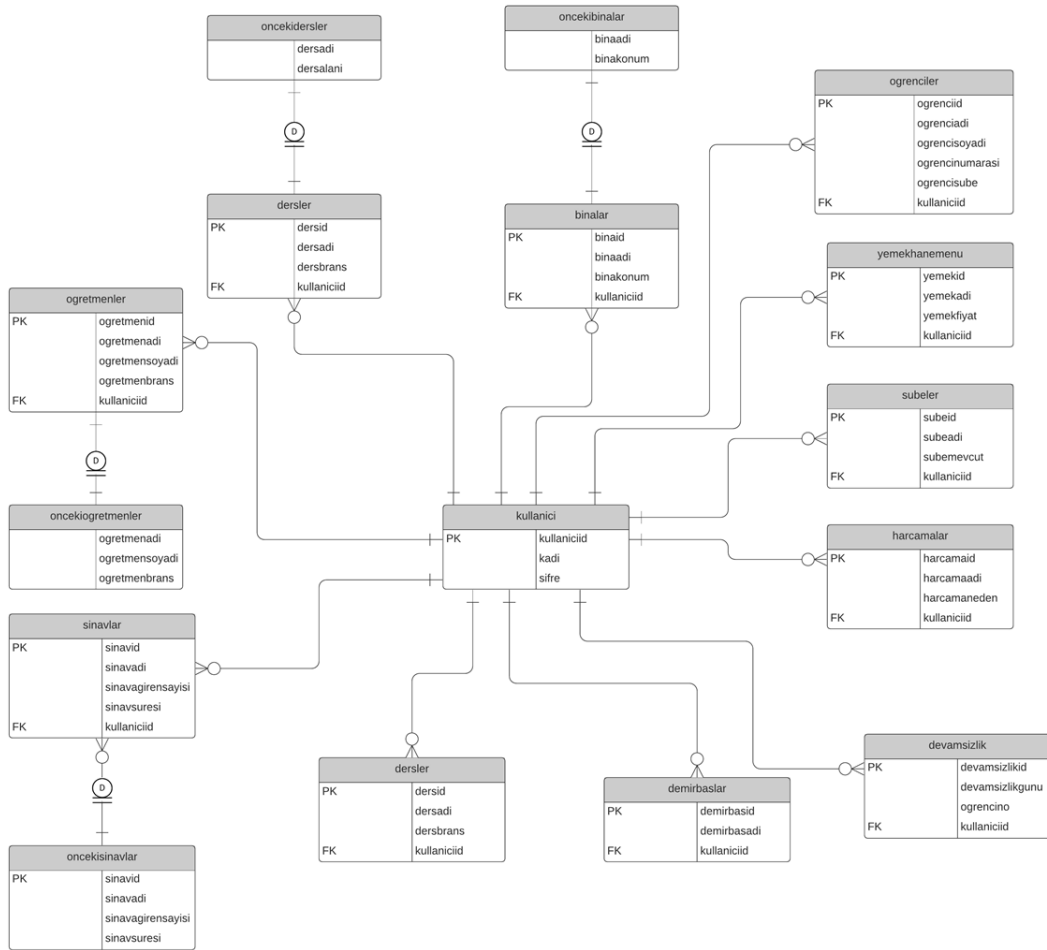
oncekibinalar(binaadi:vvarchar(255),binakonum:vvarchar(255))

oncekidersler(dersadi:vvarchar(255),dersalani:vvarchar(255))

oncekiogretmenler(ogretmenadi:vvarchar(255),ogretmensoyadi:vvarchar(255),ogretmenbrans:vvarchar(255))

oncekisinavlar(sinavadi:vvarchar(255),sinavsuresi:vvarchar(255),sinavagirensayi:vvarchar(255))

b) Varlık Bağntı Modeli



c) Veri tabanını, içerisindeki verilerle birlikte oluşturmayı sağlayan SQL ifadeleri

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 12.2
-- Dumped by pg_dump version 12rc1

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: indirim(integer, integer); Type: PROCEDURE; Schema: public; Owner: postgres
--

CREATE PROCEDURE public.indirim(yemekid integer, indirim integer)
  LANGUAGE plpgsql
  AS $$
begin
  update yemekhanemenu
  set yemekfiyat = yemekfiyat - indirim
  where id = yemekid;
  commit;
end;$$;

ALTER PROCEDURE public.indirim(yemekid integer, indirim integer) OWNER TO postgres;

--
-- Name: oncekibina(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.oncekibina() RETURNS trigger
  LANGUAGE plpgsql
  AS $$
BEGIN
INSERT into oncekibinalar VALUES (OLD.binaadi,OLD.binakonum);
RETURN NEW;
END;
$$;
```

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

```
--  
-- Name: oncekiders(); Type: FUNCTION; Schema: public; Owner: postgres  
--
```

```
CREATE FUNCTION public.oncekiders() RETURNS trigger  
  LANGUAGE plpgsql  
  AS $$  
BEGIN  
INSERT into oncekidersler VALUES (OLD.dersadi,OLD.dersbrans);  
RETURN NEW;  
END;  
$$;
```

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

```
--  
-- Name: oncekiogretmen(); Type: FUNCTION; Schema: public; Owner: postgres  
--
```

```
CREATE FUNCTION public.oncekiogretmen() RETURNS trigger  
  LANGUAGE plpgsql  
  AS $$  
BEGIN  
INSERT into oncekiogretmenler VALUES  
(OLD.ogretmenadi,OLD.ogretmensoyadi,OLD.ogretmenbrans);  
RETURN NEW;  
END;  
$$;
```

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

```
--  
-- Name: oncekisinav(); Type: FUNCTION; Schema: public; Owner: postgres  
--
```

```
CREATE FUNCTION public.oncekisinav() RETURNS trigger  
  LANGUAGE plpgsql  
  AS $$  
BEGIN  
INSERT into oncekisinavlar VALUES (OLD.sinavadi,OLD.sinavsuresi,OLD.sinavagirensayisi);  
RETURN NEW;  
END;  
$$;
```

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

```
--  
-- Name: subeartis(integer, integer); Type: PROCEDURE; Schema: public; Owner: postgres  
--
```

```
CREATE PROCEDURE public.subeartis(subeid integer, artimevcut integer)  
  LANGUAGE plpgsql  
  AS $$  
begin  
  update subeler  
  set subemevcut = subemevcut + artimevcut  
  where id = subeid;  
  commit;  
end;$$;
```

```
ALTER PROCEDURE public.subeartis(subeid integer, artimevcut integer) OWNER TO postgres;
```

```
--  
-- Name: subeazalis(integer, integer); Type: PROCEDURE; Schema: public; Owner: postgres  
--
```

```
CREATE PROCEDURE public.subeazalis(subeid integer, azalimevcut integer)  
  LANGUAGE plpgsql  
  AS $$  
begin  
  update subeler  
  set subemevcut = subemevcut - azalimevcut  
  where id = subeid;  
  commit;  
end;$$;
```

```
ALTER PROCEDURE public.subeazalis(subeid integer, azalimevcut integer) OWNER TO  
postgres;
```

```
--  
-- Name: zamlanma(integer, integer); Type: PROCEDURE; Schema: public; Owner: postgres  
--
```

```
CREATE PROCEDURE public.zamlanma(yemekid integer, zam integer)  
  LANGUAGE plpgsql  
  AS $$  
begin  
  update yemekhanemenu  
  set yemekfiyat = yemekfiyat + zam  
  where id = yemekid;  
  commit;  
end;$$;
```

```
ALTER PROCEDURE public.zamlanma(yemekid integer, zam integer) OWNER TO postgres;
```



```
SET default_tablespace = '';
```

```
SET default_table_access_method = heap;
```

```
--  
-- Name: binalar; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.binalar (  
    binaid integer NOT NULL,  
    binaadi character varying(255) NOT NULL,  
    binakonum character varying(255) NOT NULL,  
    kullaniciid integer DEFAULT 1 NOT NULL  
);
```

```
ALTER TABLE public.binalar OWNER TO postgres;
```

```
--  
-- Name: binalar_binaid_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
--
```

```
CREATE SEQUENCE public.binalar_binaid_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE public.binalar_binaid_seq OWNER TO postgres;
```

```
--  
-- Name: binalar_binaid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres  
--
```

```
ALTER SEQUENCE public.binalar_binaid_seq OWNED BY public.binalar.binaid;
```

```
--  
-- Name: demirbaslar; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.demirbaslar (  
    demirbasid integer NOT NULL,  
    demirbasadi character varying(255) NOT NULL,  
    kullaniciid integer DEFAULT 1 NOT NULL  
);
```

```
ALTER TABLE public.demirbaslar OWNER TO postgres;
```

```
--  
-- Name: demirbaslar_demirbasid_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
--
```

