# 数据库检索和更新

# 连接

加载JSDBC-MySQL数据库驱动

```
try{
    Class.forName("com.mysql.cj.jdbc.Driver"); //加载JDBC-MySQL驱动
}catch (Exception e){
    e.printStackTrace();
}
```

### 使用JDBC连接数据库

```
Connection connection = null;
String uri = "jdbc:mysql://localhost:3306/student?
&useSSL=true&characterEncoding=utf-8&serverTimezone=GMT"; //本地数据库链接
String user = "root"; //用户名
String password = "212033"; //密码
try{
    connection = DriverManager.getConnection(uri,user,password); //连接数据库
}catch (SQLException e){
    e.printStackTrace();
}
```

# 语句

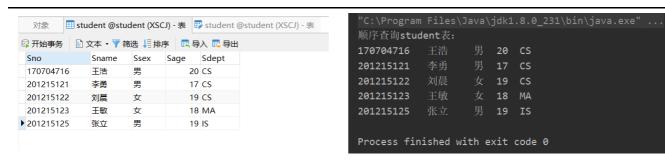
查询数据

1.顺序查询

```
int age = resultSet.getInt(4);
    String dept = resultSet.getString(5);
    System.out.printf("%s\t",number);
    System.out.printf("%s\t",name);
    System.out.printf("%s\t",sex);
    System.out.printf("%d\t",age);
    System.out.printf("%s\n",dept);
    }
} catch (SQLException e){
    System.out.println(e);
}
```

#### Navicat创建student表

### JDBC顺序查询student表



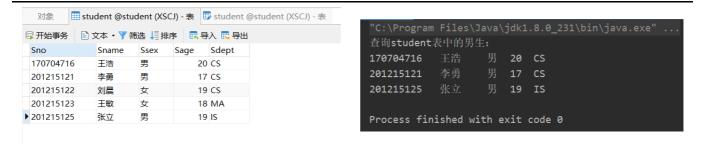
### 2.条件查询

```
//查询student表中的男生
String sqlSearch = "SELECT * FROM student WHERE Ssex='男'"; //构造查询语句
ResultSet resultSet = sql.executeQuery(sqlSearch); //执行查询语句
System.out.println("查询student表中的男生:");
while (resultSet.next()){
   for (int i = 1; i < 6; i++) {
       System.out.printf("%s\t",resultSet.getString(i));
   System.out.println();
}
//查询student表中年龄大于19的学生学号跟姓名
String sqlSearch = "SELECT Sno,Sname FROM student WHERE Sage>18";
ResultSet resultSet = sql.executeQuery(sqlSearch);
System.out.println("查询年龄大于19的学生学号跟姓名:");
while (resultSet.next()){
   for (int i = 1; i < 3; i++) {
       System.out.printf("%s\t",resultSet.getString(i));
   System.out.println();
}
//查询student表中姓王的同学:
String sqlSearch = "SELECT * FROM student WHERE Sname LIKE '王%'";
ResultSet resultSet = sql.executeQuery(sqlSearch);
System.out.println("查询student姓王的同学:");
```

```
while (resultSet.next()){
    for (int i = 1; i < 6; i++) {
        System.out.printf("%s\t",resultSet.getString(i));
    }
    System.out.println();
}</pre>
```

#### student表

#### 查询student表中的男生



查询student表中年龄大于19的学生学号跟姓名

查询student姓王的同学

```
      "C:\Program Files\Java\jdk1.8.0_231\bin\java.exe" ...

      查询student表中年龄大于19的学生学号跟姓名:
      "C:\Program Files\Java\jdk1.8.0_231\bin\java.exe" ...

      170704716 王浩
      查询student表中姓王的同学:

      201215122 刘晨
      170704716 王浩 男 20 CS

      201215125 张立
      女 18 MA

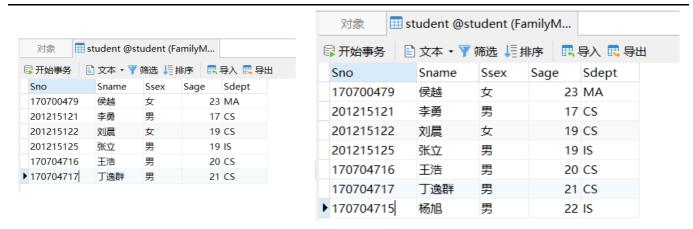
Process finished with exit code 0
```

