



— İSTANBUL —
OKAN ÜNİVERSİTESİ

İSTANBUL OKAN ÜNİVERSİTESİ
MESLEK YÜKSEKOKULU
BİLİŞİM TEKNOLOJİLERİ BİLGİSAYAR
PROGRAMCILIĞI BÖLÜMÜ
JAVA PROGRAMLAMA
Sinema Otomasyonu
Danışman: NİLGÜN İNCEREİS
18MY03009 – GÖRKEM ACAR

İÇİNDEKİLER

Giriş:	2
Genel Bilgiler:	2
1. Java nedir?	2
2. Java nerelerde kullanılır?	2
3. Java'nın avantajları nelerdir?	2
4. NetBeans nedir?	2
5. JFrame nedir?	2
6. Veri tabanı nedir?	2
7. Veri tabanı ne işe yarar?	2
8. Paket yapısı ve import nedir?	3
9. JAR dosyası nedir?	3
Gelişme:	3
1. Nelere ihtiyaç duydum?	3
2. Formlarımı nasıl oluşturdum?	4
3. Formlar arası geçiş	4
4. JAR dosyasını projeme nasıl dahil ettim?	4
5. Veri tabanı	5
• Ekleme	5
• Silme	7
• Güncelleme	7
• Veri çekme	8
• Arama	10
• Resim çekme-ekleme	11
6. Backup(yedek) nasıl aldım?	11
• Java program yedeği	11
• Veri tabanı yedeği	11
Sonuç:	12
Zaman Çizelgesi:	12
Ekler:	12
Kaynakça:	15

Giriş:

Sinema Otomasyonu programında sadece yönetici(admin) girişi bulunuyor. İlk olarak ana bir panel karşılıyor yöneticiyi buradan isterse, admin girişi veya direkt sinemalar formlarına geçiş yapabiliyor. Panel formunda film ekleme, silme, güncelleme ve filmleri görüntüleyebiliyor. Sinemalar formundaysa eklenen filmleri aratabiliyor veya tür puana göre de arama yapabiliyor.

Genel Bilgiler:

1. Java nedir?

Java SunMicrosystems tarafından 1995 yılında piyasaya sürülen bir programlama dilidir.

2. Java nerelerde kullanılır?

Java programlama dili; bilgisayarlarda, cep telefonlarında, veri depolama merkezlerinde, oyun konsollarında, internet uygulamalarında, akıllı televizyonlarında vb. platformlarda kullanılmaktadır.

3. Java'nın avantajları nelerdir?

1. Güvenilir olması
2. Taşınabilir olması
3. Nesne yönelimli olması
4. Güçlü olması
5. Yorumlanabilir olması

4. NetBeans nedir?

Netbeans başta Java olmak üzere PHP, C++ ve birçok programlama dilini kullanarak yazılımlar geliştirmeye imkan sunan, tümleşik geliştirme ortamı (IDE) yazılımıdır.

Oracle tarafından geliştirilmiştir.

5. JFrame nedir?

Java swing kütüphanesinde bir pencere oluşturmak için kullanılacak ana sınıf JFrame'dir.

6. Veri tabanı nedir?

İngilizce anlamı database olan veri tabanı düzenli bilgiler topluluğudur. Birbirleriyle ilişkili olan verilerin birlikte tutulduğu yönetilebilir, güncellenebilir, taşınabilir ve anlamlandırılabilir kullanım amacına uygun olarak sistemli bir şekilde toplanmış düzenli bilgiler topluluğudur.

7. Veri tabanı ne işe yarar?

Veri tabanı bizlere kolaylık sağlar. Veri tabanında bulunan bilgilere kolayca erişim sağlayabiliriz ve içerisinde bulunan bilgiler arasında hesaplama gibi işlemleri kolaylıkla yapabiliriz.

SQL Management Studio nedir?

SQL Management Studio ile veri tabanı ile ilgili işlemlerimizi yapabiliriz. Sorgular, tablolar, fonksiyon gibi daha fazla işlemlerimizi bu program üzerinde yapabiliriz.

8. Paket yapısı ve import nedir?

Java programları paketlerden oluşmaktadır. Paketlerin içinde sınıflar ve arayüzler bulunur. İstenirse var olan paketlerden başka (import) paketleri de kullanabiliriz. Paket içine yer alan sınıflar, arayüzler ve importlar için bir sayı sınırlaması yoktur.

```
import static com.sun.xml.internal.fastinfoset.alphabet.  
import java.awt.Image;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.sql.*;  
import java.util.ArrayList;  
import javax.swing.ImageIcon;  
import javax.swing.JOptionPane;  
import javax.swing.table.DefaultTableModel;
```

9. JAR dosyası nedir?

JAR dosyası bir java paket dosyasıdır. Amacı birden fazla sınıfı tek bir paket içinde toplamasıdır. JAR dosyalarını sıkıştırma olarak düşünmek yerine bir arşivleme olarak düşünmek gerekir. Buradaki amaç tek bir dosyada bütün sınıfların toparlanmasıdır.

