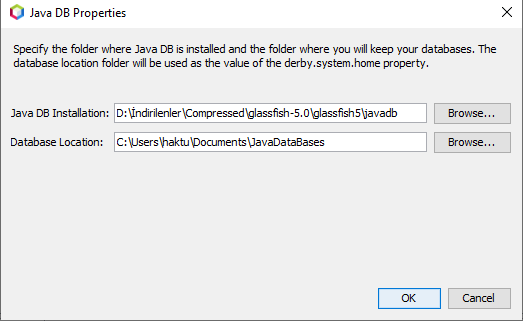


Bu **proje ismine** ve **main class ismine** sahip bir proje oluşturuyoruz.(Java with Ant Türünde)

Ardından netbeans Services sekmesinden Java DB yi kontrol ediyoruz,

**Java DB seçeneğine sağ tık yaptığımızda Create Database, Start Database vb. Tıklanamaz durumda ise**:

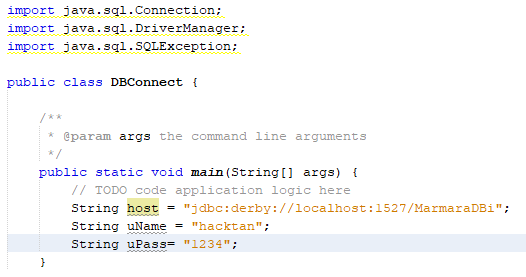
aynı menüden Properties kısmından javadb klasörünün yolunu göstermemiz gerekiyor. Oracle javadb klasörünün jdk kurulumunun yanında geldiğini iddia etmiş ancak baştan kurmama rağmen kendi bilgisayarımda böyle bir klasör yok. Bu sorunun çözümü olarak internette GlassFish klasörünü buldum ve bu dökümantasyonun yanına da ekledim. GlassFish’in içindeki javadb klasörünü kullanacağız.

Properties sekmesinde

**Java DB seçeneğine sağ tık yaptığımızda Create Database, Start Database vb. Seçilebilir durumda ise:**

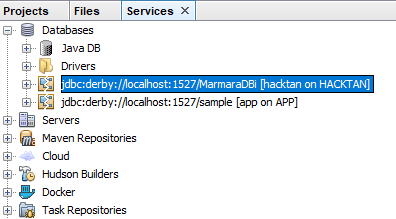
Java DB sağ click -> Create Database diyip Veri Tabanı ismi , kullanıcı adı ve şifresi girilip oluşturulur.

Java DB sağ click -> start server



Kodlara bu Kısmı ekliyoruz

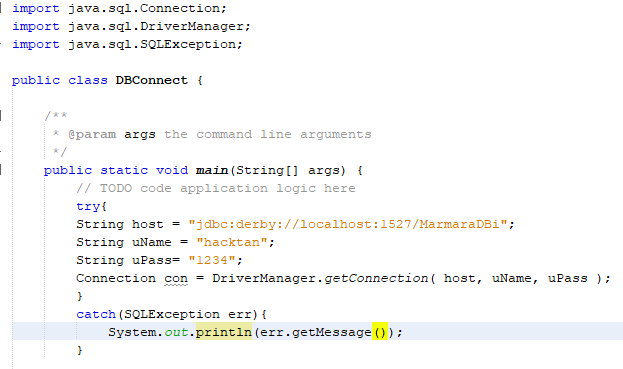
String host = "jdbc:derby://localhost:1527/MarmaraDBi";

Bu kısmı açtığımız veri tabanından alıyoruz, Services sekmesinden veri tabanını bulup isminin sondaki köşeli parantez kısmına kadar olan kısmını alıyoruz

Sonra kodun devamına şunu ekliyoruz:

Connection con = DriverManager.getConnection( host, uName, uPass );

Kod buraya kadar hata verebilir, try catch bloğuna almak gerekiyor.



Çalıştırınca Bu hatayı alabilirsiniz:

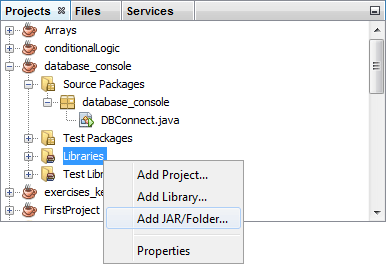
**"java.net.ConnectException : Error connecting to server localhost on port 1527 with message Connection refused: connect."**

Services sekmesinden Java DB sağ click Start Server.

Sonrasında her şey düzgünse çalıştırınca şu mesaj gelmeli:

**"No suitable driver found for jdbc:derby://localhost:1527/MarmaraDBi"**

Sonra,

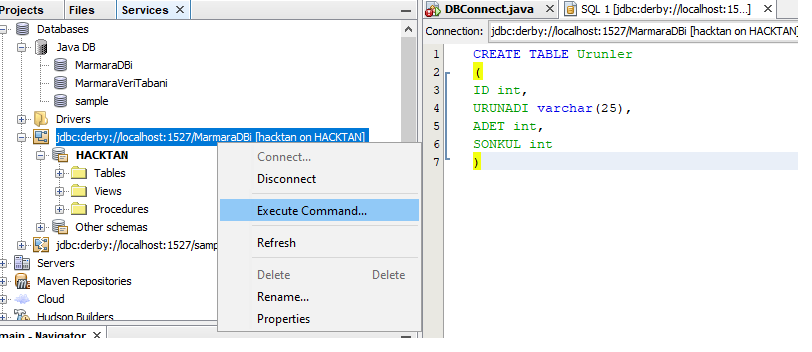


**C:\Program Files\Java\jdk1.8.0\_45\db\lib klasöründen**

**veya GlassFish kullanmak gerektiyse \glassfish-5.0\glassfish5\javadb\lib**

klasöründen, derbyclient.jar dosyası seçilmelidir.

Şimdi çalıştırıldığında Build Successful.Demelidir.

Şimdi sekmelerden veri tabanımıza gidip örnek bir tablo oluşturalım

Execute command diyip yandaki kodları girip run SQL yapalım(ctr+ shift + e)

Tablomuz oluşmuş olmalıdır.

Aynı execute command kısmından birkaç örnek veri ekliyoruz

INSERT INTO URUNLER

VALUES (15, 'ayran', 15, 2020)

INSERT INTO URUNLER

VALUES (5, ‘çitos’, 15, 2020)

Gibi veriler ekledik.

Sonra java dosyasına kodlar ekliyoruz. Son halini direkt ekledim:

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package database\_console;

/\*\*

\*

\* @author haktu

\*/

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import java.sql.ResultSet;

public class DBConnect {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

try{

String host = "jdbc:derby://localhost:1527/MarmaraDBi";

String uName = "hacktan";

String uPass= "1234";

Connection con = DriverManager.getConnection( host, uName, uPass );

Statement stmt = con.createStatement();

String SQL = "SELECT \* FROM URUNLER";

ResultSet rs = stmt.executeQuery( SQL );

while(rs.next( )){

int id = rs.getInt("ID");

String name = rs.getString("URUNADI");

String amount = rs.getString("ADET");

String lastUse = rs.getString("SONKUL");

System.out.println( id + " " + name + " " + amount + " " + lastUse );

}

}

catch(SQLException err){

System.out.println(err.getMessage());

}

}

}

}

Çıktı:

