

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 oylanabilir ya da hiç oylanmayabilir 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 filmdir.', '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: kullanici; 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.silinensayma();

--

-- 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: kullaniciSil 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

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

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.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

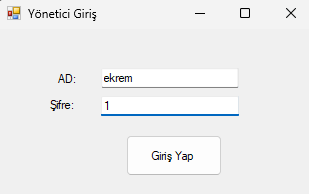
metin içeren bir resim

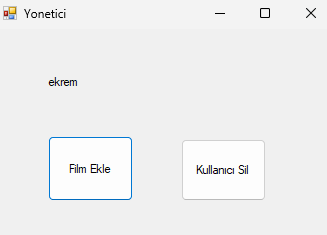
Açıklama otomatik olarak oluşturuldu

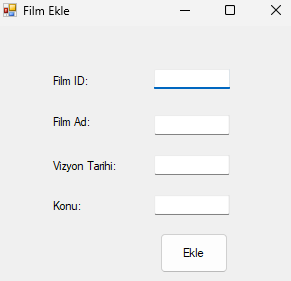
Ekran Görüntüleri

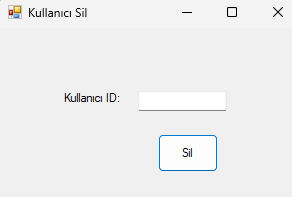
metin içeren bir resim

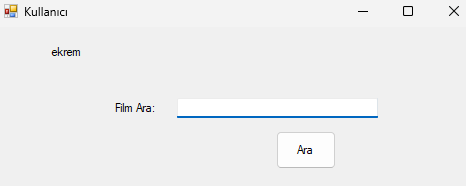
Açıklama otomatik olarak oluşturuldugiriş ekranında yönetici ve kullanıcı olarak ayırdım.

giriş ekranına giriyorum false değer verirsem hata veriyor.

yönetici olarak film ekleyebiliyor ve kullanıcı silebiliyorum.

burdan film ekleyebiliyorum.

kullanıcı id sini verip kullanıcıyı silebiliyorum.

kullanıcı olarak girdikten sonra film arıyabiliyorum.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldudoğru değer döndürürsem cıktı böyle oluyor false girersem reddediyor.

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