



SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİTABANI YÖNETİM SİSTEMLERİ DERSİ
PROJE ÖDEVİ

AD SOYAD.....EKREM SELÇUK ÇELİK
NUMARA.....: G211210587
E-POSTA.....ekrem.celik4@ogr.sakarya.edu.tr

Senaryo:

Film bilgilerini tutan bir veri tabanı tasarlanması istenmektedir. Kullanıcıların aradıkları filmlerle ilgili bilgilere kolayca ulaşabilmesi beklenmektedir.

İş Kuralları:

- Her kişinin bir ID'si adı ve şifresi vardır.
- Her kullanıcının kayıt tarihi ve son giriş tarihi kaydedilir.
- Her yöneticinin bir yönetici ID'si vardır
- Kullanıcı silme işlemi bir numara ile kayıt altına alınır. Silinen kullanıcıID'si ve silen yöneticinin ID'si kayıt edilir.
- Her filmin bir türü vardır, her türün bir idsi ve adı vardır.
- Her kullanıcı bir ya da birden fazla film arayabilir. Bir kullanıcı hiç film aramayabilir de.
- Her kullanıcının bir ya da birden fazla favorisi olabilir, ya da hiç favorisi olmayabilir.
- Her kullanıcı bir ya da birden fazla yorum yapabilir ya da hiç yorum yapmayabilir
- Her filmin bir film türü vardır. Bir film türüne ait birden fazla film olabilir
- Her film bir ya da birden fazla kez oynanabilir ya da hiç oynanmayabilir de.
- Oylama işlemi bir id ve içerdiği değerle birlikte kayıt altına alınmalıdır.
- Oylama işleminde bir oy veren kullanıcı bir de oy verdiği film vardır.

İlişkisel Şema:

- Kisi(kisiID:int, ad:string, sifre:string)
- Yonetici(yoneticiID:int, silinenKullaniciSayisi:int)
- Kullanici(kullaniciID:int, kayitTarihi:date, sonGirisTarihi:date)
- Film(filmID:int, filmKonu:varchar, filmAd:string, vizyonTarihi:date)
- FilmDurum(filmDurumAd:string, filmID:int)
- Tur(turID:int, turAd:string)
- FilmTur(filmTurID:int, filmID:int, turID:int)
- FilmArama(filmAramaID:int, filmID:int, kullaniciID:int)
- Favori(favoriID:int, filmID:int, kullaniciID:int)
- Oylama(oylamaID:int, oylamaDeger:int, filmID:int, kullaniciID:int)
- Yorum(yorumID:int, yorumIcerik:varchar, filmID:int, kullaniciID:int)
- KullaniciSil(kullaniciSilID:int, yoneticiID:int, kullaniciID:int)

```

--
-- PostgreSQL database dump
--

-- Dumped from database version 15.1
-- Dumped by pg_dump version 15rc2

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: film_ara(character varying); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.film_ara(_filmad character varying) RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    if(SELECT count(*) from "film" where "filmad"=_filmad)> 0 THEN
        return 1;
    else
        return 0;
    end if;
end
$$;

ALTER FUNCTION public.film_ara(_filmad character varying) OWNER TO postgres;

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

CREATE FUNCTION public.film_ekle(_filmad character varying, _filmid integer)
RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    INSERT INTO film(filmad, filmid)
    VALUES(_filmad, _filmid);
    if found then
        return 1;

```

```

        else
            return 0;
        end if;
    end
    $$;

ALTER FUNCTION public.film_ekle(_filmad character varying, _filmid integer)
OWNER TO postgres;

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

CREATE FUNCTION public.filmsayma() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
begin
update toplamfilm set sayi=sayi+1;
return new;
end;
$$;

ALTER FUNCTION public.filmsayma() OWNER TO postgres;

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

CREATE FUNCTION public.filmsilme() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
begin update toplamfilmsilme set silinenfilm=silinenfilm-1;
return new;
end;
$$;

ALTER FUNCTION public.filmsilme() OWNER TO postgres;

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

CREATE FUNCTION public.kisi_sil(_kisiid integer) RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    DELETE FROM "kisi" WHERE "kisiID" = _kisiID;
    if found THEN
        return 1;
    else