```
CREATE SEQUENCE public.demirbaslar_demirbasid_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE public.demirbaslar_demirbasid_seq OWNER TO postgres;
```

```
--  
-- Name: demirbaslar_demirbasid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:  
postgres  
--
```

```
ALTER SEQUENCE public.demirbaslar_demirbasid_seq OWNED BY  
public.demirbaslar.demirbasid;
```

```
--  
-- Name: dersler; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.dersler (  
    dersid integer NOT NULL,  
    dersadi character varying(255) NOT NULL,  
    dersbrans character varying(255) NOT NULL,  
    kullaniciid integer DEFAULT 1 NOT NULL  
);
```

```
ALTER TABLE public.dersler OWNER TO postgres;
```

```
--  
-- Name: dersler_dersid_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
--
```

```
CREATE SEQUENCE public.dersler_dersid_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```

ALTER TABLE public.dersler_dersid_seq OWNER TO postgres;

--
-- Name: dersler_dersid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public.dersler_dersid_seq OWNED BY public.dersler.dersid;

--
-- Name: devamsizlik; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.devamsizlik (
    devamsizlikid integer NOT NULL,
    ogrencino character varying(255) NOT NULL,
    devamsizlikgunu character varying(10) NOT NULL,
    kullaniciid integer DEFAULT 1 NOT NULL
);

ALTER TABLE public.devamsizlik OWNER TO postgres;

--
-- Name: devamsizlik_devamsizlikid_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public.devamsizlik_devamsizlikid_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public.devamsizlik_devamsizlikid_seq OWNER TO postgres;

--
-- Name: devamsizlik_devamsizlikid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public.devamsizlik_devamsizlikid_seq OWNED BY
public.devamsizlik.devamsizlikid;

--
-- Name: harcamalar; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.harcamalar (

```

```
    harcamaid integer NOT NULL,  
    harcamaadi character varying(255) NOT NULL,  
    harcamaneden character varying(255) NOT NULL,  
    kullaniciid integer DEFAULT 1 NOT NULL  
);
```

```
ALTER TABLE public.harcamalar OWNER TO postgres;
```

```
--  
-- Name: harcamalar_harcamaid_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
--
```

```
CREATE SEQUENCE public.harcamalar_harcamaid_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
ALTER TABLE public.harcamalar_harcamaid_seq OWNER TO postgres;
```

```
--  
-- Name: harcamalar_harcamaid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres  
--
```

```
ALTER SEQUENCE public.harcamalar_harcamaid_seq OWNED BY  
public.harcamalar.harcamaid;
```

```
--  
-- Name: kullanici; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.kullanici (  
    kullaniciid integer NOT NULL,  
    kadi character varying(255) NOT NULL,  
    sifre character varying(255) NOT NULL  
);
```

```
ALTER TABLE public.kullanici OWNER TO postgres;
```

```
--  
-- Name: kullanici_kullaniciid_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
--
```

```
CREATE SEQUENCE public.kullanici_kullaniciid_seq  
    AS integer
```

```
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE public.kullanici_kullaniciid_seq OWNER TO postgres;
```

```
--
-- Name: kullanici_kullaniciid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:
postgres
--
```

```
ALTER SEQUENCE public.kullanici_kullaniciid_seq OWNED BY public.kullanici.kullaniciid;
```

```
--
-- Name: ogrenciler; Type: TABLE; Schema: public; Owner: postgres
--
```

```
CREATE TABLE public.ogrenciler (
    ogrenciid integer NOT NULL,
    ogrenciadi character varying(255) NOT NULL,
    ogrencisoyadi character varying(255) NOT NULL,
    ogrencinumarasi character varying(255) NOT NULL,
    ogrencisinif character varying(255) NOT NULL,
    ogrencisube character varying(255) NOT NULL,
    kullaniciid integer DEFAULT 1 NOT NULL
);
```

```
ALTER TABLE public.ogrenciler OWNER TO postgres;
```

```
--
-- Name: ogrenciler_ogrenciid_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE public.ogrenciler_ogrenciid_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
```

```
ALTER TABLE public.ogrenciler_ogrenciid_seq OWNER TO postgres;
```

```
--
-- Name: ogrenciler_ogrenciid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:
postgres
```

--

ALTER SEQUENCE public.ogrenciler_ogrenciid_seq OWNED BY public.ogrenciler.ogrenciid;

--

-- Name: ogretmenler; Type: TABLE; Schema: public; Owner: postgres

--

```
CREATE TABLE public.ogretmenler (  
    ogretmenid integer NOT NULL,  
    ogretmenadi character varying(255) NOT NULL,  
    ogretmensoyadi character varying(255) NOT NULL,  
    ogretmenbrans character varying(255) NOT NULL,  
    kullaniciid integer DEFAULT 1 NOT NULL  
);
```

ALTER TABLE public.ogretmenler OWNER TO postgres;

--

-- Name: ogretmenler_ogretmenid_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

```
CREATE SEQUENCE public.ogretmenler_ogretmenid_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

ALTER TABLE public.ogretmenler_ogretmenid_seq OWNER TO postgres;

--

-- Name: ogretmenler_ogretmenid_seq; Type: SEQUENCE OWNED BY;
Schema: public; Owner: postgres

--

ALTER SEQUENCE public.ogretmenler_ogretmenid_seq OWNED BY
public.ogretmenler.ogretmenid;

--

-- Name: oncekibinalar; Type: TABLE; Schema: public; Owner: postgres

--

```
CREATE TABLE public.oncekibinalar (  
    binaadi character varying(255) NOT NULL,  
    binakonum character varying(255) NOT NULL  
);
```

```
ALTER TABLE public.oncekibinalar OWNER TO postgres;
```

```
--  
-- Name: oncekidersler; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.oncekidersler (  
    dersadi character varying(255) NOT NULL,  
    dersalani character varying(255) NOT NULL  
);
```

```
ALTER TABLE public.oncekidersler OWNER TO postgres;
```

```
--  
-- Name: oncekiogretmenler; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.oncekiogretmenler (  
    ogretmenadi character varying(255) NOT NULL,  
    ogretmensoyadi character varying(255) NOT NULL,  
    ogretmenbrans character varying(255) NOT NULL  
);
```

```
ALTER TABLE public.oncekiogretmenler OWNER TO postgres;
```

```
--  
-- Name: oncekisinavlar; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.oncekisinavlar (  
    sinavadi character varying(255) NOT NULL,  
    sinavsuresi character varying(255) NOT NULL,  
    sinavagirensayi character varying(255) NOT NULL  
);
```

```
ALTER TABLE public.oncekisinavlar OWNER TO postgres;
```

```
--  
-- Name: sinavlar; Type: TABLE; Schema: public; Owner: postgres  
--
```

```
CREATE TABLE public.sinavlar (  
    sinavid integer NOT NULL,  
    sinavadi character varying(255) NOT NULL,  
    sinavsuresi character varying(255) NOT NULL,  
    sinavagirensayisi character varying(255) NOT NULL,  
    kullaniciid integer DEFAULT 1 NOT NULL
```

```
);
```

```
ALTER TABLE public.sinavlar OWNER TO postgres;
```

```
--
```

```
-- Name: sinavlar_sinavid_seq; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE SEQUENCE public.sinavlar_sinavid_seq  
  AS integer  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE public.sinavlar_sinavid_seq OWNER TO postgres;
```

```
--
```

```
-- Name: sinavlar_sinavid_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
```

```
--
```

```
ALTER SEQUENCE public.sinavlar_sinavid_seq OWNED BY public.sinavlar.sinavid;
```

```
--
```

```
-- Name: siniflar; Type: TABLE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE TABLE public.siniflar (  
  sinifid integer NOT NULL,  
  sinifadi character varying(255) NOT NULL,  
  sinifsube character varying(10) NOT NULL,  
  kullanicIID integer DEFAULT 1 NOT NULL  
);
```

```
ALTER TABLE public.siniflar OWNER TO postgres;
```

```
--
```

```
-- Name: siniflar_sinifid_seq; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE SEQUENCE public.siniflar_sinifid_seq  
  AS integer  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```



```
yemekid integer NOT NULL,  
yemekadi character varying(255) NOT NULL,  
yemekfiyat character varying(10) NOT NULL,  
kullaniciid integer DEFAULT 1 NOT NULL  
);
```

```
ALTER TABLE public.yemekhanemenu OWNER TO postgres;
```

```
--  
-- Name: yemekhaneMenu_yemekid_seq; Type: SEQUENCE; Schema: public; Owner: postgres  
--
```

```
CREATE SEQUENCE public."yemekhaneMenu_yemekid_seq"  
  AS integer  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE public."yemekhaneMenu_yemekid_seq" OWNER TO postgres;
```

```
--  
-- Name: yemekhaneMenu_yemekid_seq; Type: SEQUENCE OWNED BY; Schema: public;  
Owner: postgres  
--
```