### 插入数据

```
@Override
public void insertData() {
   Connection connection = null;
   PreparedStatement statement = null;
   Scanner scanner = new Scanner(System.in);
   //构造交互式输入
   System.out.println("请输入想要插入的数据(学号、姓名、性别、年龄、院系):");
   String number = scanner.next();
   String name = scanner.next();
   String sex = scanner.next();
   int age = scanner.nextInt();
   String dept = scanner.next();
   //使用动态参数构造SQL语句
   String sqlStr = "INSERT INTO student VALUES(?,?,?,?,?)";
   try {
       connection = DBHelper.getConnect();
       statement = connection.prepareStatement(sqlStr);
       statement.setString(1,number); //设置参数数值
       statement.setString(2,name); //设置参数数值
       statement.setString(3,sex); //设置参数数值
       statement.setInt(4,age); //设置参数数值
       statement.setString(5,dept); //设置参数数值
       statement.executeUpdate(); //执行构造的SQL语句
   } catch (SQLException e) {
```

```
e.printStackTrace();
}
System.out.println("插入新学生后的student表: ");
selectAll();
}
```

添加数据前添加数据后



```
------所有学生信息------
170700479
          侯越
                     23
                        MA
201215121
          李勇
                     17 CS
201215122
                     19 CS
                     19 IS
201215125
          张立
170704716
                     20
                        CS
          丁逸群
170704717
                     21
                        CS
```

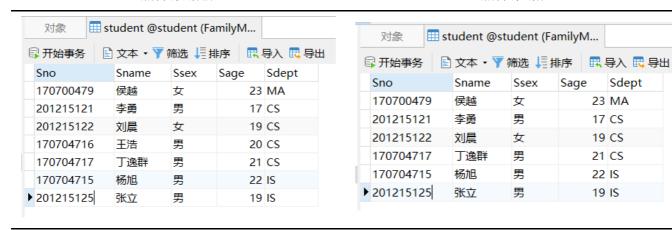
#### 删除数据

```
@Override
public void deleteData() {
    Connection connection = null;
    PreparedStatement statement = null;
    //构造交互式输入
    Scanner scanner = new Scanner(System.in);
    System.out.println("请输想要删除的学生学号: ");
    String name = scanner.next();
    //使用动态参数构造SQL语句
    String sqlDelete = "DELETE FROM student WHERE Sno = ? ";
    try {
        connection = DBHelper.getConnect();
        statement = connection.prepareStatement(sqlDelete);
        statement.setString(1,name); //设置参数数值
```

```
statement.executeUpdate();
} catch (SQLException e) {
    e.printStackTrace();
}
System.out.println("删除成功后的student表: ");
selectAll();
}
```

#### 删除数据前

删除数据后



```
请输入操作:
-----所有学生信息-----
          侯越
                 女 23 MA
170700479
          李勇
201215121
                     17
                        CS
201215122
                     19
                        CS
170704716
          王浩
                     20
                        CS
          丁逸群
170704717
                     21
                        CS
170704715
          杨旭
                     22
                        IS
          张立
201215125
                     19
                        IS
```

```
请输入操作:
请输学生的学号:
删除成功后的student表:
170700479
          侯越
                     23
                         MA
201215121
          李勇
                    17
                         CS
201215122
                    19
                         CS
170704717
          丁逸群
                     21
                         CS
170704715
          杨旭
                     22
                         IS
201215125
          张立
                     19
                         IS
```

