## 2调用方法向数据库插入数据

public static boolean addOne(Student stu)

## 3 设计方法,通过id删除用户数据

public static int deleteById(int id)

返回值是受影响的行数

## 4 设计方法,通过id更改用户数据

public static int updateById(Student stu)

注意:根据id更新,即参数Student对象中一定有id属性值

更新Student表的数据,根据Student对象的属性更新,如果属性值是null则不跟更新,不为null就更新对应的字段

返回值是受影响的行数

## 5 设计方法,查询全部数据

public static List<Student> findAll()

返回值是List集合,集合中是全部学生数据

|  |
| --- |
| **package** com.zhiyou100.jdbc;  **import** java.sql.\*; **import** java.text.SimpleDateFormat; **import** java.util.ArrayList; **import** java.util.Date; **import** java.util.List;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/13  \** ***@desc*** *\*/* **public class** Zuoye {   **public static void** main(String[] args) **throws** Exception{  *// boolean r = addOne(new Student(24,"宋江",35,"男",98,1,new Date())); // System.out.println("插入是否成功? "+r ); // Student student = new Student( ); // student.setId(24); // student.setName("及时雨" ); // student.setAge(40); // student.setBirthday(new Date((long)1000\*60\*60\*24\*365\*10)); // int i = updateById(student); // System.out.println("更新后:返回受影响的行数 "+i );* List<Student> all = *findAll*( );  **for** (Student student : all) {  System.***out***.println(student );  }  }   */\*\*  \* 工具方法: 获得数据库连接对象  \*/* **public static** Connection getConnection() {  Connection conn = **null**;  **try** {  Class.*forName*(**"com.mysql.jdbc.Driver"**);  String url = **"jdbc:mysql://localhost:3306/java3201"**;  String username = **"root"**;  String password = **"123456"**;  conn = DriverManager.*getConnection*(url,username,password);  } **catch** (Exception e) {  e.printStackTrace( );  }  **return** conn;  }   */\*\*  \* 更新  \*/* **public static int** updateById(Student stu) **throws** SQLException {  Connection conn = *getConnection*( );  Statement statement = conn.createStatement( );   StringBuilder sb = **new** StringBuilder(**"update student set "**);  **if** (stu.getName() != **null**) {  sb.append(**"sname = '"**+stu.getName()+**"',"**);  }  **if** (stu.getAge() != 0) {  sb.append(**"sage = "**+stu.getAge()+**","**);  }  **if** (stu.getSex() != **null**) {  sb.append(**"sex = '"**+stu.getSex()+**"',"**);  }  **if** (stu.getScore() != 0.0) {  sb.append(**"score = "**+stu.getScore());  }  **if**(stu.getSclass() != 0) {  sb.append(**"sclass = "**+stu.getSclass()+**", "**);  }  **if** (stu.getBirthday() != **null**) {  SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  String birthday = sdf.format(stu.getBirthday( ));  sb.append(**"sbirthday = '"**+birthday+**"',"**);  }  sb.append(**" where sid = "**+stu.getId());   **int** index = sb.indexOf(**"where"**);  sb.deleteCharAt(index-2);   **return** statement.executeUpdate(sb.toString());  }    */\*\*  \* 插入  \*/* **public static boolean** addOne(Student stu) **throws** Exception {  Connection conn = *getConnection*( );  Statement statement = conn.createStatement( );   SimpleDateFormat sdf = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  String birthday = sdf.format(stu.getBirthday( ));  String sql = **"insert into student values ("**+stu.getId()+**",'"**+stu.getName()+**"',"**+stu.getAge()+**",'"**+stu.getSex()+**"',"**+stu.getScore()+**","**+stu.getSclass()+**",'"**+birthday+**"')"**;   **int** i = statement.executeUpdate(sql);   statement.close();  conn.close();   **if** (i > 0) {  **return true**;  }  **return false**;  }    **public static** List<Student> findAll() **throws** SQLException {  Connection conn = *getConnection*( );  Statement statement = conn.createStatement( );  String sql = **"select** *\** **from student"**;  ResultSet rs = statement.executeQuery(sql);  ArrayList<Student> list = **new** ArrayList<>( );  **while** (rs.next()) {  Student student= **new** Student( );  student.setId(rs.getInt(**"sid"**));  student.setName(rs.getString(**"sname"**));  student.setScore(rs.getDouble(**"score"**));  student.setBirthday(rs.getDate(**"sbirthday"**));   *// 加入集合* list.add(student);  }  **return** list;  } } |

# 练习: 模拟登录

create table user (

id int,

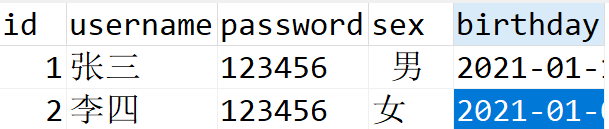
username varchar(10),

password varchar(10),

sex char(2),

birthday date

);



1 键盘输入用户名密码

2 提示出登录成功与否?