```

```

        return 0;
    end if;
end
$$;

ALTER FUNCTION public.kisi_sil(_kisiid integer) OWNER TO postgres;

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

CREATE FUNCTION public.kisi_sil(_kisiad character varying) RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    DELETE FROM "kisi" WHERE "kisiAd" = _kisiAd;
    if found THEN
        return 1;
    else
        return 0;
    end if;
end
$$;

ALTER FUNCTION public.kisi_sil(_kisiad character varying) OWNER TO postgres;

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

CREATE FUNCTION public.kisisayma() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
begin
update toplamkisi set kisisayisi=kisisayisi+1;
return new;
end;
$$;

ALTER FUNCTION public.kisisayma() OWNER TO postgres;

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

CREATE FUNCTION public.oylama_hesapla(_filmid integer) RETURNS integer
    LANGUAGE plpgsql
    AS $$

```

```

DECLARE
    miktar INTEGER;
BEGIN
    miktar := (SELECT SUM(oylamaDeger) FROM "oylama" WHERE "filmID" = _filmID);
    DELETE FROM kisi WHERE "kisiID" = _kisiID;
    if found THEN
        return miktar;
    else
        return 0;
    end if;
end
$$;

ALTER FUNCTION public.oylama_hesapla(_filmid integer) OWNER TO postgres;

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

CREATE FUNCTION public.silinensayma() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
BEGIN
    UPDATE TOPLAM SET silinensayi=silinensayi-1;
RETURN NEW;
END;
$$;

ALTER FUNCTION public.silinensayma() OWNER TO postgres;

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

CREATE FUNCTION public.u_giris(_username character varying, _password character
varying) RETURNS integer
    LANGUAGE plpgsql
    AS $$
BEGIN
    if(SELECT count(*) from "kisi" where "ad"=_username and "sifre"
=_password)> 0 THEN
        return 1;
    else
        return 0;
    end if;
end
$$;

```

```
ALTER FUNCTION public.u_giris(_username character varying, _password character varying) OWNER TO postgres;
```

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

```
SET default_table_access_method = heap;
```

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

```
CREATE TABLE public.favori (  
    "favoriID" integer NOT NULL,  
    "filmID" integer NOT NULL,  
    "kullaniciID" integer NOT NULL  
);
```

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

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

```
CREATE TABLE public.film (  
    filmid integer NOT NULL,  
    "filmKonu" text,  
    filmad character varying(2044) NOT NULL,  
    "vizyonTarihi" text  
);
```

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

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

```
CREATE TABLE public."filmArama" (  
    "filmAramaID" integer NOT NULL,  
    "filmID" integer NOT NULL,  
    "kullaniciID" integer NOT NULL  
);
```

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

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

```
CREATE TABLE public."filmDurum" (  
    "filmDurumAd" character varying(2044) NOT NULL,
```

```
        "filmID" integer NOT NULL
    );

ALTER TABLE public."filmDurum" OWNER TO postgres;

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

CREATE TABLE public."filmTur" (
    "filmTurID" integer NOT NULL,
    "filmID" integer NOT NULL,
    "turID" integer NOT NULL
);

ALTER TABLE public."filmTur" OWNER TO postgres;

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

CREATE TABLE public.kisi (
    "kisiID" integer NOT NULL,
    ad character varying(2044) NOT NULL,
    sifre character varying(2044) NOT NULL
);

ALTER TABLE public.kisi OWNER TO postgres;

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

CREATE TABLE public.kullanici (
    "kullaniciID" integer NOT NULL,
    "kayitTarihi" date NOT NULL,
    "sonGirisTarihi" date NOT NULL
);

ALTER TABLE public.kullanici OWNER TO postgres;

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

CREATE TABLE public."kullaniciSil" (
    "kullaniciSilNo" integer NOT NULL,
    "yoneticiID" integer NOT NULL,
    "kullaniciID" integer NOT NULL
```



```

);

ALTER TABLE public."kullaniciSil" OWNER TO postgres;

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

CREATE TABLE public.oylama (
    "oylamaID" integer NOT NULL,
    "oylamaDeger" integer NOT NULL,
    "filmID" integer NOT NULL,
    "kullaniciID" integer NOT NULL
);

ALTER TABLE public.oylama OWNER TO postgres;

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

CREATE TABLE public.toplamfilm (
    sayi integer
);

ALTER TABLE public.toplamfilm OWNER TO postgres;

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

CREATE TABLE public.toplamfilmsilme (
    silinenfilm integer
);

ALTER TABLE public.toplamfilmsilme OWNER TO postgres;

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

CREATE TABLE public.toplamkisi (
    kisisayisi integer
);

ALTER TABLE public.toplamkisi OWNER TO postgres;

--

