Sakarya Üniversitesi

Veri Tabanı Yönetim Sistemleri
Proje Ödevi

İzzet İlker Durdu

B181210033 1-B

izzet.durdu@ogr.sakarya.edu.tr

Uygulama Tanıtımı:

Merhabalar. Bu projemizde bizlerden veri tabanı yönetim sistemini aktif olarak kullandığımız bir proje hazırlamamız istendi. Ben bir kütüphane uygulaması yazmaya çalıştım elimden geldiğince optimize bir şekilde veri tabanını oluşturdum ve uygulamamı .net ortamında geliştirdim. Projemin detayları aşağıdadır.

İş Kuralları:

1- Kişiler, siteye giriş yaparken kullanılacak şifreye ve maile sahiptir.

2- Kişiler görevli veya müşteri olabilir.

3- Görevlinin kendine özel araçları vardır.

4- Her kitap bir numaraya, isme, basım tarihine, yayınevine ve adede sahiptir.

5- Kitaplar birbirlerinden numaraları ile ayrılırlar.

6- Her yayınevi, yayınevi koduna, isme, telefon numarasına, web ve mail adresi bilgilerine sahiptir.

7- Basım türü bilgisi, kitaplar için basıldıkları tür bilgisini kod ve ad özellikleri ile tutmaktadır.

8- Her yazar yazar numarasına, isme, soyada, telefon numarası ve mail bilgilerine
sahiptir.

9- Yazarlar birbirlerinden yazar numaraları ile ayrılırlar.

10- Her kitap için tür bilgisi bulunur ve tür bilgisi tür adi ile birbirlerinden ayrılır.

11- Her müşteri için satın alma işlemi mevcuttur.

12- Fatura numarası bilgisi ile satın alma işlemleri birbirlerinden ayrılır.

13- Bir kişi birden fazla kitap alabilir, hiç kitap almayabilir.

14- Bir kitap birden fazla kişi tarafından alınmış olabilir, hiç alınmamış da olabilir.

15- Bir müşteri birden fazla satın alma işlemi yapabilir, hiç yapmamış da olabilir.

16- Bir satın alma işlemi en az bir en çok bir kişi tarafından yapılmış olmalıdır.

17- Bir kitaba ait birden fazla yazar olabilir, en az bir yazar olmalıdır.

18- Bir yazar birden fazla kitap yazmış olabilir, hiç kitap yazmamış da olabilir.

19- Bir yayınevi birden fazla kitaba sahip olabilir, hiç kitabı olmayabilir.

20- Bir kitap en az bir en çok bir yayınevine sahip olmalıdır.

21- Bir kitap birden fazla basım türüne sahip olabilir, hiç basılmamış olabilir.

22- Bir basım türü birden fazla kitap için kullanılmış olabilir, hiç kullanılmamış da olabilir.

23- Bir kitap birden fazla türde olabilir, en bir türde olmak zorundadır.

24- Bir türe ait birden fazla kitap olabilir, en az bir kitap olmak zorundadır.

İlişkisel Şema:

1- Kitap(**kitapNo:integer**, ad:varchar, basimTarihi:date, **yayinEvi:varchar**, adet:integer)
2- Yayinevi(**yayineviKodu:varchar**, ad:varchar, telefonNo:varchar, webAdresi:varchar, mail:varchar)
3- KitapYazar(**kitapNo:integer**, **yazarNo:integer**)

4- Yazar(**yazarNo:integer**, ad:varchar, soyad:varchar, telefonNo:varchar, mail:varchar)
5- KitapTur(**kitapNo:integer**, **turAdi:varchar**)

6- Tur(**turAdi:varchar**)

7- KitapBasim(**kitapNo:integer**, **turKodu:varchar**, fiyat:int)

8- BasimTuru(**turKodu:varchar**, ad:varchar)

9- KisiKitap(**mail:varchar**, **kitapNo:integer**)

