

Veri Tabanı Yönetim Sistemi

Database Management System

Berkay Efe ÖZCAN
Mühendislik Fakültesi, Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
berkayefeozen@gmail.com

Özetçe—Bu projenin amacı veritabanı yönetim sistemlerinin yapısını anlamak ve bu sistemlerin çözümünü sağlamaktır.Bu projede birçok film ,dizi vb. programları izleme platformu geliştirilmiştir.Platforma kullanıcılar kayıt olup giriş yapabilmektedir. Kullanıcılar ve izlediği programlar veri tabanında ilişkisel olarak tutulmuştur.Veritabanında oluşturulan tablolar ve varlıklar arasındaki ilişkiler ileriki bölümlerde anlatılacaktır.Proje ms access veritabanı yönetim sistemi kullanılarak ve java programlama dili ile geliştirilmiştir.Grafik kısmı ise swing kütüphanesi kullanılarak yazılmıştır.

Anahtar Kelimeler— Veritabanı, Veritabanı yönetim sistemleri, kullanıcı kayıt tabanlı sistemler, İlişkisel Veritabanı

Abstract — The purpose of this project is to understand the structure of database management systems and to provide solutions for these systems. Programs monitoring platform has been developed. Users can log in and login. The relations between the users and the programs they watch are kept in the database. The relationships between the tables and assets created in the database will be explained in the following sections. The project has been developed using the ms access database management system and the java programming language. The graphic part has been written using the swing library.

Keywords— Database, Database Management Systems ,User Registration Based Systems, Relational Database.

I. GİRİŞ

Veritabanı birbiriyle ilişkili ve organize olmuş verilerin elektriksel olarak bilgisayar ortamında tutulmasıdır. Günümüzde kullanılan en yaygın veritabanı türleri içindeki veriler, işleme ve veri sorgulamayı verimli hale getirmek için genellikle bir dizi tablodaki satır ve sütunlarda modellenir. Veritabanları gerçekte var olan ve birbirleriyle ilişkileri olan nesneleri ve ilişkilerini modeller. Çok sayıda veritabanı türü bulunur. Belirli bir organizasyon için ideal veritabanı, organizasyonun veriyi nasıl kullanmayı amaçladığına bağlı olarak değişiklik gösterir. Günümüzde veriyi kullanma biçimlerine göre veritabanları ilişkisel veritabanları,nesne odaklı veritabanları,dağıtılmış veritabanları,veri ambarı,No SQL veritabanları , grafik veritabanları ,OLTP veritabanları sayılabilir. Veritabanı genellikle bir veritabanı yönetim sistemi tarafından kontrol edilir. Veritabanı yönetim sistemi (VTYS), yeni veritabanları oluşturmak, veritabanını düzenlemek, geliştirmek ve bakımını yapmak gibi çeşitli

karmaşık işlemlerin gerçekleştirildiği birden fazla programdan oluşmuş bir yazılım sistemidir. Veritabanı yönetim sistemi, kullanıcı ile veritabanı arasında bir arabirim oluşturur ve veritabanına her türlü erişimi sağlar.

Bu projede ise bir dizi, film izleme platformunu bir veritabanı yönetim sistemi kullanılarak gerçekleştirmemiz istenmiştir.Bu platforma kullanıcılar kayıt olup sisteme giriş yapabilmekte ayrıca program izleyebilmektedir.-İzleme işlemi sembolik olarak yapılmaktadır- Bu projede veritabanı yönetim sistemi olarak ms access kullanılmıştır.Sorgu dili olarak SQL (**Structured Query Language**) kullanılmıştır.

II. YÖNTEM

a. Tabloların ve İlişkilerin Tasarlanması :

Bu proje için veritabanında 5 tane tablo oluşturulmuştur. Bunlardan ilki kullanıcıların verilerini tutan Kullanıcı tablosudur.Tablonun alanları Şekil 1 deki gibidir.

Alan Adı	Veri Türü
kullanıcıID	Otomatik Sayı
ad_soyad	Kısa Metin
sifre	Kısa Metin
dogum_tarihi	Tarih/Saat
email	Kısa Metin

Şekil 1: Kullanıcı tablosunun tasarımı

Bu tablonun primary key olarak veri eklendiğinde artan tipinde veri tipi tanımlanmıştır. Email de primary key olabilirdi ancak emailin değişmesi gibi durumlar için email primary key olarak tanımlanmaz bunun yerine bir kullanıcıyı diğerlerinden ayırt etmek için ID tanımlanır.¹

İkinci tablo izlenecek programların verilerini tutan Program tablosudur.

Alan Adı	Veri Türü
programID	Otomatik Sayı
ad	Kısa Metin
uzunluk	Sayı
bolum_sayisi	Sayı
tip	Kısa Metin

Şekil 2: Program tablosu

Program tablosunun primary keyi (birincil anahtarı) program idsidir ve bu integer alandaki veri otomatik olarak artmaktadır. Uzunluk alanı programın dakika cinsinden süresidir. Film tipindeki programın bölüm sayısı 1 olarak belirlenmiştir.

Üçüncü tablo ise bir programın sahip olabileceği türleri tutmaktadır. Bir program birden fazla türde sahip olabilir. Bir tür de birden fazla programa sahip olacağından aralarındaki ilişki ProgramTur tablosunda tutulmuştur. Bu ilişki şekil 4 gösterilmiştir. Bir programın sahip olabileceği türlerin verilerini tutan tablo şekil 3 te gösterilmiştir.