2.1 接收用户名密码

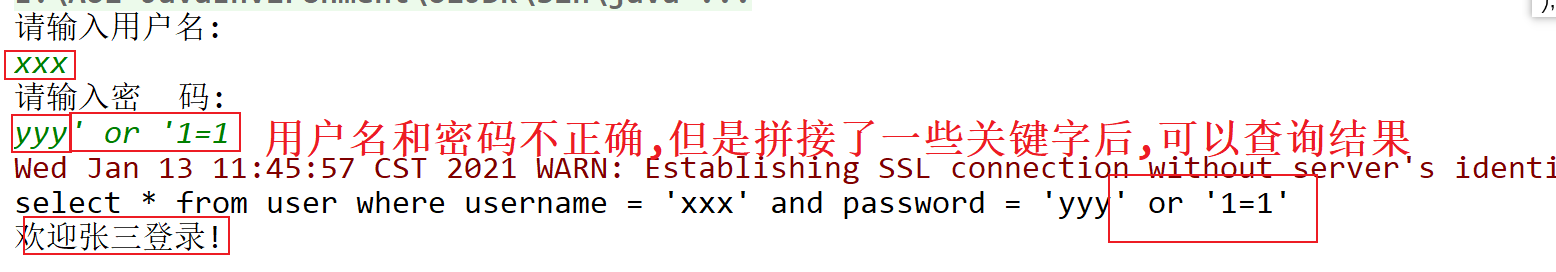
2.2 根据用户名密码查数据库

2.3 查到用户,提示登录成功

2.4 查不到,提示用户名或密码错误

|  |
| --- |
| **package** com.zhiyou100.jdbc;  **import** java.sql.\*; **import** java.util.Scanner;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/13  \** ***@desc*** *模拟登录  \*/* **public class** Login {   **public static void** main(String[] args) **throws** SQLException {  Scanner scanner = **new** Scanner(System.***in***);  System.***out***.println(**"请输入用户名: "** );  String username = scanner.nextLine( );  System.***out***.println(**"请输入密 码: "** );  String password = scanner.nextLine( );   User user = *findUserByUsernameAndPassword*(username,password);  **if** (user != **null**) {  System.***out***.println(**"欢迎"**+user.getUsername()+**"登录!"** );  }**else** {  System.***out***.println(**"用户名或密码错误"** );  }   }   **private static** User findUserByUsernameAndPassword(String username, String password) **throws** SQLException {  Connection conn = *getConnection*( );  Statement statement = conn.createStatement( );   String sql = **"select** *\** **from user where username = '"**+username+**"' and password = '"**+password+**"'"**;  User user = **null**;   ResultSet rs = statement.executeQuery(sql);  **if**(rs.next()) {  user = **new** User();  user.setId(rs.getInt(**"id"**));  user.setUsername(rs.getString(**"username"**));  user.setPassword( rs.getString(**"password"**));  user.setSex(rs.getString(**"sex"**));  user.setBirthday(rs.getDate(**"birthday"**));  }  rs.close();  statement.close();  conn.close();  **return** user;  }   */\*\*  \* 工具方法: 获得数据库连接对象  \*/* **public static** Connection getConnection() {  Connection conn = **null**;  **try** {  Class.*forName*(**"com.mysql.jdbc.Driver"**);  String url = **"jdbc:mysql://localhost:3306/java3201"**;  String username = **"root"**;  String password = **"123456"**;  conn = DriverManager.*getConnection*(url,username,password);  } **catch** (Exception e) {  e.printStackTrace( );  }  **return** conn;  }  } |