Gelişme:

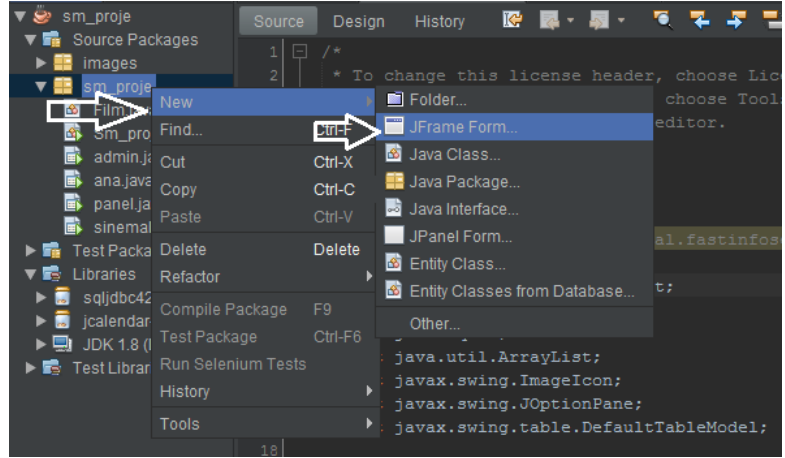
1. Nelere ihtiyaç duydum?

Java formlarımı oluşturmak ve kodlarımı yazmak için NetBeans 8.2 kullandım. Veri tabanı tablolarımı SQL Management Studio ile hazırladım.



2. Formlarımı nasıl oluşturdum?

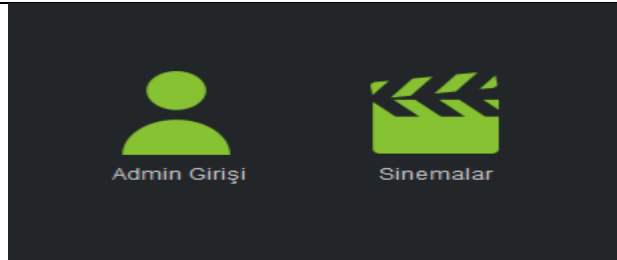
Projeme sağ tıklayarak new, JFrame Form diyerek formlarımı oluşturdum. Toplamda 4 adet formum bulunmakta bunlar; ana.java, admin.java, panel.java ve sinemalar.java



3. Formlar arası geçiş

Formlar arası geçiş yapmamız için, gideceğimiz formdan nesne almalıyız.

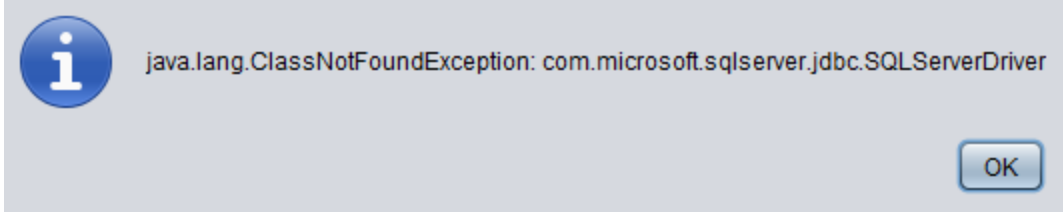
Admin.java formundan adm adında nesne ürettik ve admin'e tıklandığı zaman, `this.setVisible(false)` diyerek bulunduğumuz formu kapadık. `Adm.setVisible(true)` yaparak admin formunu açmış olduk.