Alan Adı	Veri Türü
TurID	Otomatik Sayı
ad	Kısa Metin

Şekil 3 : Tür tablosunun tasarımı

Dördüncü tablo program ile tur arasındaki sahiplik ilişkisini tutan tablodur. Bu tablonun tasarımı şekil 4 de verilmiştir.

Alan Adı	Veri Türü
TurID	Sayı
ProgramID	Sayı

Şekil 4 : ProgramTur tabosunun tasarımı

Bu tablonun primary keyi turId ve ProgramId dir. Aynı zamanda TurId ve ProgramId foreign keydir (dışarıdaki bir tablonun primary keynin referans alır). Bu primary key tanımı tur ve program varlıkları arasındaki many to many ilişkisini sağlamak için oluşturulmuştur.

Beşinci tablo ise kullanıcı ve program varlıkları arasındaki ilişkiyi temsil eden KullaniciProgram tablosudur. Bu tablonun tasarımı şekil 5 teki gibidir.

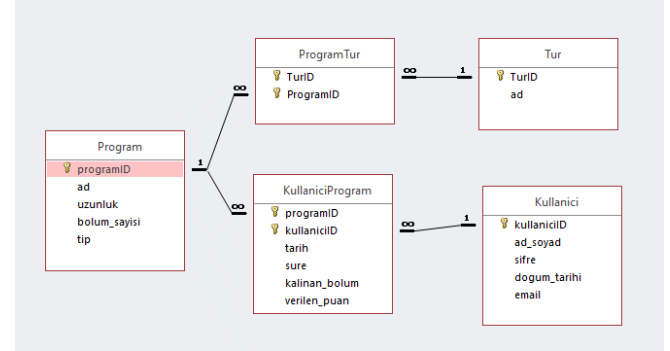
Alan Adı	Veri Türü
programID	Sayı
kullaniciID	Sayı
tarih	Tarih/Saat
sure	Sayı
kalinan_bolum	Sayı
verilen_puan	Sayı

Şekil 5: KullaniciProgram tablosunun alanları

ProgramID ve kullaniciID birlikte primary keyi oluşturur. Her iki alan da aynı zamanda foreign dir. Bu ilişki şekil 6 da gösterilmiştir. Dış anahtarlardan referans alındığı için bilgi tutarlığı zorunludur. Yani program tablosunda mevcut olmayan bir program ID kullanıcıProgram tablosuna eklenemez. Bu tablodaki süre alanında süre saniye cinsinden tutulmaktadır. Verinin bu şekilde tutulmasının sebebi kullanıcı

nın bir dakikadan daha az sürede programı izleyebileceği düşüncesidir.

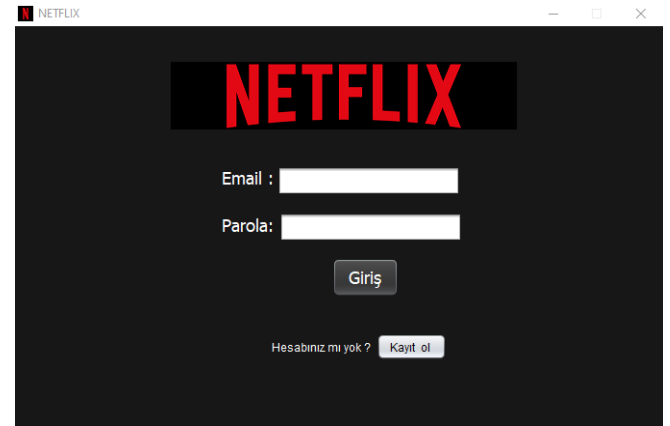
Bu projedeki tablolar arası ilişki şeması şekil 6 da verilmiştir. Bu ilişkiler veri tekrarının engellenmesi (normalizasyon) için ayrı tablolara bölünmüştür. Örneğin program tablosunda tür bilgisi de tutulabilirdi ancak tür ile program arasında çok a çok ilişki bulunduğu için ayrı tablolara bölündü ki veri tekrarları oluşmasın.



Şekil 6 : Tablolar arasındaki ilişkiler

I. Arayüzün Tasarlanması ve Programın Çalıştırılması :

Program çalıştırıldığında daha önceden giriş yapılmışsa ana sayfa açılır. Kullanıcı ana sayfasından çıkış yapıldığı takdirde giriş sayfası açılır. Eğer programda halihazırda açık bir oturum yoksa direkt kullanıcı giriş sayfası açılır. Oturum kontrolü oturumKontrolü.conf adlı dosyadan okunan bilgiyle yapılır. Grafik java swing kütüphanesi kullanılarak geliştirilmiştir.



