# JDBC实验

## 1.实验目的：

使用编程语言连接数据库并实现批量化插入删除查询等功能。

注意事项：本次实验不限定编程语言，但仅提供windows环境下的java用例，如果使用其他环境或其他语言连接亦可，我们更推荐使用java(JDBC)或c++(ODBC)，想要选用其他语言的同学需自行查阅相关语言的数据库连接文档。

## 2.实验步骤

### 2.1 安装JAVA环境

[手把手教会你在 windows10 下安装、配置 Java JDK 14](https://blog.csdn.net/weixin_44496396/article/details/105478734)

请根据上面链接进行java14的安装

**大夏学堂上提供了java14的安装包，以方便大家下载**

### 2.2 下载JDBC驱动JAR包

#### 2.2.1

1.在电脑中新建一个空白文件夹（英文路径，路径中文件名中不要出现空格和除下划线以外的其他符号）

2.将jdbc驱动程序 mysql-connector-j-8.4.0 保存到当前工作文件夹；

**大夏学堂上提供了mysql-connector-j-8.4.0.jar文件，以方便大家下载**

3. 在当前工作文件夹下创建一个 jdbcConnect.java 文件，与数据库建立连接 ：

图形用户界面, 应用程序

描述已自动生成

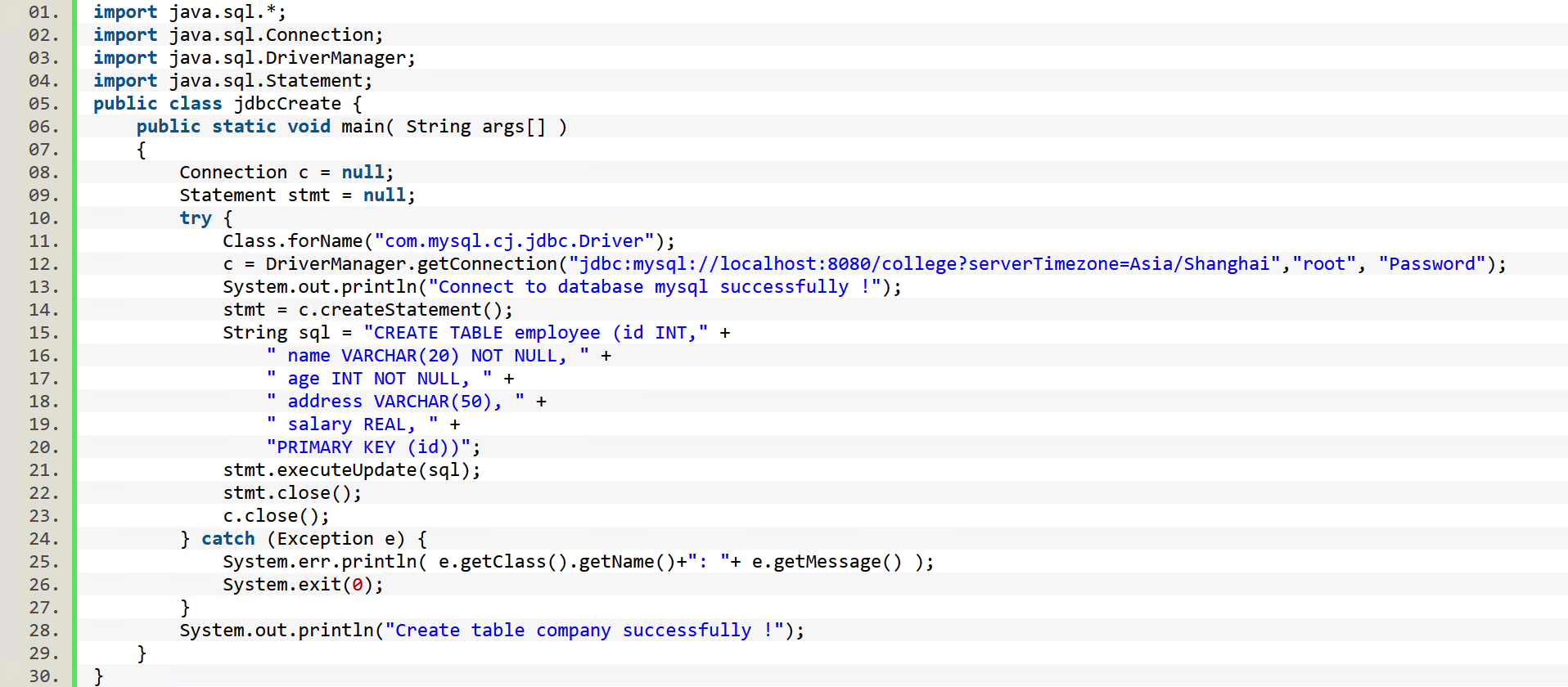
1. **import** java.sql.Connection;
2. **import** java.sql.DriverManager;
3. **public** **class** jdbcConnect {
4. **public** **static** **void** main(String args[]) {
5. Connection c = **null**; **try** {
6. Class.forName("com.mysql.cj.jdbc.Driver");
7. c = DriverManager.getConnection("jdbc:mysql://localhost:8080/college?serverTimezone=Asia/Shanghai","root", "Password"); }
8. **catch** (Exception e) {
9. e.printStackTrace();
10. System.err.println(e.getClass().getName()+": "+e.getMessage()); System.exit(0);
11. }
12. System.out.println("Connect to database mysql successfully !");
13. }
14. }

