****

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTASİ**

**BİLGİSAYAR MÜHANDİSLİĞİ BÖLÜMÜ**

VERİTABANI YÖNETİM SİSTEMLERİ

Öğrenci adı : abdulrahman rashid

Öğrenci No : g201210552

Grup : 2C

**UYGULAMANIN KISA TANITIM**

*Yaptığım uygulama postgresql, c# dilini ve form elemanlarını içeren sinema otomasyonudur. Uygulamayı sadece sinema gişesindeki personelin kullanımı için tasarladım. Bu uygulamada film ekle butonuna tıkladığımızda; vizyondaki filmleri görebiliriz, Vizyondaki filmleri güncelleyebiliriz, vizyona film ekleyip çıkarabiliriz. Film ara butonuna tıkladığımızda vizyondaki film listesindeki filmleri buluruz. Bilet sat butonuna tıkladığımızda ise boş koltuklardan istediğimizi, izlemek istenilen filmi, bilet tarihi ve seans saatini, bilet numarasını belirtip bilet kesebiliyoruz.*

***IŞ KURALLARI***

Programı kullanan kişi sinemada çalışan kasiyerdir.

Kasiyer; film ekleyebilir,

vizyondaki filmleri güncelleyebilir,film silebilir.

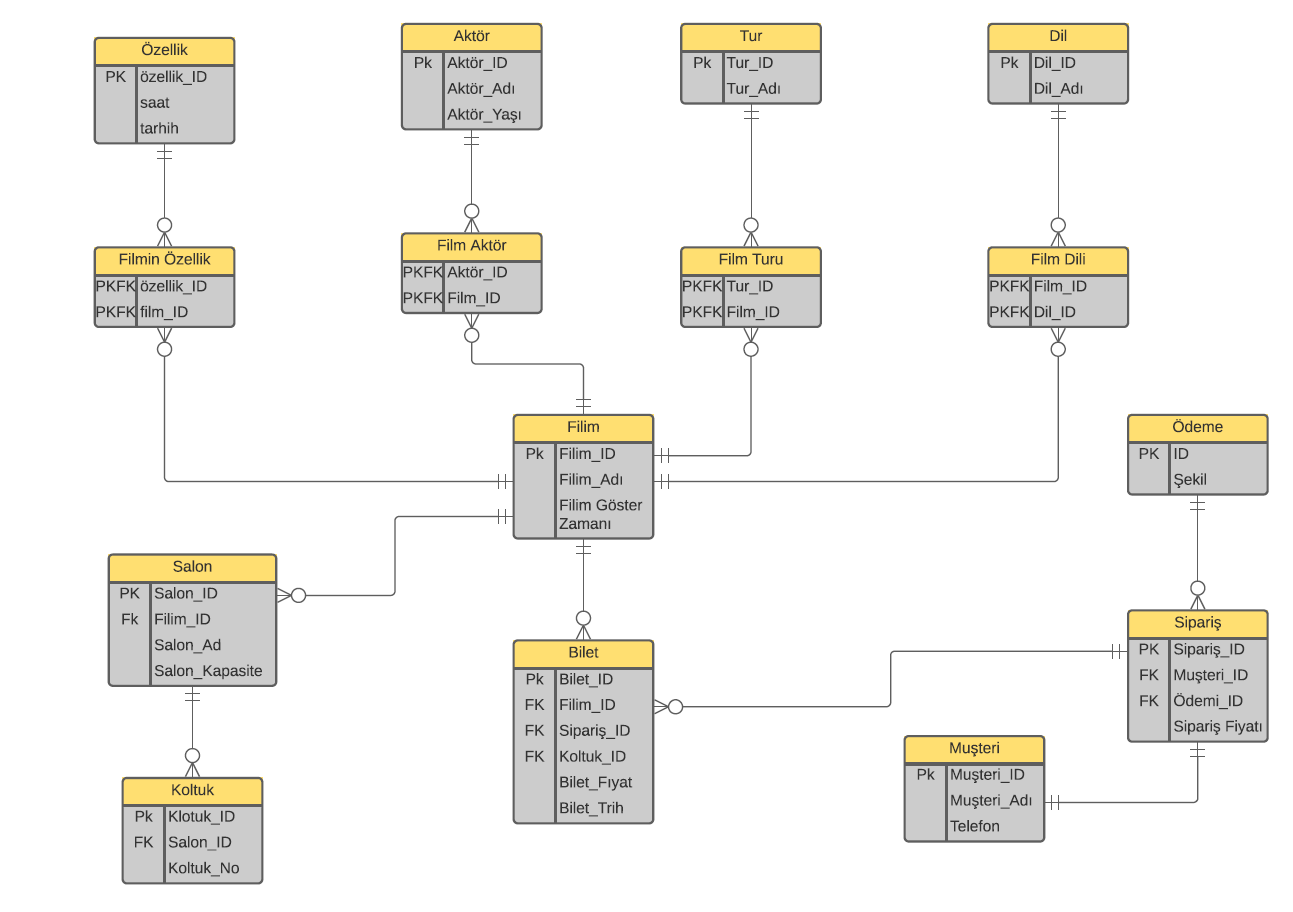
Kasiyer vizyondaki filmleri arayabilir. Bir türe ait birden çok film olabilir.