# SQL注入



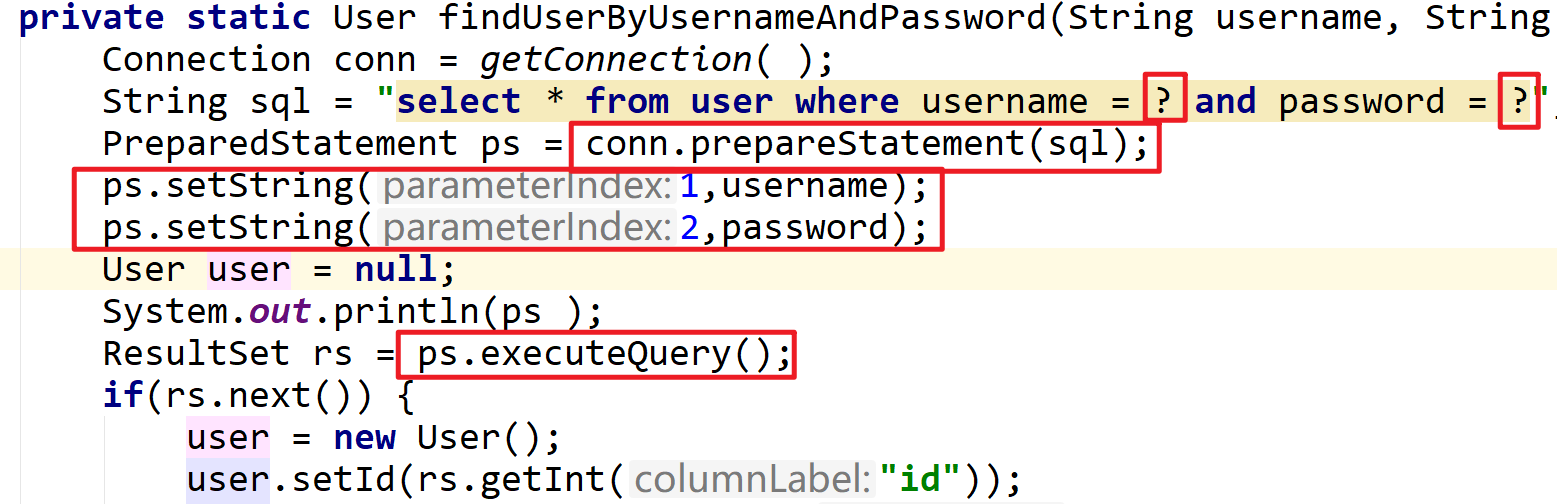
sql注入: 利用数据库关键字,在sql语句中恶意拼接,出现一些非常规操作,导致严重问题!

# 预处理语句

使用预处理语句进行sql操作,就可以避免sql注入问题.



改造【登录】练习,使用预处理语句避免sql注入



# 预处理语句-增

|  |
| --- |
| **package** com.zhiyou100.presql;  **import** java.sql.Connection; **import** java.sql.DriverManager; **import** java.sql.PreparedStatement; **import** java.sql.Date;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/13  \** ***@desc*** *\*/* **public class** Demo2 {   **public static void** main(String[] args) **throws** Exception{  *// 加载驱动* Class.*forName*(**"com.mysql.jdbc.Driver"**);  *// 获得连接* Connection conn = DriverManager.*getConnection*(**"jdbc:mysql://localhost:3306/java3201"**, **"root"**, **"123456"**);   String sql = **"insert into student values (?,?,?,?,?,?,?)"**;  PreparedStatement ps = conn.prepareStatement(sql);   ps.setInt(1,25);  ps.setString(2,**"卢俊义"**);  ps.setInt(3,40);  ps.setString(4,**"男"**);  ps.setDouble(5,98.2);  ps.setInt(6,1);  *// setDate(int index,java.sql.Date date)* ps.setDate(7,**new** Date((**long**)1000\*60\*60\*24\*365\*10));   **int** i = ps.executeUpdate( );  System.***out***.println(i );  ps.close();  conn.close();  } } |

