public class P1\_ReadData {  
  
  
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
  
 ***/\*  
 Soru : JDBC Nedir ?  
 - Veritabanlarının automation yapmamıza yardımcı oluyor  
 - Sorgularımı gonderıyorum, gelen sonuclarla, UI da ki gozuken sonucları karsılastırıyorum (UI : user interface)  
  
  
 Soru : JDBC yi nasıl kullanırsın ?  
 - URL , username ve password vererek bir connection olusturur.  
 Bu connection uzerinden statement olusturarak, sorgularimi calistiririm.  
 Sonuclari ResultSet e atarak kullanirim.  
  
 \*/ //MySql de day3 de..*** Class.*forName*("com.mysql.cj.jdbc.Driver");  
  
 Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/batch60\_61?serverTimezone=UTC", "root", "elif.12345"); ***//database baglantisi icin*** Statement st = con.createStatement(); ***//query calistirmak icin  
  
  
 // // SORU1: Talebeler tablosundan tum kayıtları listeleyiniz.  
 // ResultSet kayitlar = st.executeQuery("select \* from talebeler");  
//  
 // while(kayitlar.next()) {  
 // System.out.printf("%-6d %-15.15s %-8s\n", kayitlar.getInt(1),kayitlar.getString(2),kayitlar.getString(3),kayitlar.getInt(4)) ;  
 // } id isim veli\_isim yazili\_notu  
//  
 // // SORU2: Talebeler tablosunda notları 90 uzeri olan kayitları listeleyiniz.  
 // ResultSet kayitlarBuyuk90 = st.executeQuery("select \* from talebeler where yazili\_notu>90");  
//  
 // System.out.printf("%-16s %-16s \n", "id", "isim");  
 // while(kayitlarBuyuk90.next()) {  
 // System.out.printf("%-16s %-16s \n", kayitlarBuyuk90.getInt(1) ,kayitlarBuyuk90.getString(2));  
 // }***

***// SORU3: Talebeler tablosunda id'si 124 olan öğrencilerin tüm bilgilerini listeleyiniz.  
  
 // ResultSet id124 = st.executeQuery("SELECT \* FROM talebeler WHERE id = 124");  
 // System.out.printf("%-6s %-15.15s %-8s %-8s\n","id", "isim","veli\_isim","yazili\_notu");  
 // while(id124.next()) {  
 // System.out.printf("%-6s %-15.15s %-8s %-8s\n", id124.getObject(1), id124.getObject(2), id124.getObject(3), id124.getObject(4));  
 // System.out.println( id124.getObject(1) +"\t\t"+ id124.getObject(2)+ "\t\t" + id124.getObject(3)+"\t\t" + id124.getObject(4));  
 // }  
  
  
 // SORU4: Talebeler tablosunda notu 70 ile 90 arasindaki kisilerin isimlerini listeleyiniz.  
  
 // ResultSet not7090 = st.executeQuery("SELECT isim,yazili\_notu FROM talebeler WHERE yazili\_notu BETWEEN '70' AND '90'");  
 // System.out.printf("%-16s %-16s \n", "isim", "not");  
 // while (not7090.next()) {  
 // System.out.printf("%-16s %-16s \n", not7090.getObject(1) , not7090.getObject(2));  
 // }  
  
  
  
 // SORU5: Talebeler tablosunda isminin 2. harfi e olan kisilerin veli\_isimleri ile birlikte listeleyiniz.  
  
 // ResultSet isim2E\_VeliISim = st.executeQuery("SELECT isim,veli\_isim FROM talebeler WHERE isim LIKE '\_e%'");  
  
 // System.out.printf("%-16s %-16s \n", "isim", "veli\_isim");  
 // while (isim2E\_VeliISim.next()) {  
 // System.out.printf("%-16s %-16s \n", isim2E\_VeliISim.getObject(1) , isim2E\_VeliISim.getObject(2));  
 // }* con.close();  
 st.close();**  
  
  
 }

public class P2\_ReadData\_Different {  
   
 public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 ***//MySql de day5 de..*** Class.*forName*("com.mysql.cj.jdbc.Driver");  
  
 Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/batch60\_61?serverTimezone=UTC", "root", "elif.12345"); ***//database baglantisi icin*** Statement st = con.createStatement(ResultSet.*TYPE\_SCROLL\_INSENSITIVE*, ResultSet.*CONCUR\_READ\_ONLY*); ***//result set scroll duyarsiz, sadece okuma*** String query = "SELECT \* FROM personel";  
 ResultSet resultSet = st.executeQuery(query);  
 System.*out*.println(resultSet.getRow()); ***//0  
 // resultSet.first() bizi ilk kayida goturur, eger gecerli bir kayda gitmeden  
 // data ile ilgili islem yapmaya calisirsak  
 // "The result set has no current row." hatasi aliriz  
  
 //ilk satirdaki ilk objeyi yazdiralim*** resultSet.first();  
 Object ilkSAtirdakiObje = resultSet.getObject(1);  
 System.*out*.println(ilkSAtirdakiObje.toString()); ***//123456789  
  
 //ikinci satirdaki ilk objeyi yazdiralim*** resultSet.next();  
 Object ikinciSatirdakiObje = resultSet.getObject(1);  
 System.*out*.println(ikinciSatirdakiObje.toString()); ***//234567890  
  
 // ilk sutundaki 5.degeri yazdiralim*** resultSet.next();  
 resultSet.absolute(5);  
 Object besinciDeger = resultSet.getObject(1);  
 System.*out*.println(besinciDeger.toString());  
  
 ***// bu satirdan sonra tum listeyi yazdirmak icin bir dongu kullanirsam  
 // resultSet 5.satirda oldugu icin  
 // bize 5.satirdan sonrasini yazdirir  
 // Eger ilk satirdan itibaren yazdirmak istiyorsak once ilk satira gitmemiz gerekir*** System.*out*.println("=======================");  
 resultSet.absolute(0);  
 ***// 0.satira gitmemizin sebebi while sorgusu icerisinde bulunan resultSet.next() in bizi  
 // birinci kayda goturmesi icindir. Eger 1.satira gidecek olsak while icindeki kod yuzunden yazdirmaya  
 // 2.satirdan baslar, tum listeyi alamayiz  
  
 // Tum listeyi yazdirmak istersek*** while (resultSet.next()) { ***// while'in icine yazdigimiz kod boolean olarak true/false dondurur*** Object idler = resultSet.getObject(1); ***//bu kolondaki, 1.kolondaki tum objeleri verir...*** System.*out*.println(idler.toString());  
 }  
  
 System.*out*.println("=======================");  
 resultSet.absolute(0);  
 while (resultSet.next()) { ***// while'in icine yazdigimiz kod boolean olarak true/false dondurur*** Object isimler = resultSet.getObject("isim"); ***//isim kolondaki tum objeleri verir...*** System.*out*.println(isimler.toString()); ***//toString olmasa da yazar..*** }  
  
 System.*out*.println("=======================");  
 resultSet.absolute(0);  
 while (resultSet.next()) { ***// while'in icine yazdigimiz kod boolean olarak true/false dondurur*** Object sirketler = resultSet.getObject("sirket");  
 System.*out*.println(sirketler);  
 }  
  
 System.*out*.println("=======================");  
 resultSet.absolute(0);  
 while (resultSet.next()) {  
 Object sehirler = resultSet.getObject("sehir");  
 System.*out*.println(sehirler);  
 }  
 System.*out*.println("=======================");  
 resultSet.absolute(0);  
 while (resultSet.next()) {  
 Object maaslar = resultSet.getObject(4);  
 System.*out*.println(maaslar);  
 }  
  
 System.*out*.printf("%-6s %-15.15s %-8s %-8s\n","idler", "isimler","sirketler","sehirler", "maaslar");  
  
 con.close();  
 st.close();

public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
  
 Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/batch60\_61?serverTimezone=UTC", "root", "elif.12345"); ***//database baglantisi icin*** Statement st = con.createStatement(); ***//query calistirmak icin  
  
 // SORU: markalar adında bir tablo oluşturunuz. marka\_id int, marka\_isim VARCHAR(15), calisan\_sayisi int*** String createQuery = "create table markalar (marka\_id int, marka\_isim VARCHAR(15), calisan\_sayisi int)";  
  
 ***//1.yontem : execute () methodu ile*** boolean s1 = st.execute(createQuery);  
 System.*out*.println("markalar tablosu olusturuldu -> " +s1 ); ***//False  
 // // execute(), Boolean bir deger dondurur.  
 // //- DDL islemlerin false dondururken(ResultSet olusturmadigi için), DML islemlerinde true deger dondurur.(ResultSet olusturur)  
  