```

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

CREATE TABLE public.toplamsilinen (
    silinen integer
);

ALTER TABLE public.toplamsilinen OWNER TO postgres;

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

CREATE TABLE public.tur (
    "turID" integer NOT NULL,
    "turAd" character varying(2044) NOT NULL
);

ALTER TABLE public.tur OWNER TO postgres;

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

CREATE TABLE public.yonetici (
    "yoneticiNo" integer NOT NULL,
    "silinenKullanıcıSayisi" integer NOT NULL
);

ALTER TABLE public.yonetici OWNER TO postgres;

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

CREATE TABLE public.yorum (
    "yorumID" integer NOT NULL,
    "yorumIcerik" character varying(2044) NOT NULL,
    "filmID" integer NOT NULL,
    "kullaniciID" integer NOT NULL
);

ALTER TABLE public.yorum OWNER TO postgres;

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

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

INSERT INTO public.film VALUES
    (4, 'Zekâ seviyesi, klasik devlet okuluna bile kabul edilmeyecek kadar düşük olan ve fiziksel olarak da sorunları olan bir çocuğun, hayatın ona getirdiği acı ve tatlı sürprizlerden, inanılmaz başarılar elde etmesini konu alan kitaptan uyarlanmış bir filmidir.', 'Forrest Gump', NULL),
    (5, 'Sevginin Gücü, konuşamayan küçük bir kız ile yolları kesişen bir adamın hikayesini anlatıyor. ', 'Sevginin Gücü', NULL),
    (2, 'New York''ta yaşayan güçlü bir İtalyan mafya ailesinin hikâyesini anlatır.', 'Godfather ', NULL),
    (1, 'Ege''deki çiftlikten gazetecilik okumak için ayrılan Sadık''ın, yıllar sonra oğluyla beraber yeniden çiftliğe dönüşünün, 12 Eylül Darbesi arka planında aktarıldığı filmin senaryosunu da yine Çağan Irmak yazdı.', 'Babam ve Oglum', NULL),
    (3, 'Sokakta bulduğu bir cüzdanın Antalyalı çok önemli bir iş adamına ait olduğunu öğrenince güneye doğru yola koyulur. Yol boyunca birbirinden komik sürprizlerle karşılaşan Recep sonunda Antalya''ya varır ve cüzdanı turizmci Muhsin Bey''e teslim eder.', 'Recep Ivedik', NULL),
    (8, NULL, 'Godfather 2', NULL),
    (7, NULL, 'Godfather 3', NULL),
    (6, NULL, 'Recep Ivedik 2', NULL);

--

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

--

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

--

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

--

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

--

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

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

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

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

INSERT INTO public.toplamfilm VALUES
    (NULL),
    (NULL),
    (NULL),
    (8);

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

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

INSERT INTO public.toplamkisi VALUES
    (1);

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

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

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

--

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

--

-- Name: favori favori_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.favori
    ADD CONSTRAINT favori_pkey PRIMARY KEY ("favoriID");

--

-- Name: filmArama filmArama_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."filmArama"
    ADD CONSTRAINT "filmArama_pkey" PRIMARY KEY ("filmAramaID");

--

-- Name: filmDurum filmDurum_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."filmDurum"
    ADD CONSTRAINT "filmDurum_pkey" PRIMARY KEY ("filmDurumAd");