* Bir filmin birçok türü olabilir.
* Bir aktör birçok filmde olabilir.
* Bir filmde birçok aktör olabilir.
* Bir bilet bir filme aittir.
* Bir birçok bilete aittir.
* Bir filmin birçok salonu olabilir
* Bir salonun bir filmi olabilir.
* Bir salonda birçok koltuk olabilir.
* Bir koltuk bir salonda olabilir.
* Bir sipariş birçok bilet içerebilir.
* Bir bilet bir siparişe aittir.
* Bir ödeme türü birçok siparişin olabilir.
* Bir sipariş bir ödeme türünün olabilir.
* Bir personel birçok sipariş alabilir.
* Bir siparişi bir personel alabilir.
* Bir sipariş birçok menünün olabilir.
* Bir menü bir siparişin olabilir.
* Bir filmin çok dil içerebilir
* Bir dil en az bir film içerisinde olabilir
* Bir film çok sayıda özelliği olabilir
* Bir özellik bir film içerisinde olabilir

***Ilişkisel Şema***

* aktor(aktor\_id:int, aktor\_yasi:int, aktor\_adi:varchar aktor
* film\_aktor(aktor id:int, film id:int)
* tur(tur\_id : int , tur\_yasi: int )
* film\_turu(film id:int, tur id:int)
* dil(dil id : int , dil adi : varchar aktor )
* film\_dili( film\_id : int , dil\_id :int )
* özellik(özellik\_id : int , saat : time ,tarih : data )
* film\_özellik(özellik\_id :int , film\_id :int )
* bilet(bilet\_id:int,film id:int, bilet fiyat:int, bilet\_tarih:date, siparis id:int, koltuk no:text)
* film (film\_id:int, film\_gösterim\_zamanı:date, film\_adi:varchar)
* koltuk(koltuk id:int, koltuk\_no:text, salon id:int)
* müşteri(müşteri\_id:int, müşteri\_tel:text, müşteri\_adi:varchar)
* odeme(odeme\_id:int, odeme\_sekli:varchar)
* salon (salon\_id:int, film id:int, salon\_adi:varchar,salon\_kapasitesi:int)
* siparis(siparis\_id:int, müsteri id:int, odeme id:int,

***varlik baginti diyagrami***



***SQL IVADELARI***

-- PostgreSQL database dump

-- Dumped from database version 14.1

-- Dumped by pg\_dump version 14.0

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;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

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

--

CREATE TABLE public."Aktor" (

"AktorID" integer NOT NULL,

"AktorAdi" character varying,

"AktorYasi" integer

);

ALTER TABLE public."Aktor" OWNER TO postgres;

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

--

CREATE TABLE public."Bilet" (

"BiletID" integer NOT NULL,

"FilmID" integer NOT NULL,

"SiparisID" integer,

"KoltukID" integer NOT NULL,

"Saat" character varying NOT NULL,

"Tarih" character varying

);

ALTER TABLE public."Bilet" OWNER TO postgres;

--

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

--

CREATE TABLE public."Dil" (

"DilID" integer NOT NULL,

"DilAdi" character varying

);

ALTER TABLE public."Dil" OWNER TO postgres;

--

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

--

CREATE TABLE public."Film" (

"FilmID" integer NOT NULL,

"FilmAdi" character varying,

"FilmGosterimTarihi" character varying

);

ALTER TABLE public."Film" OWNER TO postgres;