```
ALTER SEQUENCE public."yemekhaneMenu_yemekid_seq" OWNED BY  
public.yemekhanemenu.yemekid;
```

```
--  
-- Name: binalar binaid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.binalar ALTER COLUMN binaid SET DEFAULT  
nextval('public.binalar_binaid_seq'::regclass);
```

```
--  
-- Name: demirbaslar demirbasid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.demirbaslar ALTER COLUMN demirbasid SET DEFAULT  
nextval('public.demirbaslar_demirbasid_seq'::regclass);
```

```
--  
-- Name: dersler dersid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.dersler ALTER COLUMN dersid SET DEFAULT  
nextval('public.dersler_dersid_seq'::regclass);
```

```
--  
-- Name: devamsizlik devamsizlikid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.devamsizlik ALTER COLUMN devamsizlikid SET DEFAULT  
nextval('public.devamsizlik_devamsizlikid_seq'::regclass);
```

```
--  
-- Name: harcamalar harcamaid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.harcamalar ALTER COLUMN harcamaid SET DEFAULT  
nextval('public.harcamalar_harcamaid_seq'::regclass);
```

```
--  
-- Name: kullanıcı kullaniciid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.kullanici ALTER COLUMN kullaniciid SET DEFAULT  
nextval('public.kullanici_kullaniciid_seq'::regclass);
```

```
--  
-- Name: ogrenciler ogrenciid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.ogrenciler ALTER COLUMN ogrenciid SET DEFAULT  
nextval('public.ogrenciler_ogrenciid_seq'::regclass);
```

```
--  
-- Name: ogretmenler ogretmenid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.ogretmenler ALTER COLUMN ogretmenid SET DEFAULT  
nextval('public.ogretmenler_ogretmenid_seq'::regclass);
```

```
--  
-- Name: sinavlar sinavid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.sinavlar ALTER COLUMN sinavid SET DEFAULT  
nextval('public.sinavlar_sinavid_seq'::regclass);
```

```
--  
-- Name: siniflar sinifid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.siniflar ALTER COLUMN sinifid SET DEFAULT  
nextval('public.siniflar_sinifid_seq'::regclass);
```

```
--  
-- Name: subeler subeid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.subeler ALTER COLUMN subeid SET DEFAULT  
nextval('public.subeler_subeid_seq'::regclass);
```

```
--  
-- Name: yemekhanemenu yemekid; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE ONLY public.yemekhanemenu ALTER COLUMN yemekid SET DEFAULT  
nextval('public."yemekhaneMenu_yemekid_seq"'::regclass);
```

```
--  
-- Data for Name: binalar; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.binalar VALUES (1, 'Fen Bilimleri Binası', 'Yemekhane Karşısı', 1);  
INSERT INTO public.binalar VALUES (2, 'Spor Bilimleri Binası', 'Girişin Sağında', 1);
```

```
--  
-- Data for Name: demirbaslar; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.demirbaslar VALUES (1, 'Sıralar', 1);  
INSERT INTO public.demirbaslar VALUES (2, 'Masalar', 1);  
INSERT INTO public.demirbaslar VALUES (3, 'Tahtalar', 1);  
INSERT INTO public.demirbaslar VALUES (4, 'Kalemler', 1);
```

```
--  
-- Data for Name: dersler; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.dersler VALUES (1, 'matematik2', 'mat', 1);  
INSERT INTO public.dersler VALUES (4, 'Kimya', 'kim', 1);  
INSERT INTO public.dersler VALUES (2, 'Fizik', 'fiz', 1);  
INSERT INTO public.dersler VALUES (5, 'Kimya2', 'kim', 1);
```

```
--  
-- Data for Name: devamsizlik; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.devamsizlik VALUES (1, '123456', '4', 1);  
INSERT INTO public.devamsizlik VALUES (2, '123564', '7', 1);  
INSERT INTO public.devamsizlik VALUES (3, '12321', '5', 1);  
INSERT INTO public.devamsizlik VALUES (4, '123659', '2', 1);  
INSERT INTO public.devamsizlik VALUES (5, '456987', '2', 1);  
INSERT INTO public.devamsizlik VALUES (6, '12321', '3', 1);
```

```
--  
-- Data for Name: harcamalar; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.harcamalar VALUES (1, 'Temizlik malzemeleri', 'Malzeme ihtiyacı vardı', 1);  
INSERT INTO public.harcamalar VALUES (2, 'Bayrak alındı', 'Tören için', 1);
```

```
--  
-- Data for Name: kullanıcı; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.kullanici VALUES (1, 'admin', '123456');
```

```
--  
-- Data for Name: ogrenciler; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.ogrenciler VALUES (2, 'Ogrenci2', 'Ogrenci2S', '123565', '11', 'A', 1);  
INSERT INTO public.ogrenciler VALUES (4, 'Ogrenci4S', 'Ogrenci4S', '231213', '11', 'B', 1);  
INSERT INTO public.ogrenciler VALUES (1, 'Ogrenci1', 'Ogrenci1S', '123215', '12', 'A', 1);  
INSERT INTO public.ogrenciler VALUES (7, 'OgrenciYeni', 'SoyadYeni', '156987', '12', 'B', 1);
```

```
--  
-- Data for Name: ogretmenler; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.ogretmenler VALUES (1, 'Harry', 'Kane', 'Beden Eğitimi', 1);  
INSERT INTO public.ogretmenler VALUES (3, 'Mahmut', 'Hoca', 'Edebiyat', 1);  
INSERT INTO public.ogretmenler VALUES (5, 'Walter', 'White2', 'Kimya', 1);
```

```
--  
-- Data for Name: oncekibinalar; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.oncekibinalar VALUES ('Güzel Sanatlar Binası', 'Yemekhane Karşısı');
```

```
--  
-- Data for Name: oncekidersler; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.oncekidersler VALUES ('matematik1', 'mat');  
INSERT INTO  
public.oncekidersler VALUES ('fizik2', 'fiz');
```

```
--  
-- Data for Name: oncekiogretmenler; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.oncekiogretmenler VALUES ('Lionel', 'Messi', 'Beden Eğitimi');  
INSERT INTO public.oncekiogretmenler VALUES ('Kanye', 'West', 'Güzel Sanatlar');  
INSERT INTO public.oncekiogretmenler VALUES ('Walter', 'White', 'Kimya Bilimi');
```

```
--  
-- Data for Name: oncekisinavlar; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.oncekisinavlar VALUES ('Matematik2 Kısa Sınav', '125dk ', '55');
```

```
--  
-- Data for Name: sinavlar; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.sinavlar VALUES (1, 'Matematik Kısa Sınav', '120dk', '40', 1);  
INSERT INTO public.sinavlar VALUES (2, 'Fizik Kısa Sınav', '100dk', '35', 1);
```

```
--  
-- Data for Name: siniflar; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.siniflar VALUES (1, '12', 'A', 1);  
INSERT INTO public.siniflar VALUES (2, '12', 'B', 1);  
INSERT INTO public.siniflar VALUES (3, '11', 'A', 1);  
INSERT INTO public.siniflar VALUES (4, '11', 'B', 1);
```

```
--  
-- Data for Name: subeler; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.subeler VALUES (1, '12A', '45', 1);  
INSERT INTO public.subeler VALUES (2, '12B', '38', 1);  
INSERT INTO public.subeler VALUES (3, '11A', '29', 1);
```

```
INSERT INTO public.subeler VALUES (4, '11B', '25', 1);
```

```
--  
-- Data for Name: yemekhanemenu; Type: TABLE DATA; Schema: public; Owner: postgres  
--
```

```
INSERT INTO public.yemekhanemenu VALUES (1, 'Mercimek Çorbası', '2tl', 1);  
INSERT INTO public.yemekhanemenu VALUES (2, 'Pilav', '3tl', 1);  
INSERT INTO public.yemekhanemenu VALUES (3, 'Tavuk Sote', '4tl', 1);  
INSERT INTO public.yemekhanemenu VALUES (4, 'Salata', '2tl', 1);  
INSERT INTO public.yemekhanemenu VALUES (5, 'Karnıyarık', '4', 1);
```

```
--  
-- Name: binalar_binaid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres  
--
```

```
SELECT pg_catalog.setval('public.binalar_binaid_seq', 1, false);
```

```
--  
-- Name: demirbaslar_demirbasid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres  
--
```

```
SELECT pg_catalog.setval('public.demirbaslar_demirbasid_seq', 4, true);
```

```
--  
-- Name: dersler_dersid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres  
--
```

```
SELECT pg_catalog.setval('public.dersler_dersid_seq', 5, true);
```

```
--  
-- Name: devamsizlik_devamsizlikid_seq; Type: SEQUENCE SET; Schema: public; Owner:  
postgres  
--
```

