**JDBC DERS NOTLARI**

**1-Veri tabanı sürücü sınıfını kaydetme**

**Class.forName("com.mysql.cj.jdbc.Driver"); //Java 7’den sonraki sürümlerde bunu yazma zorunluluğu kalktı**

**2-DBMS bağlantısı kurma**

**Connection con=DriverManager.getConnection ( “jdbc:hostName:databaseName ", username , password);**

**3-Statement oluşturma**

**Statement st = con.createStatement();**

**4-Sorguyu çalıştırma**

**ResultSet veri = st.executeQuery("select \* from table\_name");**

**5-Bağlantıyı kapatma**

**con.close();**

**İntellij ile başlıyoruz:**

**İNTELLİJ**

**File => New => Project => New Project**

**=> Name : jdbcLesson, Location, Language:Java, Build system : Maven, JDK,**

**=> Advenced Settings**

**-> GroupId : com.tpe (Firma ismi)**

**-> ArtifactId : jdbcLesson (Bu firmanın içinde bulunduğu proje ismi) => CREATE**

**Kütüphane Oluşturma :**

**Pom.xml < işaretini yazınca dependencies üzerine çift tıkla. (properties’in dışında,Project’in içinde olamlı)**

**< dependencies>**

**Tarayıcıya “maven repository” yaz ->** <https://mvnrepository.com> **gir -> Arama çubuğuna ‘postgresql jdbc driver’ yaz -> En yeni güncellenmiş kütüphaneyi seç ve tıkla -> Versiyon seç (en çok kullanılanlardan seç 42.6.0) -> Kutu içinde sol tıkla; metin seçilmiş olacak -> BURAYA YAPIŞTIR -> Postgresql lehçesi entegre olmuş oldu. JDBC’yi kullanarak Postgresql’e istediğimiz işlemleri yapabilecek duruma geldik -> Pom.xml’yi kapat**

**</dependencies>**

**Yazacğımız kodları src -> main-> java altında oluşturuyoruz. Önce Java altında kısa yoldan package’lerimizi (com.tpe.jdbc.) ve clas’ımızı oluşturalım.**

**Java => New => Class : com.tpe.jdbc.Executed01 => ENTER**

**Main method oluşturduktan sonra java ile pgAdminde lod oluşturmak için 5 adımlı bir işlem yapmak gerekiyor =>**

**1.ADIM : Driver'i ekleme**

**A) pgAdmin’de bir user (Kullanıcı) oluşturmamız gerekiyor. Bunun için;**

**Login/Group Roles sağ tık => Create => Login/Group Roles => General sekmesinde Name : techprodt => Definition sekmesinde Password : password => Privileges sekmesinde Can login (Kullanıcı sisteme login olsun mu? ) aktif yapıyoruz , Superuser aktif yapıyoruz => SAVE**

**B) pgAdmin’de database oluşturuyoruz**

**Databases sağ tık => Create => Database -> jdbc\_dt, Owner -> techprodt => SAVE**

**pgAdmin’de işimiz bitti. İntellije geçiyoruz ve driver ekliyoruz**

**Class.forName("org.postgresql.Driver"); // Driver’ın adresini vermiş olduk. JDK 7 ile birlikte buna gerek kalmadi**  
**2.ADIM : Hangi DB , Hangi kullanici ve sifre  
Connection con =  
 DriverManager.getConnection("jdbc:postgresql://localhost:5432/jdbc\_dt","techprodt","password");**  
**3.ADIM: statement olusturma, yazacagimiz query'leri statement nesnesi ile DB ye gonderecegiz**  
**Statement st = con.createStatement();  
System.out.println("success");**  
**4.ADIM: sorgulari oluşturma (execute ile)**

**ÖRNEK 1:"workers" adında bir tablo oluşturup "worker\_id,worker\_name,salary" sütunlarını ekleyiniz.**  
**boolean sql1 =  
st.execute("CREATE TABLE IF NOT EXISTS workers(worker\_id INT,worker\_name VARCHAR(50),salary REAL)");  
System.out.println("sql1 : " + sql1);**

**ORNEK 2 : "workers" tablosuna VARCHAR(20) tipinde "city" sutununu ekleyelim**  
**st.execute("ALTER TABLE workers ADD COLUMN city VARCHAR(20)”);**

**ORNEK 3 : "workers" tablosunu silelim //Ornek 2'yi comment'e almamız gerekiyor  
st.execute("DROP TABLE WORKERS”);**

**execute:tüm sorguları çalıştırmak için kullanılır  
sorgunun sonucunda ResultSet alınıyorsa TRUE döndürür, aksi halde FALSE döndürür.  
1-DDL (Data Definition Language) -->FALSE ( Create gibi sorgular )  
2-DQL (Data Query Language) -->TRUE (Select gibi sorgular )  
2-DML (Data Manipulation Language) --> FALSE ( update, insert gibi sorgular )  
genellikle DDL için kullanılır.**

**5.ADIM: kaynakları kapatıyoruz**

**st.close();**

**con.close();**