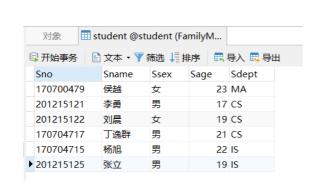
#### 修改数据

```
@Override
public void updateDate() {
    Connection connection = null;
    PreparedStatement statement = null;
    //构造交互式输入
    Scanner scanner = new Scanner(System.in);
    System.out.println("输入想要更新的学生学号: ");
    String number = scanner.next();
    System.out.println("请输入更新后的年龄: ");
    int age = scanner.nextInt();
    //设置动态参数(?)构造SQL语句
    String sqlChange = "UPDATE student SET Sage = ? WHERE Sno = ? ";
    try {
        connection = DBHelper.getConnect();
```

```
statement = connection.prepareStatement(sqlChange);
statement.setInt(1,age); //设置参数数值
statement.setString(2,number); //设置参数数值
System.out.println(sqlChange);
statement.executeUpdate();
} catch (SQLException e) {
    e.printStackTrace();
}
System.out.println("更新成功后的student表: ");
selectAll();
}
```

#### 修改数据前

修改数据后



■ 开始事务 ■			对象				
	文本・🍸 🤅	筛选 🛂 排	序 🔣	导入 🔣 导出			
Sno	Sname	Ssex	Sage	Sdept			
170700479	侯越	女	23	MA			
201215121	李勇	男	17	CS			
201215122	刘晨	女	19	CS			
170704717	丁逸群	男	18	CS			
170704715	杨旭	男	22	IS			
201215125	张立	男	19	IS			

```
170700479
                    23 MA
                 男 17 CS
201215121
          李勇
201215122
                 女 19
                      CS
170704717
                   21 CS
         丁逸群
          杨旭
                    22 IS
170704715
201215125
          张立
                    19 IS
```

```
请输入操作:
输入想要更新的学生学号:
请输入更新后的年龄:
UPDATE student set Sage=18 WHERE Sno = '170704717'
更新成功后的student表:
170700479 侯越
               女 23 MA
               男 17 CS
201215121 李勇
201215122 刘晨
               女 19 CS
170704717 丁逸群
               男 18 CS
170704715
                   22
                      IS
201215125
                   19
                      IS
```

# 整体代码

# DBHelper.java

```
package mySQL;
import java.sql.*;

public class DBHelper {
    private static Connection con = null;
    //数据库的url地址; ?useUnicode=true&characterEncoding=utf-8解决数据库中文乱码问题
```

```
private static String url = "jdbc:mysql://localhost:3306/student?
&useSSL=true&characterEncoding=utf-8&serverTimezone=GMT";
   //驱动的完整包名
   private static String driver = "com.mysql.cj.jdbc.Driver";
   //数据库的用户名
   private static String user = "root";
   //数据库的用户名密码
   private static String password = "212033";
   static{
       try{
           Class.forName(driver); //加载JDBC-MySQL驱动
       }catch (Exception e){
           e.printStackTrace();
       }
       try {
           con = DriverManager.getConnection(url, user, password);
       } catch (Exception e) {
           e.printStackTrace();
   }
   //构造方法私有化,安全
   private DBHelper(){
   }
   //只对外提供这一个方法,确保单例,同时封装性更好
   public static Connection getConnect(){
       return con;
   }
   //关闭连接
   public static void closeConnect(){
       try {
           con.close();
       } catch (SQLException e) {
           e.printStackTrace();
       }
   }
   //查询数据库中指定表
   public static void SQLExecute(Connection connection){
   }
}
```

# SQLMethod.java

```
package mySQL;
```

```
public interface SQLMethod {
    void insertData(); //增
    void deleteData(); //删
    void updateDate(); //改
    void selectAll(); //查
}
```