--

-- Name: filmTur filmTur_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."filmTur"
    ADD CONSTRAINT "filmTur_pkey" PRIMARY KEY ("filmTurID");

--

-- Name: film film_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--
```

```
ALTER TABLE ONLY public.film
    ADD CONSTRAINT film_pkey PRIMARY KEY (filmid);

--
-- Name: kisi kisi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.kisi
    ADD CONSTRAINT kisi_pkey PRIMARY KEY ("kisiID");

--
-- Name: kullaniciSil kullaniciSil_pkey; Type: CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."kullaniciSil"
    ADD CONSTRAINT "kullaniciSil_pkey" PRIMARY KEY ("kullaniciSilNo");

--
-- Name: kullanici kullanici_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.kullanici
    ADD CONSTRAINT kullanici_pkey PRIMARY KEY ("kullaniciID");

--
-- Name: oylama oylama_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.oylama
    ADD CONSTRAINT oylama_pkey PRIMARY KEY ("oylamaID");

--
-- Name: tur tur_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.tur
    ADD CONSTRAINT tur_pkey PRIMARY KEY ("turID");

--
-- Name: yonetici yonetici_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.yonetici
```

```

    ADD CONSTRAINT yonetici_pkey PRIMARY KEY ("yoneticiNo");

--
-- Name: yorum yorum_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.yorum
    ADD CONSTRAINT yorum_pkey PRIMARY KEY ("yorumID");

--
-- Name: film filmsayma; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER filmsayma AFTER INSERT ON public.film FOR EACH ROW EXECUTE
FUNCTION public.filmsayma();

--
-- Name: kisi kisisaymatrig; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER kisisaymatrig AFTER INSERT ON public.kisi FOR EACH ROW EXECUTE
FUNCTION public.kisisayma();

--
-- Name: film silinenfilmsayma; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER silinenfilmsayma AFTER DELETE ON public.film FOR EACH ROW
EXECUTE FUNCTION public.filmsilme();

--
-- Name: kisi silinentrig; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER silinentrig AFTER DELETE ON public.kisi FOR EACH ROW EXECUTE
FUNCTION public.silinsayma();

--
-- Name: favori lnk_film_favori; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.favori
    ADD CONSTRAINT lnk_film_favori FOREIGN KEY ("filmID") REFERENCES
public.film(filmid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

```

```
--
-- Name: filmArama lnk_film_filmArama; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."filmArama"
    ADD CONSTRAINT "lnk_film_filmArama" FOREIGN KEY ("filmID") REFERENCES
public.film(filmid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: filmDurum lnk_film_filmDurum; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public."filmDurum"
    ADD CONSTRAINT "lnk_film_filmDurum" FOREIGN KEY ("filmID") REFERENCES
public.film(filmid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: filmTur lnk_film_filmTur; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."filmTur"
    ADD CONSTRAINT "lnk_film_filmTur" FOREIGN KEY ("filmID") REFERENCES
public.film(filmid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: oylama lnk_film_oylama; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.oylama
    ADD CONSTRAINT lnk_film_oylama FOREIGN KEY ("filmID") REFERENCES
public.film(filmid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: yorum lnk_film_yorum; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.yorum
    ADD CONSTRAINT lnk_film_yorum FOREIGN KEY ("filmID") REFERENCES
public.film(filmid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: kullanici lnk_kisi_kullanici; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
```



```

--

ALTER TABLE ONLY public.kullanici
    ADD CONSTRAINT lnk_kisi_kullanici FOREIGN KEY ("kullaniciID") REFERENCES
public.kisi("kisiID") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: yonetici lnk_kisi_yonetici; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.yonetici
    ADD CONSTRAINT lnk_kisi_yonetici FOREIGN KEY ("yoneticiNo") REFERENCES
public.kisi("kisiID") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: favori lnk_kullanici_favori; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

ALTER TABLE ONLY public.favori
    ADD CONSTRAINT lnk_kullanici_favori FOREIGN KEY ("kullaniciID") REFERENCES
public.kullanici("kullaniciID") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: filmArama lnk_kullanici_filmArama; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."filmArama"
    ADD CONSTRAINT "lnk_kullanici_filmArama" FOREIGN KEY ("kullaniciID")
REFERENCES public.kullanici("kullaniciID") MATCH FULL ON UPDATE CASCADE ON
DELETE CASCADE;

