

Java ile Veri Tabanı Bağlantısı

Bildiğiniz üzere projenizde veri tabanı bağlantısı yapmanız gerekmektedir. Aşağıda java ile veri tabanı bağlantısının nasıl yaptığı açıklanmaktadır. Veri tabanı yönetim sistemi olarak MSSQL kullanılmıştır. Diğer VTYS ler (Oracle, MySQL, vb.) ile de çalışabilirsiniz fakat diğerlerinin bağlantısı temelde aynı olup birtakım farklılıklar göstermektedir. Dolayısıyla MSSQL ile çalışmak yararınıza olacaktır.

Adım-1:

İlk önce MSSQL kurulumunu yapınız.

Adım-2:

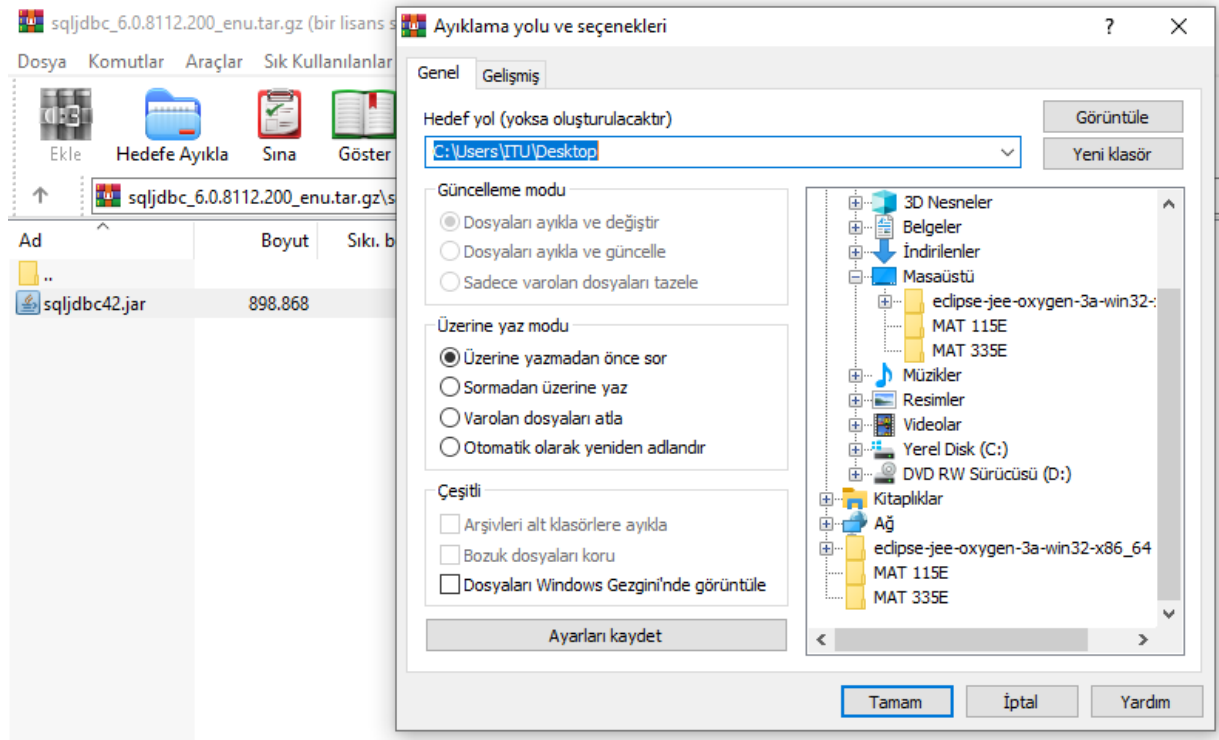
<https://www.microsoft.com/en-US/Download/details.aspx?id=11774>

bağlantısını kullanarak bu bağlantıda yer alan

sqljdbc_6.0.8112.200_enu.tar.gz

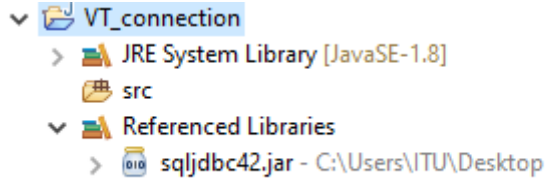
zip dosyasını indiriniz. Bu zip dosyasının içerisinde **jre7** ve **jre8** olmak üzere 2 farklı klasör bulunmaktadır. Eclipse üzerinde hangi jre'yi kullanıyorsanız ilgili klasörün içerisinde bulunan .jar uzantılı dosyayı dizine çıkartınız.

Bende jre8 yüklü olduğundan sqljdbc42.jar dosyasını dizine çıkartıyorum. jre7 veya jre8 den farklı bir jre kullanıyorsanız kullandığınız jre ile uyumlu bir sqljdbc(driver) zip dosyası indirmelisiniz.