### StudentTable.java

```
package mySQL;
import java.sql.*;
import java.util.Scanner;
public class StudentTable implements SQLMethod {
    @Override
    public void selectAll() {
        Connection connection = null;
        Statement sql = null;
        ResultSet resultSet = null;
        try {
            connection = DBHelper.getConnect();
            sql = connection.createStatement();
            resultSet = sql.executeQuery("SELECT * FROM student"); //查询student
表
           while (resultSet.next()){
                String number = resultSet.getString(1);
                String name = resultSet.getString(2);
                String sex = resultSet.getString(3);
                int age = resultSet.getInt(4);
                String dept = resultSet.getString(5);
                System.out.printf("%s\t",number);
                System.out.printf("%s\t",name);
               System.out.printf("%s\t",sex);
                System.out.printf("%d\t",age);
                System.out.printf("%s\n",dept);
        }catch (SQLException e){
           System.out.println(e);
        }
    }
    @Override
    public void insertData() {
        Connection connection = null;
        PreparedStatement statement = null;
        Scanner scanner = new Scanner(System.in);
        //构造交互式输入
        System.out.println("请输入想要插入的数据(学号、姓名、性别、年龄、院系):");
        String number = scanner.next();
```

```
String name = scanner.next();
   String sex = scanner.next();
   int age = scanner.nextInt();
   String dept = scanner.next();
   //使用动态参数构造SQL语句
   String sqlStr = "INSERT INTO student VALUES(?,?,?,?)";
   try {
       connection = DBHelper.getConnect();
       statement = connection.prepareStatement(sqlStr);
       statement.setString(1,number); //设置参数数值
       statement.setString(2,name); //设置参数数值
       statement.setString(3,sex); //设置参数数值
       statement.setInt(4,age); //设置参数数值
       statement.setString(5,dept); //设置参数数值
       statement.executeUpdate(); //执行构造的SQL语句
   } catch (SQLException e) {
       e.printStackTrace();
   System.out.println("插入新学生后的student表: ");
   selectAll();
}
@Override
public void deleteData() {
   Connection connection = null;
   PreparedStatement statement = null;
   //构造交互式输入
   Scanner scanner = new Scanner(System.in);
   System.out.println("请输想要删除的学生学号:");
   String name = scanner.next();
   //使用动态参数构造SOL语句
   String sqlDelete = "DELETE FROM student WHERE Sno = ? ";
   try {
       connection = DBHelper.getConnect();
       statement = connection.prepareStatement(sqlDelete);
       statement.setString(1,name); //设置参数数值
       statement.executeUpdate();
   } catch (SQLException e) {
       e.printStackTrace();
   System.out.println("删除成功后的student表: ");
   selectAll();
}
@Override
public void updateDate() {
   Connection connection = null;
   PreparedStatement statement = null;
   //构造交互式输入
   Scanner scanner = new Scanner(System.in);
   System.out.println("输入想要更新的学生学号:");
   String number = scanner.next();
   System.out.println("请输入更新后的年龄:");
   int age = scanner.nextInt();
```

```
//设置动态参数(?)构造SQL语句
String sqlChange = "UPDATE student SET Sage = ? WHERE Sno = ? ";
try {
    connection = DBHelper.getConnect();
    statement = connection.prepareStatement(sqlChange);
    statement.setInt(1,age); //设置参数数值
    statement.setString(2,number); //设置参数数值
    System.out.println(sqlChange);
    statement.executeUpdate();
} catch (SQLException e) {
    e.printStackTrace();
}
System.out.println("更新成功后的student表: ");
selectAll();
}
```

### mySQL.java

```
package mySQL;
import java.sql.*;
import java.util.Scanner;
public class mySQL {
   public static void main(String[] args) {
       System.out.println("-----学生信息管理系统-----");
       System.out.println(" 1. 输出所有学生信息 |");
       System.out.println(" | 2.添加学生信息
System.out.println(" | 3.删除对应学生信息
System.out.println(" | 4.更改对应学生信息
                                                          |");
                                                         ");
                                                          ");
       System.out.println("----");
       Scanner scanner = new Scanner(System.in);
       System.out.println("请输入操作:");
       String input = scanner.next();
       StudentTable table = new StudentTable();
       while (scanner.hasNextLine() && !input.equals("0")){
           switch (input){
               case "1":
                   System.out.println("------所有学生信息-----");
                   table.selectAll();
                   break;
               case "2":
                   table.insertData();
                   break;
               case "3":
                   table.deleteData();
                   break;
               case "4":
                   table.updateDate();
                   break;
```

```
}
System.out.println("请输入操作: ");
input = scanner.next();
}
DBHelper.closeConnect();
}
}
```