```
SELECT pg_catalog.setval('public.devamsizlik_devamsizlikid_seq', 6, true);
```

```
--  
-- Name: harcamalar_harcamaid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres  
--
```

```
SELECT pg_catalog.setval('public.harcamalar_harcamaid_seq', 1, false);
```

```
--  
-- Name: kullanıcı_kullaniciid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
```

--

```
SELECT pg_catalog.setval('public.kullanici_kullaniciid_seq', 1, false);
```

--

-- Name: ogrenciler_ogrenciid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.ogrenciler_ogrenciid_seq', 7, true);
```

--

-- Name: ogretmenler_ogretmenid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.ogretmenler_ogretmenid_seq', 5, true);
```

--

-- Name: sinavlar_sinavid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.sinavlar_sinavid_seq', 2, true);
```

--

-- Name: siniflar_sinifid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.siniflar_sinifid_seq', 4, true);
```

--

-- Name: subeler_subeid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public.subeler_subeid_seq', 4, true);
```

--

-- Name: yemekhaneMenu_yemekid_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

```
SELECT pg_catalog.setval('public."yemekhaneMenu_yemekid_seq"', 5, true);
```

--

-- Name: binalar BinalarPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--


```
ALTER TABLE ONLY public.binalar
  ADD CONSTRAINT "BinalarPK" PRIMARY KEY (binaid);
```

```
--
-- Name: demirbaslar DemirbaslarPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.demirbaslar
  ADD CONSTRAINT "DemirbaslarPK" PRIMARY KEY (demirbasid);
```

```
--
-- Name: dersler DerslerPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.dersler
  ADD CONSTRAINT "DerslerPK" PRIMARY KEY (dersid);
```

```
--
-- Name: devamsizlik DevamsizlikPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.devamsizlik
  ADD CONSTRAINT "DevamsizlikPK" PRIMARY KEY (devamsizlikid);
```

```
--
-- Name: harcamalar HarcamalarPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.harcamalar
  ADD CONSTRAINT "HarcamalarPK" PRIMARY KEY (harcamaid);
```

```
--
-- Name: kullanıcı Kullanıcılar; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.kullanici
  ADD CONSTRAINT "Kullanıcılar" PRIMARY KEY (kullaniciid);
```

```
--
-- Name: ogrenciler OgrencilerPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.ogrenciler
  ADD CONSTRAINT "OgrencilerPK" PRIMARY KEY (ogrenciid);
```

```
--
-- Name: ogretmenler OgretmenlerPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.ogretmenler
  ADD CONSTRAINT "OgretmenlerPK" PRIMARY KEY (ogretmenid);

--
-- Name: sinavlar SinavlarPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.sinavlar
  ADD CONSTRAINT "SinavlarPK" PRIMARY KEY (sinavid);

--
-- Name: siniflar SiniflarPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.siniflar
  ADD CONSTRAINT "SiniflarPK" PRIMARY KEY (sinifid);

--
-- Name: subeler SubelerPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.subeler
  ADD CONSTRAINT "SubelerPK" PRIMARY KEY (subeid);

--
-- Name: yemekhanemenu YemekhanePK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.yemekhanemenu
  ADD CONSTRAINT "YemekhanePK" PRIMARY KEY (yemekid);

--
-- Name: index_dersid; Type: INDEX; Schema: public; Owner: postgres
--

CREATE INDEX index_dersid ON public.dersler USING btree (dersid);

--
-- Name: index_devamsizlikid; Type: INDEX; Schema: public; Owner: postgres
--

CREATE INDEX index_devamsizlikid ON public.devamsizlik USING btree (devamsizlikid);
```

```
--  
-- Name: index_ogrenciid; Type: INDEX; Schema: public; Owner: postgres  
--
```

```
CREATE INDEX index_ogrenciid ON public.ogrenciler USING btree (ogrenciid);
```

```
--  
-- Name: index_ogretmenid; Type: INDEX; Schema: public; Owner: postgres  
--
```

```
CREATE INDEX index_ogretmenid ON public.ogretmenler USING btree (ogretmenid);
```

```
--  
-- Name: index_sinavid; Type: INDEX; Schema: public; Owner: postgres  
--
```

```
CREATE INDEX index_sinavid ON public.sinavlar USING btree (sinavid);
```

```
--  
-- Name: binalar_bina; Type: TRIGGER; Schema: public; Owner: postgres  
--
```

```
CREATE TRIGGER bina AFTER DELETE ON public.binalar FOR EACH ROW EXECUTE  
FUNCTION public.oncekibina();
```

```
--  
-- Name: dersler_ders; Type: TRIGGER; Schema: public; Owner: postgres  
--
```

```
CREATE TRIGGER ders AFTER DELETE ON public.dersler FOR EACH ROW EXECUTE  
FUNCTION public.oncekididers();
```

```
--  
-- Name: ogretmenler_ogretmen; Type: TRIGGER; Schema: public; Owner: postgres  
--
```

```
CREATE TRIGGER ogretmen AFTER DELETE ON public.ogretmenler FOR EACH ROW  
EXECUTE FUNCTION public.oncekiogretmen();
```

```
--  
-- Name: sinavlar_sinav; Type: TRIGGER; Schema: public; Owner: postgres  
--
```

```
CREATE TRIGGER sinav AFTER DELETE ON public.sinavlar FOR EACH ROW EXECUTE
```

```
FUNCTION public.oncekisinav();
```

```
--
```

```
-- Name: binalar BinalarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.binalar
```

```
ADD CONSTRAINT "BinalarFK" FOREIGN KEY (kullaniciid) REFERENCES  
public.kullanici(kullaniciid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: demirbaslar DemirbaslarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.demirbaslar
```

```
ADD CONSTRAINT "DemirbaslarFK" FOREIGN KEY (kullaniciid) REFERENCES  
public.kullanici(kullaniciid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: dersler DerslerFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.dersler
```

```
ADD CONSTRAINT "DerslerFK" FOREIGN KEY (kullaniciid) REFERENCES  
public.kullanici(kullaniciid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: devamsizlik DevamsizlikFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.devamsizlik
```

```
ADD CONSTRAINT "DevamsizlikFK" FOREIGN KEY (kullaniciid) REFERENCES  
public.kullanici(kullaniciid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
```

```
-- Name: ogrenciler HarcamalarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.ogrenciler
```

```
ADD CONSTRAINT "HarcamalarFK" FOREIGN KEY (kullaniciid) REFERENCES  
public.kullanici(kullaniciid);
```

```
--
```

```
-- Name: ogretmenler HarcamalarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
```

```
--
```

```
ALTER TABLE ONLY public.ogretmenler
```

```
ADD CONSTRAINT "HarcamalarFK" FOREIGN KEY (kullaniciid) REFERENCES
public.kullanici(kullaniciid);
```

```
--
-- Name: harcamalar HarcamalarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.harcamalar
ADD CONSTRAINT "HarcamalarFK" FOREIGN KEY (kullaniciid) REFERENCES
public.kullanici(kullaniciid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

```
--
-- Name: sinavlar HarcamalarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.sinavlar
ADD CONSTRAINT "HarcamalarFK" FOREIGN KEY (kullaniciid) REFERENCES
public.kullanici(kullaniciid);
```

```
--
-- Name: siniflar HarcamalarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.siniflar
ADD CONSTRAINT "HarcamalarFK" FOREIGN KEY (kullaniciid) REFERENCES
public.kullanici(kullaniciid);
```

```
--
-- Name: subeler HarcamalarFK; Type: FK CONSTRAINT;
Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.subeler
ADD CONSTRAINT "HarcamalarFK" FOREIGN KEY (kullaniciid) REFERENCES
public.kullanici(kullaniciid);
```

```
--
-- Name: yemekhanemenu HarcamalarFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.yemekhanemenu
ADD CONSTRAINT "HarcamalarFK" FOREIGN KEY (kullaniciid) REFERENCES
public.kullanici(kullaniciid);
```

```
--
-- PostgreSQL database dump complete-
-
```

d) Saklı yordam (4 adet) ve tetikleyici (4 adet)

Tanımladığım İşlev/Saklı Yordamlar

1)

```
create or replace procedure zamlanma(  
    yemekid int,  
    zam int  
)  
language plpgsql  
as $$  
begin  
    update yemekhanemenu  
    set yemekfiyat = yemekfiyat + zam  
    where id = yemekid;  
    commit;  
end;$$
```

2)

```
create or replace procedure indirim(  
    yemekid int,  
    indirim int  
)  
language plpgsql  
as $$  
begin  
    update yemekhanemenu  
    set yemekfiyat = yemekfiyat - indirim  
    where id = yemekid;  
    commit;  
end;$$
```

3)

```
create or replace procedure subeartis(  
    subeid int,  
    artimevcut int  
)
```