Şekil 7 : Kullanıcı giriş sayfasının görünümü

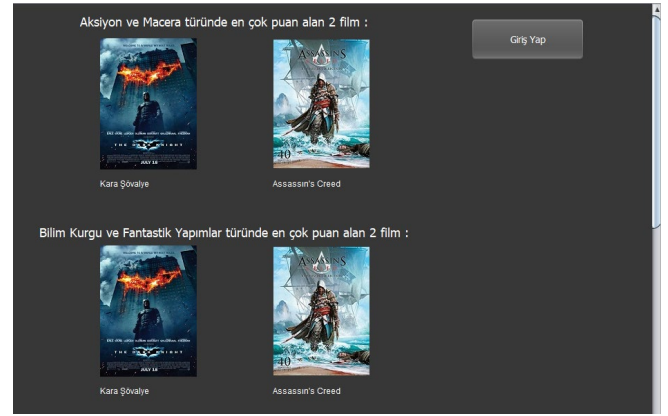
Kullanıcı giriş yaparken sisteme kayıtlı olmayan email yazıldığında bu email sistemde kayıtlı değildir hatası verilir. Kullanıcı kayıtlı butonundan kayıt olabilmektedir.

Şekil 8: Kayıt olma ekranının görüntüsü

Eğer tüm alanlar doğru doldurulursa kayıt ol butonu aktif olacaktır. Emailin doğrulanması regex kullanılarak gerçekleştirilmiştir. Ayrıca kayıt ol butonuna tıklandığında kullanıcı veritabanına kayıt edilir ve seçilen türlere göre her program için en çok puan alan 2 film toplamda 6 film listelenir .

Şekil 9 : Kullanıcı kayıt formunu doldurduktan sonra alınan ekran görüntüsü

Parola ve parola tekrarlarının birbiriyle uyumu ayrıca kontrol edilir. Email girilirken @ işaretiyle bir domain adresi de girilmelidir aksi taktirde doğrulanma sağlanmamaktadır. Doğum tarihi seçilirken mümkün olmayan bir tarih girilirse hata verilebilir. Bu olası hata denetlenmemiştir. Kullanıcı kayıt ol butonuna bastığında eğer daha önce girilen email ile bir kayıt yapılmamışsa başarı ile kayıt edildiğine dair kullanıcıya mesaj verilir. Aksi taktirde bu kullanıcı zaten kayıtlı diye hata gösterilir.



Şekil 10 : Kayıt esnasında kullanıcının seçtiği türlere göre listelenen filmlerin bir örneğini

Şekil 10 da dört tane film gösterilmiş gibi gözüküyor ancak filmler listelendiğinde scrollview sayesinde aşağı doğru kaydırılabilir.

Sisteme bir kullanıcı giriş yaptığında kullanıcının izlediği filmler listelenmiş bir halde bir panel açılır ve kullanıcı ana sayfasının üstünde tüm programları listeleyebileceği bir buton , türe göre arama yapma paneli , ana ekranı aktif eden buton , isme göre ara yapıldığında listelenen sonuç panelini açmaya yarayan bir buton bulunur. İsme göre arama, ana sayfada bulunan arama butonuna basılarak yapılabilir. Sistemden kullanıcı çıkışı kişi simgesine basılarak yapılabilir. Ayrıca listelenen filmlerin üstündeki play butonuna basıldığında program izleme paneli açılır. Program izleme panelinden puan vermeden çıkılamaz. Burada bahsedilenler ileriki şekillerde anlatılacaktır. Ekran fotoğraflarının büyüklüğü münasebeti nedeniyle buraya eklenememiştir. Program izleme panelindeki sonraki bölüm ve önceki bölüm butonları sayesinde eğer program filmse bölümler arasında geçiş yapılabilir. Şekil 11-17 arası kullanıcıyı anasayfası görsellerle anlatılmıştır. Şekil 18 de ER diyagramı verilmiştir.

III. DENEYSEL SONUÇLAR

Yapılan testler sonucunda veritabanına sorunsuz bir şekilde kullanıcı ekleme yapılabildiği, kayıt sırasında email ile kullanıcının var olup olmadığını sorgulanabildiği, kullanıcı sisteme girerek kayıt yapılabildiği gözlemlendi. İsme ve türe göre arama yapıp filmlerin listelendiği doğrulandı. Veri tabanının varlıklar arasındaki ilişkiler sayesinde daha efektif çalıştığı gözlemlendi. Java ile nesneye yönelik programlamanın avantajları bir kez daha gözlemlendi. Program izleme paneli ve butonlara tıklamak için thread kullanıldı. Bir programda threadler çalışırken bekleme süresi olmadan sürekli çalışırsa işlemciyi bir hayli yorduğu sonucuna varıldı. Bir program izledikten sonra program kapatılsa bile kullanıcı kaldığı yerden tekrar edebildiği test edilip doğrulandı. Veritabanı yönetim sistemiyle veritabanına verileri kolay bir şekilde eklenip sorgulanabildiği gözlenmiş

ve deneyimlenmiş oldu. Ms acces ile programın dizininde veri tabanının tutulabildiği gözlemlendi.

IV. SONUÇ

Sonuç olarak veri tabanı yönetim sistemlerinin işleyişi öğrenildi. İlişkisel veri tabanında varlıklar arasında ilişkiler kurulup verilerin daha iyi nasıl işlenebileceği deneyimlendi. Java swing kütüphanesinde uygulama yapılarak kütüphane pekiştirildi.