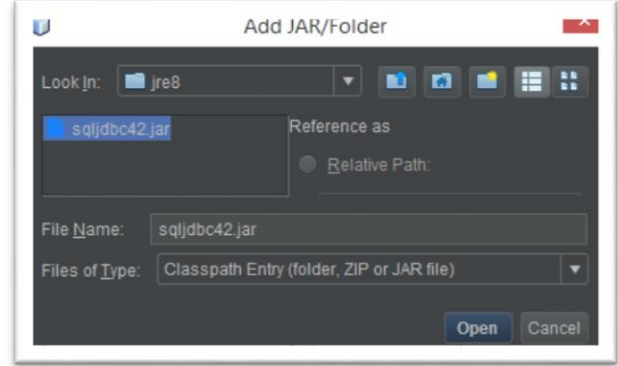
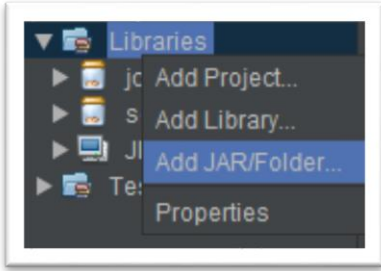
```
admin adm = new admin();  
this.setVisible(false);  
adm.setVisible(true);
```

4. JAR dosyasını projeme nasıl dahil ettim?

SQL' de işlemlerimi yapmam için sqljdbc42.jar adlı dosyayı indirip projeme dahil etmem gerekiyordu. Aksi takdirde şöyle bir hata alıyordum.



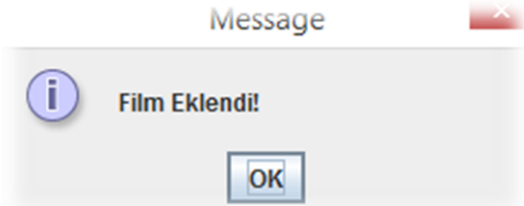
Gelelim nasıl dahil edeceğimize. Projemizin altında Libraries adında bir klasör var ona sağ tıklayıyoruz ve sonrasında Add JAR/Folder seçeneğini seçiyoruz. Açılan sekmede .jar uzantılı dosyamızı bulup projeye dahil ediyoruz.



5. Veri tabanı

- Ekleme

Yandaki görselde olduğu gibi yönetici boş alanlara bilgileri giriyor ve film ekle butonuna basılınca film ekleniyor.



Peki bu ekleme işlemini nasıl yapıyoruz? Şimdi onu inceleyelim.

A screenshot of the "Film Ekle" (Add Film) form. The form has a dark gray background with a green header bar containing the title "Film Ekle" and four buttons: "Film Ekle", "Film Sil", "Film Güncelle", and "Filmleri Göster". The form fields include: "Film Adı" (text input), "Film Konu" (text input with a scroll bar), "Film Yönetmen" (text input), "Film Yıl" (dropdown menu with "1920" selected), "Film Tür" (dropdown menu with "Aksiyon" selected), "Film Puan" (dropdown menu with "5" selected), and "Film ID" (text input with "356161" entered). There is a "Resim Ekle" button below the "Film ID" field and a large green "Film Ekle" button at the bottom.

Film Ekle butonuna çift tıklıyoruz ve Netbeans' da Source kısmına gidiyoruz. Butonun içerisine girip Try-Catch blokları yaratalım ki, programımız bir hata durumunda çökmesin. Try içerisinde ekleme işlemleri olacak. Catch içerisindeyse ekleme esnasında bir hata olursa onu görüntüleyebileceğiz.

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

Veritabanı bağlantı sınıfı projeye dahil ediliyor.

```
String url="jdbc:sqlserver://localhost:1433;user=sa;password=123;databaseName=sineDB;";
```

Veritabanımızın yolunu veriyoruz.

```
Connection con = DriverManager.getConnection(url);
```

Bağlantı sağlanıyor.

```
String sql = "INSERT INTO  
filmler(FilmAd,FilmKonu,FilmYonetmen,FilmYil,FilmTur,FilmPuan,filmID,FilmResimler)  
values(?,?,?,?,?,?,?,?)";
```

Sorgumuzu alıyoruz; ilk olarak ekleme işlemi yapacağımız için, INSERT INTO şeklinde yazıyoruz ve alanlarımızı ekliyoruz.

```
PreparedStatement pst = con.prepareStatement(sql);
```

SQL komutlarını veri tabanına göndermek için, Statement arayüzünü kalıtım alarak çeşitli metotları tanımlar. SetInt, SetString gibi.

```
pst.setString(1,txt_FilmAdi.getText());
```

Buradaysa alacağım ilk değer String bu yüzden setString diyerek değerimi alıyorum. Eğer int veya float alacak olsaydım setInt veya setFloat olacaktı.

```
pst.executeUpdate();
```

Buradaysa executeUpdate diyerek tablomuza yeni verimizi eklemiş oluyoruz.

```
JOptionPane.showMessageDialog(null,"Film Eklendi!");
```

İşin en güzel kısmıysa burası her şey düzgün çalışıyorsa, Film Eklendi mesajını görüyoruz. Genel bilgiler kısmında paketlere kısaca değinmiştim. İşte burada bir örneği JOptionPane.

import javax.swing.JOptionPane; diyerek projeme dahil ettim.