# 预处理语句 – 更新

|  |
| --- |
| **package** com.zhiyou100.presql;  **import** java.sql.Connection; **import** java.sql.Date; **import** java.sql.DriverManager; **import** java.sql.PreparedStatement;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/13  \** ***@desc*** *\*/* **public class** Demo3 {   **public static void** main(String[] args) **throws** Exception{  *// 加载驱动* Class.*forName*(**"com.mysql.jdbc.Driver"**);  *// 获得连接* Connection conn = DriverManager.*getConnection*(**"jdbc:mysql://localhost:3306/java3201"**, **"root"**, **"123456"**);   String sql = **"update student set sname = ? where sid = ?"**;  PreparedStatement ps = conn.prepareStatement(sql);   ps.setString(1,**"玉麒麟"**);  ps.setInt(2,25);   **int** i = ps.executeUpdate( );  System.***out***.println(i );  ps.close();  conn.close();  } } |

# 预处理语句-删除

自行完成。。

# JDBC操作事务

|  |
| --- |
| **package** com.zhiyou100.presql;  **import** java.sql.Connection; **import** java.sql.DriverManager; **import** java.sql.PreparedStatement; **import** java.sql.SQLException;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/13  \** ***@desc*** *事务  \*/* **public class** Demo4 {   */\*\*  \* 转账操作:  \* A向B转账100  \* A 减少100  \* B 增加100  \*/* **public static void** main(String[] args) {  Connection conn = **null**;  PreparedStatement ps1 = **null**;  PreparedStatement ps2 = **null**;  **try** {   *// 加载驱动* Class.*forName*(**"com.mysql.jdbc.Driver"**);  *// 获得连接* conn = DriverManager.*getConnection*(**"jdbc:mysql://localhost:3306/java3201"**, **"root"**, **"123456"**);   *// 将事务自动提交设置false,即相当于手动开启事务* conn.setAutoCommit(**false**);   String sql1 = **"update account set money = money - 100 where id = ?"**;  ps1 = conn.prepareStatement(sql1);  ps1.setInt(1, 1);   String sql2 = **"update account set money = money + 100 where id = ?"**;  ps2 = conn.prepareStatement(sql2);  ps2.setInt(1, 2);   **int** i1 = ps1.executeUpdate( );  *// 模拟出错* System.***out***.println(1 / 0);  **int** i2 = ps2.executeUpdate( );  System.***out***.println(i1);  System.***out***.println(i2);   *// 提交* conn.commit( );  } **catch** (Exception e) {  System.***out***.println(**"sql执行出错!"** );  **try** {  *// 有异常,回滚事务* conn.rollback();  } **catch** (Exception e1) {  e1.printStackTrace( );  }  } **finally** {  **try** {  ps1.close( );  ps2.close( );  conn.close( );  } **catch** (SQLException e) {  e.printStackTrace( );  }   }  }  } |

# 封装DBUtil

|  |
| --- |
| **package** com.zhiyou100.presql;  **import** java.sql.\*;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/13  \** ***@desc*** *数据库工具类  \*/* **public class** DBUtil {   **private static final** String ***URL*** = **"jdbc:mysql://localhost:3306/java3201"**;  **private static final** String ***USER*** = **"root"**;  **private static final** String ***PASSWORD*** = **"123456"**;  *// 静态代码块,随着类加载而执行.且其中内容在内存只存在一份* **static** {  **try** {  Class.*forName*(**"com.mysql.jdbc.Driver"**);  } **catch** (ClassNotFoundException e) {  e.printStackTrace( );  }  }  */\*\*  \* 获得连接对象  \*/* **public static** Connection getConnection() {  Connection conn = **null**;  **try** {  conn = DriverManager.*getConnection*(***URL***, ***USER***, ***PASSWORD***);  } **catch** (Exception e) {  e.printStackTrace( );  }  **return** conn;  }   */\*\*  \* 关流  \*/* **public static void** closeAll(Connection conn, Statement s, ResultSet rs) {  **if** (conn != **null**) {  **try** {  conn.close();  } **catch** (SQLException e) {  e.printStackTrace( );  }  }  **if** (s != **null**) {  **try** {  s.close();  } **catch** (SQLException e) {  e.printStackTrace( );  }  }  **if** (rs != **null**) {  **try** {  rs.close();  } **catch** (SQLException e) {  e.printStackTrace( );  }  }  }  **public static void** closeAll(Connection conn, Statement s) {  **if** (conn != **null**) {  **try** {  conn.close();  } **catch** (SQLException e) {  e.printStackTrace( );  }  }  **if** (s != **null**) {  **try** {  s.close();  } **catch** (SQLException e) {  e.printStackTrace( );  }  }  } } |

# 作业

昨日5题，使用预处理语句完成。

使用DBUtil再完成一遍。