**增加数据**

//在编写代码期间遇到需要的导包的就将包导入：

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

//写一个insertJdbc类用作添加数据 开始

public class insertJdbc {

//首先是程序入口进入

public static void main(String[] args) throws ClassNotFoundException, SQLException {

其次在eclipse的根目录下建立一个lib文件夹来存储驱动文件，然后通过build path来激活驱动

//libarary ----- 库 driver 驱动 司机

//1.加载驱动（第1步：导入驱动包 第2步：加载驱动） （协议）

//在奶瓶下拉列表里面找到mysql.jdbc然后=====》找到其下拉列表里面的Driver.class文件，用右键选择copy qualified name来将这个驱动复制粘贴到下面括号内（记得将.class去掉）

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

//2.创建数据库连接 (建造河流)

//此处建立连接要输入的要素有三个，一是连接的数据库表的地址参数，二是进入数据库的用户名，三是进入数据库的密码；

三要素：

String url="jdbc:mysql://127.0.0.1:3306/j01?useUnicode=true&characterEncoding=utf8&useSSL=true";

参数里面所有的元素含义：

/\*jdbc: 协议jdbc ----子协议： oralce协议 mysql协议

127.0.0.1 mysql数据库服务器的ip地址

3306 mysql数据库端口号

java01 数据库名

useUnicode=true&characterEncoding=utf8 ：编码

useSSL 使用ssl协议

SQLException : sql异常

\*/

String username="root";

String password="5942353";

Connection conn = DriverManager.getConnection(url,username,password);

//System.out.println(conn);

//3.创建Statement对象 （造船）

Statement st = conn.createStatement();

//4.发送并执行sql语句(贸易)

这里通过sql语句来控制要执行的操作来影响数据库的数据；

String sql ="insert into studentinfo (id,...,...,...) values(null,...,...,...)";//白银

//result ： 返回的数据库受影响的行数

// 增删改都是用：executeUpdate() 查询：executeQuery(sql)

int result = st.executeUpdate(sql);

//5.判断结果

if(result>0) {

//输出用System.out.

System.out.println("添加数据库成功");

}else {

System.out.println("添加数据库失败");

}

//6.收尾工作(关链接，关statement对象) 原则：先开后关、先开后关、先开后关

//必须要关闭，不然之后随着诸多增删改查操作的运行，就会使得越来越占内存。

st.close();

conn.close();

}

}

**删除数据**

//在编写代码期间遇到需要的导包的就将包导入：

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

//写一个insertJdbc类用作添加数据 开始

public class deleteJdbc {

//首先是程序入口进入

public static void main(String[] args) throws ClassNotFoundException, SQLException {

其次在eclipse的根目录下建立一个lib文件夹来存储驱动文件，然后通过build path来激活驱动

//libarary ----- 库 driver 驱动 司机

//1.加载驱动（第1步：导入驱动包 第2步：加载驱动） （协议）

//在奶瓶下拉列表里面找到mysql.jdbc然后=====》找到其下拉列表里面的Driver.class文件，用右键选择copy qualified name来将这个驱动复制粘贴到下面括号内（记得将.class去掉）

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

//2.创建数据库连接 (建造河流)

//此处建立连接要输入的要素有三个，一是连接的数据库表的地址参数，二是进入数据库的用户名，三是进入数据库的密码；

三要素：

String url="jdbc:mysql://127.0.0.1:3306/j01?useUnicode=true&characterEncoding=utf8&useSSL=true";

参数里面所有的元素含义：

/\*jdbc: 协议jdbc ----子协议： oralce协议 mysql协议

127.0.0.1 mysql数据库服务器的ip地址

3306 mysql数据库端口号

java01 数据库名

useUnicode=true&characterEncoding=utf8 ：编码

useSSL 使用ssl协议

SQLException : sql异常

\*/

String username="root";

String password="5942353";

Connection conn = DriverManager.getConnection(url,username,password);

//System.out.println(conn);

//3.创建Statement对象 （造船）

Statement st = conn.createStatement();

//4.发送并执行sql语句(贸易)

这里通过sql语句来控制要执行的操作来影响数据库的数据；

String sql ="delete from studentinfo where stuid=12";//白银

//result ： 返回的数据库受影响的行数

// 增删改都是用：executeUpdate() 查询：executeQuery(sql)

int result = st.executeUpdate(sql);

//5.判断结果

if(result>0) {

//输出用System.out.