--

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

--

CREATE TABLE public."FilmAktor" (

"AktorID" integer NOT NULL,

"FilmID" integer NOT NULL

);

ALTER TABLE public."FilmAktor" OWNER TO postgres;

--

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

--

CREATE TABLE public."FilmDili" (

"FilmID" integer NOT NULL,

"dilID" integer NOT NULL

);

ALTER TABLE public."FilmDili" OWNER TO postgres;

--

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

--

CREATE TABLE public."FilmOzellik" (

"OzellikID" integer NOT NULL,

"FilmID" integer NOT NULL

);

ALTER TABLE public."FilmOzellik" OWNER TO postgres;

--

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

--

CREATE TABLE public."FilmTuru" (

"TurID" integer NOT NULL,

"FilmID" integer NOT NULL

);

ALTER TABLE public."FilmTuru" OWNER TO postgres;

--

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

--

CREATE TABLE public."Koltuk" (

"KoltukID" integer NOT NULL,

"SalonID" integer NOT NULL

);

ALTER TABLE public."Koltuk" OWNER TO postgres;

--

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

--

CREATE TABLE public."Musteri" (

"MusteriID" integer NOT NULL,

"MusteriADi" character varying,

"MusteriTELNO" integer,

"MusteriYasi" integer

);

ALTER TABLE public."Musteri" OWNER TO postgres;

--

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

--

CREATE TABLE public."Odeme" (

"OdemeID" integer NOT NULL,

"OdemeSekli" character varying

);

ALTER TABLE public."Odeme" OWNER TO postgres;

--

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

--

CREATE TABLE public."Ozellik" (

"OzellikID" integer NOT NULL,

"Saat" integer,

"Tarih" integer

);

ALTER TABLE public."Ozellik" OWNER TO postgres;

--

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

--

CREATE TABLE public."Salon" (

"SalonID" integer NOT NULL,

"FilmID" integer NOT NULL,

"SalonKapasite" integer

);

ALTER TABLE public."Salon" OWNER TO postgres;

--

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

--

CREATE TABLE public."Siparis" (

"SiparisID" integer NOT NULL,

"MusteriID" integer NOT NULL,

"OdemeID" integer NOT NULL,

"SiparisFiati" integer

);

ALTER TABLE public."Siparis" OWNER TO postgres;

--

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

--

CREATE TABLE public."Tur" (

"TurID" integer NOT NULL,

"TurAdi" character varying

);

ALTER TABLE public."Tur" OWNER TO postgres;

--

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

--

--

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

--

INSERT INTO public."Bilet" VALUES

(1, 1, 1, 2, '10', '12/12/2020');

--

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

--

--

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

--

INSERT INTO public."Film" VALUES

(1, 'recep ivedik', '12/12/2020'),

(2, 'agora', '11/11/2020'),

(3, 'uyanis', '10/10/2020'),

(4, 'teskilat', '09/09/2020'),

(5, 'aykut', '08/08/2020');

--

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

--

--

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

--

--

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

--

--

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

--

--

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

--

INSERT INTO public."Koltuk" VALUES

(1, 1),

(2, 1),

(3, 1),

(4, 1),

(5, 1),

(6, 1),

(7, 1),

(8, 1),

(9, 1),

(10, 1),

(11, 2),

(12, 2),

(13, 2),

(14, 2),

(15, 2),

(16, 2),

(17, 2),

(18, 2),

(19, 2),

(20, 2),

(21, 3),

(22, 3),

(23, 3),

(24, 3),

(25, 3),

(26, 3),

(27, 3),

(28, 3),

(29, 3),

(30, 3);

--

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

--

INSERT INTO public."Musteri" VALUES

(1, 'bilal', 1150801946, 18),

(2, 'marzoq', 1036378485, 22),

(3, 'masod', 1334489322, 24),

(4, 'abdo', 693506995, 25),

(5, 'bura', 292761282, 25);

--

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

--

INSERT INTO public."Odeme" VALUES

(1, 'online'),

(2, 'nakit'),

(3, 'nakit'),

(4, 'nakit'),

(5, 'online');

--

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

--

--

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

--

INSERT INTO public."Salon" VALUES

(1, 1, 100),

(2, 2, 100),

(3, 3, 50);

--

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

--

INSERT INTO public."Siparis" VALUES

(1, 1, 1, 50),

(2, 2, 2, 50),

(3, 3, 3, 50),

(4, 4, 4, 50);