V. KAYNAKÇA

[1]<https://www.oracle.com/tr/database/what-is-database.html>

[2]<https://www.techopedia.com/definition/24361/database-management-systems-dbms>

[3] <https://www.w3schools.com/sql/>

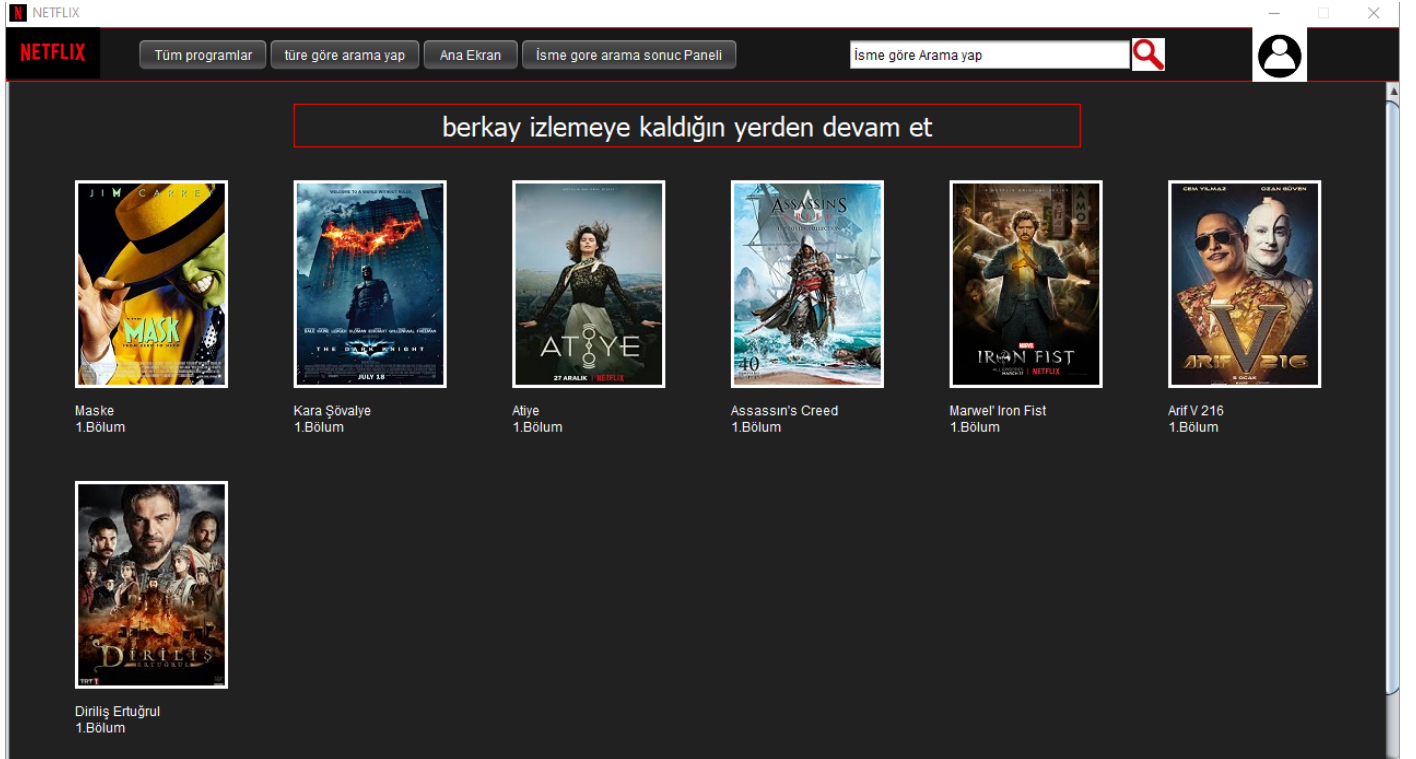
[4]<https://stackoverflow.com/questions/45300178/how-to-connect-java-to-ms-access-database>