--
-- Name: kullaniciSil lnk_kullanici_kullaniciSil; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."kullaniciSil"
    ADD CONSTRAINT "lnk_kullanici_kullaniciSil" FOREIGN KEY ("kullaniciSilNo")
REFERENCES public.kullanici("kullaniciID") MATCH FULL ON UPDATE CASCADE ON
DELETE CASCADE;

--
-- Name: oylama lnk_kullanici_oylama; Type: FK CONSTRAINT; Schema: public;
Owner: postgres
--

```

```
ALTER TABLE ONLY public.oylama
    ADD CONSTRAINT lnk_kullanici_oylama FOREIGN KEY ("kullaniciID") REFERENCES
public.kullanici("kullaniciID") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: yorum lnk_kullanici_yorum; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public.yorum
    ADD CONSTRAINT lnk_kullanici_yorum FOREIGN KEY ("kullaniciID") REFERENCES
public.kullanici("kullaniciID") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: filmTur lnk_tur_filmTur; Type: FK CONSTRAINT; Schema: public; Owner:
postgres
--

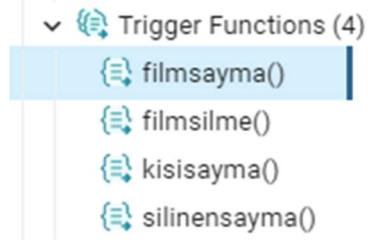
ALTER TABLE ONLY public."filmTur"
    ADD CONSTRAINT "lnk_tur_filmTur" FOREIGN KEY ("turID") REFERENCES
public.tur("turID") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: kullanıcıSil lnk_yonetici_kullaniciSil; Type: FK CONSTRAINT; Schema:
public; Owner: postgres
--

ALTER TABLE ONLY public."kullaniciSil"
    ADD CONSTRAINT "lnk_yonetici_kullaniciSil" FOREIGN KEY ("yoneticiID")
REFERENCES public.yonetici("yoneticiNo") MATCH FULL ON UPDATE CASCADE ON DELETE
CASCADE;

