**《Java程序设计》实验报告**

**学生姓名： 耿立博 班级： 19计算机3班 学号： 2019207320123**

**实验日期： 2021.6.15 指导教师： 胡继礼、谷宗运、李芳芳**

**实验环境：win7+Jdk1.8**

**一、实验目的**

**掌握** 1.掌握JDBC数据库操作方法

2.掌握Connection、Statement、PreparedStatement、ResultSet类常用方法

**熟悉 1.** JDBC原理

**了解** 1.了解数据库连接池

2. MySQL和SQL Server数据库

**二、实验内容**

1.安装并创建MySQL数据库testdb要求如下：

（1）安装mysql(字符集设置为utf-8)和navicate；

（2）利用navicate可视化界面，在mysql里创建数据库testdb,字符集编码UTF-8；

（3）新建t\_user表，字段和初始数据如下

|  |  |  |
| --- | --- | --- |
| id(int,自动递增) | name（varchar） | password（varchar） |
| 1 | 张三 | 123456 |
| 2 | 李四 | abcdef |

2.定义一个TestMySQL类要求如下：

（1）加载mysql数据库驱动,获取连接，功能实现获取一个testdb数据库连接;

（2）在main方法里，利用statement语句查询遍历打印输出t\_user的所有记录;

（3）利用Statement语句向表中插入一条记录，name为“王二”，密码为“123456”，且密码使用md5加密存放;

（4）利用PreparedStatement将李四的密码更换为“123456”；

（5）删除id为1的记录；

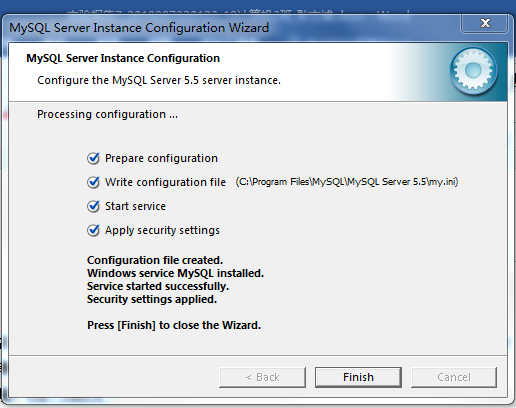
（6）关闭结果集ResultSet，关闭Statement、PreparedStatement语句，关闭连接connection；

（7）对可能出现的异常进行捕获

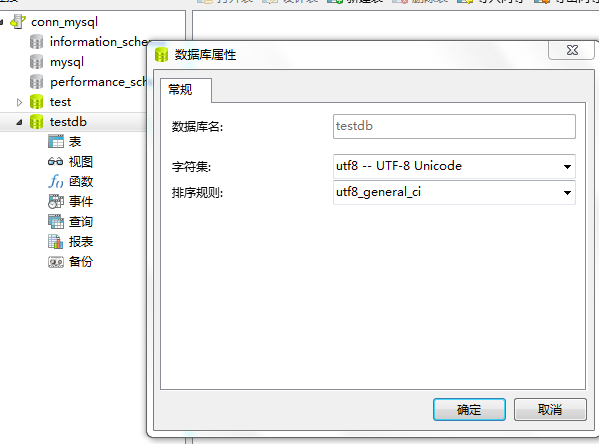
**三、实验过程（按上述实验内容要求写出源代码及运行结果）**

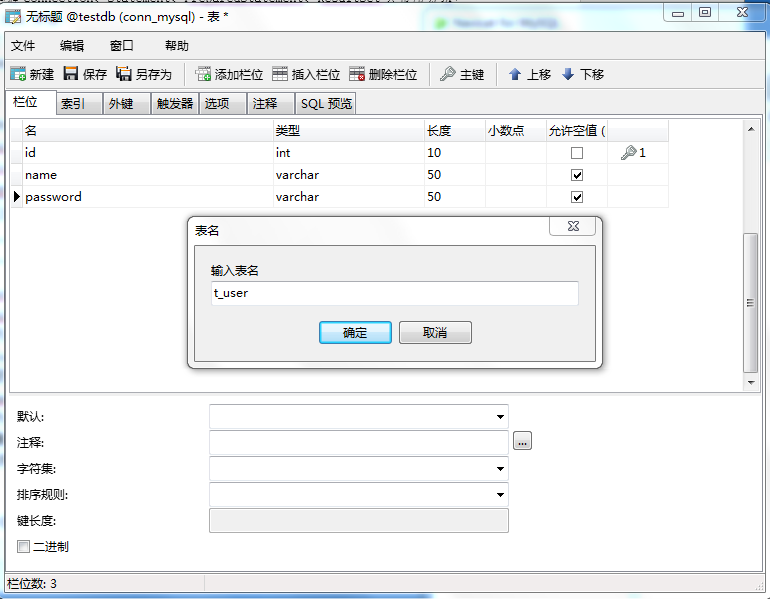
**1.实验-1**

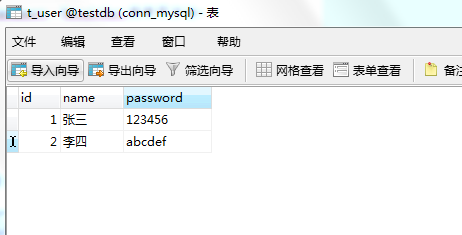
**【安装截图和表截图】**

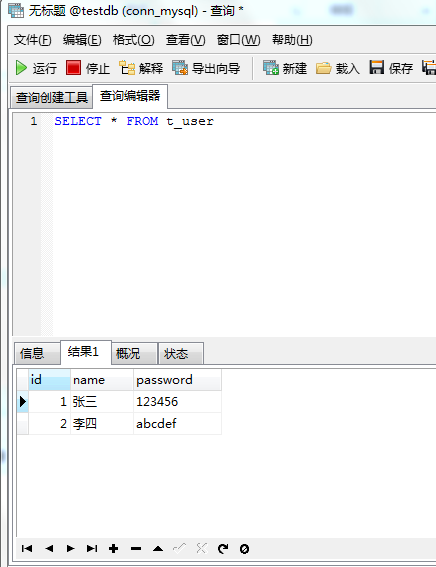












**2.实验-2**

**【源码】**

**package** cn.edu.ahtcm.db;

**import** java.math.BigInteger;

**import** java.security.MessageDigest;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

/\*

\* Java连接MySQL5.5 示例,

\* blog 数据库

\* users(id,name,password)表

\* id 为自增字段s

\*/

**public** **class** TestMySQL {

//注册MySQL驱动

**static**{

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

System.***out***.println("加载驱动成功！");

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

System.***out***.println("加载驱动失败！");

}

}

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

String connectionUrl = "jdbc:mysql://localhost:3306/testdb?user=root&password=123456&useUnicode=true&characterEncoding=UTF8";

Connection connection = **null**;

**try** {

connection = DriverManager.*getConnection*(connectionUrl);

System.***out***.println("Done.");

} **catch** (SQLException e) {

e.printStackTrace();

}

// 读取

String sql = "SELECT id,name,password from t\_user";

**try** {

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(sql);

**while** (resultSet.next()) {

System.***out***.println(resultSet.getInt(1) + " " + resultSet.getString(2) + " " + resultSet.getString(3));

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

//插入

sql = "insert into t\_user(name,password) values(?,?)";

PreparedStatement pst;

**try** {

pst = connection.prepareStatement(sql);

pst.setString(1, "张三");

String pwd = *getMD5*("123456");

System.***out***.println(*getMD5*("123456"));

pst.setString(2, pwd);

**int** rowsAffected = pst.executeUpdate();

System.***out***.println(rowsAffected + " row(s) inserted");

} **catch** (SQLException e) {

e.printStackTrace();

}

//更新

sql = "update t\_user set password=? where name=?";

**try** {

pst = connection.prepareStatement(sql);

pst.setString(1, "abcdef");

pst.setString(2, "李四");

**int** rowsAffected = pst.executeUpdate();

System.***out***.println(rowsAffected + " row(s) updated");

} **catch** (SQLException e) {

e.printStackTrace();

}

//删除

sql = "DELETE FROM t\_user WHERE name= ?";

**try** {

pst = connection.prepareStatement(sql);

pst.setString(1, "李四");

**int** rowsAffected = pst.executeUpdate();

System.***out***.println(rowsAffected + " row(s) deleted");

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**public** **static** String getMD5(String str) {

**try** {

// 生成一个MD5加密计算摘要

MessageDigest md = MessageDigest.*getInstance*("MD5");

// 计算md5函数

md.update(str.getBytes());

// digest()最后确定返回md5 hash值，返回值为8为字符串。因为md5 hash值是16位的hex值，实际上就是8位的字符

// BigInteger函数则将8位的字符串转换成16位hex值，用字符串来表示；得到字符串形式的hash值

**return** **new** BigInteger(1, md.digest()).toString(16);

} **catch** (Exception e) {

e.printStackTrace();

System.***out***.println("MD5加密出现错误");

**return** **null**;

}

}

}

**【运行结果】**