[5]<https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/>

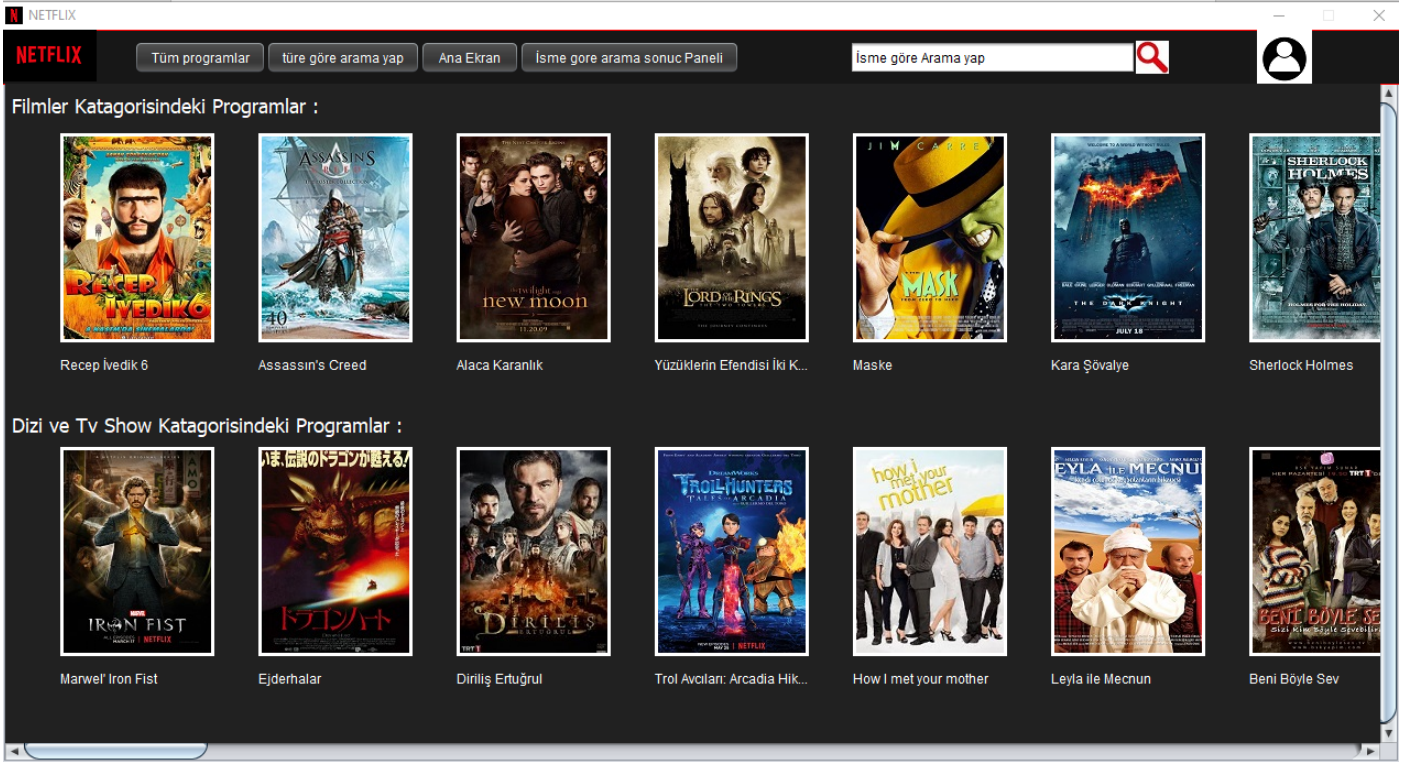
[6] <https://www.youtube.com/watch?v=zYuToCeHSbk>

VI. DİPNOT

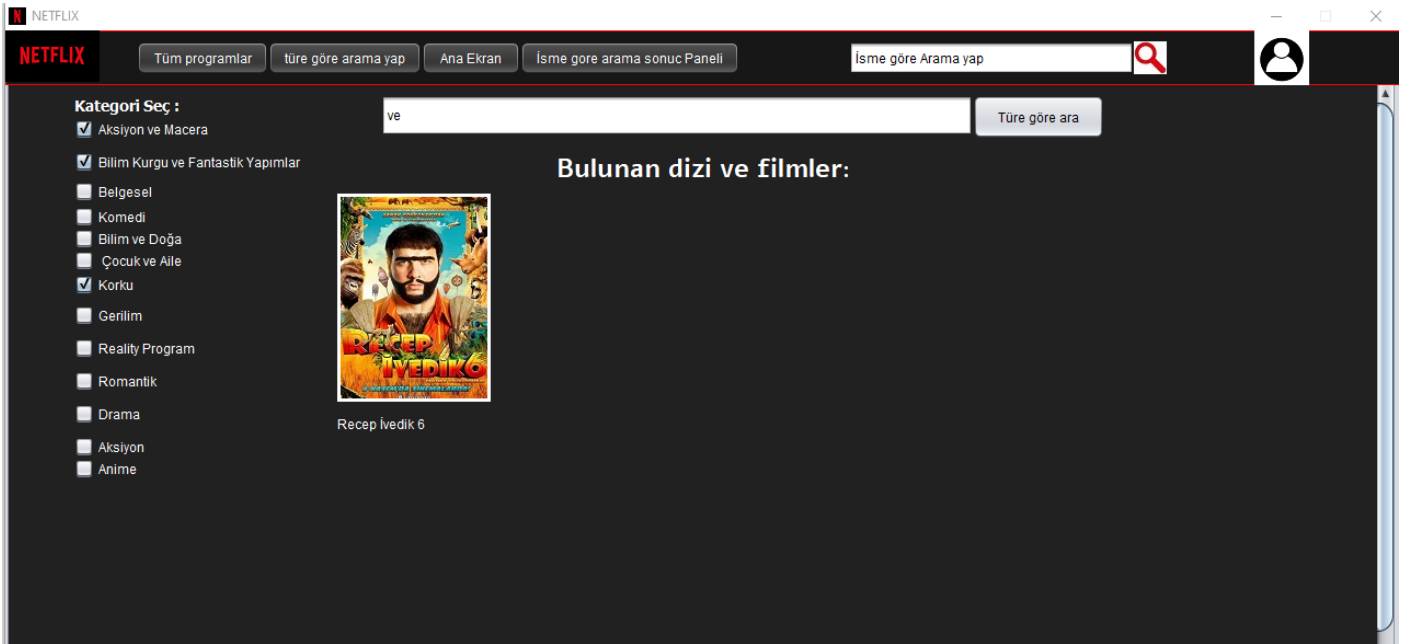
1-> <https://support.office.com/en-us/article/add-or-change-a-table-s-primary-key-in-access-07b4a84b-0063-4d56-8b00-65f2975e4379>