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

Trigger ve Saklı Yordamlar

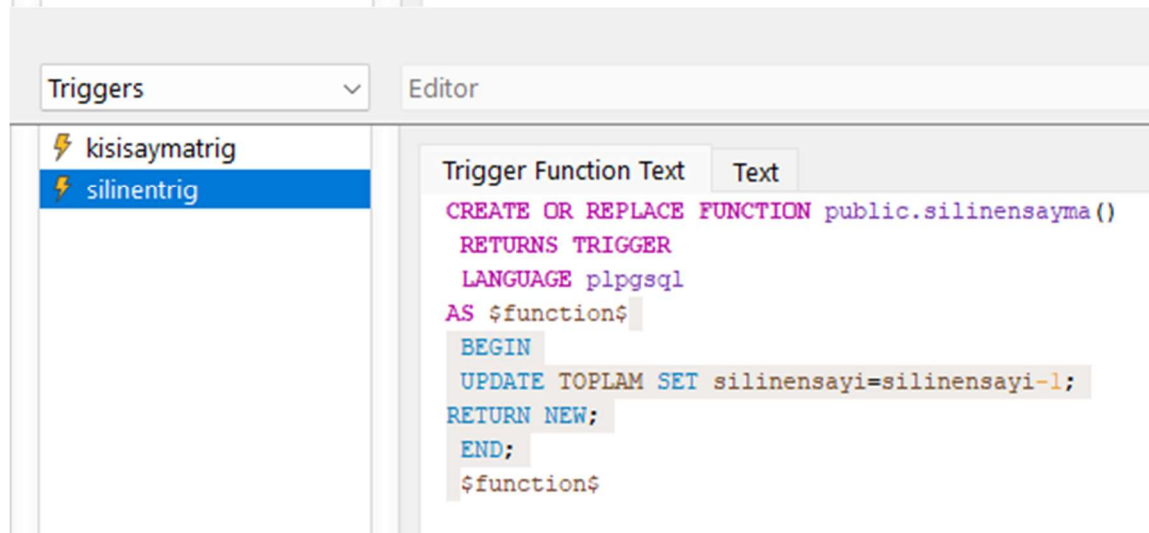
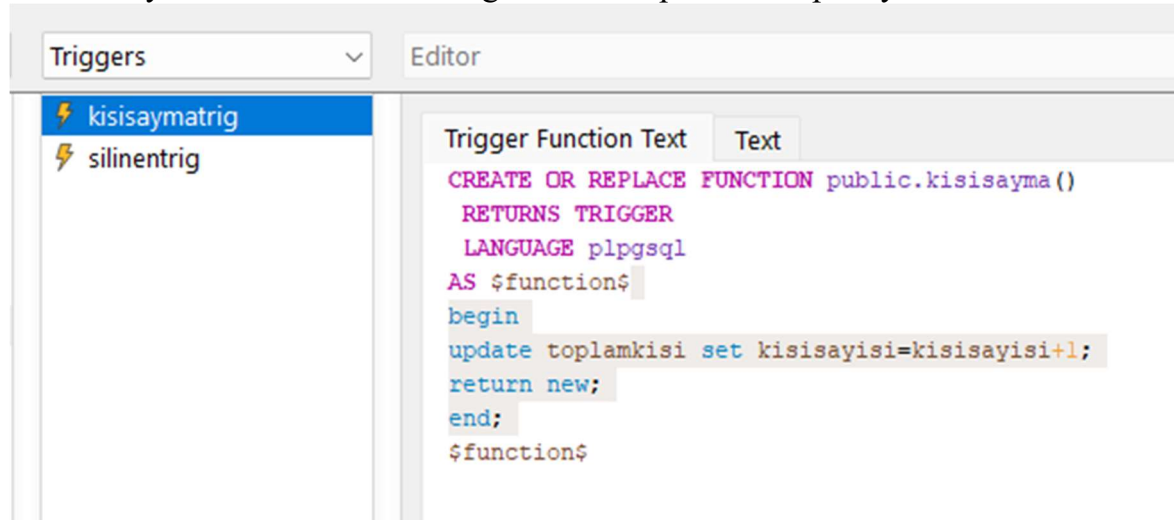


Film sayma eklenen film sayısını arka planda takip ediyor.

Film silme silinen film sayısını arka planda takip ediyor.

Kişi sayma eklenen kullanıcı ve yönetici bilgilerini arka planda takip ediyor.

Silinen sayma silinen kullanıcı bilgilerini arka planda takip ediyor.



Triggers

filmsayma

silinenfilmsayma

Trigger Function Text

Text

```
CREATE OR REPLACE FUNCTION public.filmsayma()  
  RETURNS TRIGGER  
  LANGUAGE plpgsql  
AS $function$  
begin  
update toplamfilm set sayi=sayi+1;  
return new;  
end;  
$function$
```

Triggers

filmsayma

silinenfilmsayma

Trigger Function Text

Text

```
CREATE OR REPLACE FUNCTION public.filmsilme()  
  RETURNS TRIGGER  
  LANGUAGE plpgsql  
AS $function$  
begin update toplamfilmsilme set silinenfilm=silinenfilm-1;  
return new;  
end;  
$function$
```

Functions

film_ara(varchar)

film_ekle(...ar, int4)

filmsayma()

filmsilme()

kisi_sil(int4)

kisi_sil(varchar)

kisisayma()

oylama_hes...la(int4)

silinensayma()

u_giris(v... varchar)

CREATE OR REPLACE FUNCTION public.silinensayma()
 RETURNS TRIGGER
 LANGUAGE plpgsql
AS \$function\$
 BEGIN
 UPDATE TOPLAM SET silinensayi=silinensayi-1;
 RETURN NEW;
 END;
\$function\$

