# VERİ TABANI DERS NOTLARI

**Veri Tabanı İşlemleri**

**Veri Çağırma**

* SELECT \* FROM ogrenciler;(ogrenciler tablosundaki tüm verileri getir.)
* SELECT adi,soyadi FROM ogrenciler;(ogrenciler tablosundaki kişilerin adi ve soyadini getir.)
* SELECT adi,soyadi FROM ogrenciler WHERE ogrenciID=2; (ogrenciler tablosundan ogrenciID si 2 olan kişilerin adi ve soyadini getir.)
* SELECT \* FROM ogrenciler WHERE ogrenciID=1; (ogrenciler tablosundan ogrenciID si 1 olan kişilerin tüm verilerini getir.)
* SELECT \* FROM ogrenciler WHERE adi=’Samet’ AND soyadi=’Çavur’; (ogrenciler tablosundan adi Samet ve Soyadi Çavur olan kişilerin tüm verilerini getir.)
* SELECT adi,soyadi FROM ogrenciler WHERE adi=’Samet’; (ogrenciler tablosundan adi Samet olan kişilerin adi ve soyadini getir.)

---------------------------------------------------------------

**Veri Ekleme**

* INSERT INTO ogrenciler(ogrenciID,adi,soyadi,dogumYili,okulNo)

VALUES (3,'Mehmet','Hakan',1990,'1236');(ogrenciler tablosuna yandaki girilen verileri ekle)

---------------------------------------------------------------

**Veri Silme**

* DELETE FROM ogrenciler WHERE ogrenciID=3;(ogrenciler tablosundan ogrenciID si 3 olan kisinin verilerini sil.)

---------------------------------------------------------------

**Veri Güncelleme**

* UPDATE ogrenciler SET adi="AbdulSamet" WHERE ogrenciID=1; (ogrenciID si 1 olan verinin adini AbdulSamet olarak değiştir,where burada ‘Samet’ yazamayız çünkü Samet adında bir çok veri olabilir ve bunların hepsinin adını AbdulSamet yapmak yanlış olabilir.)
* UPDATE ogrenciler SET adi="Hakkı", soyadi="Baş", dogumYili=1991 ,okulNo="1532" WHERE ogrenciID=2; (ogrenciID si 2 olan kisinin verilerini set komutunda girdiğimiz veriler ile değiştir.)

---------------------------------------------------------------

**Tablo Oluşturma**

* CREATE TABLE adres(adresID int PRIMARY KEY NOT NULL,sokak VARCHAR(15),mahalle VARCHAR(15),sehir VARCHAR(15)); (adres adında,primary keyi adresID olan ve boş bırakılamaz olan,max 15 harfli sokak,mahalle,sehir sütunları olan bir tablo oluşturdur)

# JDBC DERS NOTLARI

**1.Veri Tabanına Bağlanma**

Bağlantıyı kurmamız için ilk önce çalışacağımız java projesine kütüphane olarak connector jar bağlamamız gerek bunun içinse soldan projenin üsteki sağ tıkladık BuildPath>AddLibraries>UserLibraries>UserLibraries>New dedik ve Bir sql paket ismi verdik sonra sağdaki Add ExternalJARS’a basıp aşşağıdaki linten indirdiğimiz JARI ekledik. https://dev.mysql.com/downloads/connector/j/8.0.html (Select Operating Independent=Platform Independent-ZIP Archvive)

**\*Statement yerine PreparedStatement kullanmak uygulamayı daha dinamik hale getirir.**

**---dbHelper Classı---**

Bu classın amacı veri tabanını javaya bağlarken sürekli uğraşmamak,böyle bir class oluşturuyoruz ve içine sqle bağlanma metodunu yazıyoruz.