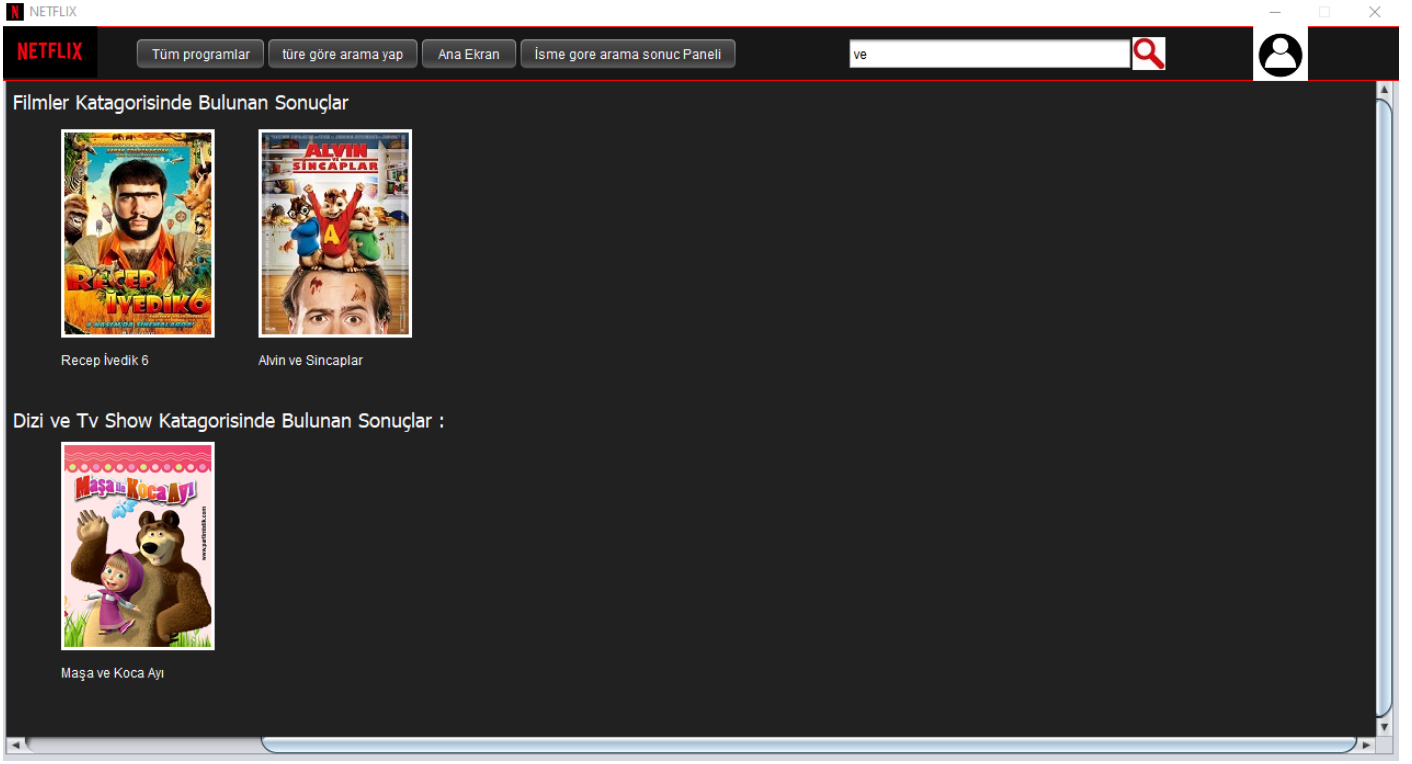
Şekil 11: Kullanıcın ana ekranı



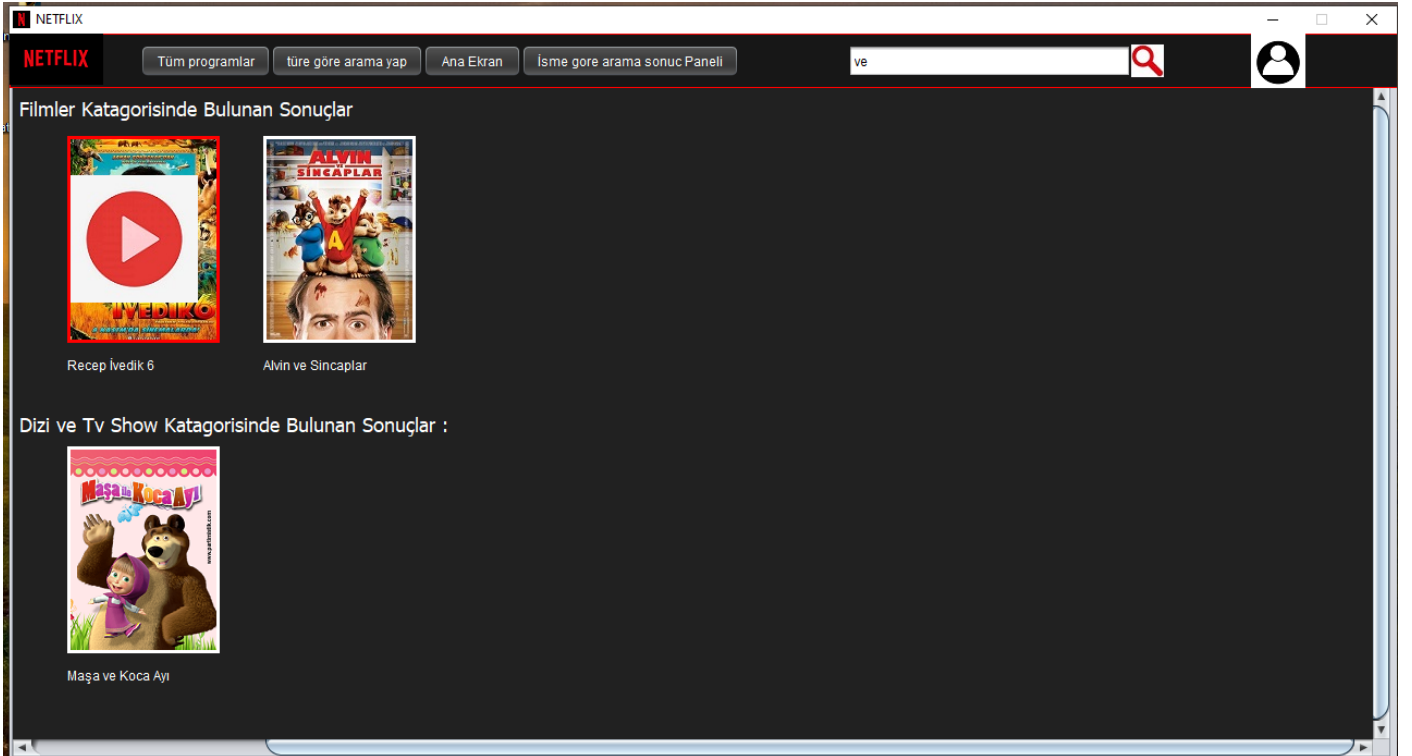
Şekil 12 : Tüm programların gösterildiği ekran