Functions	Editor
<ul style="list-style-type: none"> film_ara(varchar) film_ekle(...ar, int4) filmsayma() filmsilme() kisi_sil(int4) kisi_sil(varchar) kisisayma() oylama_hes...la(int4) silinensayma() u_giris(v... varchar) 	<pre>CREATE OR REPLACE FUNCTION public.kisisayma() RETURNS TRIGGER LANGUAGE plpgsql AS \$function\$ begin update toplamkisi set kisisayisi=kisisayisi+1; return new; end; \$function\$</pre>

Functions	Editor
<ul style="list-style-type: none"> film_ara(varchar) film_ekle(...ar, int4) filmsayma() filmsilme() kisi_sil(int4) kisi_sil(varchar) kisisayma() oylama_hes...la(int4) silinensayma() u_giris(v... varchar) 	<pre>CREATE OR REPLACE FUNCTION public.filmsilme() RETURNS TRIGGER LANGUAGE plpgsql AS \$function\$ begin update toplamfilmsilme set silinenfilm=silinenfilm-1; return new; end; \$function\$</pre>

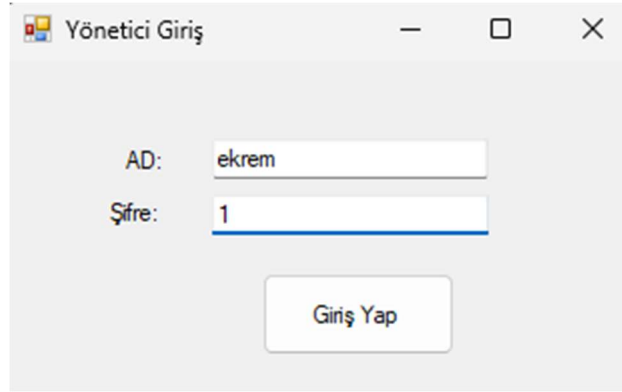
Functions	Editor
<ul style="list-style-type: none"> film_ara(varchar) film_ekle(...ar, int4) filmsayma() filmsilme() kisi_sil(int4) kisi_sil(varchar) kisisayma() oylama_hes...la(int4) silinensayma() u_giris(v... varchar) 	<pre>CREATE OR REPLACE FUNCTION public.filmsayma() RETURNS TRIGGER LANGUAGE plpgsql AS \$function\$ begin update toplamfilm set sayi=sayi+1; return new; end; \$function\$</pre>

Ekran Görüntüleri



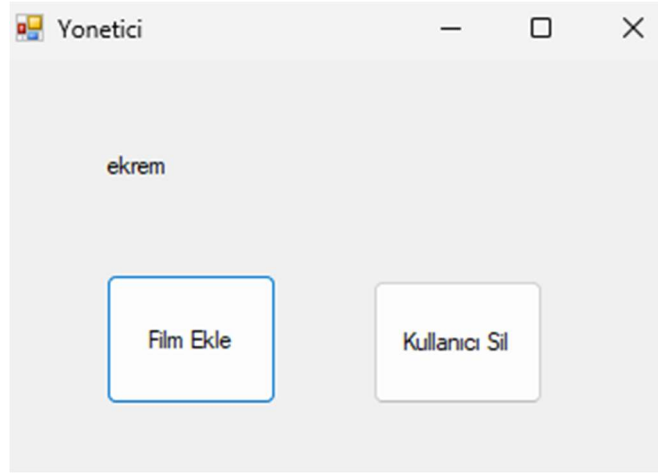
ayırdım.

giriş ekranında yönetici ve kullanıcı olarak



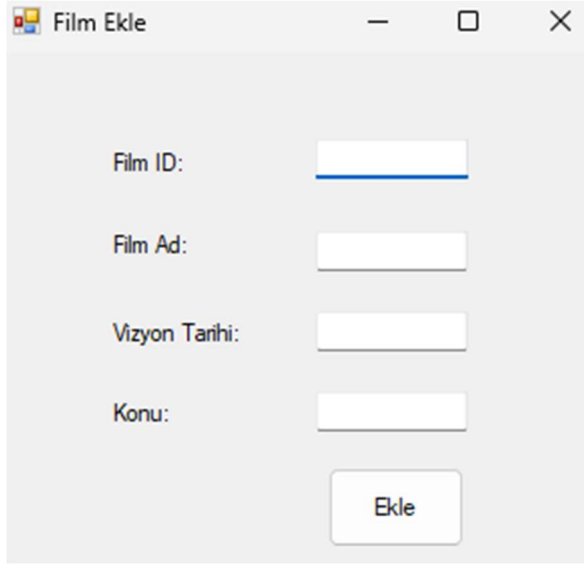
verirsem hata veriyor.