Yukarıdaki kodlar try bloğu için geçerliydi. Peki hata aldığımızı varsayarsak programımız otomatik olarak catch bloğuna gidecekti.

```
catch (Exception e) {  
    JOptionPane.showMessageDialog(null,e);  
}
```

Buradaysa bir hata alsaydık, hatamızın nereden kaynaklandığını görebilecektik.

- **Silme**

Yukarıda film eklemiştik. Şimdiyse onu nasıl sileriz onu inceleyelim, programımda her filme özel bir ID atıyorum ve o ID e göre siliyorum. Veri eklemekten tek farkı sorgular değişecek yani;

```
String sql = "DELETE FROM filmler WHERE filmID="+s_filmID.getText();
```

Burada ise veri tabanında bulunan filmID ile text olarak girdiğim veriye göre sil diyorum.



- **Güncelleme**

Silme işleminde olduğu gibi değişen tek kısım yine sorgumuz olacak.

```
String sql = "UPDATE filmler SET  
FilmAd=?,FilmKonu=?,FilmYonetmen=?,FilmYil=?,FilmTur=?,FilmPuan=? WHERE  
filmID="+g_txtFilmID.getText();
```

Silme işleminde olduğu gibi veri tabanımda bulunan filmID alanına göre güncelleme yapıyorum.

Film Ekle Film Sil Film Güncelle Filmleri Göster

Güncellenecek Film ID

487420

Film Adı

ads

Film Konu

daasd

Film Yönetmen

adstda

Film Yılı

1920

Film Tür

Aksiyon

Film Puan

5

Film Güncelle

Message

Film Güncellendi!

OK

- **Veri çekme**

Eklediğim filmleri görmek için öncelikle film.java adında sınıf oluşturdum. Bu sınıf içerisinde veri tabanımda bulunan alanlar vardı. Verileri çekmek için bir ArrayList oluşturdum <> içerisine ise oluşturduğum film.java yı yazdım.

```
ArrayList<Film> filmsList = new ArrayList<>();
```

```
String sql = "SELECT * FROM filmler";
```

İlk olarak yine sorgumuzu değiştiriyoruz. Sorguda tablodaki bütün alanları çağırıyoruz.

```
ResultSet rs = st.executeQuery(sql);
```

SQL Sorgusundan dönecek sonuç rs sonuç kümesi içinde tutuluyor.

```
Film film;
```

```
while(rs.next()){ Kayıt olduğu sürece her işlem sonunda 1 satır atla
```

```
film = new
```

```
Film(rs.getInt("id"),rs.getString("filmAd"),rs.getString("filmKonu"),rs.getString("filmYonetmen"),rs.getString("FilmYil"),rs.getString("FilmTur"),rs.getString("filmPuan"),rs.getString("filmID"),rs.getBytes("FilmResimler")); değerleri alıyoruz.
```

```
filmsList.add(film); her dönen sonucu listeye ekliyoruz.
```

```
}
```

```
return filmsList;
```

Son olaraksa listeyi return ediyoruz.

Film çekme işlemimiz başarılı peki bu filmleri nasıl göstereceğiz? Bir metot oluşturdum ve form açıldığında filmler de beraberinde geliyor.

Şimdi de bu metot içindeki kodlara bakalım...

```
ArrayList<Film> list = FilmList(); yukarıdan çekiyoruz.
```

```
DefaultTableModel model = (DefaultTableModel)jTable_Display_Movies.getModel(); satır ve sütunları tutmak için DefaultTableModel oluşturduk.
```

```
Object[] row = new Object[8]; bir dizi oluşturduk 8 olmasının sebebi 8 adet sütun gösterecek olmam.
```

Sonrasında bir for döngüsü oluşturuyoruz döngümüz 0 dan başlıyor ve list in uzunluğu kadar dönüyor. Döngünün içerisineyse 0 dan 7 ye kadar olan sütunlarımızı yazıyoruz. Örnek 1 tane kullanımı aşağıda görüldüğü gibi.

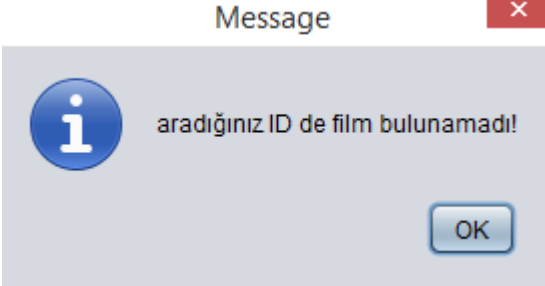
```
row[0]=list.get(i).getId();
```

```
model.addRow(row); Döngü her döndüğündeyse satırları ve sütunları tablomuza ekliyoruz.
```

Metodumuz artık hazır ister butona tıklandığında ister form açıldığında çağırabiliriz. Ben form açıldığında çağırıyorum. “show_new_movies();” şeklinde.