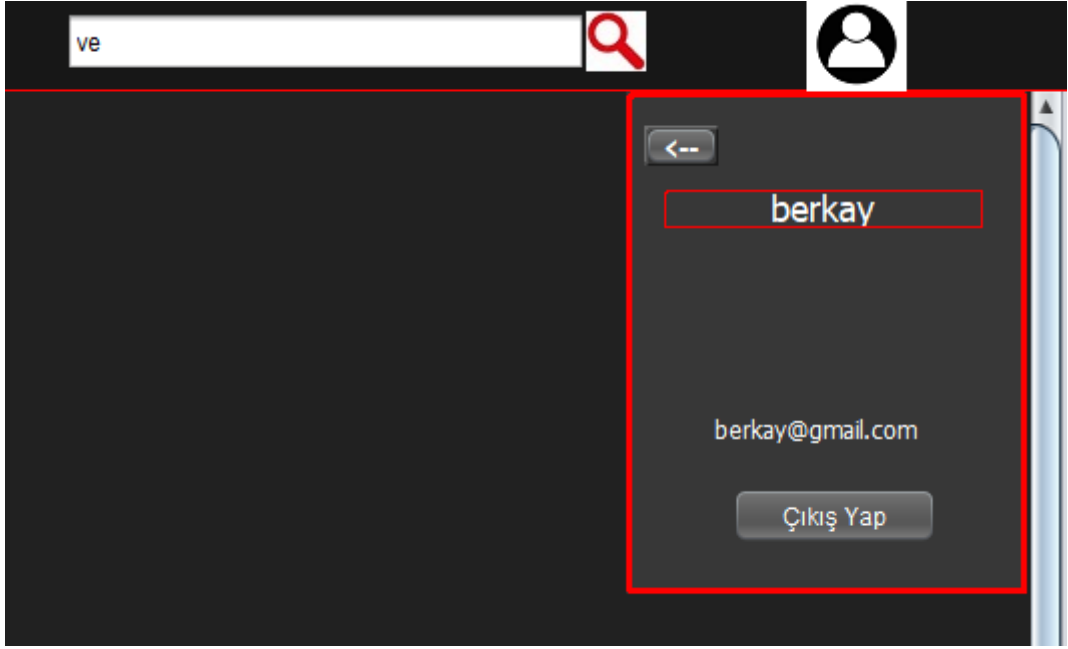
Şekil 13 : Türe göre arama yapılan panelin görünümü