4. 打开一个新的cmd窗口，在当前工作文件夹下，执行下面命令运行该java代码（需要确保已经创建了名为college的数据库）

**java -cp .\mysql-connector-j-8.4.0.jar .\jdbcConnect.java**

运行后，会显示“Connect to database mysql successfully !”，表示成功连接到mysql数据库

5. 在当前工作文件夹下创建并运行 jdbcCreate.java 文件，此代码的功能是在数据库中创建employee表，其内容如下：



1. **import** java.sql.\*;
2. **import** java.sql.Connection;
3. **import** java.sql.DriverManager;
4. **import** java.sql.Statement;
5. **public** **class** jdbcCreate {
6. **public** **static** **void** main( String args[] )
7. {
8. Connection c = **null**;
9. Statement stmt = **null**;
10. **try** {
11. Class.forName("com.mysql.cj.jdbc.Driver");
12. c = DriverManager.getConnection("jdbc:mysql://localhost:8080/college?serverTimezone=Asia/Shanghai","root", "Password");
13. System.out.println("Connect to database mysql successfully !");
14. stmt = c.createStatement();
15. String sql = "CREATE TABLE employee (id INT," +
16. " name VARCHAR(20) NOT NULL, " +
17. " age INT NOT NULL, " +
18. " address VARCHAR(50), " +
19. " salary REAL, " +
20. "PRIMARY KEY (id))";
21. stmt.executeUpdate(sql);
22. stmt.close();
23. c.close();
24. } **catch** (Exception e) {
25. System.err.println( e.getClass().getName()+": "+ e.getMessage() );
26. System.exit(0);
27. }
28. System.out.println("Create table company successfully !");
29. }
30. }

**java -cp .\mysql-connector-j-8.4.0.jar .\jdbcCreate.java**

运行命令和上面类似，只需要修改java文件名即可

6.请参考上面的例子，使用JDBC在college数据库上完成以下sql操作

**创建表**

“CREATE TABLE employee (id INT," +

" name VARCHAR(20) NOT NULL, " +

" age INT NOT NULL, " +

" address VARCHAR(50), " +

" salary REAL, " +

"PRIMARY KEY (id))”

**插入数据**

INSERT INTO employee VALUES (1, 'Gong', 48, '2075 Kongjiang Road', 20000.00 );

INSERT INTO employee VALUES (2, 'Luan', 25, '3663 Zhongshan Road(N)', 15000.00 );

INSERT INTO employee VALUES (3, 'Hu', 23, '3663 Zhongshan Road(N)', 15000.00 );

INSERT INTO employee VALUES (4, 'Jin', 24, '3663 Zhongshan Road(N)', 15000.00 );

INSERT INTO employee VALUES (5, 'Yi', 24, '3663 Zhongshan Road(N)', 15000.00 );

**查询数据**

SELECT \* FROM employee;

**更新数据**

UPDATE employee set SALARY = 50000.00 where ID=1;

检查更新结果：SELECT \* FROM employee;

**删除数据**

DELETE from employee where ID=2;

检查删除结果：SELECT \* FROM employee;

**使用prepare语句批量插入和更新**

import java.math.BigDecimal;

CREATE TABLE GPA(grade CHAR(2), grade\_point DECIMAL(3,2));

String[] strArray = new String[11];

strArray[0] = "A";

strArray[1] = “A-";

strArray[2] = "B+";

strArray[3] = "B";

strArray[4] = "B-";

strArray[5] = "C+";

strArray[6] = "C";

strArray[7] = "C-";

strArray[8] = "D";

strArray[9] = "D-";

strArray[10] = "F";

double[] doubleArray = new double[] {4.0,3.7,3.3,3.0,2.7,2.3,2.0,1.5,1.3,1.0,0};

pst = conn.prepareStatement( "INSERT INTO customer VALUES (?,?)");

for ( int i = 0; i < 12; i++) {

pst.setString(1, strArray[i]);

pst.setBigDecimal (2, BigDecimal.valueOf(doubleArray[i]));

pst.addBatch();

}

pst.executeBatch();

## 3.实验任务：

小项目作业：

1. 连接SQL(实验报告1)中使用的college数据库。

2. 现在，你就是学校信息系统的程序员，使用JDBC编写程序，完成以下任务：

2.1 通过输入的登陆ID和密码，开启与数据库的连接

2.2连接成功后，要求用户输入一个字符串。返回所有名字中含有该子串的学生的信息，包括（ID，name，dept\_name, tot\_cred）4个字段。

2.3 然后要求用户输入一个整数（0～99999），并显示ID与之完全匹配的的学生的信息，内容要求同上。

2.4 若2.3中学生查找成功，则在用户输入1后，输出学生所修读的所有课程信息。包括以下信息：

1. 课程ID

2. 上课年份

3. 上课学期

4. 课程名称

5. 开课院系

6. 成绩等级

7. 课程学分数

2.5 若2.4中修读课程不为空，则在用户输入1后，计算该学生的计算平均绩点，并显示。

3. 以上错误处理要求如下：

3.1 若2.1连接失败，提示用户错误并允许重试

3.2 若2.2查询结果为空，提示用户无相关学生并允许重新输入

3.3 若2.3查询结果为空，提示用户无该学生并允许重试

3.4 若2.4和2.5中用户输入0，则停止本次查询

3.5 连接数据库成功后，请捕获所有数据库有关异常并处理，不要抛出停止程序。