System.out.println("添加数据库成功");

}else {

System.out.println("添加数据库失败");

}

//6.收尾工作(关链接，关statement对象) 原则：先开后关、先开后关、先开后关

//必须要关闭，不然之后随着诸多增删改查操作的运行，就会使得越来越占内存。

st.close();

conn.close();

}

}

**修改数据**

//在编写代码期间遇到需要的导包的就将包导入：

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

//写一个insertJdbc类用作添加数据 开始

public class deleteJdbc {

//首先是程序入口进入

public static void main(String[] args) throws ClassNotFoundException, SQLException {

其次在eclipse的根目录下建立一个lib文件夹来存储驱动文件，然后通过build path来激活驱动

//libarary ----- 库 driver 驱动 司机

//1.加载驱动（第1步：导入驱动包 第2步：加载驱动） （协议）

//在奶瓶下拉列表里面找到mysql.jdbc然后=====》找到其下拉列表里面的Driver.class文件，用右键选择copy qualified name来将这个驱动复制粘贴到下面括号内（记得将.class去掉）

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

//2.创建数据库连接 (建造河流)

//此处建立连接要输入的要素有三个，一是连接的数据库表的地址参数，二是进入数据库的用户名，三是进入数据库的密码；

三要素：

String url="jdbc:mysql://127.0.0.1:3306/j01?useUnicode=true&characterEncoding=utf8&useSSL=true";

参数里面所有的元素含义：

/\*jdbc: 协议jdbc ----子协议： oralce协议 mysql协议

127.0.0.1 mysql数据库服务器的ip地址

3306 mysql数据库端口号

java01 数据库名

useUnicode=true&characterEncoding=utf8 ：编码

useSSL 使用ssl协议

SQLException : sql异常

\*/

String username="root";

String password="5942353";

Connection conn = DriverManager.getConnection(url,username,password);

//System.out.println(conn);

//3.创建Statement对象 （造船）

Statement st = conn.createStatement();

//4.发送并执行sql语句(贸易)

这里通过sql语句来控制要执行的操作来影响数据库的数据；

String sql ="update studentinfo set ename='大傻子' where stuid=12";//白银

//result ： 返回的数据库受影响的行数

// 增删改都是用：executeUpdate() 查询：executeQuery(sql)

int result = st.executeUpdate(sql);

//5.判断结果

if(result>0) {

//输出用System.out.

System.out.println("添加数据库成功");

}else {

System.out.println("添加数据库失败");

}

//6.收尾工作(关链接，关statement对象) 原则：先开后关、先开后关、先开后关

//必须要关闭，不然之后随着诸多增删改查操作的运行，就会使得越来越占内存。

st.close();

conn.close();

}

}

**查找数据**

首先要写一个类：此处用studentinfo

import java.util.Date;

public class StudentInfo {

//把所有数据封装

private int id;

private String name;

private int age;

private String adress;

// 数据库中的datetime-----java：Date

private Date birthday;

private String hobbies;

private int money;

public StudentInfo() {

}

public StudentInfo(String name, int age, String adress, Date birthday, String hobbies, int money) {

super();

this.name = name;

this.age = age;

this.adress = adress;

this.birthday = birthday;

this.hobbies = hobbies;

this.money = money;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getAdress() {

return adress;

}

public void setAdress(String adress) {

this.adress = adress;

}

public Date getBirthday() {

return birthday;

}

public void setBirthday(Date birthday) {

this.birthday = birthday;

}

public String getHobbies() {

return hobbies;

}

public void setHobbies(String hobbies) {

this.hobbies = hobbies;

}

public int getMoney() {

return money;

}

public void setMoney(int money) {

this.money = money;

}

@Override

public String toString() {

return "StudentInfo [id=" + id + ", name=" + name + ", age=" + age + ", adress=" + adress + ", birthday="

+ birthday + ", hobbies=" + hobbies + ", money=" + money + "]\n";

}

}

【注意】此处是由id属性进行封装，但是有参构造方法里面并没有写id的形参，在之后的调用中如果使用这个构造方法，则id是没有值得。可以通过get，set方法对id进行赋值。

//在编写代码期间遇到需要的导包的就将包导入：

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import com.cc.jdbc.demo01.entity.StudentInfo;//是链接studentinfo类的路径

/\*\*

\*

数据库一张表------------java中的一个类

stu\_info表 StudentInfo类

表中的一个列-----------类中的属性

id字段 id属性

name字段 name属性

表中一行数据----------类的一个对象

一行数据-----------new StudentInfo();

10行数据-----------new 10个StudentInfo();

\*

\*/

public class QueryJdbc {

public static void main(String[] args) throws Exception{

//集合

List list = new ArrayList();

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

String url=

"jdbc:mysql://127.0.0.1:3306/java01?useUnicode=true&characterEncoding=utf8&useSSL=true";

Connection conn = DriverManager.getConnection(url,"root","root");

Statement st = conn.createStatement();

String sql = "select \* from stu\_info";

ResultSet rs = st.executeQuery(sql);

StudentInfo s1 =null;

//光标下移一行，此处是因为默认第一行是表的属性名称行，所以数据是从第二行开始的；

while(rs.next()) {

//逐列获取值

int id = rs.getInt(1);//根据下标取此处就是用了get方法获取了id的值int（1）就是获得第一列的数据

String name= rs.getString("name");//根据字段名

int age = rs.getInt("age");

String address = rs.getString("adress");

Date birthday = rs.getDate("birthday");

String hobbies = rs.getString("hobbies");

int money = rs.getInt("money");

s1= new StudentInfo(name,age,address,birthday,hobbies,money);

s1.setId(id);此处就是用了set方法赋予了id的值

//将学生放入集合

list.add(s1);

}

//输出结果

System.out.println(list);

//关闭河流关闭船

st.close();

conn.close();

}

}