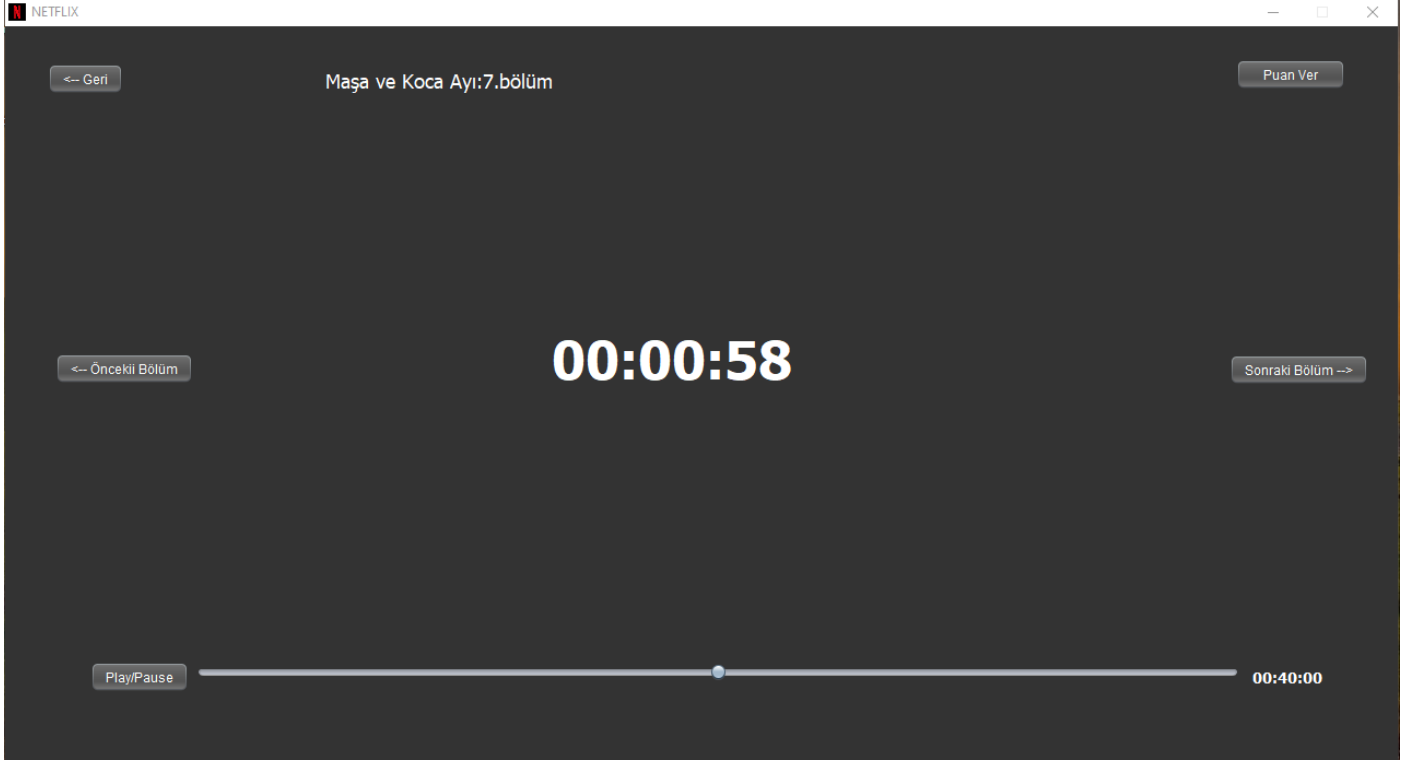
Şekil 14 : İsme göre arama yapma örneği



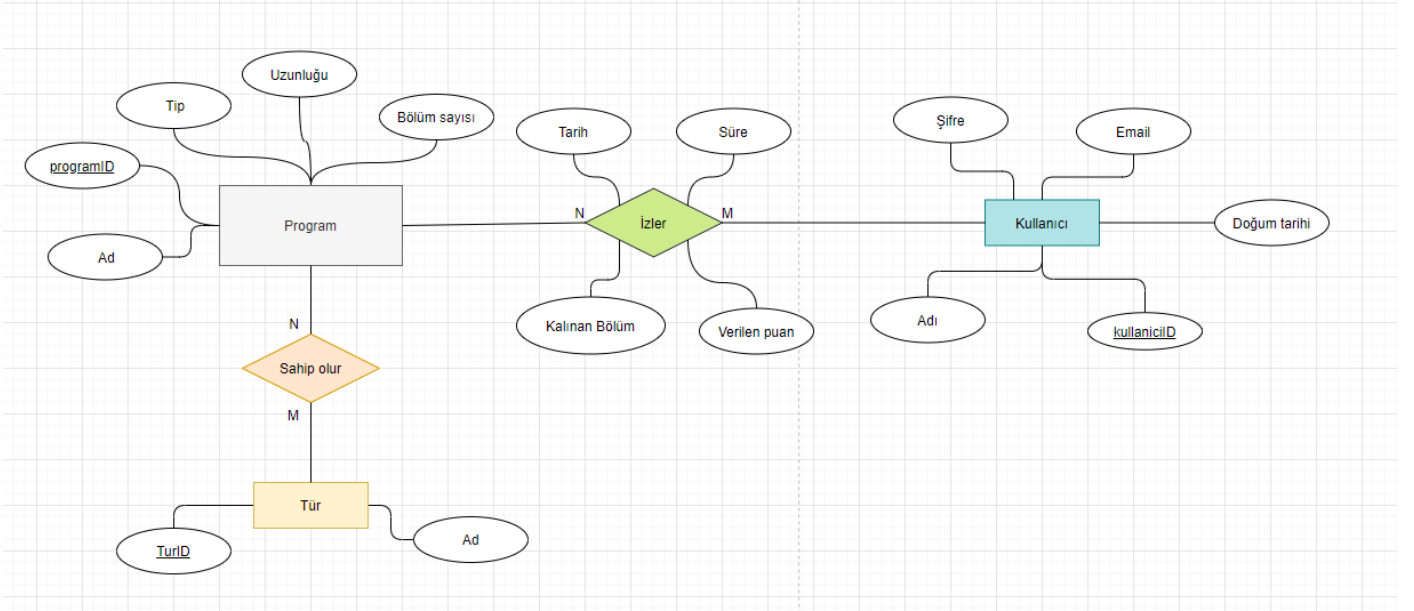
Şekil 15 : Play tuşunun gösterimi



Şekil 16 : Çıkış panelinin gösterimi



Şekil 17 : Program izleme ekranı



Şekil 18 : Veri tabanının ER diyagramı