--

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

--

--

-- Name: FilmAktor FilmAktor\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmAktor"

ADD CONSTRAINT "FilmAktor\_pkey" PRIMARY KEY ("AktorID", "FilmID");

--

-- Name: FilmDili FilmDili\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmDili"

ADD CONSTRAINT "FilmDili\_pkey" PRIMARY KEY ("dilID", "FilmID");

--

-- Name: FilmOzellik FilmOzellik\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmOzellik"

ADD CONSTRAINT "FilmOzellik\_pkey" PRIMARY KEY ("OzellikID", "FilmID");

--

-- Name: FilmTuru FilmTuru\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmTuru"

ADD CONSTRAINT "FilmTuru\_pkey" PRIMARY KEY ("TurID", "FilmID");

--

-- Name: Ozellik Ozellik\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Ozellik"

ADD CONSTRAINT "Ozellik\_pkey" PRIMARY KEY ("OzellikID");

--

-- Name: Siparis Siparis\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Siparis"

ADD CONSTRAINT "Siparis\_pkey" PRIMARY KEY ("SiparisID");

--

-- Name: Aktor unique\_Aktor\_AktorID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Aktor"

ADD CONSTRAINT "unique\_Aktor\_AktorID" PRIMARY KEY ("AktorID");

--

-- Name: Bilet unique\_Bilet\_BietID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bilet"

ADD CONSTRAINT "unique\_Bilet\_BietID" PRIMARY KEY ("BiletID");

--

-- Name: Dil unique\_Dil\_DilID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Dil"

ADD CONSTRAINT "unique\_Dil\_DilID" PRIMARY KEY ("DilID");

--

-- Name: Film unique\_Film\_FilmID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Film"

ADD CONSTRAINT "unique\_Film\_FilmID" PRIMARY KEY ("FilmID");

--

-- Name: Koltuk unique\_Koltuk\_KoltukID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Koltuk"

ADD CONSTRAINT "unique\_Koltuk\_KoltukID" PRIMARY KEY ("KoltukID");

--

-- Name: Musteri unique\_Musteri\_MusteriID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Musteri"

ADD CONSTRAINT "unique\_Musteri\_MusteriID" PRIMARY KEY ("MusteriID");

--

-- Name: Odeme unique\_Odeme\_OdemeID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Odeme"

ADD CONSTRAINT "unique\_Odeme\_OdemeID" PRIMARY KEY ("OdemeID");

--

-- Name: Salon unique\_Salon\_FilmID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Salon"

ADD CONSTRAINT "unique\_Salon\_FilmID" UNIQUE ("FilmID");

--

-- Name: Salon unique\_Salon\_SalonID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Salon"

ADD CONSTRAINT "unique\_Salon\_SalonID" PRIMARY KEY ("SalonID");

--

-- Name: Tur unique\_Tur\_TurID; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Tur"

ADD CONSTRAINT "unique\_Tur\_TurID" PRIMARY KEY ("TurID");

--

-- Name: index\_FilmID; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "index\_FilmID" ON public."FilmDili" USING btree ("FilmID");

--

-- Name: index\_dilID; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "index\_dilID" ON public."FilmDili" USING btree ("dilID");

--

-- Name: FilmAktor AktorTOFilmAktor; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmAktor"

ADD CONSTRAINT "AktorTOFilmAktor" FOREIGN KEY ("AktorID") REFERENCES public."Aktor"("AktorID") MATCH FULL;

--

-- Name: FilmDili DilTOFilmDili; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmDili"

ADD CONSTRAINT "DilTOFilmDili" FOREIGN KEY ("dilID") REFERENCES public."Dil"("DilID") MATCH FULL;

--

-- Name: Bilet FilmTOBilet; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bilet"

ADD CONSTRAINT "FilmTOBilet" FOREIGN KEY ("FilmID") REFERENCES public."Film"("FilmID") MATCH FULL;

--

-- Name: FilmAktor FilmTOFilmAktor; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmAktor"

ADD CONSTRAINT "FilmTOFilmAktor" FOREIGN KEY ("FilmID") REFERENCES public."Film"("FilmID") MATCH FULL;

--

-- Name: FilmDili FilmTOFilmDili; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmDili"