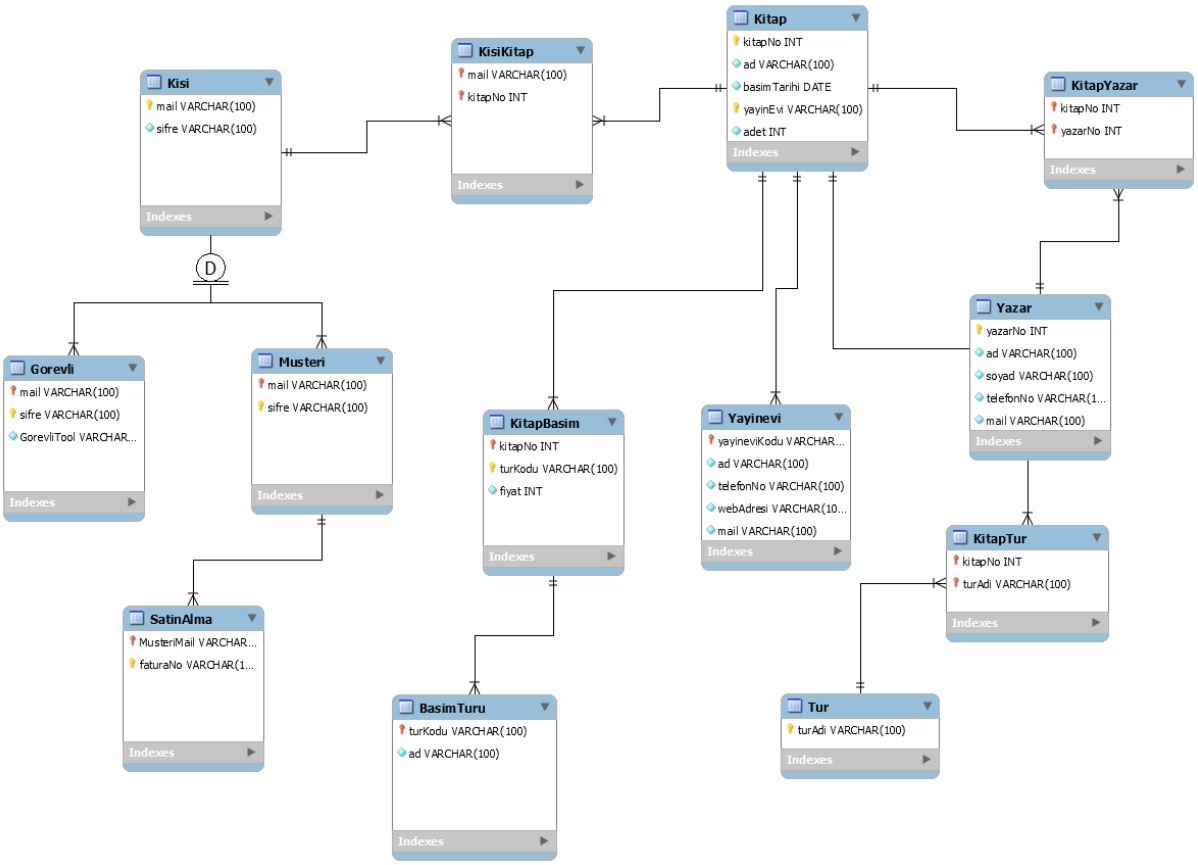
10- Kisi(**mail:varchar**, sifre:varchar)

11- Gorevli(**mail:varchar**, **sifre:varchar**, gorevliTool:varchar)

12- Musteri(**mail:varchar**, **sifre:varchar**)

13- SatinAlma(**MusteriMail:varchar**, **faturaNo:integer**)

Varlık Bağıntı Modeli:



Sql İfadeleri:

--

-- PostgreSQL database dump

--

-- Dumped from database version 13.1

-- Dumped by pg_dump version 13.1

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: girisgorevlikontrol(character varying, character varying); Type:
FUNCTION; Schema: public; Owner: postgres