- **Arama**

Film arama işlemini sinemalar formumda yapıyorum. Boş alana film ID si giriliyor ve film varsa film adı resmi ve yönetmenini getiriyor. Eğer yoksa



Peki bu işlemleri nasıl yapıyoruz şimdi buna göz atalım.

Yukarıdaki görselde olduğu gibi jTextField alanına bir sayı giriliyor ve anında gösteriliyor. Bu özellik ise jTextField in KeyReleased özelliği oluyor. Tuşa basıldığı anda değişim yaşanıyor.

Biz de kodlarımızı buraya yazacağız.

Diğer veri tabanı işlemlerinde olduğu gibi aynı mantık işliyor. Bu yüzden değişen kısımlara göz atalım.

*String sql = "SELECT * FROM filmler WHERE Id=?";* bütün tabloyu çağırıyorum ID ye göre
pst.setString(1, txt_FilmAra.getText()); buradaysa jTextField a ID giriyorum

ResultSet rs = pst.executeQuery();

SQL Sorgusundan dönecek sonuç rs sonuç kümesi içinde tutuluyor.

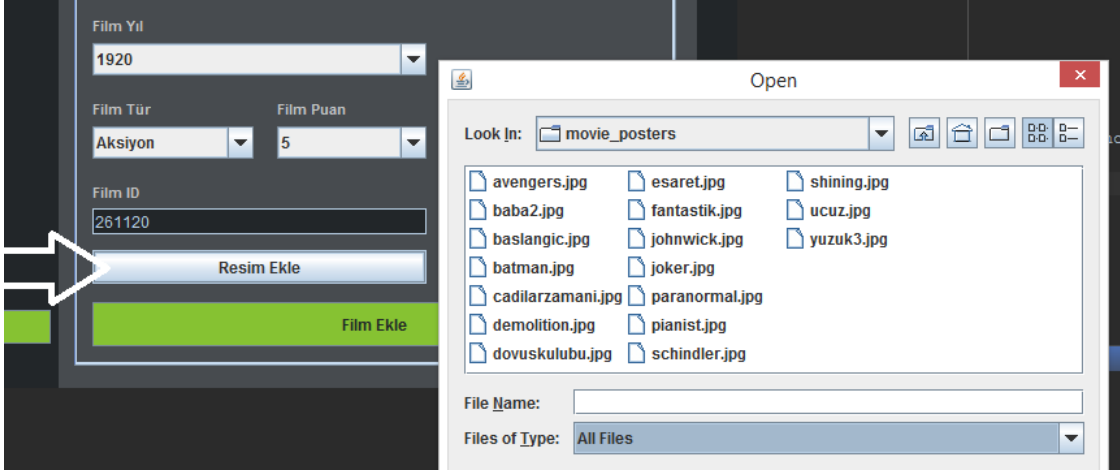
if(rs.next()){ **ID sütununa göre değerlerimiz değişiyor**

Buranın içerisindeyse 3 label koydum dönen değerleri getString ile alıyorum, setText olarak yazdırıyorum.

```
String setAd=rs.getString("FilmAd");  
lbl_filmAdi.setText("Film Adı: "+setAd);  
}
```

- **Resim çekme-ekleme**

Resim eklememiz için tabii bir yerlerden seçmemiz gerekiyor. Bunun için FileChooser oluşturdum. Resim Seç butonuna tıklayınca bir resim seçmemizi istiyor.



Resim işlemleri için öncelikle veri tabanında 'FilmResimler' adında sütun oluşturdum ve veri tipini image olarak kaydettim.

Film.java sınıfıma ise *private byte[] picture;* **byte tipinden Picture adında değişken oluşturdum.**

String fileName=null; **dosya yolu için değişken**

byte[] person_image =null; **byte tipinde resmi tutmak için değişken**

pst.setBytes(8, person_image); **person image yi setByte ederek veri tabanına ekliyoruz**

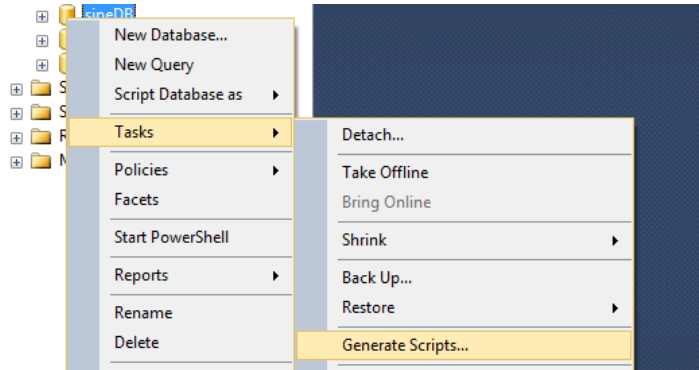
6. Backup(yedek) nasıl aldım?