ADD CONSTRAINT "FilmTOFilmDili" FOREIGN KEY ("FilmID") REFERENCES public."Film"("FilmID") MATCH FULL;

--

-- Name: FilmTuru FilmTOFilmTuru; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmTuru"

ADD CONSTRAINT "FilmTOFilmTuru" FOREIGN KEY ("FilmID") REFERENCES public."Film"("FilmID") MATCH FULL;

--

-- Name: Salon FilmTOSalon; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Salon"

ADD CONSTRAINT "FilmTOSalon" FOREIGN KEY ("SalonID") REFERENCES public."Film"("FilmID") MATCH FULL;

--

-- Name: FilmOzellik FilmToFilmOzellik; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmOzellik"

ADD CONSTRAINT "FilmToFilmOzellik" FOREIGN KEY ("FilmID") REFERENCES public."Film"("FilmID") MATCH FULL;

--

-- Name: Bilet KoltukTOBilet; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bilet"

ADD CONSTRAINT "KoltukTOBilet" FOREIGN KEY ("KoltukID") REFERENCES public."Koltuk"("KoltukID") MATCH FULL;

--

-- Name: Siparis MusteriTOSiparis; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Siparis"

ADD CONSTRAINT "MusteriTOSiparis" FOREIGN KEY ("MusteriID") REFERENCES public."Musteri"("MusteriID") MATCH FULL;

--

-- Name: Siparis OdemeTOSiparis; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Siparis"

ADD CONSTRAINT "OdemeTOSiparis" FOREIGN KEY ("OdemeID") REFERENCES public."Odeme"("OdemeID") MATCH FULL;

-- Name: FilmOzellik OzellikTOFilmOzellik; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."FilmOzellik"

ADD CONSTRAINT "OzellikTOFilmOzellik" FOREIGN KEY ("OzellikID") REFERENCES public."Ozellik"("OzellikID") MATCH FULL;

--

-- Name: Koltuk SalonTOKoltuk; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Koltuk"

ADD CONSTRAINT "SalonTOKoltuk" FOREIGN KEY ("SalonID") REFERENCES public."Salon"("SalonID") MATCH FULL;

-- Name: Bilet SiparisTOBilet; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bilet"

ADD CONSTRAINT "SiparisTOBilet" FOREIGN KEY ("SiparisID") REFERENCES public."Siparis"("SiparisID") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

-- Name: FilmTuru turTOFilmTuru; Type: FK CONSTRAINT; Schema: public; Owner: postgres

ALTER TABLE ONLY public."FilmTuru"

ADD CONSTRAINT "turTOFilmTuru" FOREIGN KEY ("TurID") REFERENCES public."Tur"("TurID") MATCH FULL;

-- PostgreSQL database dump complete

***FONKİSYONLAR***

CREATE OR REPLACE FUNCTION bilet.kdv("bilet\_fiyat" double precision)

RETURNS double precision

LANGUAGE plpgsql

AS $function$

BEGIN

bilet\_fiyat: = bilet\_fiyat\*0.04 + bilet\_fiyat

RETURN bilet\_fiyat;

end;

$function$

CREATE OR REPLACE FUNCTION filim.zam("film id" double precision)

RETURNS double precision

LANGUAGE plpgsql

AS

$function$

BEGIN

IF "filmID" >1290 then

\*0.06 + bilet\_fiyat;

end IF;

RETURN bilet\_fiyat;

end;

$function$

create FUNCTION film ("film name" Character Varying)

RETURNS float

language plpgsql

as

$$

BEGIN

IF "film name"="" THEN

"film Name"="film ID";

end if;

RETURNS "film Name";

end;

$$

CREATE OR REPLACE FUNCTION check\_about\_film."film\_gösterim\_zamanı"()

RETURNS boolean

LANGUAGE plpgsql

AS $function$

DECLARE BEGIN

If "tarih" >= "film\_gösterim\_zamanı" Then

"film\_id" = "Arama";

ENd IF;

END;

$function$

***TRİGGERS***

CREATE TRİGGER check\_about

BEFORE UPDATE ON film

FOR EACT ROW

EXECUTE PROCEDURE check\_about\_film() ;

\*\*\*\*\*\*

CREATE TRIGGER "film\_name\_diğişmek”

BEFORE UPDATE ON "film"

FOR EACH ROW

EXECUTE PROCEDURE "film"();