--


CREATE FUNCTION public.girisgorevlikontrol(_mail character varying, _sifre
character varying) RETURNS integer
   LANGUAGE plpgsql
   AS $$
BEGIN
   SELECT * FROM "Gorevli" where mail = _mail and sifre = _sifre;


   if find then
      return 1;
   else
      return 0;
   end if;
END;
$$;
```

```
ALTER FUNCTION public.girisgorevlikontrol(_mail character varying, _sifre
character varying) OWNER TO postgres;


--

-- Name: giriskontrol(character varying, character varying); Type: FUNCTION;
Schema: public; Owner: postgres

--


CREATE FUNCTION public.giriskontrol(_mail character varying, _sifre character
varying) RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    if (SELECT * FROM "Gorevli" where mail = _mail and sifre = _sifre)>0 then
        return 1;
    else
        return 0;
    end if;
END;
$$;



ALTER FUNCTION public.giriskontrol(_mail character varying, _sifre character
varying) OWNER TO postgres;


--
```

```
-- Name: urunara(integer, character varying, date, character varying, integer); Type: FUNCTION; Schema: public; Owner: postgres

--


CREATE FUNCTION public.urunara(_kitapno integer, _ad character varying, _basimtarihi date, _yayinevi character varying, _adet integer) RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    if (SELECT * FROM "Gorevli" where kitapNo = _kitapNo and ad = _ad and basimTarihi = _basimTarihi and yayinevi = _yayinevi and adet = _adet)>0 then
        return 1;
    else
        return 0;
    end if;
END;
$$;


ALTER FUNCTION public.urunara(_kitapno integer, _ad character varying, _basimtarihi date, _yayinevi character varying, _adet integer) OWNER TO postgres;


--

-- Name: urunekle(integer, character varying, date, character varying, integer); Type: FUNCTION; Schema: public; Owner: postgres

--
```

```
CREATE FUNCTION public.urunekle(_kitapno integer, _ad character varying,
_basimtarihi date, _yayinevi character varying, _adet integer) RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    INSERT into Kitap(kitapNo, ad, basimTarihi, yayinevi, adet)
    VALUES(_kitapNo, _ad, _basimTarihi, _yayinevi, _adet);
    if found then
        return 1;
    else
        return 0;
    end if;
END;
$$;


ALTER FUNCTION public.urunekle(_kitapno integer, _ad character varying,
_basimtarihi date, _yayinevi character varying, _adet integer) OWNER TO
postgres;


--
-- Name: urunsil(integer, character varying, date, character varying, integer);
Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.urunsil(_kitapno integer, _ad character varying,
_basimtarihi date, _yayinevi character varying, _adet integer) RETURNS integer
    LANGUAGE plpgsql
```

```
    AS $$
BEGIN
    DELETE from Kitap WHERE kitapNo = _kitapNo and ad = _ad and basimTarihi
= _basimTarihi and yayinevi = _yayinevi and adet = _adet;
    if found then
        return 1;
    else
        return 0;
    end if;
END;
$$;



ALTER FUNCTION public.urunsil(_kitapno integer, _ad character varying,
_basimtarihi date, _yayinevi character varying, _adet integer) OWNER TO
postgres;


SET default_tablespace = '';


SET default_table_access_method = heap;


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


CREATE TABLE public."BasimTuru" (
    "turKodu" character varying(100) NOT NULL,
```

```
    ad character varying(100) NOT NULL
);


ALTER TABLE public."BasimTuru" OWNER TO postgres;

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

CREATE TABLE public."Gorevli" (
    mail character varying(100) NOT NULL,
    sifre character varying(100) NOT NULL,
    "gorevliTool" character varying(100) NOT NULL
);


ALTER TABLE public."Gorevli" OWNER TO postgres;

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

CREATE TABLE public."Kisi" (
    mail character varying(100) NOT NULL,
    sifre character varying(100) NOT NULL
```

```
);


ALTER TABLE public."Kisi" OWNER TO postgres;