```
language plpgsql
as $$
begin
    update subeler
    set subemevcut = subemevcut + artimevcut
    where id = subeid;
    commit;
end;$$
```

4)

```
create or replace procedure subeazalis(
    subeid int,
    azalismevcut int
)
language plpgsql
as $$
begin
    update subeler
    set subemevcut = subemevcut - azalismevcut
    where id = subeid;
    commit;
end;$$
```

Triggarlar İçin Tanımlanan İşlevler

1)

```
CREATE OR REPLACE FUNCTION oncekiders()
    RETURNS trigger AS
$$
BEGIN
INSERT into oncekidersler VALUES (OLD.dersadi,OLD.dersbrans);
RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';
```

2)

```
CREATE OR REPLACE FUNCTION oncekibina()  
  RETURNS trigger AS  
$$  
BEGIN  
INSERT into oncekibinalar VALUES (OLD.binaadi,OLD.binakonum);  
RETURN NEW;  
END;  
$$  
LANGUAGE 'plpgsql';
```

3)

```
CREATE OR REPLACE FUNCTION oncekiogretmen()  
  RETURNS trigger AS  
$$  
BEGIN  
INSERT into oncekiogretmenler VALUES  
(OLD.ogretmenadi,OLD.ogretmensoyadi,OLD.ogretmenbrans);  
RETURN NEW;  
END;  
$$  
LANGUAGE 'plpgsql';
```

4)

```
CREATE OR REPLACE FUNCTION oncekisinav()  
  RETURNS trigger AS  
$$  
BEGIN  
INSERT into oncekisinavlar VALUES  
(OLD.sinavadi,OLD.sinavsuresi,OLD.sinavagirensayisi);  
RETURN NEW;  
END;  
$$  
LANGUAGE 'plpgsql';
```


Kullandığım Tetikleyiciler:

1)

```
CREATE TRIGGER ders
  AFTER DELETE
  ON dersler
  FOR EACH ROW
  EXECUTE PROCEDURE oncekidders();
```

2)

```
CREATE TRIGGER bina
  AFTER DELETE
  ON binalar
  FOR EACH ROW
  EXECUTE PROCEDURE oncekibina();
```

3)

```
CREATE TRIGGER ogretmen
  AFTER DELETE
  ON ogretmenler
  FOR EACH ROW
  EXECUTE PROCEDURE oncekiogretmen();
```

4)

```
CREATE TRIGGER sinav
  AFTER DELETE
  ON sinavlar
  FOR EACH ROW
  EXECUTE PROCEDURE oncekiSinav();
```

e) Arama, Ekleme, Silme, Güncelleme işlemlerine ait ekran görüntüleri

Tüm ekleme, arama, silme, güncelleme işlemleri aşağıdakine benzer bir şekilde çalışmaktadır.

Öğretmen ekle kodu:

```
def ogretmenEkle(self):
    print("\n----- [Öğretmen Ekle] -----")
    ogrtmAdi = input("Öğretmen Adı: ")
    ogrtmSoyadi = input("Öğretmen Soyadı: ")
    ogrtmBrans = input("Öğretmen Brans: ")
    ogrtmSql = "INSERT INTO ogretmenler(ogretmenadi,ogretmensoyadi,ogretmenbrans) VALUES ('{}','{}','{}').format(ogrtmAdi,ogrtmSoyadi,ogrtmBrans)
    ogrtmEklemeIslemi = db.cursor()
    ogrtmEklemeIslemi.execute(ogrtmSql)
    db.commit()
```

Öğretmen ekle uygulama içinde:

```
Yapılabilecek İşlemler:
1)Öğrenciler ile ilgili işlemler
2)Öğretmenler ile ilgili işlemler
3)Dersler ile ilgili işlemler
4)Sınavlar ile ilgili işlemler
5)Devamsızlık Kayda Geç
6)Demirbaş Kayda Geç
7)Sınıf Ekle
8)Şube Ekle
9)Yemekhane Menüsüne Yemek Ekle
0)Çıkış
```

Yukarıdaki işlemlerden birini seçiniz: 2

Aşağıdaki Listedeki Bir İşlem Seçiniz:

- a)Öğretmen Ekle
- b)Öğretmen Güncelle
- c)Öğretmen Listele
- d)Öğretmen Sil

Seçiminiz: a

```
----- [Öğretmen Ekle] -----
Öğretmen Adı: Walter
Öğretmen Soyadı: White
Öğretmen Brans: Kimya
```

Öğretmen güncelle kodu:

```
def ogretmenGuncelle(self):
    print("\n----- [Öğretmen Güncelle] -----")
    ogrtmId = input("Güncellenecek Öğretmen Id: ")
    ogrtmAdi = input("Öğretmenin Adı: ")
    ogrtmSoyadi = input("Öğretmenin Soyadı: ")
    ogrtmBrans = input("Öğretmenin Bransı: ")
    ogrtmSql = "UPDATE ogretmenler SET ogretmenadi='{ }',ogretmensoyadi='{ }',ogretmenbrans='{ }' WHERE ogretmenid={ }".format(ogrtmAdi,ogrtmSoyadi,ogrtmBrans,ogrtmId)
    ogrtmGuncellemeIslemi = db.cursor()
    ogrtmGuncellemeIslemi.execute(ogrtmSql)
    db.commit()
```

Öğretmen güncelle uygulama içinde:

Yapılabilecek işlemler:

- 1)Öğrenciler ile ilgili işlemler
- 2)Öğretmenler ile ilgili işlemler
- 3)Dersler ile ilgili işlemler
- 4)Sınavlar ile ilgili işlemler
- 5)Devamsızlık Kayda Geç
- 6)Demirbaş Kayda Geç
- 7)Sınıf Ekle
- 8)Şube Ekle
- 9)Yemekhane Menüsüne Yemek Ekle
- 0)Çıkış

Yukarıdaki işlemlerden birini seçiniz: 2

Aşağıdaki Listedeki Bir İşlem Seçiniz:

- a)Öğretmen Ekle
- b)Öğretmen Güncelle
- c)Öğretmen Listele
- d)Öğretmen Sil

Seçiminiz: b

```
----- [Öğretmen Güncelle] -----
Güncellenecek Öğretmen Id: 4
Öğretmenin Adı: Walter
Öğretmenin Soyadı: White
Öğretmenin Bransı: Kimya Bilimi
```

Öğretmen listeleme kodu:

```
def ogretmenListele(self):
    print("\n----- [Öğretmen İsim-Soyisimleri] -----")
    ogrtmSql = "SELECT * FROM ogretmenler"
    ogrtmListelemeIslemi = db.cursor()
    ogrtmListelemeIslemi.execute(ogrtmSql)
    ogretmenler = ogrtmListelemeIslemi.fetchall()
    for ogretmen in ogretmenler:
        print("İsim:",ogretmen[1],"Soyisim:",ogretmen[2])
```

Öğretmen listele uygulama içinde:

```
Yapılabilecek İşlemler:  
1)Öğrenciler ile ilgili işlemler  
2)Öğretmenler ile ilgili işlemler  
3)Dersler ile ilgili işlemler  
4)Sınavlar ile ilgili işlemler  
5)Devamsızlık Kayda Geç  
6)Demirbaş Kayda Geç  
7)Sınıf Ekle  
8)Şube Ekle  
9)Yemekhane Menüüne Yemek Ekle  
0)Çıkış
```

Yukarıdaki işlemlerden birini seçiniz: 2

```
Aşağıdaki Listedeki Bir İşlem Seçiniz:  
a)Öğretmen Ekle  
b)Öğretmen Güncelle  
c)Öğretmen Listele  
d)Öğretmen Sil
```

Seçiminiz: c

```
----- [Öğretmen İsim-Soyisimleri] -----  
İsim: Harry Soyisim: Kane  
İsim: Mahmut Soyisim: Hoca  
İsim: Walter Soyisim: White
```

Öğretmen silme kodu:

```
def ogretmenSil(self):  
    print("\n----- [Öğretmen Sil] -----")  
    ogrtmId = input("Silinecek Öğretmen Id: ")  
    ogrtmSql = "DELETE FROM ogretmenler WHERE ogretmenid={}".format(ogrtmId)  
    ogrtmSilmeIslemi = db.cursor()  
    ogrtmSilmeIslemi.execute(ogrtmSql)  
    db.commit()
```

Öğretmen sil uygulama içinde:

Yukarıdaki işlemlerden birini seçiniz: 2

```
Aşağıdaki Listedeki Bir İşlem Seçiniz:  
a)Öğretmen Ekle  
b)Öğretmen Güncelle  
c)Öğretmen Listele  
d)Öğretmen Sil
```

Seçiminiz: d

```
----- [Öğretmen Sil] -----  
Silinecek Öğretmen Id: 4
```

Öğrenci ekle kodu:

```
def ogrenciEkle(self):
    print("----- [Öğrenci Ekle] -----")
    ogrAdi = input("Öğrenci Adı: ")
    ogrSoyadi = input("Öğrenci Soyadı: ")
    ogrNo = input("Öğrenci Numarası: ")
    ogrSinif = input("Öğrenci Sınıfı: ")
    ogrSube = input("Öğrenci Şubesi: ")
    ogrSql = "INSERT INTO ogrenciler(ogrenciadi,ogrencisoyadi,ogrencinumarasi,ogrencisinif,ogrencisube) VALUES ('{}','{}','{}','{}','{}')".format(ogrAdi,ogrSoyadi,ogrNo,ogrSinif,ogrSube)
    ogreklemeIslemi = db.cursor()
    ogreklemeIslemi.execute(ogrSql)
    db.commit()
```

Öğrenci ekle uygulama içinde:

Okul/Kurs Bilgi Takip Uygulaması

Yapılabilecek İşlemler:

- 1)Öğrenciler ile ilgili İşlemler
- 2)Öğretmenler ile ilgili İşlemler
- 3)Dersler ile ilgili İşlemler
- 4)Sınavlar ile ilgili İşlemler
- 5)Devamsızlık Kayda Geç
- 6)Demirbaş Kayda Geç
- 7)Sınıf Ekle
- 8)Şube Ekle
- 9)Yemekhane Menüsüne Yemek Ekle
- 0)Çıkış

Yukarıdaki işlemlerden birini seçiniz: 1

Aşağıdaki Listedten Bir İşlem Seçiniz:

- a)Öğrenci Ekle
- b)Öğrenci Güncelle
- c)Öğrenci Listele
- d)Öğrenci Sil

Seçiminiz: a

----- [Öğrenci Ekle] -----

Öğrenci Adı: Oğrenci1

Öğrenci Soyadı: Oğrenci1S

Öğrenci Numarası: 12315

Öğrenci Sınıfı: 12

Öğrenci Şubesi: A

Öğrenci güncelle kodu:

```
def ogrenciGuncelle(self):
    print("\n----- [Öğrenci Güncelle] -----")
    ogrId = input("Güncellenecek Öğrenci Id: ")
    ogrAdi = input("Öğrencinin Adı: ")
    ogrSoyadi = input("Öğrencinin Soyadı: ")
    ogrNo = input("Öğrencinin Numarası: ")
    ogrSinif = input("Öğrencinin Sınıfı: ")
    ogrSube = input("Öğrencinin Subesi: ")
    ogrSql = "UPDATE ogrenciler SET ogrenciadi='{}',ogrencisoyadi='{}',ogrencinumarasi='{}',ogrencisinif='{}',ogrencisube='{}' WHERE ogrenciid={}".format(ogrAdi,ogrSoyadi,ogrNo,ogrSinif,ogrSube,ogrId)
    ogroguncellemeIslemi = db.cursor()
    ogroguncellemeIslemi.execute(ogrSql)
    db.commit()
```

Öğrenci güncelle uygulama içinde:

Yukarıdaki işlemlerden birini seçiniz: 1

Aşağıdaki Listedeki Bir İşlem Seçiniz:

- a)Öğrenci Ekle
- b)Öğrenci Güncelle
- c)Öğrenci Listele
- d)Öğrenci Sil

Seçiminiz: b

```
----- [Öğrenci Güncelle] -----
Güncellenecek Öğrenci Id: 1
Öğrencinin Adı: Oğrenci1
Öğrencinin Soyadı: Oğrenci1Ss
Öğrencinin Numarası: 123215
Öğrencinin Sınıfı: 12
Öğrencinin Şubesi: A
```

Öğrenci listeleme kodu:

```
def ogrenciliste(self):
    print("\n----- [Öğrenci İsim-Soyisimleri] -----")
    ogrSql = "SELECT * FROM ogrenciler"
    ogrListelemeIslemi = db.cursor()
    ogrListelemeIslemi.execute(ogrSql)
    ogrenciler = ogrListelemeIslemi.fetchall()
    for ogrenci in ogrenciler:
        print("İsim:",ogrenci[1],"Soyisim:",ogrenci[2])
```

Öğrenci listele uygulama içinde:

Yukarıdaki işlemlerden birini seçiniz: 1

Aşağıdaki Listedeki Bir İşlem Seçiniz:

- a)Öğrenci Ekle
- b)Öğrenci Güncelle
- c)Öğrenci Listele
- d)Öğrenci Sil

Seçiminiz: c

```
----- [Öğrenci İsim-Soyisimleri] -----
İsim: Oğrenci2 Soyisim: Oğrenci2S
İsim: Oğrenci4S Soyisim: Oğrenci4S
İsim: Oğrenci1 Soyisim: Oğrenci1S
```

Öğrenci silme kodu:

```
def ogrnciSil(self):
    print("\n----- [Öğrenci Sil] -----")
    ogrId = input("Silinecek Öğrenci Id: ")
    ogrSql = "DELETE FROM ogrenciler WHERE ogrnciid={}".format(ogrId)
    ogrSilmeIslemi = db.cursor()
    ogrSilmeIslemi.execute(ogrSql)
    db.commit()
```

Öğrenci sil uygulama içinde:

Yukarıdaki işlemlerden birini seçiniz: 1

Aşağıdaki Listedeki Bir İşlem Seçiniz:

- a)Öğrenci Ekle
- b)Öğrenci Güncelle
- c)Öğrenci Listele
- d)Öğrenci Sil

Seçiminiz: d

----- [Öğrenci Sil] -----
Silinecek Öğrenci Id: 1

Sınıf ekleme kodu:

```
def sinifEkle(self):
    print("\n----- [Sınıf Ekle] -----")
    sinifAdi = input("Sınıf Adı: ")
    sinifSube = input("Sınıf Şube: ")
    sinifSql = "INSERT INTO siniflar(sinifadi,sinifsube) VALUES ('{}','{}')".format(sinifAdi,sinifSube)
    sinifEklemeIslemi = db.cursor()
    sinifEklemeIslemi.execute(sinifSql)
    db.commit()
```

Şube ekleme kodu:

```
def subeEkle(self):
    print("\n----- [Şube Ekle] -----")
    subeAdi = input("Şube Adı: ")
    subeMevcut = input("Şubede Bulunan Öğrenci Sayısı: ")
    subeSql = "INSERT INTO subeler(subeadi,subemevcut) VALUES ('{}','{}')".format(subeAdi,subeMevcut)
    subeEklemeIslemi = db.cursor()
    subeEklemeIslemi.execute(subeSql)
    db.commit()
```

Devamsızlık ekleme kodu:

```
def devamsizlikEkle(self):
    print("\n----- [Devamsızlık Ekle] -----")
    ogrenciNo = input("Devamsızlık Yapan Öğrenci No: ")
    devamsizlikGunu = input("Kaç Gün Devamsızlık Yaptı: ")
    devamsizlikSql = "INSERT INTO devamsizlik(ogrencino,devamsizlikgunu) VALUES ('{}','{}').format(ogrenciNo,devamsizlikGunu)
    devamsizlikEklemeIslemi = db.cursor()
    devamsizlikEklemeIslemi.execute(devamsizlikSql)
    db.commit()
```

Menüye yemek ekleme kodu:

```
def menuyeYemekEkle(self):
    print("\n----- [Menüye Yemek Ekle] -----")
    yemekAdi = input("Yemek Adı: ")
    yemekFiyati = input("Yemek Fiyatı: ")
    yemekSql = "INSERT INTO yemekhanemenu(yemekadi,yemekfiyat) VALUES ('{}','{}').format(yemekAdi,yemekFiyati)
    yemekEklemeIslemi = db.cursor()
    yemekEklemeIslemi.execute(yemekSql)
    db.commit()
```

Demirbaş ekleme kodu:

```
def demirbasEkle(self):
    print("\n----- [Demirbaş Ekle] -----")
    demirbasAdi = input("Demirbaş Adı: ")
    demirbasSql = "INSERT INTO demirbaslar(demirbasadi) VALUES ('{}').format(demirbasAdi)
    demirbasEklemeIslemi = db.cursor()
    demirbasEklemeIslemi.execute(demirbasSql)
    db.commit()
```

Uygulamayı oluşturan 11 adet sınıf tasarladım bunlardan 9 adeti ekleme, arama, silme ,güncelleme gibi işlemleri içeren sınıflar oldu bu sınıflardan da nesne örnekleyerek main.py dosyasında çağırdım ve bu işlemleri gerçekleştirdim. Uygulama kaynak kodlarında da tüm hepsi mevcuttur.

f) Uygulama Kaynak Kodları

Ogrenciler.py dosyası