\*\*\*\*\*

CREATE TRIGGER "bilet\_kdv\_hesaplamak”

BEFORE UPDATE ON "bilet"

FOR EACH ROW

EXECUTE PROCEDURE "bilet.kdv"();

\*\*\*\*\*\*\*\*

CREATE TRIGGER "filim\_zam\_hisaplamak”

BEFORE UPDATE ON "film"

FOR EACH ROW

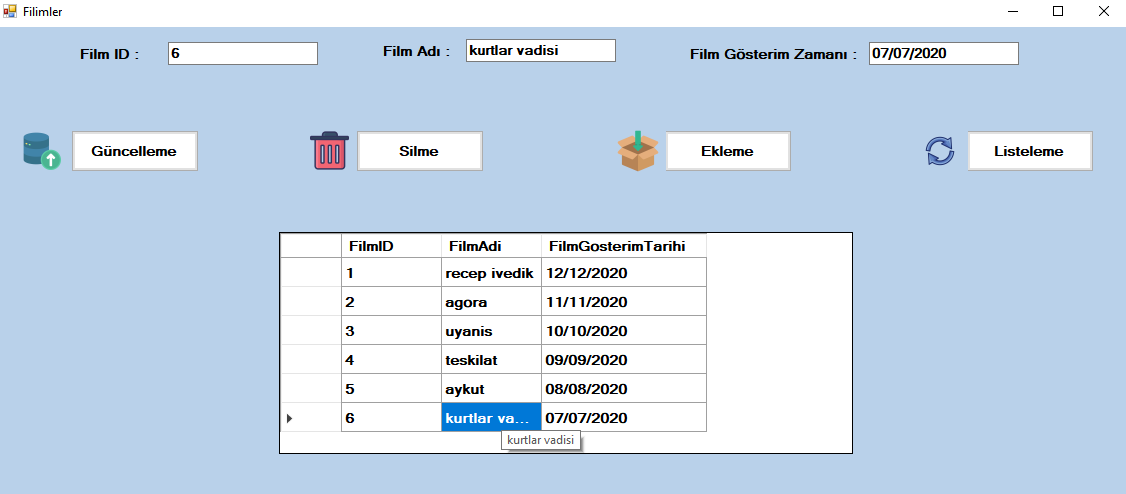
EXECUTE PROCEDURE "film\_zam"();