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

CREATE TABLE public."KisiKitap" (
    mail character varying(100) NOT NULL,
    "kitapNo" integer NOT NULL
);


ALTER TABLE public."KisiKitap" OWNER TO postgres;

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

CREATE TABLE public."Kitap" (
    "kitapNo" integer NOT NULL,
    ad character varying(100) NOT NULL,
    "basimTarihi" date NOT NULL,
    yayinevi character varying(100) NOT NULL,
```

```
    adet integer NOT NULL
);


ALTER TABLE public."Kitap" OWNER TO postgres;

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

CREATE TABLE public."KitapBasim" (
    "kitapNo" integer NOT NULL,
    "turKodu" character varying(100) NOT NULL,
    fiyat numeric NOT NULL
);


ALTER TABLE public."KitapBasim" OWNER TO postgres;

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

CREATE TABLE public."KitapTur" (
    "kitapNo" integer NOT NULL,
    "turAdi" character varying(100) NOT NULL
```

```
);


ALTER TABLE public."KitapTur" OWNER TO postgres;

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

CREATE TABLE public."KitapYazar" (
    "kitapNo" integer NOT NULL,
    "yazarNo" integer NOT NULL
);


ALTER TABLE public."KitapYazar" OWNER TO postgres;

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

CREATE TABLE public."Musteri" (
    mail character varying(100) NOT NULL,
    sifre character varying(100) NOT NULL
);
```

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

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

CREATE TABLE public."SatinAlma" (
    "MusteriMail" character varying(100) NOT NULL,
    "faturaNo" character varying(100) NOT NULL
);


ALTER TABLE public."SatinAlma" OWNER TO postgres;

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

CREATE TABLE public."Tur" (
    "turAdi" character varying(100) NOT NULL
);


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

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

CREATE TABLE public."Yayinevi" (
    "yayineviKodu" character varying(100) NOT NULL,
    ad character varying(100) NOT NULL,
    "telefonNo" character varying(100) NOT NULL,
    "webAdresi" character varying(100) NOT NULL,
    mail character varying(100) NOT NULL
);


ALTER TABLE public."Yayinevi" OWNER TO postgres;

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

CREATE TABLE public."Yazar" (
    "yazarNo" integer NOT NULL,
    ad character varying(100) NOT NULL,
    soyad character varying(100) NOT NULL,
    "telefonNo" character varying(100) NOT NULL,
    mail character varying(100) NOT NULL
);
```

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


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




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

INSERT INTO public."Gorevli" VALUES
	('durusari@gmail.com', '12345', 'Urun ekle-sil');



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

INSERT INTO public."Kisi" VALUES
	('mervetopal@gmail.com', '1234'),
	('durusari@gmail.com', '12345');
```

--

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

--

--

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

--

INSERT INTO public."Kitap" VALUES

(151, 'Kurk Mantolu Madonna', '1940-12-18', 'Yapi Kredi', 20);

--

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

--

--

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

--

--

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

--

--

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

--

INSERT INTO public."Musteri" VALUES

('mervetopal@gmail.com', '1234');

--

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

--

--

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

--

--

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

--

--

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

--

--

-- Name: BasimTuru BasimTuru_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public."BasimTuru"
    ADD CONSTRAINT "BasimTuru_pkey" PRIMARY KEY ("turKodu");
```

--

```
-- Name: Gorevli Gorevli_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."Gorevli"
    ADD CONSTRAINT "Gorevli_pkey" PRIMARY KEY (mail, sifre);



--
-- Name: KisiKitap KisiKitap_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."KisiKitap"
    ADD CONSTRAINT "KisiKitap_pkey" PRIMARY KEY (mail, "kitapNo");



--
-- Name: Kisi Kisi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Kisi"
    ADD CONSTRAINT "Kisi_pkey" PRIMARY KEY (mail);



--
```

-- Name: KitapBasim KitapBasim_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."KitapBasim"

    ADD CONSTRAINT "KitapBasim_pkey" PRIMARY KEY ("kitapNo", "turKodu");