 // boolean s2 = st.execute(createQuery);  
 // System.out.println("markalar tablosu olusturuldu -> " +s2 ); //artik Database ded tablo var, hata verir  
//  
 //2.yontem : executeUpdate() methodu ile  
  
 // int s3 = st.executeUpdate(createQuery); //DDL islemleri icin 0 degerini dondurur  
 // System.out.println("markalar tablosu olusturuldu -> " +s3 ); //);  
 // //executeUpdate(), DDL islemlerinde de kullanilabilir ve bu islemlerde 0 dondurur.  
  
  
  
 // SORU: markalar tablosunu siliniz  
  
 // String dropQuery = "DROP TABLE markalar";  
 // st.execute(dropQuery);  
 // System.out.println("markalar tablosu silindi..");  
  
 // SORU : markalar tablosuna yeni bir sutun {sube\_sayisi int} ekleyiniz  
  
 // String alterQuery = "ALTER TABLE markalar ADD sube\_sayisi int"; //default olarak sona ekler  
 // st.executeUpdate(alterQuery);***

***// // SORU : markalar tablosuna yeni bir sutun {sube\_sayisi int} ekleyiniz, ancak bu sutunun yeri marka\_id den sonra olsub  
//  
 // String alterQuery2 = "ALTER TABLE markalar ADD sube\_sayisi int AFTER marka\_id";  
 // st.executeUpdate(alterQuery2);  
//  
  
 // SORU : markalar tablosunun adini brands olarak degistiriniz  
  
 // String alterQuery3 = "ALTER TABLE markalar RENAME TO brands";  
 // System.out.println("tablo ismi brands olarak degistirildi.. " + st.execute(alterQuery3));  
//  
  
  
 // SORU : markalar tablosunda marka\_isim sutununu isim olarak degistiriniz  
 // st.execute("alter table markalar rename column marka\_isim to isim");  
 // System.out.println("markalar tablosundaki marka\_isim sutunun ismi-> isim olarak degistirildi");  
  
  
  
  
 // SORU : markalar tablosunda marka\_isim sutununun data type ini char(20) olarak degistiriniz*** st.execute("alter table markalar modify marka\_isim char(20)");  
 System.*out*.println("marka isim data type char(20) oldu");  
  
 con.close();  
 st.close();  
  
  
 ***/\*  
 CHAR - Sabit uzunlukta karakter saklar. Yani CHAR(50) ifadesi MAX 50 karakter saklayabilir.  
 Ancak içerisine atanan veriyi bakılmaksızın bellekte sabit olarak hep 50 karakterlik yer kaplar.  
 VARCHAR- Bellekte değişken uzunlukta yer kaplar.  
 Varchar(50) ifadesi MAX 50 karakter saklayabilir ancak 30 karakterlik bir atama yapıldığında bellekte 30 karakterlik yer kaplar.  
  
 SORU: varchar ile varchar2 arasındaki farklar nelerdir ?  
 CEVAP: varchar2, varchar ’ın güncellenmiş versiyonudur.  
 Varchar’a göre 2 kat daha fazla (4000) karakter saklayabilir.  
 \*/***

public class P4\_DML\_Insert {  
 ***//MySql de day6 da..*** public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
  
 String url = "jdbc:mysql://localhost:3306/batch60\_61?serverTimezone=UTC";  
 Connection con = DriverManager.*getConnection*(url, "root", "elif.12345"); ***//database baglantisi icin*** Statement st = con.createStatement(); ***//query calistirmak icin  
   
 //SORU: Calisanlar tablosuna yeni bir kayit ((10005, 'Muhammet Yenice' , 15000) ekleyelim ve eklenen kaydi teyit icin sorgulayalim.*** String insertQuery = "INSERT INTO calisanlar VALUES (10005, 'Muhammet Yenice' , 15000)";  
  
 int s1 = st.executeUpdate(insertQuery);  
 System.*out*.println(s1 + " satir eklendi"); ***//sql de select \* from calisanlar; kodunu calistir gorunecektir  
  
 //İntelijde de gorelim..*** ResultSet rs = st.executeQuery("SELECT \* FROM calisanlar");  
  
 while (rs.next()) {  
 System.*out*.printf("%-6d %-15.15s %-8s\n" ,rs.getInt(1),  
 rs.getString(2),rs.getInt(3));  
 }  
  