\*\*\*\*\*\*\*\*\*

***KOMUTLARIN ÇALIŞTIĞINA DAİR EKRAN GÖRÜNTÜLERİ***

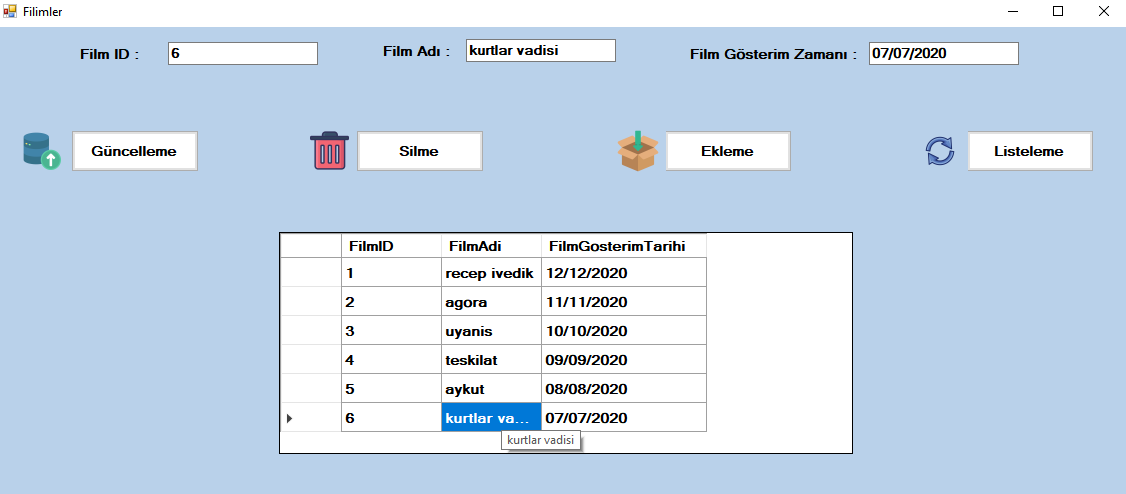
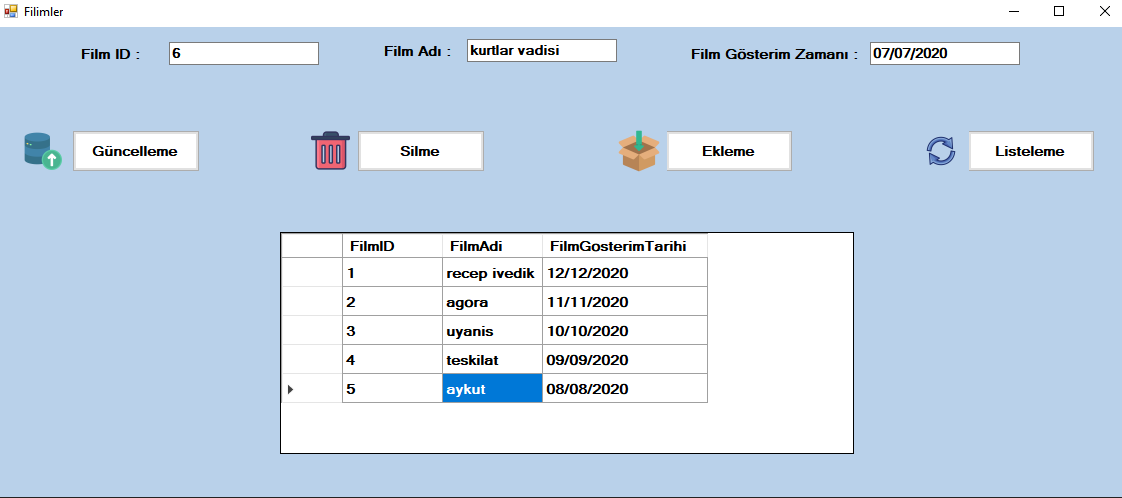
Aşağıdaki ekran görüntüsünde vizyondaki film listemin ilk hali



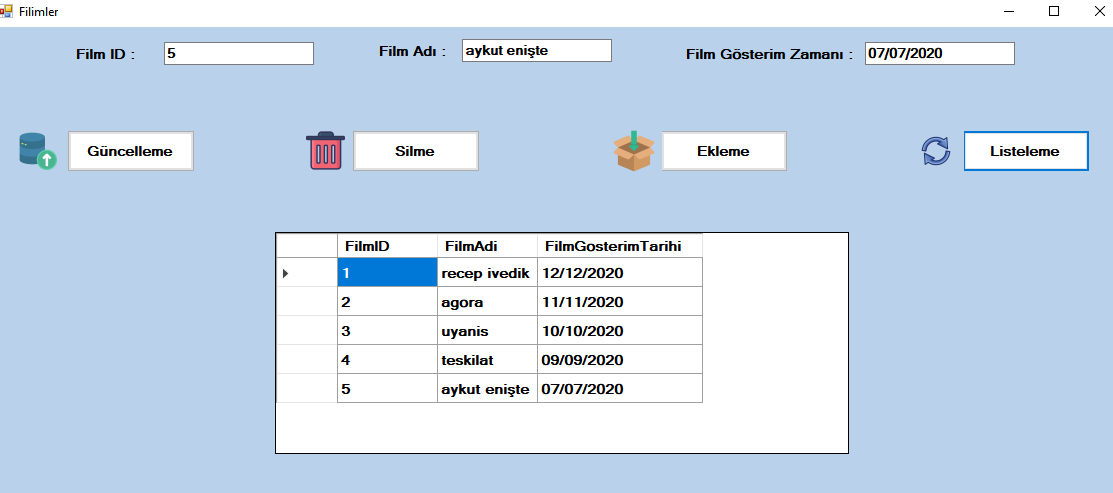
**ekleme yaptığımız zaman** 

**silme işlemleri**

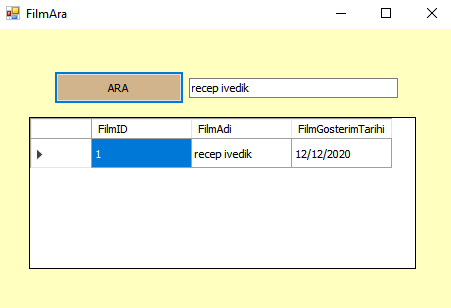
**şimdi kurtlar vadisi ekledikten sonra sileceğiz**