**import** java.sql.\*; //Bunu her zaman yazıyoruz.

**public** **class** **dbHelper** {

**public** **Connection** getConnection() **throws** **SQLException** { **return DriverManager**.*getConnection*("jdbc:mysql://localhost:3306/world?useSSL=false&useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC","root","1234");} /\*(world veritabanındaki tablomuzun adı,root UserID miz ve 1234 ise UserPasswordumuz,Veritabanını kurarken en başta belirledik.)\*/

**public** **void** showErrorMessage(**SQLException** e) { //(Bu bağlantı kurarken hata çıkmasına karşın yazdığımız metod)

**System**.*out*.println("SQL Hatası:"+ e.getLocalizedMessage());

**System**.*out*.println("Hata Kodu:"+ e.getErrorCode()); }}

**---MAİN CLASSI---**

**import** java.sql.\*; //Bunu her zaman yazıyoruz.

**public** **class** **main** {

**public** **static** **void** main(**String**[] args) **throws** **SQLException** {

**Connection** connection = **null**;

**dbHelper** dbHelper = **new** dbHelper();

**try** {

connection = dbHelper.getConnection();

**System**.*err*.println("SQL Bağlantısı Oluştu");

}

**catch** (**SQLException** e) {

dbHelper.showErrorMessage(e);

}

**finally** {

connection.close();

}

}}

**2.Veri tabanındaki verileri çağırıp Javada gösterme(SELECT)**

**MAİN CLASSI**

**public** **static** **void** main(**String**[] args) **throws** **SQLException** {

**Connection** connection = **null**;

**Statement** statement=**null**; //Şöyle bir değişken oluşturduk

**ResultSet** resultset; //Şöyle de bir değişken oluşturduk,Bunu sadece Select işleminde kullanıyoruz.

**dbHelper** dbHelper = **new** dbHelper();

**try** {

connection = dbHelper.getConnection();

**System**.*err*.println("SQL Bağlantısı Oluştu");

statement = connection.createStatement();

resultset = statement.executeQuery("SELECT Name,District FROM city;");

**while**(resultset.next()) {

**System**.*out*.print(resultset.getString("Name")); //Name Verilerini Getir

**System**.*out*.println("/"+resultset.getString("District")); //District Verilerini Getir. }}

**catch** (**SQLException** e) {

dbHelper.showErrorMessage(e);}

**finally** {

connection.close(); //En son Try Catchi kapatmayı unutmuyoruz. }}}

**dbHelper Classı**

**import** java.sql.\*; //Bunu her zaman yazıyoruz.

**public** **class** **dbHelper** {

**public** **Connection** getConnection() **throws** **SQLException** { **return DriverManager**.*getConnection*("jdbc:mysql://localhost:3306/world?useSSL=false&useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC","root","1234");} /\*(world veritabanındaki tablomuzun adı,root UserID miz ve 1234 ise UserPasswordumuz,Veritabanını kurarken en başta belirledik.)\*/

**public** **void** showErrorMessage(**SQLException** e) { //(Bu bağlantı kurarken hata çıkmasına karşın yazdığımız metod)

**System**.*out*.println("SQL Hatası:"+ e.getLocalizedMessage());

**System**.*out*.println("Hata Kodu:"+ e.getErrorCode()); }}

**3.Javadan Veri Tabanına Veri Ekleme(INSERT INTO)**

==Üstteki dbHelper Classı yine var olacak

**import** java.sql.\*;

**public** **class** **main** {

**public** **static** **void** main(**String**[] args) **throws** **SQLException** {

**Connection** connection = **null**;

**PreparedStatement** statement = **null**;

**dbHelper** db = **new** dbHelper();

**try** {

connection = db.getConnection();

**System**.*out*.println("SQL Connection is succesfully");

statement = connection.prepareStatement("insert into city (Name,CountryCode,District,Population) VALUES ('Balik','TUR','Balikesir',350000)");

statement.executeUpdate();

**System**.*out*.println("Kayıt Eklendi.");}

**catch** (**SQLException** e) {

**System**.*out*.println("SQL Connection is not succesfully");

db.showErrorMessages(e); }

**finally** {

statement.close();

connection.close(); }}}

**Eklenecek Veriyi Kullanıcıdan Alma İşlemi**

dbHelper Classı var

**public** **class** **main** {

**public** **static** **void** main(**String**[] args) **throws** **SQLException** {

**Connection** connection = **null**;

**PreparedStatement** statement = **null**;

**dbHelper** db = **new** dbHelper();

**try** {

connection = db.getConnection();