--

-- Name: KitapTur KitapTur_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."KitapTur"

    ADD CONSTRAINT "KitapTur_pkey" PRIMARY KEY ("kitapNo", "turAdi");

--

-- Name: KitapYazar KitapYazar_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."KitapYazar"

    ADD CONSTRAINT "KitapYazar_pkey" PRIMARY KEY ("kitapNo", "yazarNo");

--

```
-- Name: Kitap Kitap_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres

--


ALTER TABLE ONLY public."Kitap"

    ADD CONSTRAINT "Kitap_pkey" PRIMARY KEY ("kitapNo");



--

-- Name: Musteri Musteri_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres

--


ALTER TABLE ONLY public."Musteri"

    ADD CONSTRAINT "Musteri_pkey" PRIMARY KEY (mail, sifre);



--

-- Name: SatinAlma SatinAlma_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres

--


ALTER TABLE ONLY public."SatinAlma"

    ADD CONSTRAINT "SatinAlma_pkey" PRIMARY KEY ("MusteriMail",
"faturaNo");



--
```

```
-- Name: Tur Tur_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tur"
    ADD CONSTRAINT "Tur_pkey" PRIMARY KEY ("turAdi");



--
-- Name: Yayinevi Yayinevi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Yayinevi"
    ADD CONSTRAINT "Yayinevi_pkey" PRIMARY KEY ("yayineviKodu");



--
-- Name: Yazar Yazar_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Yazar"
    ADD CONSTRAINT "Yazar_pkey" PRIMARY KEY ("yazarNo");



--
```

```
-- Name: Kisi unique_Kisi_sifre; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Kisi"
    ADD CONSTRAINT "unique_Kisi_sifre" UNIQUE (sifre);



--
-- Name: KitapBasim unique_KitapBasim_turKodu; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."KitapBasim"
    ADD CONSTRAINT "unique_KitapBasim_turKodu" UNIQUE ("turKodu");



--
-- Name: Kitap unique_Kitap_yayinevi; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Kitap"
    ADD CONSTRAINT "unique_Kitap_yayinevi" UNIQUE (yayinevi);



--
```

-- Name: Musteri unique_Musteri_mail; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Musteri"

    ADD CONSTRAINT "unique_Musteri_mail" UNIQUE (mail);

--

-- Name: Gorevli lnk_Kisi_Gorevli; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Gorevli"

    ADD CONSTRAINT "lnk_Kisi_Gorevli" FOREIGN KEY (mail) REFERENCES public."Kisi"(mail) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: KisiKitap lnk_Kisi_KisiKitap; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."KisiKitap"

    ADD CONSTRAINT "lnk_Kisi_KisiKitap" FOREIGN KEY (mail) REFERENCES public."Kisi"(mail) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Musteri lnk_Kisi_Musteri; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Musteri"

    ADD CONSTRAINT "lnk_Kisi_Musteri" FOREIGN KEY (mail) REFERENCES public."Kisi"(mail) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: BasimTuru lnk_KitapBasim_BasimTuru; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."BasimTuru"

    ADD CONSTRAINT "lnk_KitapBasim_BasimTuru" FOREIGN KEY ("turKodu") REFERENCES public."KitapBasim"("turKodu") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: KisiKitap lnk_Kitap_KisiKitap; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."KisiKitap"

```
    ADD CONSTRAINT "lnk_Kitap_KisiKitap" FOREIGN KEY ("kitapNo")
REFERENCES public."Kitap"("kitapNo") MATCH FULL ON UPDATE CASCADE ON
DELETE CASCADE;




--

-- Name: KitapBasim lnk_Kitap_KitapBasim; Type: FK CONSTRAINT; Schema:
public; Owner: postgres

--


ALTER TABLE ONLY public."KitapBasim"

    ADD CONSTRAINT "lnk_Kitap_KitapBasim" FOREIGN KEY ("kitapNo")
REFERENCES public."Kitap"("kitapNo") MATCH FULL ON UPDATE CASCADE ON
DELETE CASCADE;




--

-- Name: KitapTur lnk_Kitap_KitapTur; Type: FK CONSTRAINT; Schema: public;
Owner: postgres

--


ALTER TABLE ONLY public."KitapTur"

    ADD CONSTRAINT "lnk_Kitap_KitapTur" FOREIGN KEY ("kitapNo")
REFERENCES public."Kitap"("kitapNo") MATCH FULL ON UPDATE CASCADE ON
DELETE CASCADE;




--
```