**silindi** 

**aykut → aykut enişteye değiştireceğiz**

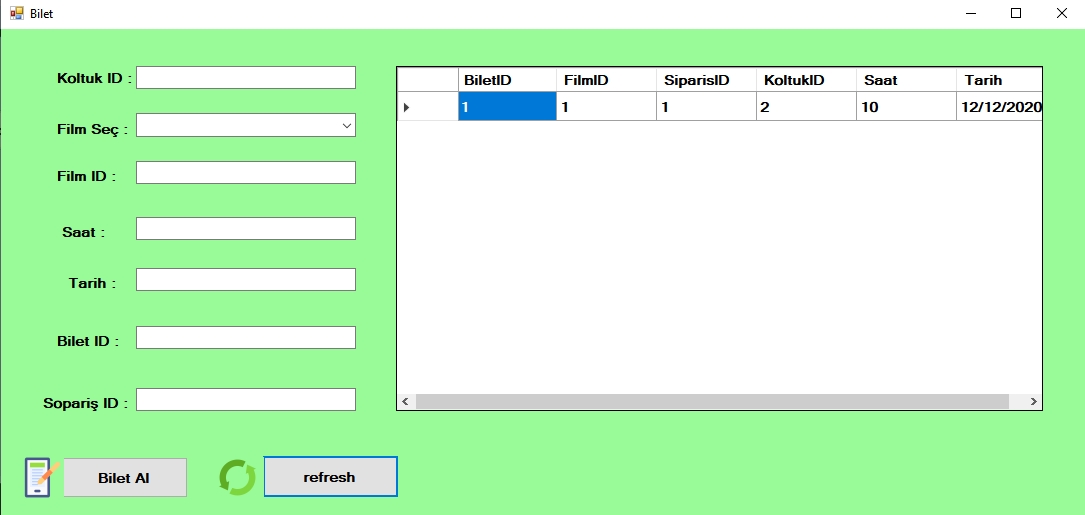
**değişterildi**

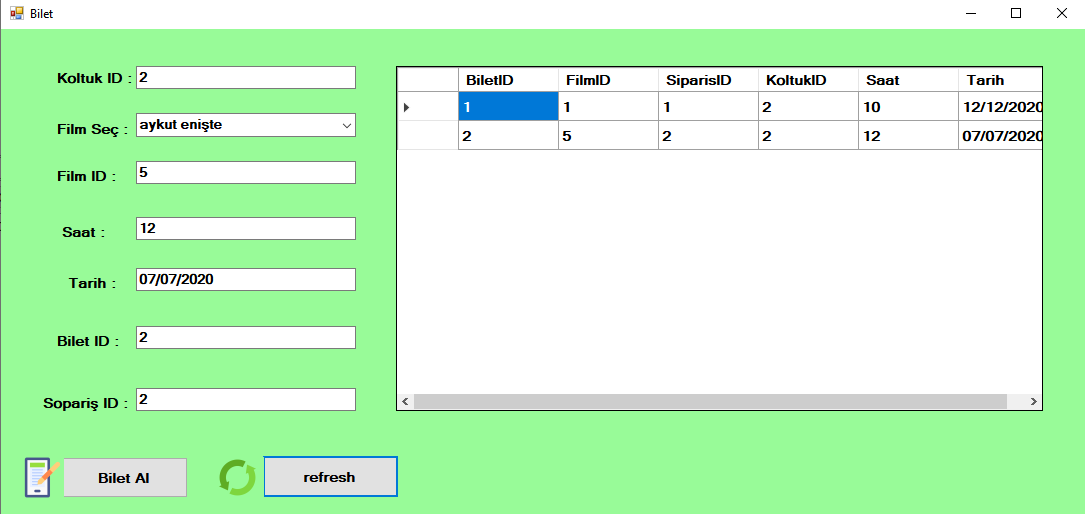
**arama işlemi için**



**Bilet alma işlemi için**

**almadan önce tablo göstereyim**



**aldıktan sonra** 

**UYGULAMA ANLATIMI İÇİN YOUTUBE VİDEO LİNKİ:**

**https://youtu.be/SVaFGhBwNQk**

**PROJE LİNKİ : https://www.mediafire.com/folder/3q4pfhqkg0fcj/VTYS\_PROJE**