 ***//SORU: Calisanlar tablosuna birden fazla yeni kayıt ekleyelim.  
  
 // 1.YONTEM  
 // Ayri ayri sorgular ile veritabanina tekrar tekrar ulasmak islemlerin yavas yapilmasina yol acar.  
 // 10000 tane veri kaydi yapildigi dusunuldugunde bu kotu bir yaklasimdir.*** String [] queries = {"INSERT INTO calisanlar VALUES (10006, 'Ahmet Yanar' , 20000)",  
 "INSERT INTO calisanlar VALUES (10007, 'Mehmet Solmaz' , 18000)"};  
  
 int data = 0;  
 for (String each : queries) {  
 data+= st.executeUpdate(each);  
 }  
 System.*out*.println(data + " data eklendi.."); ***//sql de select \* from calisanlar; kodunu calistir gorunecektir  
  
 // 2.YONTEM (addBath ve excuteBatch() metotlari ile)  
 // ----------------------------------------------------  
 // addBatch metodu ile SQL ifadeleri gruplandirilabilir ve  
 // exucuteBatch() metodu ile veritabanina bir kere gonderilebilir.  
 // \*\*\*!!!!\*\*\*\* excuteBatch() metodu bir int [] dizi dondurur.  
 // Bu dizi her bir ifade sonucunda etkilenen satir sayisini gosterir.*** String [] queries2 = {"INSERT INTO calisanlar VALUES (10008, 'Ahmet Yanar' , 20000)",  
 "INSERT INTO calisanlar VALUES (10009, 'Mehmet Solmaz' , 18000)"};  
 for (String each : queries2) {  
 st.addBatch(each);  
 }  
  
 int [] satir = st.executeBatch(); ***////satr bir array eleman sayisi (satir sayisi): length*** System.*out*.println( satir.length + " satir eklendi");  
   
 ***// 3. YONTEM  
 //-----------------------------------------------------  
 // batch metoduyla birlikte PreparedStatement kullanmak en efektif yontemdir.  
 // SQL de guvenligi icin basvurulan bir yontemdir. Unutulmamalidir...  
 // bir sonraki ornekte bunu gerceklestirecegiz.*** con.close();  
 st.close();  
 rs.close();  
 }

public static void main(String[] args) throws ClassNotFoundException, SQLException {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
  
 String url = "jdbc:mysql://localhost:3306/batch60\_61?serverTimezone=UTC";  
 Connection con = DriverManager.*getConnection*(url, "root", "elif.12345"); ***//database baglantisi icin*** Statement st = con.createStatement(); ***//query calistirmak icin  
  
 // SORU1: Dersler adında bir tablo olusturalım (id int, ders\_isim varchar(15), egitim\_suresi int)*** String createQuery = "create table dersler (id int, ders\_isim VARCHAR(15), egitim\_suresi\_gun int)";  
  
 st.execute(createQuery);  
 System.*out*.println("tablo olusturuldu..");

**Ders diye bir class oluştur.. id (int), ders\_isim (String), eğitim\_suresi(int) instance variable oluştur..**

**Generate den constructor, getter setter, toString() oluştur..**  
 List<Ders> dersler = new ArrayList<>();  
 dersler.add(new Ders (100, "Java",55));  
 dersler.add(new Ders (101, "SQL",10));  
 dersler.add(new Ders (102, "JDBC",3));  
 dersler.add(new Ders (103, "APi",12));  
 dersler.add(new Ders (104, "JENKİNS",2));  
 dersler.add(new Ders (105, "Selenium",25));  
 dersler.add(new Ders (106, "Appium",10));  
 dersler.add(new Ders (107, "SDLC",10));  
  
 PreparedStatement table = con.prepareStatement("INSERT INTO dersler VALUES (?,?,?)");  
  
 for (Ders d : dersler){  
 table.setInt(1, d.getId());  
 table.setString(2, d.getDers\_isim());  
 table.setInt(3, d.getEgitim\_suresi());  
  
 table.addBatch();  
 }  
  
 table.executeBatch();  
  
 System.*out*.println("Hersey Tamam");  
 ***//mySql de derslerin hizasindaki son simgeye basilirsa tablo SQL de gorunur..***}