```
from veritabani import *
```

```
class Ogrenciler():
    def __init__(self):
        self.ogrAdi = ""
        self.ogrSoyadi = ""
        self.ogrNo = ""
        self.ogrSinif = ""
        self.ogrSube = ""

    def ogrenciEkle(self):
        print("----- [Öğrenci Ekle] -----")
        ogrAdi = input("Öğrenci Adı: ")
        ogrSoyadi = input("Öğrenci Soyadı: ")
        ogrNo = input("Öğrenci Numarası: ")
        ogrSinif = input("Öğrenci Sınıfı: ")
        ogrSube = input("Öğrenci Şubesi: ")
        ogrSql = "INSERT INTO ogrenciler(ogrenciadi,ogrencisoyadi,ogrencinumarasi,ogrencisinif,ogrencisube) VALUES ('{}','{}','{}','{}','{}')".format(ogrAdi,ogrSoyadi,ogrNo,ogrSinif,ogrSube)
        ogrEklemeIslemi = db.cursor()
        ogrEklemeIslemi.execute(ogrSql)
        db.commit()

    def ogrenciGuncelle(self):
        print("\n----- [Öğrenci Güncelle] -----")
        ogrId = input("Güncellenecek Öğrenci Id: ")
        ogrAdi = input("Öğrencinin Adı: ")
        ogrSoyadi = input("Öğrencinin Soyadı: ")
        ogrNo = input("Öğrencinin Numarası: ")
        ogrSinif = input("Öğrencinin Sınıfı: ")
        ogrSube = input("Öğrencinin Şubesi: ")
        ogrSql = "UPDATE ogrenciler SET ogrenciadi='{}',ogrencisoyadi='{}',ogrencinumarasi='{}',ogrencisinif='{}',ogrencisube='{}' WHERE ogrenciid={}".format(ogrAdi,ogrSoyadi,ogrNo,ogrSinif,ogrSube,ogrId)
        ogrGuncellemeIslemi = db.cursor()
        ogrGuncellemeIslemi.execute(ogrSql)
        db.commit()

    def ogrenciListele(self):
        print("\n----- [Öğrenci İsim-Soyisimleri] -----")
        ogrSql = "SELECT * FROM ogrenciler"
        ogrListelemeIslemi = db.cursor()
        ogrListelemeIslemi.execute(ogrSql)
        ogrenciler = ogrListelemeIslemi.fetchall()
```

```

for ogrenci in ogrenciler:
    print("İsim:",ogrenci[1],"Soyisim:",ogrenci[2])

def ogrenciSil(self):
    print("\n----- [Öğrenci Sil] -----")
    ogrId = input("Silinecek Öğrenci Id: ")
    ogrSql = "DELETE FROM ogrenciler WHERE ogrenciid={}".format(ogrId)
    ogrSilmeIslemi = db.cursor()
    ogrSilmeIslemi.execute(ogrSql)
    db.commit()

```

Ogretmenler.py dosyası

```

from veritabani import *

```

```

class Ogretmenler():
    def __init__(self):
        self.ogrtmAdi = ""
        self.ogrtmSoyadi = ""
        self.ogrtmBrans = ""

    def ogretmenEkle(self):
        print("\n----- [Öğretmen Ekle] -----")
        ogrtmAdi = input("Öğretmen Adı: ")
        ogrtmSoyadi = input("Öğretmen Soyadı: ")
        ogrtmBrans = input("Öğretmen Branş: ")
        ogrtmSql = "INSERT INTO ogretmenler(ogretmenadi,ogretmensoyadi,ogretmenbrans) V
ALUES ('{}','{}','{}')".format(ogrtmAdi,ogrtmSoyadi,ogrtmBrans)
        ogrtmEklemeIslemi = db.cursor()
        ogrtmEklemeIslemi.execute(ogrtmSql)
        db.commit()

    def ogretmenGuncelle(self):
        print("\n----- [Öğretmen Güncelle] -----")
        ogrtmId = input("Güncellenecek Öğretmen Id: ")
        ogrtmAdi = input("Öğretmenin Adı: ")
        ogrtmSoyadi = input("Öğretmenin Soyadı: ")
        ogrtmBrans = input("Öğretmenin Branşı: ")
        ogrtmSql = "UPDATE ogretmenler SET ogretmenadi='{}',ogretmensoyadi='{}',ogretme
nbrans='{}' WHERE ogretmenid={}".format(ogrtmAdi,ogrtmSoyadi,ogrtmBrans,ogrtmId)
        ogrtmGuncellemeIslemi = db.cursor()
        ogrtmGuncellemeIslemi.execute(ogrtmSql)
        db.commit()

    def ogretmenListele(self):
        print("\n----- [Öğretmen İsim-Soyisimleri] -----")
        ogrtmSql = "SELECT * FROM ogretmenler"
        ogrtmListelemeIslemi = db.cursor()
        ogrtmListelemeIslemi.execute(ogrtmSql)

```

```

ogretmenler = ogrtmListelemeIslemi.fetchall()
for ogretmen in ogretmenler:
    print("İsim:",ogretmen[1],"Soyisim:",ogretmen[2])

def ogretmenSil(self):
    print("\n----- [Öğretmen Sil] -----")
    ogrtmId = input("Silinecek Öğretmen Id: ")
    ogrtmSql = "DELETE FROM ogretmenler WHERE ogretmenid={}".format(ogrtmId)
    ogrtmSilmeIslemi = db.cursor()
    ogrtmSilmeIslemi.execute(ogrtmSql)
    db.commit()

```

Sinavlar.py dosyası

```

from veritabani import *

class Sinavlar():

    def __init__(self):
        self.sinavAdi = ""
        self.sinavaGirenSayisi = ""
        self.sinavSuresi = ""

    def sinavEkle(self):
        print("\n----- [Sınav Ekle] -----")
        sinavAdi = input("Sınav Adı: ")
        sinavaGirenSayisi = input("Sınava Giren Öğrenci Sayısı: ")
        sinavSuresi = input("Sınav Süresi: ")
        sinavSql = "INSERT INTO sinavlar(sinavadi,sinavagirensayisi,sinavsuresi) VALUES ('{}','{}','{}').format(sinavAdi,sinavaGirenSayisi,sinavSuresi)
        sinavEklemeIslemi = db.cursor()
        sinavEklemeIslemi.execute(sinavSql)
        db.commit()

    def sinavGuncelle(self):
        print("\n----- [Sınav Güncelle] -----")
        sinavId = input("Güncellenecek Sınav Id: ")
        sinavAdi = input("Sınav Adı: ")
        sinavaGirenSayisi = input("Sınava Giren Sayısı: ")
        sinavSuresi = input("Sınav Süresi: ")
        sinavSql = "UPDATE sinavlar SET sinavadi='{}',sinavagirensayisi='{}',sinavsuresi='{}' WHERE sinavid={}".format(sinavAdi,sinavaGirenSayisi,sinavSuresi,sinavId)
        sinavGuncellemeIslemi = db.cursor()
        sinavGuncellemeIslemi.execute(sinavSql)
        db.commit()

    def sinavListele(self):
        print("\n----- [Sınav Adı-Süresi] -----")
        sinavSql = "SELECT * FROM sinavlar"

```

```

sinavListelemeIslemi = db.cursor()
sinavListelemeIslemi.execute(sinavSql)
sinavlar = sinavListelemeIslemi.fetchall()
for sinav in sinavlar:
    print("Sınav Adı:",sinav[1],"Süresi:",sinav[2])

def sinavSil(self):
    print("\n----- [Sınav Sil] -----")
    sinavId = input("Silinecek Sınav Id: ")
    sinavSql = "DELETE FROM sinavlar WHERE sinavid={}".format(sinavId)
    sinavSilmeIslemi = db.cursor()
    sinavSilmeIslemi.execute(sinavSql)
    db.commit()

```

Siniflar.py dosyası

```

from veritabani import *

class Siniflar():

    def __init__(self):
        self.sinifAdi = ""
        self.sinifSube = ""

    def sinifEkle(self):
        print("\n----- [Sınıf Ekle] -----")
        sinifAdi = input("Sınıf Adı: ")
        sinifSube = input("Sınıf Şube: ")
        sinifSql = "INSERT INTO siniflar(sinifadi,sinifsube) VALUES ('{}','{}').format(
(sinifAdi,sinifSube)
        sinifEklemeIslemi = db.cursor()
        sinifEklemeIslemi.execute(sinifSql)
        db.commit()

```

Subeler.py dosyası

```

from veritabani import *

class Subeler():

    def __init__(self):
        self.subeAdi = ""
        self.subeMevcut = ""

    def subeEkle(self):
        print("\n----- [Şube Ekle] -----")
        subeAdi = input("Şube Adı: ")
        subeMevcut = input("Şubede Bulunan Öğrenci Sayısı: ")

```