-- Name: KitapYazar lnk_Kitap_KitapYazar; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."KitapYazar"

    ADD CONSTRAINT "lnk_Kitap_KitapYazar" FOREIGN KEY ("kitapNo") REFERENCES public."Kitap"("kitapNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Yayinevi lnk_Kitap_Yayinevi; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Yayinevi"

    ADD CONSTRAINT "lnk_Kitap_Yayinevi" FOREIGN KEY ("yayineviKodu") REFERENCES public."Kitap"(yayinevi) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: SatinAlma lnk_Musteri_SatinAlma; Type: FK CONSTRAINT; Schema: public; Owner: postgres

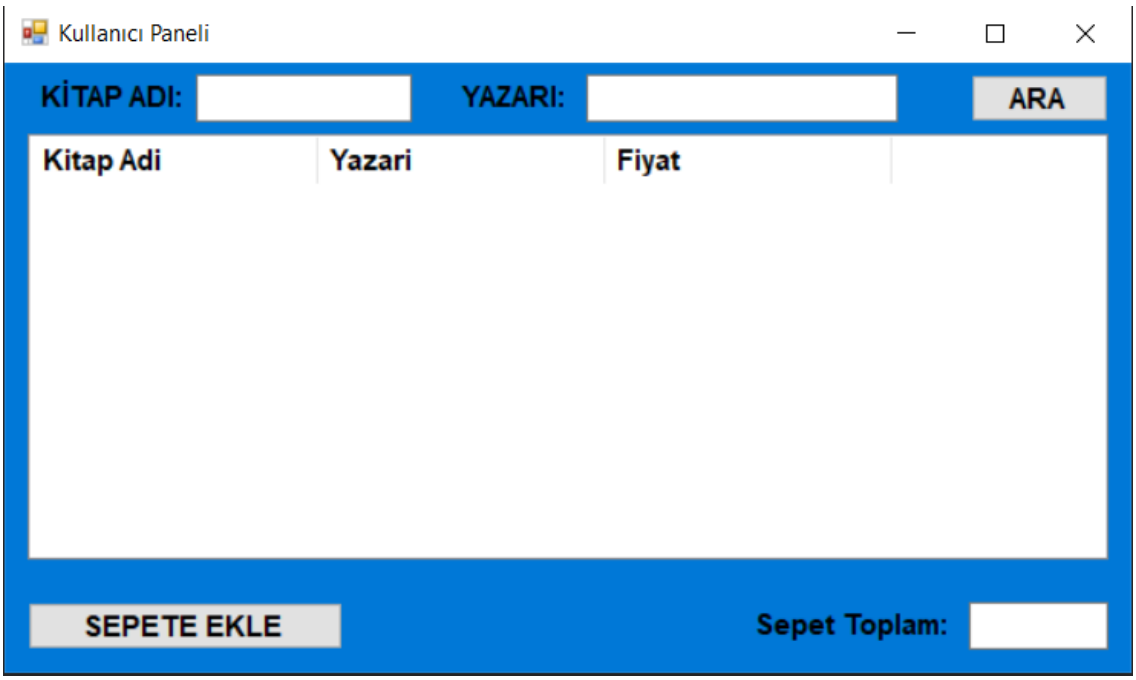
--

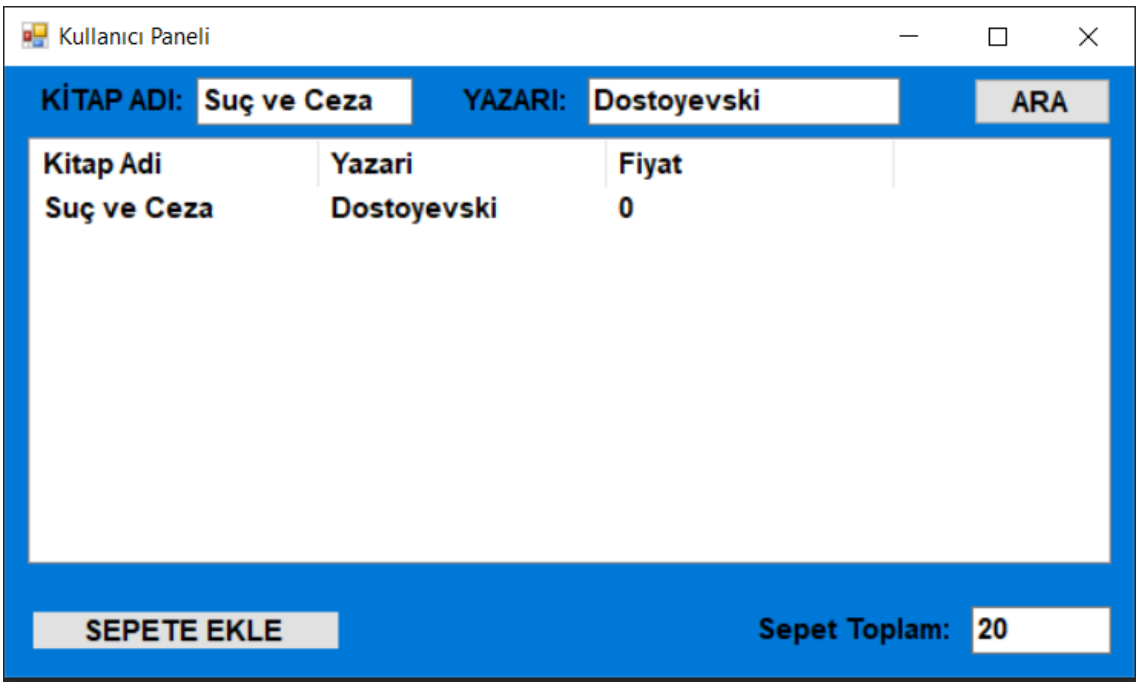
ALTER TABLE ONLY public."SatinAlma"

```
    ADD CONSTRAINT "lnk_Musteri_SatinAlma" FOREIGN KEY ("MusteriMail")
REFERENCES public."Musteri"(mail) MATCH FULL ON UPDATE CASCADE ON
DELETE CASCADE;



--

-- Name: KitapTur lnk_Tur_KitapTur; Type: FK CONSTRAINT; Schema: public;
Owner: postgres

--


ALTER TABLE ONLY public."KitapTur"

    ADD CONSTRAINT "lnk_Tur_KitapTur" FOREIGN KEY ("turAdi") REFERENCES
public."Tur"("turAdi") MATCH FULL ON UPDATE CASCADE ON DELETE
CASCADE;



--

-- Name: KitapYazar lnk_Yazar_KitapYazar; Type: FK CONSTRAINT; Schema:
public; Owner: postgres

--


ALTER TABLE ONLY public."KitapYazar"

    ADD CONSTRAINT "lnk_Yazar_KitapYazar" FOREIGN KEY ("yazarNo")
REFERENCES public."Yazar"("yazarNo") MATCH FULL ON UPDATE CASCADE ON
DELETE CASCADE;



--

-- PostgreSQL database dump complete
```

--

Uygulamaya Ait Ekran Görüntüleri:





Uygulamanın Bulunduğu Github Adresi:

https://github.com/izzetilker/DBMS

Çalışma Videosunun Bulunduğu Adres:

https://www.youtube.com/watch?v=TBW4t9C3Gw8