Adım-3:

Eclipse açıyoruz. Yeni bir proje oluşturuyoruz. Daha sonra projeye sağ tıklayıp **'Build Path' → 'Add External Archives...'** diyoruz ve dizine çıkarttığımız dosyayı seçip , ekliyoruz.



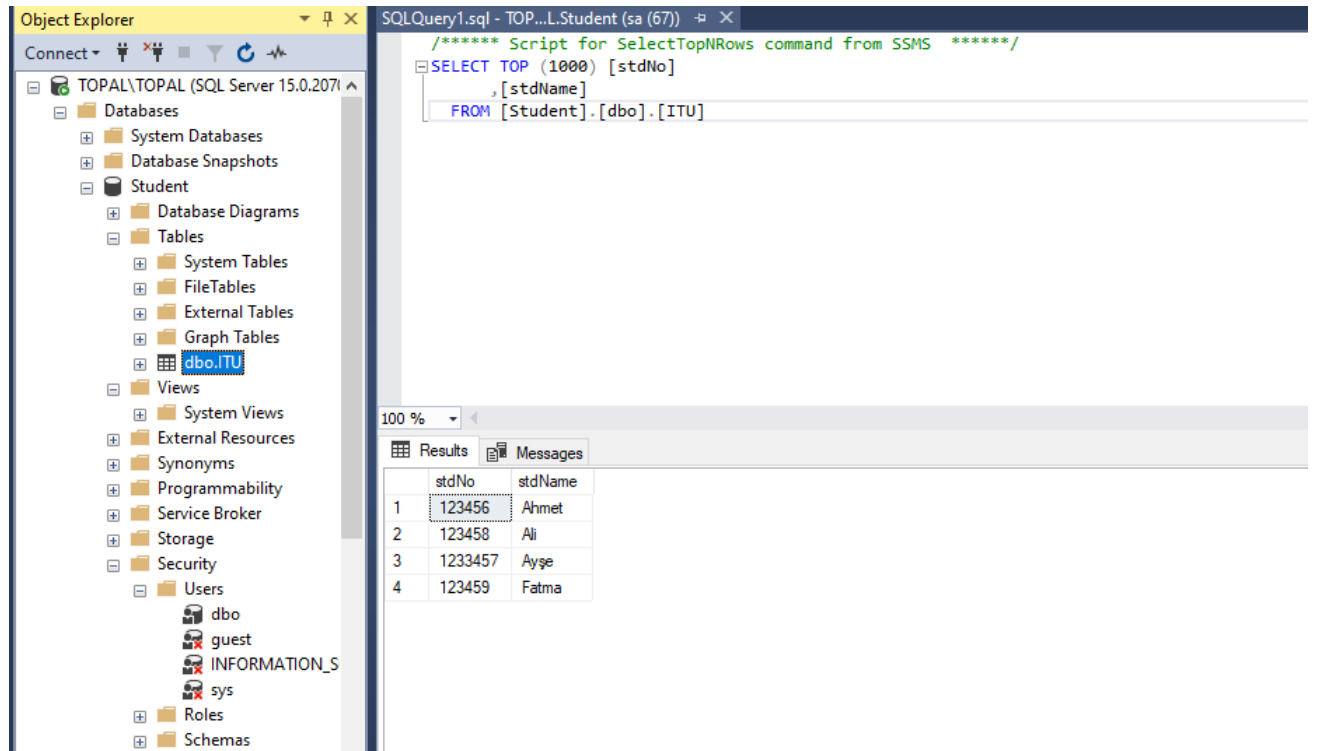
Görüldüğü üzere Referenced Libraries kısmında yüklendiğini görüyoruz. Dolayısıyla gerekli olan yüklemeleri yaptık.

Adım-4:

Şimdi sıra kod üzerinden MSSQL veri tabanı yönetim sistemine bağlanalım.

Deneme adında bir sınıf oluşturuyorum. Bu sınıfta yazacağım kodlar ile VTYS bağlantısını kuracağım ve örnek bir sorgu yazıp çalıştıracam.

Bundan önce VTYS üzerinde Student diye bir veri tabanı oluşturuyorum. Bu veri tabanının altında ITU diye bir tablo oluşturuyorum ve birkaç kayıt ekliyorum. ITU tablosunun görünümü aşağıdadır.



Adım-5:

Bağlantı yapmak ve sorguyu çalıştırmak için gerekli tüm java kodlarıdır. Bunları adım adım açıklayalım:

```
import java.sql.*;
public class deneme {
    public static void main() {
        try
        {
            //Make a connection
            String myDriver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
            String db = "jdbc:sqlserver://127.0.0.1:1433;database=Student;user=sa;password=123";
            Class.forName(myDriver);
            Connection conn = DriverManager.getConnection(db);

            //Create a query and execute it
            String sorgu = "SELECT * FROM dbo.ITU";
            Statement st = conn.createStatement();
            ResultSet rs = st.executeQuery(sorgu);

            while (rs.next())
            {
                String s_ad = rs.getString("stdNo");
                String s_soyad = rs.getString("stdName");

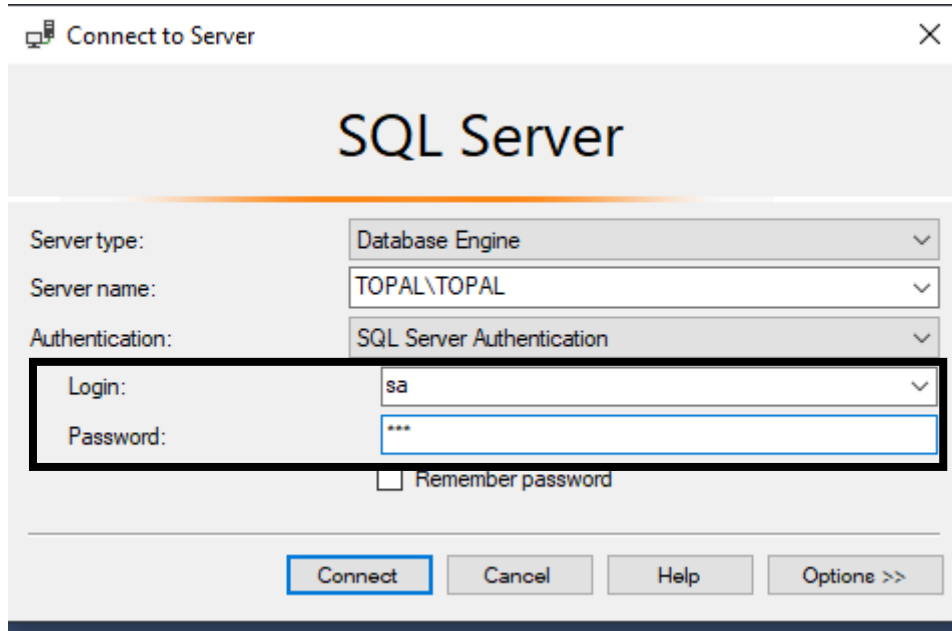
                System.out.print(s_ad + " " + s_soyad + "\n");
            }
            st.close();
        }

        catch (Exception a)
        {
            System.err.println("Hata ! ");
            System.err.println(a.getMessage());
        }
    }
}
```

```
    //Make a connection
    String myDriver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
    String db = "jdbc:sqlserver://127.0.0.1:1433;database=Student;user=sa;password=123";
    Class.forName(myDriver);
    Connection conn = DriverManager.getConnection(db);
```

- Mssql ile çalışacak arkadaşlar 1.satırı(myDriver) olduğu gibi tanımlamalıdır.
- 2.satırda 127.0.0.1:1433 olarak localhost belirtiyoruz. Daha sonra sırasıyla hangi veritabanı üzerinde işlem yapılıyorsa onun adını belirtiyoruz. Örneğin, ben Student adlı bir veritabanı ile çalışıyorum. Daha sonra veritabanına bağlanırken kullandığınız kullanıcı adı ve şifrenizi ekliyorsunuz.

Bu alanlar MSSQL üzerinde aşağıda gösterilmiştir.



- Aşağıdaki satırlarda ise örnek bir sorgu cümlesi yazıyorum. Bu sorgu çalıştığında ITU tablosundaki tüm satırları **rs** değişkeninde tutacaktır.

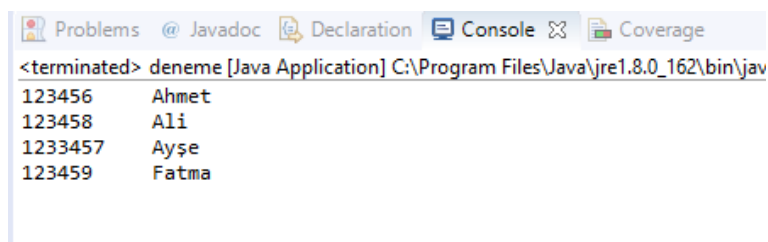
```
//Create a query and execute it
String sorgu = "SELECT * FROM dbo.ITU";
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery(sorgu);

while (rs.next())
{
    String s_ad = rs.getString("stdNo");
    String s_soyad = rs.getString("stdName");

    System.out.print(s_ad + " " + s_soyad + "\n");
}
st.close();
}
```

- While döngüsü içerisinde tablodaki ilgili değerleri satır bazında okuyup ekrana yazdırma işlemini yapıyorum.

Çıktı:



```
<terminated> deneme [Java Application] C:\Program Files\Java\jre1.8.0_162\bin\jav
123456      Ahmet
123458      Ali
1233457     Ayşe
123459      Fatma
```

Java ile MSSQL bağlantısını tamamlamış ve Basit bir statik sorgu üzerinden veri tabanından satırların nasıl çekildiğini göstermiş olduk.

Ayrıca buradaki try ve catch blokları hata işleme ile ilgili özel bloklardır. Eğer try bloğunda yer alan kodlarda herhangi bir hata meydana gelmişse o hatayı catch blokları tarafından yakalamaktır.