**System**.*out*.println("SQL Connection is succesfully");

**String** sql = "insert into city (Name,CountryCode,District,Population) VALUES (?,?,?,?)";

statement = connection.prepareStatement(sql);

statement.setString(1, "Zeytinburnu");

statement.setString(2, "TUR");

statement.setString(3, "ISTA");

statement.setInt(4, 1234567);

statement.executeUpdate();

**System**.*out*.println("Kayıt Eklendi."); }

**catch** (**SQLException** e) {

**System**.*out*.println("SQL Connection is not succesfully");

db.showErrorMessages(e); }

**finally** {

statement.close();

connection.close(); }

}

}

**4. Veri Tabanına Toplu Veri ekleme( BATCH INSERT INTO)**

**public** **class** DbProcces {

**public** Connection **connection**(List<Personal> personal) {

Logger **logger** = LogManager.*getLogger*();

PreparedStatement **statement** = **null**;

DbConnection **db** = **new** DbConnection();

Connection **connection** = db.openConnection();

String **sorgu** = "INSERT INTO personal(personalID,adi,soyadi,dogumYili,personelNo) VALUES (?,?,?,?,?)";

**try** {

statement = connection.prepareStatement(sorgu);

**for** (Personal **personel** : personal) {

statement.setInt(1, personel.getPersonalID());

statement.setString(2, personel.getAdi());

statement.setString(3, personel.getSoyadi());

statement.setInt(4, personel.getDogumYili());

statement.setInt(5, personel.getPersonalNo());

statement.addBatch();

}

statement.executeBatch();

logger.info("Sorgu Çalıştı");

} **catch** (SQLException **e**) {

logger.warn("Sorgu Çalışmadı" + e);

}

**return** connection;

}

}

**5.Javadan VeriTabanındaki Verileri Güncelleme-Değiştirme (UPDATE)**

**import** java.sql.\*;

**public** **class** **main** {

**public** **static** **void** main(**String**[] args) **throws** **SQLException** {

**Connection** connection = **null**;

**PreparedStatement** statement = **null**;

**dbHelper** db = **new** dbHelper();

**try** {

connection = db.getConnection();

**System**.*out*.println("SQL Connection is succesfully");

**String** sql = "update city set name ='Bestelsiz' where ID='4086'"; statement = connection.prepareStatement(sql);

statement.executeUpdate();

**System**.*out*.println("Kayıt Güncellendi.");

}

**catch** (**SQLException** e) {

**System**.*out*.println("SQL Connection is not succesfully");

db.showErrorMessages(e);

}

**finally** {

statement.close();

connection.close(); } }}

**Kullanıcı tarafından Veritabanındaki Veriyi Güncelleme**

**Connection** connection = **null**;

**PreparedStatement** statement = **null**;

**dbHelper** db = **new** dbHelper();

**try** {

connection = db.getConnection();

**System**.*out*.println("SQL Connection is succesfully");

**String** sql = "update city set Name=? where ID=4085"; //Name? ,SURNAME ? diyerek birden fazla veriyide güncelleyebiliriz.

statement = connection.prepareStatement(sql);

**System**.*out*.println("Girmek istediğiniz bölgeyi yazınız.");

**Scanner** scan= **new** Scanner(**System**.*in*);

**String** girilenBolge = scan.next();

statement.setString(1, girilenBolge);

statement.executeUpdate();

**System**.*out*.println("Kayıt Güncellendi."); }

**catch** (**SQLException** e) {

**System**.*out*.println("SQL Connection is not succesfully");

db.showErrorMessages(e); }

**finally** {

statement.close();

connection.close(); } }}

**6.Javadan VeriTabanındaki Verileri Silme(DELETE)**

**public** **static** **void** **main**(String[] args) **throws** SQLException {

Connection **connection** = **null**;

PreparedStatement **statement** = **null**;

dbHelper **db** = **new** dbHelper();

**try** {

connection = db.getConnection();

System.***out***.println("Bağlantı Başarılı");

String **silinenVeri** = "Delete from city where ID='4086'";

statement = connection.prepareStatement(silinenVeri);

statement.execute();

} **catch** (SQLException **e**) {

db.showErrorMessages(e);

}

**finally** {

connection.close();

statement.close();

System.***out***.println("SQL Kapatıldı");

} } }