```
subeSql = "INSERT INTO subeler(subeadı,subemevcut) VALUES ('{}','{}').format(s
ubeAdi,subeMevcut)
subeEklemeIslemi = db.cursor()
subeEklemeIslemi.execute(subeSql)
db.commit()
```

Yemekhane.py dosyası

```
from veritabani import *

class Yemekhane():

    def __init__(self):
        self.yemekAdi = ""
        self.yemekFiyati = ""

    def menuyeYemekEkle(self):
        print("\n----- [Menüye Yemek Ekle] -----")
        yemekAdi = input("Yemek Adı: ")
        yemekFiyati = input("Yemek Fiyatı: ")
        yemekSql = "INSERT INTO yemekhanemenu(yemekadi,yemekfiyat) VALUES ('{}','{}').
format(yemekAdi,yemekFiyati)
        yemekEklemeIslemi = db.cursor()
        yemekEklemeIslemi.execute(yemekSql)
        db.commit()
```

Demirbaslar.py dosyası

```
from veritabani import *

class Demirbaslar():

    def __init__(self):
        self.demirbasAdi = ""

    def demirbasEkle(self):
        print("\n----- [Demirbaş Ekle] -----")
        demirbasAdi = input("Demirbaş Adı: ")
        demirbasSql = "INSERT INTO demirbaslar(demirbasadi) VALUES ('{}').format(demir
basAdi)
        demirbasEklemeIslemi = db.cursor()
        demirbasEklemeIslemi.execute(demirbasSql)
        db.commit()
```

Dersler.py dosyası

```
from veritabani import *
```

```
class Dersler():
```

```
    def __init__(self):  
        self.dersAdi = ""  
        self.dersBrans = ""
```

```
    def dersEkle(self):  
        print("\n----- [Ders Ekle] -----")  
        dersAdi = input("Ders Adı: ")  
        dersBrans = input("Ders Branş: ")  
        dersSql = "INSERT INTO dersler(dersadi,dersbrans) VALUES ('{}','{}').format(de  
rsAdi,dersBrans)  
        dersEklemeIslemi = db.cursor()  
        dersEklemeIslemi.execute(dersSql)  
        db.commit()
```

```
    def dersGuncelle(self):  
        print("\n----- [Ders Güncelle] -----")  
        dersId = input("Güncellenecek Ders Id: ")  
        dersAdi = input("Ders Adı: ")  
        dersBrans = input("Ders Branş: ")  
        dersSql = "UPDATE dersler SET dersadi='{}',dersbrans='{}' WHERE dersid={}".form  
at(dersAdi,dersBrans,dersId)  
        dersGuncellemeIslemi = db.cursor()  
        dersGuncellemeIslemi.execute(dersSql)  
        db.commit()
```

```
    def dersListele(self):  
        print("\n----- [Ders Adı-Branş] -----")  
        dersSql = "SELECT * FROM dersler"  
        dersListeleIslemi = db.cursor()  
        dersListeleIslemi.execute(dersSql)  
        dersler = dersListeleIslemi.fetchall()  
        for ders in dersler:  
            print("İsim:",ders[1],"Branş:",ders[2])
```

```
    def dersSil(self):  
        print("\n----- [Ders Sil] -----")  
        dersId = input("Silinecek Ders Id: ")  
        dersSql = "DELETE FROM dersler WHERE dersid={}".format(dersId)  
        dersSilmeIslemi = db.cursor()  
        dersSilmeIslemi.execute(dersSql)  
        db.commit()
```

Devamsizlik.py dosyası

```
from veritabani import *

class Devamsizlik():

    def __init__(self):
        self.ogrenciNo = ""
        self.devamsizlikGunu = ""

    def devamsizlikEkle(self):
        print("\n----- [Devamsızlık Ekle] -----")
        ogrenciNo = input("Devamsızlık Yapan Öğrenci No: ")
        devamsizlikGunu = input("Kaç Gün Devamsızlık Yaptı: ")
        devamsizlikSql = "INSERT INTO devamsizlik(ogrencino,devamsizlikgunu) VALUES ('{
}', '{')' ".format(ogrenciNo,devamsizlikGunu)
        devamsizlikEklemeIslemi = db.cursor()
        devamsizlikEklemeIslemi.execute(devamsizlikSql)
        db.commit()
```

main.py dosyası

```
from Ogrenciler import *
from Ogretmenler import *
from Sinavlar import *
from Dersler import *
from Devamsizlik import *
from Demirbaslar import *
from Yemekhane import *
from Siniflar import *
from Subeler import *
```

```
while(True):
    print("""
```

Okul/Kurs Bilgi Takip Uygulaması

Yapılabilecek İşlemler:

- 1)Öğrenciler ile ilgili İşlemler
- 2)Öğretmenler ile ilgili İşlemler
- 3)Dersler ile ilgili İşlemler
- 4)Sınavlar ile ilgili İşlemler
- 5)Devamsızlık Kayda Geç
- 6)Demirbaş Kayda Geç
- 7)Sınıf Ekle
- 8)Şube Ekle
- 9)Yemekhane Menüsüne Yemek Ekle
- 0)Çıkış

```

"""
menuSecim = int(input("Yukarıdaki işlemlerden birini seçiniz: "))
if(menuSecim == 1):
    ogrenciler = Ogrenciler()
    print("""
Aşağıdaki Listedeki Bir İşlem Seçiniz:
a)Öğrenci Ekle
b)Öğrenci Güncelle
c)Öğrenci Listele
d)Öğrenci Sil
""")
    ogrI = input("Seçiminiz: ")
    if(ogrI == "a" or ogrI == "A"):
        ogrenciler.ogrenciEkle()
    elif(ogrI == "b" or ogrI == "B"):
        ogrenciler.ogrenciGuncelle()
    elif(ogrI == "c" or ogrI == "C"):
        ogrenciler.ogrenciListele()
    elif(ogrI == "d" or ogrI == "D"):
        ogrenciler.ogrenciSil()

elif(menuSecim == 2):
    ogretmenler = Ogretmenler()
    print("""
Aşağıdaki Listedeki Bir İşlem Seçiniz:
a)Öğretmen Ekle
b)Öğretmen Güncelle
c)Öğretmen Listele
d)Öğretmen Sil
""")
    ogrtmI = input("Seçiminiz: ")
    if(ogrtmI == "a" or ogrtmI == "A"):
        ogretmenler.ogretmenEkle()
    elif(ogrtmI == "b" or ogrtmI == "B"):
        ogretmenler.ogretmenGuncelle()
    elif(ogrtmI == "c" or ogrtmI == "C"):
        ogretmenler.ogretmenListele()
    elif(ogrtmI == "d" or ogrtmI == "D"):
        ogretmenler.ogretmenSil()

elif(menuSecim == 3):
    dersler = Dersler()
    print("""
Aşağıdaki Listedeki Bir İşlem Seçiniz:
a)Ders Ekle
b)Ders Güncelle
c)Ders Listele
d)Ders Sil
""")
    dersI = input("Seçiminiz: ")
    if(dersI == "a" or dersI == "A"):

```



```
dersler.dersEkle()
elif(dersI == "b" or dersI == "B"):
    dersler.dersGuncelle()
elif(dersI == "c" or dersI == "C"):
    dersler.dersListele()
elif(dersI == "d" or dersI == "D"):
    dersler.dersSil()

elif(menuSecim == 4):
    sinavlar = Sinavlar()
    print("""
Aşağıdaki Listedeki Bir İşlem Seçiniz:
a)Sınav Ekle
b)Sınav Güncelle
c)Sınav Listele
d)Sınav Sil
""")
    sinavI = input("Seçiminiz: ")
    if(sinavI == "a" or sinavI == "A"):
        sinavlar.sinavEkle()
    elif(sinavI == "b" or sinavI == "B"):
        sinavlar.sinavGuncelle()
    elif(sinavI == "c" or sinavI == "C"):
        sinavlar.sinavListele()
    elif(sinavI == "d" or sinavI == "D"):
        sinavlar.sinavSil()

elif(menuSecim == 5):
    devamsizlik = Devamsizlik()
    devamsizlik.devamsizlikEkle()

elif(menuSecim == 6):
    demirbaslar = Demirbaslar()
    demirbaslar.demirbasEkle()

elif(menuSecim == 7):
    siniflar = Siniflar()
    siniflar.sinifEkle()

elif(menuSecim == 8):
    subeler = Subeler()
    subeler.subeEkle()

elif(menuSecim == 9):
    yemekhanemenu = Yemekhane()
    yemekhanemenu.menuyeYemekEkle()

elif(menuSecim == 0):
    print("Uygulama Sonlanıyor...")
    exit()
else:
```

```
print("Listede Olan Bir İşlemi Seçiniz!")
```

g) Çalışmanın anlatıldığı video adresi:

https://www.youtube.com/watch?v=cYGw3dir_DM