- **Java program yedeği**

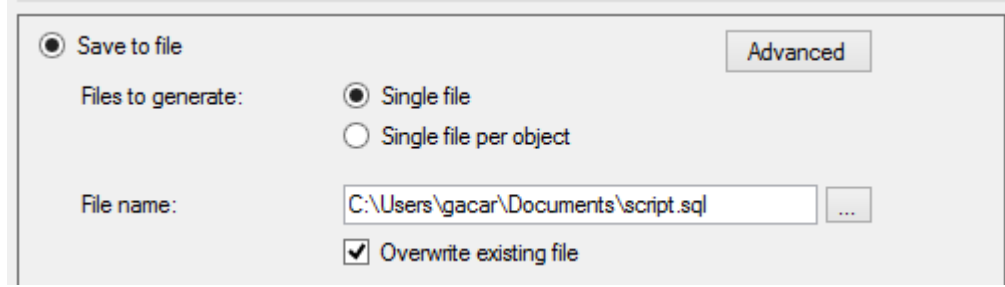
Projem NetBeansProjects adlı klasörde yer alıyor, projemi buradan kopyaladım.

- **Veri tabanı yedeği**

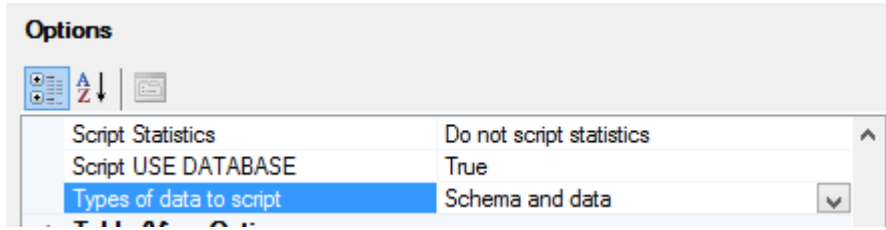
Tablolarımın yedeğini almak için öncelikle SQL Management Studio programını açtım sonrasında Databases kısmında veri tabanımı seçtim 'sineDB' sağ tıkladım>Tasks>Generate Scripts



Açılan sekmede 2 kere Next diyoruz.



Advanced butonuna tıklıyoruz.



Types of data to script kısmını Schema and data diyoruz ve işlemimizi bitiriyoruz.

Sonuç:

Sinema Otomasyonu olan sinego adlı projemde içerik olarak çok detaylı olmasada veri tabanı işlemleri bulunuyor. Ekleme, silme, güncelleme, veri çekme ve yönetici girişi gibi temel işlemler yer alıyor.

Zaman Çizelgesi:

Proje başlangıç tarihi: 16 Kasım

Proje bitiş tarihi: 9 Aralık

Toplam süre: 25-35 saat.

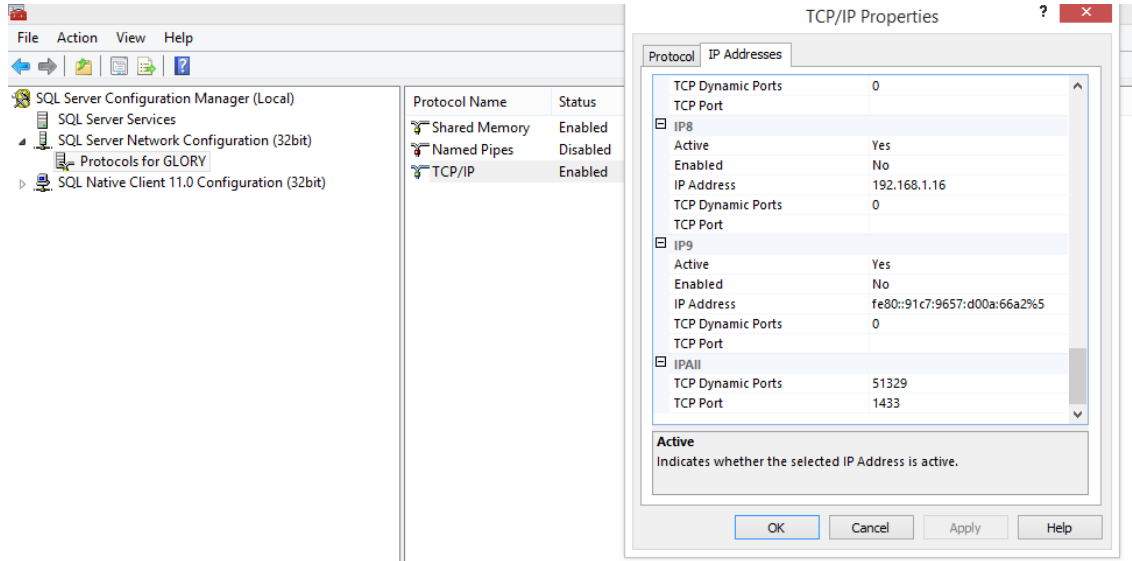
Ekler:

Veri tabanı bağlantı sırasında birçok hata aldım bunları nasıl çözdüm biraz bunları görsellerle anlatalım.

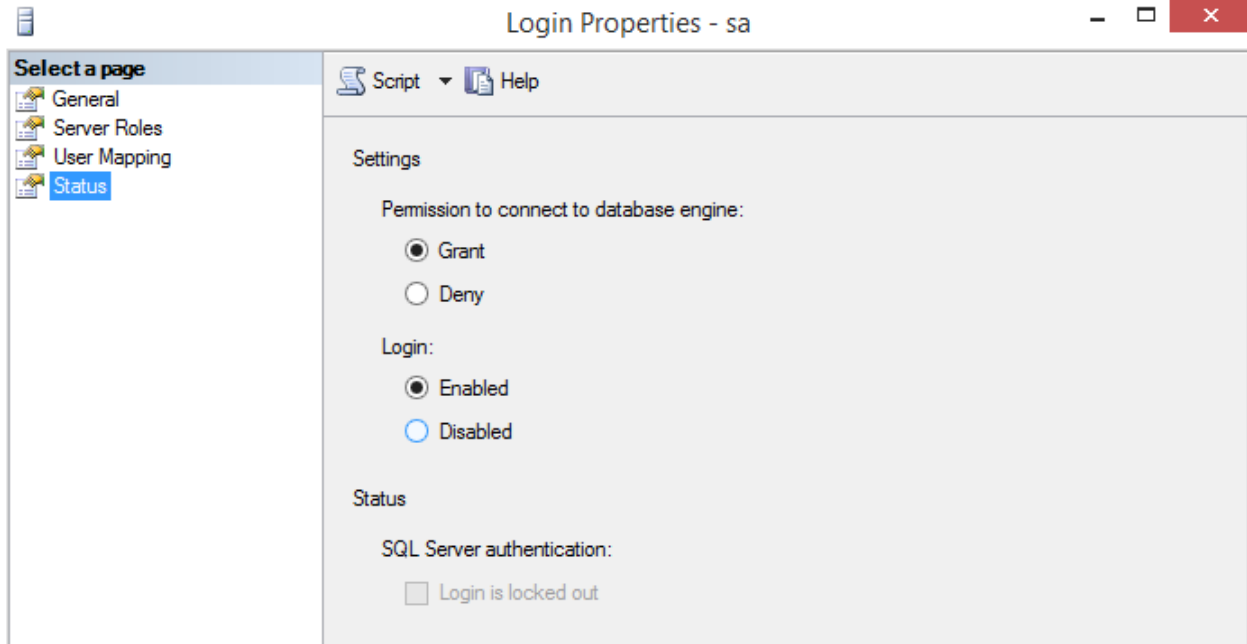
İlk olarak Configuration Manager kısmında SQL Server ımızın açık olması gerekiyor.

SQL Server (GLORY)	Running
SQL Full-text Filter Daemon Launcher (GLORY)	Stopped
SQL Server Reporting Services (GLORY)	Stopped
SQL Server Agent (GLORY)	Stopped
SQL Server Browser	Stopped

String url verirken localhost:1433 demiştik peki bunu nasıl yapıyoruz? Yine Configuration Manager kısmına giriyoruz. TCP/IP bölümünü enabled yapıyoruz ve çift tıklıyoruz, IP Addresses bölümüne giriyoruz en aşağı inip IPII kısmını görseldeki gibi yapıyoruz.



