JDBC二

课程目标:

1. 业务实现登入
2. 预执行对象
3. 实现JDBC的封装(理解)
4. 实现登入业务
5. 创建用户表:

|  |
| --- |
| CREATE TABLE `users` (  `uid` int(11) NOT NULL AUTO\_INCREMENT,  `uname` varchar(15) NOT NULL,  `upwd` varchar(15) NOT NULL,  `tel` char(11) DEFAULT NULL,  `email` varchar(25) DEFAULT NULL,  PRIMARY KEY (`uid`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8; |

2. 分析登入

-- 实现登入：通过用户名密码作为条件去查询用户表

-- 有记录 登入成功 否则登入失败

SELECT \* FROM users WHERE uname='wjb' AND upwd='123';

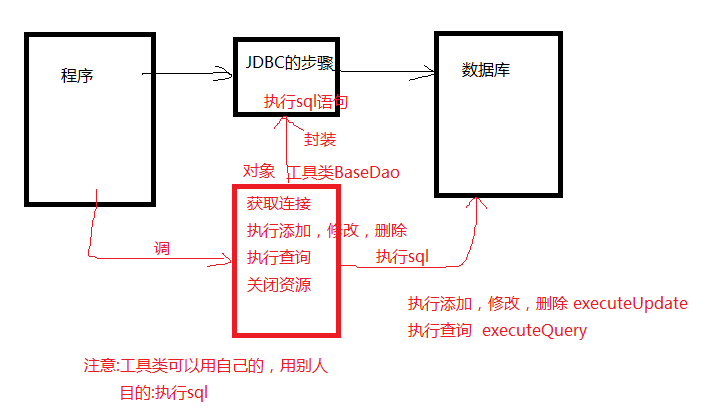
1. 实现登入

|  |
| --- |
| **public static void** main(String[] args) {  *//实现登入功能:* Scanner input=**new** Scanner(System.***in***);  System.***out***.println(**"请输入用户名:"**);  String name=input.next();  System.***out***.println(**"请输入密码:"**);  String pwd=input.next();   *//实现登入 拼sql语句:存在sql注入问题* String sql=**"SELECT** *\** **FROM users WHERE uname='"**+name+**"' AND upwd='"**+pwd+**"'"**;   **boolean** b=**false**;  *//执行sql* **try** {  Class.*forName*(**"com.mysql.jdbc.Driver"**);  Connection con=DriverManager.*getConnection*(**"jdbc:mysql://localhost:3306/k9051db"**,**"root"**,**"123"**);  Statement st=con.createStatement();  *//执行查询* ResultSet rs=st.executeQuery(sql);  **if**(rs.next())  {  b=**true**;  }  rs.close();  st.close();  con.close();  rs=**null**;  st=**null**;  con=**null**;  **if**(b){  System.***out***.println(**"登入成功"**);  }**else** System.***out***.println(**"登入失败"**);   } **catch** (ClassNotFoundException e) {  e.printStackTrace();  } **catch** (SQLException e) {  e.printStackTrace();  } } |

1. 预执行对象(PreparedSatement):理解为执行参数化的sql
2. 作用:
3. 提高安全性，防止sql注入
4. 提高代码的可读性和可维护性
5. 提高sql语句执行效率
6. 使用PreparedSatement执行参数化的sql
   1. 定义参数化sql *使用?号占位置*String sql=**"SELECT** *\** **FROM users WHERE uname=? AND upwd=?"**;
   2. 执行参数化sql的步骤:

|  |
| --- |
| *1.基于连接对象创建预执行对象执行sql* PreparedStatement ps=con.prepareStatement(sql); *2.使用setXXX方法给参执行对象传参 参数位置从1开始* ps.setObject(1,name); ps.setObject(2,pwd); *3.执行查询* ResultSet rs=ps.executeQuery();  *//ps.executeUpdate(); 执行添加，修改，删除* |

1. 理解JDBC的封装



1. 使用jdbc工具类BaseDao
2. 使用BaseDao执行添加，修改，删除

|  |
| --- |
| **public static void** main(String[] args) {  *//添加年级* Scanner input=**new** Scanner(System.***in***);  System.***out***.println(**"请输入编号:"**);  **int** gid=input.nextInt();  System.***out***.println(**"请输入名称:"**);  String gname=input.next();   *//调用工具类实现添加* String sql=**"insert into grade(gid,gname) values(?,?)"**;  BaseDao baseDao=**new** BaseDao();  **int** temp=baseDao.executeUpdate(sql,**new** Object[]{gid,gname});   **if**(temp>0)  System.***out***.println(**"添加成功"**);  **else** System.***out***.println(**"添加失败"**);  } |

1. 使用工具类执行查询

|  |
| --- |
| **public static void** main(String[] args) **throws** SQLException {  *//查询所有年级* String sql=**"select \* from grade"**;  *//调工具类* BaseDao baseDao=**new** BaseDao();  ResultSet rs=baseDao.executeQuery(sql,**null**);  *//显示* System.***out***.println(**"编号\t名称"**);  **while**(rs.next()){  System.***out***.println(rs.getObject(1)+**"\t"**+rs.getObject(2));  }  *//关闭* baseDao.closeAll(); } |