giriş ekranına giriyorum false değer

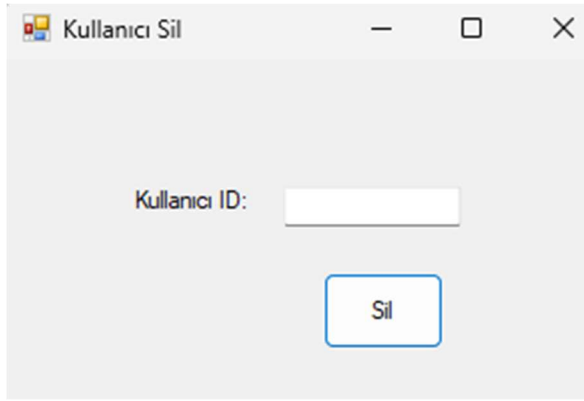


kullanıcı silebiliyorum.

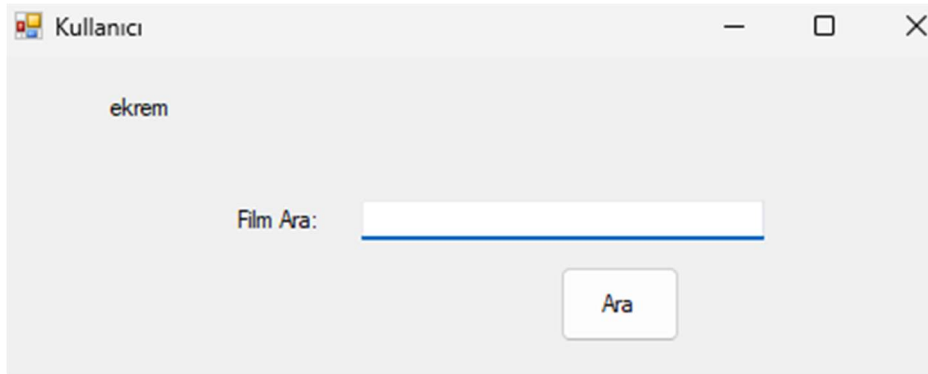
yönetici olarak film ekleyebiliyor ve



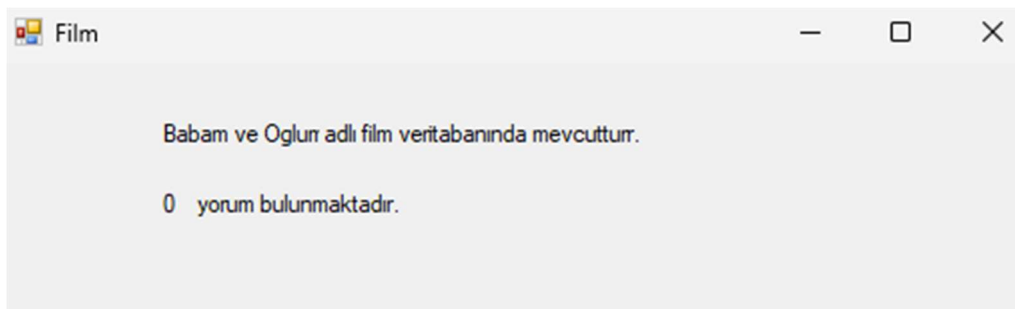
burdan film ekleyebiliyorum.



kullanıcı id sini verip kullanıcıyı
silebiliyorum.



kullanıcı olarak
girdikten sonra film arıyabiliyorum.



doğru değer
döndürürsem çıktı böyle oluyor false girersem reddediyor.

Kaynak kodlar : <https://github.com/esc444/veritabanzartzurt>