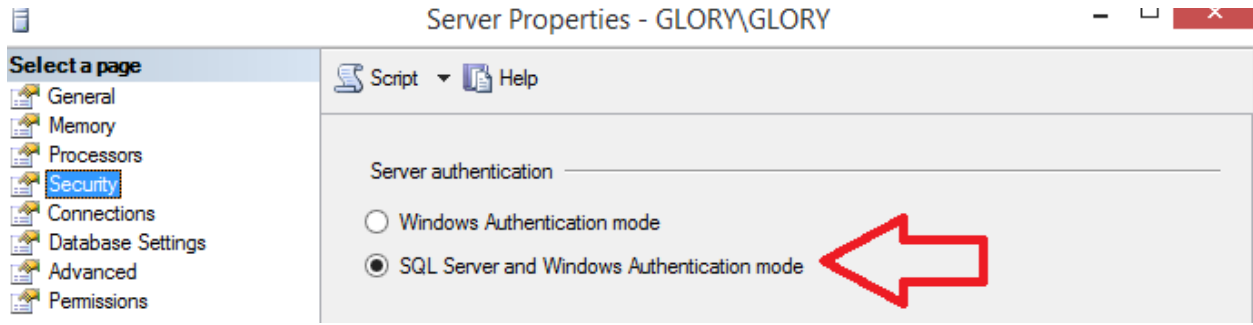
Aldığım başka bir hata ise sa adlı kullanıcı girişini yapamıyordum bunun için login kısmını enabled yapmamız yetiyor.



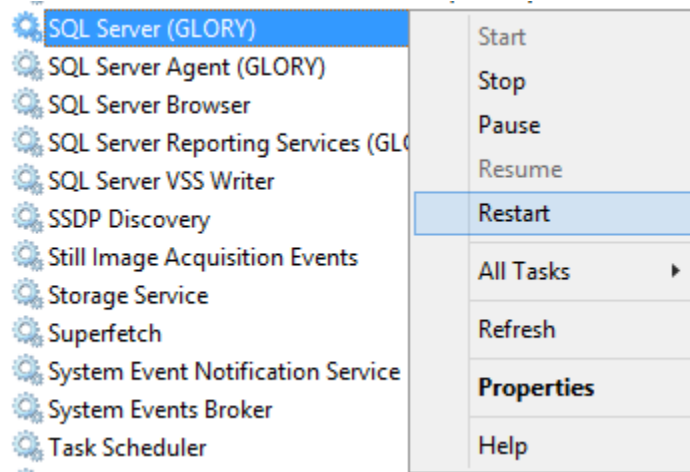
Peki sa kullanıcı adı ile giriş yaptık bu sefer de veri tabanı oluşturamıyoruz, bunun içinse SQL Management Studio'yu yeniden başlatıyoruz normal Windows Authentication olarak giriyoruz. Sonrasında kullanıcı adımızın olduğu üst kısma sağ tıklıyoruz...



Security kısmına giriyoruz ve SQL Server and Windows Authentication mode seçeneğini seçiyoruz.



Bu işlemleri yaptıysak en son olarak services (hizmetler) giriyoruz. Sağ tıklayıp yeniden başlatıyoruz. Artık her şey problemsiz çalışabilir halde.



FileChooser kullanımı

```
private void btn_resimSEActionPerformed(java.awt.event.ActionEvent evt) {  
    //resim ekleme butonu -- file chooser  
  
    JFileChooser chooser = new JFileChooser();  
    chooser.showOpenDialog(null);  
    File f = chooser.getSelectedFile();  
    fileName= f.getAbsolutePath();  
    ImageIcon imageIcon;  
    imageIcon= new ImageIcon(new ImageIcon(fileName).getImage().getScaledInstance  
    (lbl_image.getWidth(),lbl_image.getHeight(),Image.SCALE_SMOOTH));  
    lbl_image.setIcon(imageIcon);  
    try{  
        File image = new File(fileName);  
        FileInputStream FIS = new FileInputStream(image);  
        ByteArrayOutputStream BOS = new ByteArrayOutputStream();  
        byte[] buf = new byte [1024];  
        for(int readNum; (readNum=FIS.read(buf)) !=-1;){  
            BOS.write (buf,0, readNum);  
        }  
        person_image=BOS.toByteArray();  
    }  
    catch(Exception e){  
        JOptionPane.showConfirmDialog(null, e);  
    }  
}
```

Kaynakça:

<https://stackoverflow.com>

<https://medium.com>

<https://kod5.org>

<https://www.yusufsezer.com.tr>

<https://www.yazilimkodlama.com>

<https://www.youtube.com/channel/UCi-uTy8YErITPyljHOuu3yw>

<https://www.youtube.com/user/ProgrammingKnowledge>